Personalised Health Monitoring and Decision Support Based on Artificial Intelligence and Holistic Health Records

D4.2 – Personalised health modelling and predictions II

WP4 Knowledge Management and Utilisation in the iHelp Platform

Dissemination Level:	Public
Document type:	Report
Version:	1.0
Date:	October 27, 2022

The project iHelp has received funding from the European Union's Horizon 2020 Programme for research, technological development, and demonstration under grant agreement no 101017441.

Document Details

Project Number	101017441
Project Title	iHelp - Personalised Health Monitoring and Decision Support Based on Artificial Intelligence and Holistic Health Records
Title of deliverable	Personalised health modelling and predictions II
Work package	WP4 Knowledge Management and Utilisation in the iHelp Platform
Due Date	October 31, 2022
Submission Date	October 27, 2022
Start Date of Project	January 1, 2021
Duration of project	36 months
Main Responsible Partner	ATC
Deliverable nature	Report
Author name(s)	Giorgos Giotis (ATC), Thanos Kalligeris (ATC) Konstantina Liagkou (ATC), Spyros Papafragkos (ATC), Maritini Kalogerini (ATC), Kenneth Muir, Ke Te-Min, Artitaya Lophatananon (UNIMAN)
Reviewer name(s)	Oscar Garcia (ICE), George Manias (UPRC)

Document Revision History

Version History			
Version	**Date**	**Author(s)**	**Changes made**
0.1	2022-09-21	Giorgos Giotis (ATC), Spyros Papafragkos (ATC), Maritini Kalogerini (ATC), Thanos Kaligeris (ATC), Kenneth Muir (UNIMAN), Artitaya Lophatananon (UNIMAN), Andrea Damiani (FPG), Oscar Garcia Perales (ICE), Eshita Dhar (TMU), Vincent Moncho Mas (HDM), Harm Op den Akker (iSprint), Aristodemos Pnevmatikakis (iSprint)	Creation of 1st version. ToCs and initial content
0.2	2022-09-23	Spyros Papafragkos (ATC)	Updates in Chapters 1 and 7
0.3	2022-10-03	Giorgos Giotis (ATC), Spyros Papafragkos (ATC), Thanos Kaligeris (ATC)	Updates in Sections 6.2.4, 6.4.4, 6.4.5 and conclusions

0.4	2022-10-12	Kenneth Muir (UNIMAN), Artitaya Lophatananon (UNIMAN), Ke Te-Min (UNIMAN)	Updates in Section 6.1.3
0.5	2022-10-17	George Manias (UPRC)	1st Internal Review
0.6	2022-10-19	Oscar Garcia (ICE)	2nd Internal Review
0.7	2022-10-21	Maritini Kalogerini (ATC), Giorgos Giotis (ATC)	Resolving comments in all sections
0.8	2022-10-25	Pavlos Kranas (LXS)	Quality review
1.0	2022-10-27	Dimosthenis Kyriazis (UPRC)	Final version for submission

Table of Contents

List of Figures

Executive summary

This document summarizes the actions performed under T4.1 - “Personalized Health Modelling and Predictions” in the context of WP4 “Knowledge Management and Modelling in the iHelp Platform” at this phase of the project. The first version of this series of deliverables, i.e., D4.1 - “Personalised health modelling and predictions I”, provided an extended description of the mechanisms and Artificial Intelligence (AI) models that will be implemented for the realisation of personalised health and risk prediction models. The implementation of the AI algorithms that are being created during the project’s lifecycle highly depend on the provided datasets on which they are trained. At the previous phase of the project, the description of the datasets that are going to be used, drove the actions under this Task towards a concrete description and specification of the AI algorithms and models that will be utilized.

In this updated version D4.2 - “Personalised health modelling and predictions II”, the primary data were available for two out of five pilots -namely UNIMAN and MUP- and a sample dataset was provided by the FPG pilot. In this respect, an initial approach of the development of the AI models and some preliminary results are provided in the corresponding sub-sections of Section 6. As in the previous version, this document encapsulates the necessary and relevant information that was researched from recent bibliography to facilitate the manipulation of the available datasets, setting the basis for the design and implementation of the models. Finally, in this deliverable are analysed the main concepts behind the models, also in compliance with an analysis of the importance of known and unknown risk factors based on the description of the clinical/primary data.

1 Introduction

Over many decades, massive effort from numerous experts has been devoted to cancer research globally. Despite notable progress in this field, specific cancer types -such as Pancreatic Cancer- are highly associated with poor prognosis and low survival rates. In Pancreatic Cancer, symptoms typically occur late in the course of the disease, so early detection can result not only to a more accurate diagnosis but also in increased the survival rates of the patients (K., C., K., + 21).

Among other technologies, current applications of AI, Machine Learning (ML) and Deep Learning (DL) in cancer research and clinical care, have emerged as highly successful tools for early cancer detection and risk identification. The applications of AI/ML in cancer research and care are already highly diverse and will continue to expand. Data modelling tools and techniques allow advanced AI algorithms to develop personalised health models that enable the identification of pancreatic cancer, the relevant contributing factors and the associated risks. The validity of AI models highly depends on the datasets on which they are trained, however, the extraction of high-quality data for research uses from real-world sources has proven complex. Applying AI approaches to real world data - such as Electronic Health Records (EHRs) or Holistic Health Records (HHRs), clinical notes, -omics and patient generated health data from wearables, smart phones, and social media- require a very careful and well-organized pre-processing of data.

For the case of Pancreatic Cancer, several risk factors have been identified either they are modifiable (i.e., smoking, alcohol, obesity, dietary factors) or non-modifiable (such as gender, age, ethnicity, diabetes mellitus, family history). iHelp, focusing on the modifiable risk factors, aims to make use of the applications of the AI algorithms in combination with the benefits of mobile and wearable technologies, for improving individuals' adherence to risk mitigation strategies, delivery of targeted health advice and even supporting lifestyle changes. The personalized healthcare and the identification of high-risk individuals for early detection, via the AI algorithms that will be implemented under iHelp, is of huge importance, not only for the possible patients themselves, but also for the Health Care Professionals (HCPs) and policy makers.

1.1 Scope of the document

D4.2 - "Personalised health modelling and predictions II" is developed under T4.1 "Personalized Health Modelling and Predictions", with the main goal to further update the deliverable of previous version (D4.1 - "Personalised health modelling and predictions I"). More specifically, the scope of this document is to provide description of the mechanisms that are being implemented to realise the creation of personalised health and risk prediction models. The approaches, the techniques and the AI algorithms that are - and will continue to be - realized during the iHelp project, will be described in this series of deliverables. This series of deliverables, started with D4.1 - "Personalised health modelling and predictions I", which is followed by this updated version II, while one third version will follow on M32. On the one hand, the majority of the context of this document is based on the available retrospective primary and secondary data, that the Pilots have shared before sharing the "real" prospective data. On the other hand, the algorithms created specifically for MUP pilot, are based on their real data. As the project progresses and more real data are becoming available, the deliverable will be updated as it will evolve and follow the directions of iHelp project on cancer risk assessment.

1.2 Structure of the document

This document is divided into seven main sections structured as follows:

- Section 1 includes a short introduction, the main description of this document, a reference to the relevance of this task with other WPs in the iHelp project, as well as the updates since the D4.1 – "Personalised health modelling and predictions I".
- Section 2 gives an overview of the advances of ML in the domain of healthcare, with special focus on the approach of "Federated Learning (FL)", "clustering analysis" and their importance in healthcare systems.
- Section 3, summarizes the procedure of the pre-processing of data, including methods for data cleaning, initial feature selection, imputation, and labelling annotation techniques.
- Section 4, refers to the risk factors importance, including a description of the Interpretable models and the model agnostic interpretation methods among others.
- Section 5 discusses the risk predictions including the SotA algorithms, the relevant hyperparameter optimization and the models evaluation.
- Section 6 introduces the iHelp pilot cases and aims to describe the work done, before the initiation of this project. More specifically, all pilots, (namely: UNIMAN, FPG, HDM, MUP and TMU), present their connection to T4.2 "Personalized Health Modelling and Predictions", via describing in general the data that they are going to share, any pre-existing algorithms and their results, and their specific needs. Moreover, updates and new algorithms have been described for UNIMAN, FPG and MUP pilots.
- Finally, Section 7 highlights the main outcomes and concludes this report.

1.3 Relevance with other Work Packages

This report summarizes the approaches, the techniques and the AI algorithms that will be realized during the iHelp project, for the creation of the risk prediction models. The data that will be imported to the initial algorithms in T4.1 - "Personalized Health Modelling and Predictions", will be derived from WP3 – "Personalised Holistic Health Records", after the mapping activities into HHR format and the respective storage in the HHR platform. The models that will be produced here, will be further developed, and specified under T5.1 - "Techniques for Early Risk Identification, Predictions & Assessment", under WP5 – "AI for Early Risk Assessment and Personalised Recommendations". After that, specific recommendations and proposed Intervention measures will be drafted in WP5 - "AI for Early Risk Assessment and Personalised Recommendations".

1.4 Background

iHelp project builds upon the innovations coming out of previous EU projects such as CrowdHEALTH (https://www.crowdhealth.eu/), where the concept of HHRs was firstly investigated. CrowdHEALTH is an international research project partially funded by the Horizon 2020 Programme of the European Commission that worked on integrating high volumes of health-related heterogeneous data from multiple sources with the aim of supporting policy making decisions. The project started in March 2017 and finished in February 2020. Today's rich digital information environment is characterized by the multitude of data sources. There are extremely large amounts of medical data. But currently collected data are heterogeneous, spread across different health care providers and systems that operate independently. Due

to this fact it is quite common that important events related to health are missed. CrowdHEALTH delivered a secure ICT platform that was able to collect and aggregate high volumes health data from multiple information sources in Europe. CrowdHEALTH also proposed the evolution of Patient Health Records (PHRs) towards HHRs enriched to become more "Social HHRs" to capture the clinical, social and human factors.

In addition, ATC brings the experience on Health-related Big Data Analytics, from BD2Decide project (PHC-21-2015 call). More specifically, prediction models have been published for head-and-neck cancer cases, while the innovation that will be utilized via this project includes the incorporation of adaptive learning to update, improve and refine the models using routinely collected data.

1.5 Updates since D4.1 - "Personalised health modelling and predictions I"

The updates since D4.1 - "Personalised health modelling and predictions I" resulted in the creation of D4.2 - "Personalised health modelling and predictions II" that incorporates all the work done during the second year of the project. The changes concerning the structure of the document include:

- the transfer of Section 3 "Connection with the Pilots: State of the art" to the end of the document, named Section 6 "Connection with the Pilots: State of the art".
- the additions on the updated version were made in (new) Section 6 and specifically in sub-sections 6.1, 6.2 and 6.4 that are related to UNIMAN, FPG and MUP pilots respectively. Therein, is presented the pre-processing of the data, the development of the AI algorithms that were utilized for the predictions of each requirement as well as the corresponding evaluation methods and related results. Moreover, methods and tools in the context of eXplainable AI (XAI) are provided in an effort to interpret these research outcomes.

2 Machine Learning in Healthcare

Machine Learning (ML), simply put, is a type of AI in which computers are programmed to learn information without human intervention. In ML, the development of the underlying algorithms relies on computational statistics. Computers are provided data and then the computers “learn” from that data. The data actually “teaches" the computer by revealing its complex patterns and underlying algorithms. The larger the sample of data the “machine" is provided, the more precise the machine's output becomes.

The AI technologies are steadily being applied to the healthcare domain. The use of AI in healthcare has the potential to assist healthcare providers and professionals in many aspects of patient care, especially for illness detection and treatment selection. Most AI and healthcare technologies have strong relevance to the domain, where they can perform just as well or better than humans at certain procedures. ML is one of the most common forms of AI in healthcare. It is a broad technique at the core of many approaches to AI and healthcare technology and there are many versions of it.

Using AI in healthcare, the most widespread utilisation of traditional ML is precision medicine, where the algorithms are able to predict what treatment procedures are likely to be successful with patients based on their make-up and the treatment framework is a huge leap forward for many healthcare organisations. In this sense, the precision medicine uses algorithms that fail under supervised learning, where at the training stage, the results are known so the algorithm is able to know which is the expected result.

Diagnosis and treatment of diseases have been at the core of AI in healthcare for the last 50 years. Early rule-based systems had potential to accurately diagnose and treat disease but were not totally accepted for clinical practice. They were not significantly better at diagnosing than humans, and the integration was less than ideal with clinician workflows and health record systems.

But whether rules-based or algorithmic, using AI in healthcare for diagnosis and treatment plans can often be difficult to integrate with clinical workflows and EHR systems. Integration issues have been a greater barrier to widespread adoption of AI in healthcare when compared to the accuracy of suggestions. Much of the AI and healthcare capabilities for diagnosis and treatment from medical software vendors are standalone and address only a certain area of care. Some EHR software vendors are beginning to build limited healthcare analytics functions with AI into their product offerings but are in the elementary stages. To take full advantage of the use of AI in healthcare using a standalone EHR system providers will either have to undertake substantial integration projects themselves or leverage the capabilities of third-party vendors that have AI capabilities and can integrate with their EHR.

2.1 Federated Learning

Standard ML approaches require centralizing the training data on one machine or in a data centre. However, for models trained from distributed datasets or user interaction with mobile devices, we're introducing an additional approach: *Federated Learning*.

FL enables IoT devices (such as wearables or mobile phones) to collaboratively learn a shared prediction model while keeping all the training data on device, decoupling the ability to do ML from the need to store the data in the cloud. This goes beyond the use of local models that make predictions on these devices by bringing model *training* to the device as well and then distributing the computation efforts across different computation agents.

The overall functionality can be described as follows. The device downloads the current model, improves it by learning from data gathered by the device, and then summarizes the changes as a small focused update. Only this update to the model is sent to the cloud, using encrypted communication, where it is immediately averaged with other user updates to improve the shared model. All the training data remains on the device, and no individual updates are stored in the cloud.

FL allows for smarter models, lower latency, and less power consumption, all while ensuring privacy. And this approach has another immediate benefit: in addition to providing an update to the shared model, the improved model on the device can also be used immediately, powering experiences personalized by the way the device is used.

2.1.1 Scalability

Producing a scalable federated learning approach currently depends on how the partitions are partitioned and by looking at optimisations in the sometimes-high communications overhead of FL algorithms. Works reported at (Y., K., L., + 21) and (Z., W., B., 21) explore the scalability for FL although the examples described there are from other than healthcare domains.

2.1.2 Security

FL is preferred in use-cases where security and privacy are the key concerns, and certainly healthcare domain is one of these. Having a clear view and understanding of risk factors enable an implementer/adopter of FL to successfully build a secure environment and gives researchers a clear vision on possible research areas.

Moreover, it enables clients to collaboratively learn a shared global model without sharing their local training data with a cloud server. However, malicious clients can corrupt the global model to predict incorrect labels for testing examples. Existing defences against malicious clients leverage Byzantine-robust FL methods. However, these methods cannot provably guarantee that the predicted label for a testing example is not affected by malicious clients. The latter is addressed through the utilization of ensemble FL. In particular, given any base FL algorithm, we use the algorithm to learn multiple global models, each of which is learnt using a randomly selected subset of clients. When predicting the label of a testing example, we take majority vote among the global models.

2.2 Clustering

Cluster analysis, or clustering, is an unsupervised ML task. It involves automatically discovering natural grouping in data. Unlike supervised learning (like predictive modelling), clustering algorithms only interpret the input data and find natural groups or clusters in feature space.

It can be defined as ***"A way of grouping the data points into different clusters, consisting of similar data points to a given feature. The objects with the possible similarities remain in a group that has less or no similarities with another group"***.

The algorithm tries to find similar patterns in the unlabelled dataset such as shape, size, colour, behaviour, combination of these, or other factors such as %-presence of a given protein, etc., and divides them as per the presence and absence of those similar patterns. By being an unsupervised learning method, it deals with the unlabelled dataset. After applying this clustering technique, each cluster or group is provided with

a cluster-ID for further processing. ML system can use this id to simplify the processing of large and complex datasets.

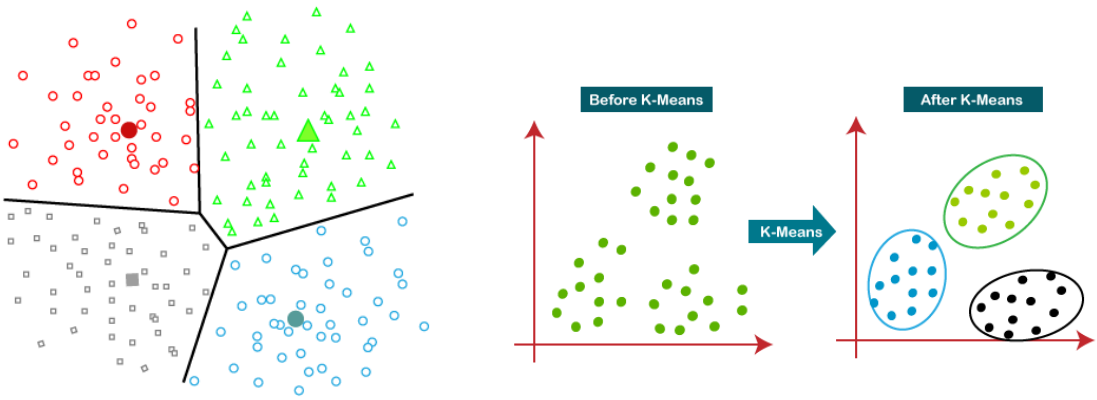

Figure 1: Examples of grouping the data points into different clusters

The main algorithm going to be used within iHelp of all the families that belong to Clustering is K-Means, which is of *Partioning Clustering* type, and it is useful when it is not 100% sure what we are looking for. Partitioning Clustering divides the data into non-hierarchical groups. It is also known as the centroid-based method. Within K-Means, the dataset is divided into a set of *k* groups, where K is used to define the number of pre-defined groups (K=2 means there will be 2 clusters, K=3 means 3 clusters, etc.). The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid. By being an iterative algorithm, the data is initially divided in the past clusters and then the different data is being accommodated to one cluster or another depending on their distance to the different centroids of the clusters.

One of the typical uses of clustering algorithms within the healthcare domain is in the identification of Cancer Cells where they are widely used. The algorithm divides the cancerous and non-cancerous data sets into different groups to assess the likeliness of a cell being affected by cancer. In this sense, the information from the patients is stored in the EHRs so it should be possible to create clusters based on the information present there to learn about the progression of the diseases.

2.2.1 Scalability

Scalability in clustering means that as you increase the number of data objects, the time to perform clustering should roughly scale to the order of complexity of the algorithm. For example, if you perform K-means clustering, we know it is O(n), where n is the number of data objects. If you increase the number of data objects 10 folds, then the time to cluster them should also 'roughly' increase 10 time i.e. there should be a linear relationship.

2.3 Combination of Algorithms

Sometimes the solution to a given problem cannot be assessed by using one single ML algorithm; instead, a combination of algorithms might be more appropriate rather than using one. A recent work published by JAMA Oncology (YAN, 21) in one of his journals, showed how combining ML algorithms with Behavioural Nudges was beneficial to increase rates of serious illness conversations in patients with cancer

3 Pre-processing of data

3.1 Initial feature selection

A common belief among the researchers is that more features equals better model performance, however, this is far from being true. Less features usually means faster training models: for parametric models like linear or Logistic Regression (LR), it means there are less weights to calculate, and for non-parametric models like Random Forest (RF) of DT, it means there are less features to evaluate at each split.

Feature selection consists of automatically selecting the best features for our models and algorithms, by taking these insights from the data, and without the need to use expert knowledge or other kinds of external information. Automatically means that we do not handpick the features, but instead we use some algorithm or procedure that keeps only the most important features for our model and its application domain.

Nevertheless, it is important to know here that expert knowledge of the application domain that the model is being built for is very important, as it allows us to better understand the data that is going to be used, and therefore gain some intuition about which features will probably be important, and which features should probably be discarded.

The study of features to eliminate from our models is very important, since we can remove irrelevant features that would not be affecting or changing the output of our model. If for example, we try to predict the survival rate of breast cancer in females between 20-30 years old, using variables that include the traffic accident, these variables will probably not be very useful. These kinds of irrelevant features can actually decrease the performance of your model by introducing noise.

In the context of iHelp, the clinicians in close consultation with the data scientists (see also Figure 2) will decide which features can be considered relevant in the corresponding pilots'/use cases. It will be the first level of feature selection/elimination and it will help significantly to boost the process since many irrelevant features can lead to significant delays.

3.2 Imputation

Clinical trials and studies may concern diseases with low incidence rates and different types of exams, which lead to medical datasets with low prevalence categories and various features that are missing because the people involved were never subjected to the corresponding exams. Such datasets are problematic for ML because most data processing techniques and algorithms require a complete dataset in order to yield results. Therefore, it is important to handle missing data in order to ensure good performance of the models.

Missing data can be categorized into 3 types according to the underlying mechanism: Missing completely at random (MCAR), Missing at random (MAR) and Missing not at Random (MNAR). When the mechanism that is causing the missingness is irrelevant to the observed and the missing data we have MCAR. In this case the missingness does not cause any bias, only larger errors. When the mechanism only depends on the observed data and not on the missing data then we have MAR. In this case it is possible to predict the missing data based on other data which are filled. Finally, when the mechanism depends on the missing data itself, we have MNAR.

The suggested solutions to the missing data problem depend on the type of data that is missing. Since data MCAR does not introduce bias, it is possible to use the subset of the dataset that does not contain missing data. This is advised to be applied only when missingness does not exceed 10~15%. However, in datasets with low prevalence classes this may cause the loss of critical information. It is thus better to try to estimate the missing values or replace them with a specific value.

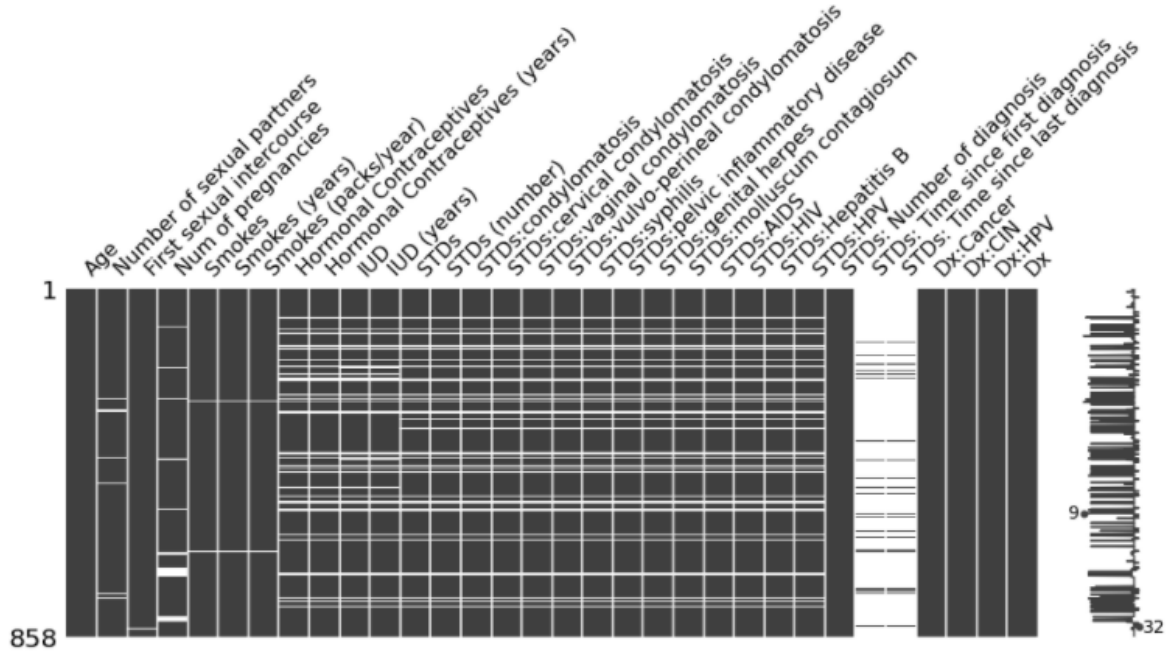

Figure 2: Data missingness in UCI Cervical cancer Data Set.

In the above figure (Figure 2), we visualize the missing data of a dataset regarding cervical cancer. Rows represent observations columns represent features. White spaces denote missing features from observations. The plot on the right side shows the total available features for each observation. Each column represents the feature that is written above the column and each row represents a patient's data. White colour denotes that a feature is missing for that patient and on the right, we see the total available features for each patient. As we can see a patient only has 9 out of 32 features and features like time since STD tests are missing even though patients have had the relevant STD tests. In such a case it may be more convenient to drop the features all together. However, it might be important to keep patients who are missing some features since their available information may prove important to the analysis.

Since medical datasets contain a variety of diseases, conditions and test results it is only natural that many fields are missing from most patients and thus most rows will have missing data. Furthermore, it is not possible to estimate the type of missingness for the data because each field derives from a different exam or process at a different time and perhaps even a different hospital. Especially in the case of diseases with low incidence, rates or different examinations compared to others there are going to be significantly fewer examples and they will have missing values. By dropping the data with missing values, it is thus more probable to throw away such data which hold significant information. For the above reasons the complete case analysis may be practically impossible when handling medical data.

Thus, it is important in the scope of this project to have some techniques that estimate and replace missing values. Such techniques are generally split into two categories:

- univariate imputation
- multivariate imputation
- multiple imputation

Univariate imputation techniques generally replace a feature's missing values using only non-missing values of that feature. In such cases all missing values are replaced with the same value (usually mean, median or mode) depending on the type of the feature. Since all the missing values will be replaced with the same value, if the amount of missing data is high, such techniques have a great impact on the distributions of the features.

Multivariate imputation estimating the missing values of a feature using the entire set of available features. Such imputation techniques utilize classical ML regression, classification or clustering algorithms to estimate the missing values. For example, the missing values may be filled as a function of the k nearest neighbours or with the value of a centroid of a clustering technique. It is also possible to apply more sophisticated techniques that estimate each next missing feature from the already imputed features. Such algorithms iterate over the features and use a classical ML technique, such as Logistic Regression or Random Forests, to impute each feature. They then loop over the dataset and try to make better estimations of the missing values based on their previous iteration's results until a convergence criterion is met.

Finally, **multiple imputation** is the process of using an imputation algorithm that contains stochasticity (s.a. iterative imputation) to input the dataset multiple times. Each of the resulting datasets is analyzed so that the final analysis can yield better results since it accounts for the stochasticity introduced by the missing data.

Compared to Single imputation, ML approaches may lead to more realistic results as they take into consideration the available features when estimating the missing ones. In the context of the iHelp project that would mean that a patient's missing feature's value would be influenced by that patient's available data instead of relying only on the general population's characteristics. It is however very important not to ensure that the models do not interfere too much with the original data and does not introduce too much bias to the dataset.

In the figure below (Figure 3) we see the results of some implemented imputation algorithms when trying to estimate a feature of the dataset plotted against the patient's age. Mean imputation and Iterative imputation which did not converge, heavily influence the distribution of the data. However, KNN and MICE with RF Kernel manage to perform better.

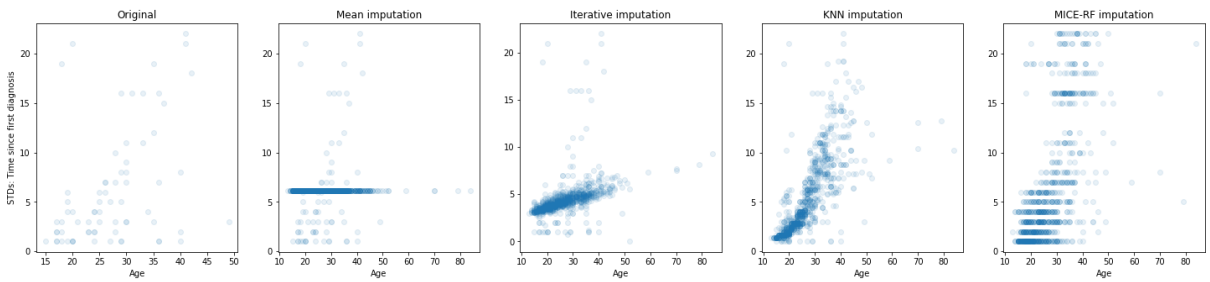

Figure 3: Plots of imputation technique results.

For practical reasons we split the implemented imputation methods according to the type of the data. For categorical data we need imputation to give discrete values and mode imputation and kNN with discrete values is more suited. For numeric data we can use mean, median, iterative imputation and kNN. The proposed imputation procedure in case it is needed is first dropping features with a high amount of missingness (e.g., 40%). The dataset will be split into train and test datasets, making sure that no patients are common between the datasets to avoid data leaking. It is also important to keep the prevalence of the categories in the two datasets as close as possible to have valid predictions. The remaining features will be imputed according to their type using the exact same method for both datasets. The imputed train dataset will be used to fit models to make risk predictions and the test dataset will be available for evaluation.

4 Risk factors importance

The cancer risk predictive analytics models will be used by HCPs in the iHelp pilots and therefore there is the need that they can understand the cause of a prediction. A clear requirement from the clinical pilot's perspective is to provide ways to understand the knowledge extracted from the models developed regarding relationships either contained in data or learned by the model.

In summary, there are model specific, and model agnostic interpretation methods based on whether the explanations are separated by the model itself. The former approach has the disadvantage that it binds the explanations to a specific model type, and it is difficult to switch to another to evaluate. The latter approach results in more flexibility since it can be applied to any model. Finally, there are models that provide a native way to explain their results/predictions. An overview of the various approaches follows.

4.1 Interpretable models

The Logistic Regression (LR) algorithm, a quite commonly used method in the medical AI area, belongs to the interpretable model's family. The interpretation of the weights in logistic regression is based on the odds function wrapped in the logarithm (log odds), where the odds function refers to the probability of event divided by probability of no event (i.e., probability to develop or not pancreatic cancer). Then we compare what happens when we increase one of the feature values by 1, but instead of looking at the difference, we look at the ratio of the two predictions:

$$\frac{odds_{x_j+1}}{odds} = exp(b_j)$$

In essence, the feature importance corresponds to the exponential of the feature weight. This way, a change in feature x by one unit increases the log odds ratio by the value of the corresponding weight. In the list that follows we summarize the interpretations of a LR model based on the feature type:

- Numerical feature: increasing the value of feature x_j by one unit results in a change in the estimated odds by a factor of $exp(b_j)$.
- Categorical feature (binary): changing the feature x_j from one category to the other changes the estimated odds by a factor of $exp(b_j)$.
- Categorical feature (multiple categories): one way to deal with multiple categories is to apply one-hot encoding so that the interpretation for each category is equivalent to the interpretation of binary features.

	0	Odds Ratio
6	Hormonal.Contraceptives	1.338926
1	Number.of.sexual.partners	1.271521
4	Smokes	1.250461
13	STDs..Time.since.first.diagnosis	1.235289
11	STDs..number.	1.112246
3	Num.of.pregnancies	1.019719
2	First.sexual.intercourse	0.998276
0	Age	0.990558
9	IUD..years.	0.961708
5	Smokes..years.	0.933887
7	Hormonal.Contraceptives..years.	0.900247
14	STDs..Time.since.last.diagnosis	0.837685

Figure 4: Example of Logistic Regression algorithm

The LR models suffer from the nonlinearity they impose in the relationship between features and outcome or where features are correlated. This can be addressed by tree-based models like DT algorithm. Tree based models work by splitting the data multiple times according to certain cut-off values in the features. During the splitting process different subsets of the dataset are created, resulting in each instance belonging to one subset. The predicted outcome in the edge nodes is based on the average outcome of the training data that belong to that node. DT takes a feature and determines which cut-off point minimizes the variance of y for a regression task.

In terms of interpretation the process is the following: by traversing the tree starting from the root node the edges reveal the subsets involved. At the end, the last node reached reveals the predicted outcome. All the edges are connected by “AND”. Part of a DT regression tree, with allowed depth set to 2, fitted on the cervical cancer dataset is shown in Figure 5. The overall importance of a feature in a DT can be computed by going through all the splits for which the feature was used and measuring how much it has reduced the variance compared to its parent node. The sum of all importance is scaled to 100 in a way that each importance can be interpreted as a share of the overall model importance.

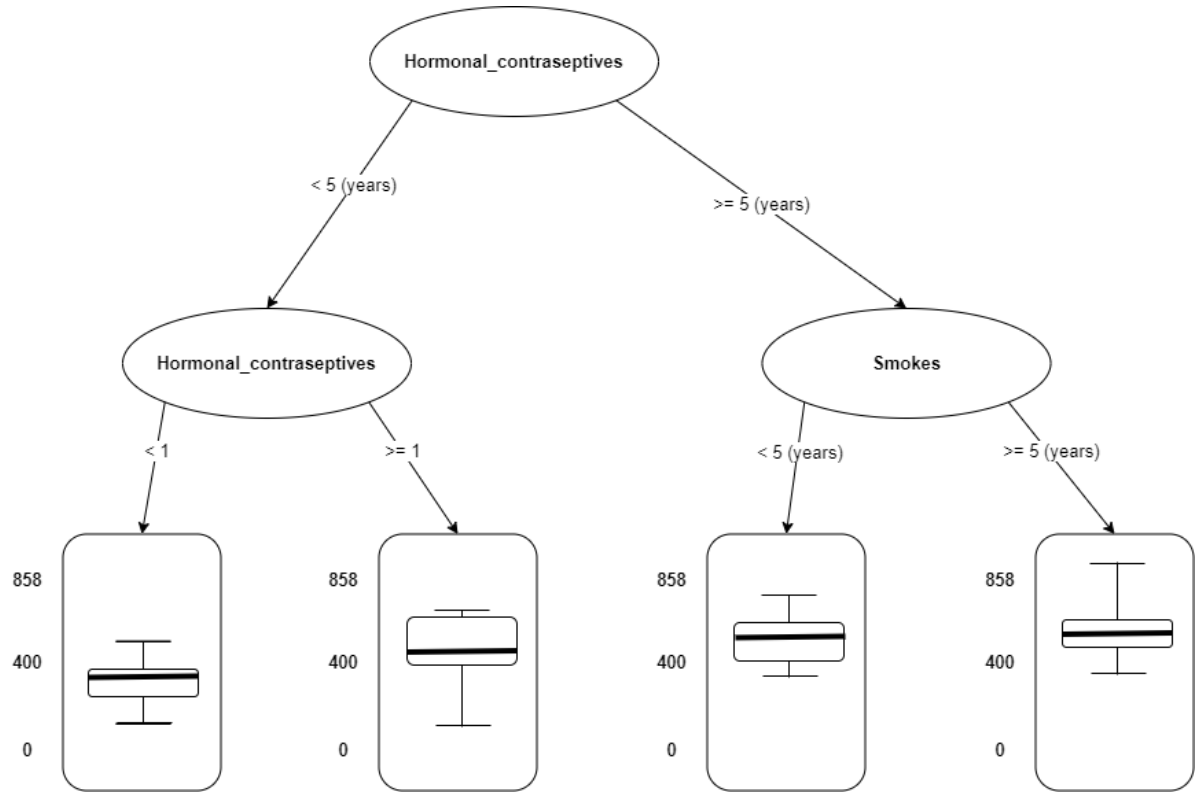

Figure 5: Regression tree fitted on the cervical dataset

4.2 Model agnostic interpretation methods

The model agnostic interpretation methods are independent of the underlying ML model, separating the explanations from the ML model itself. Model-agnostic interpretation methods can be further distinguished into local and global methods. Global methods describe how features affect the prediction on average while, in contrast, local methods aim to explain individual predictions.

The **SHAP (SHapley Additive exPlanations)** method is a quite popular model agnostic method that supports both local and global explanations and enhance the explainability of an AI model. It computes the contribution of each feature to the prediction, thus offering a better explanation and interpretability of the prediction itself.

In the context of global interpretability, the SHAP values can show how much each predictor contributes, either positively or negatively, to the target variable. This way it reveals not only the variable importance but also the positive or negative relationship between each variable with the target, as it is shown in Figure 6. In the case of local interpretability each observation gets its own SHAP values. Each of the participants in the iHelp clinical pilots can have a personalised interpretation of their case that explains why they received that risk prediction and the contributions of the predictors. It is important to present the feature importance on each individual case rather than across the entire population.

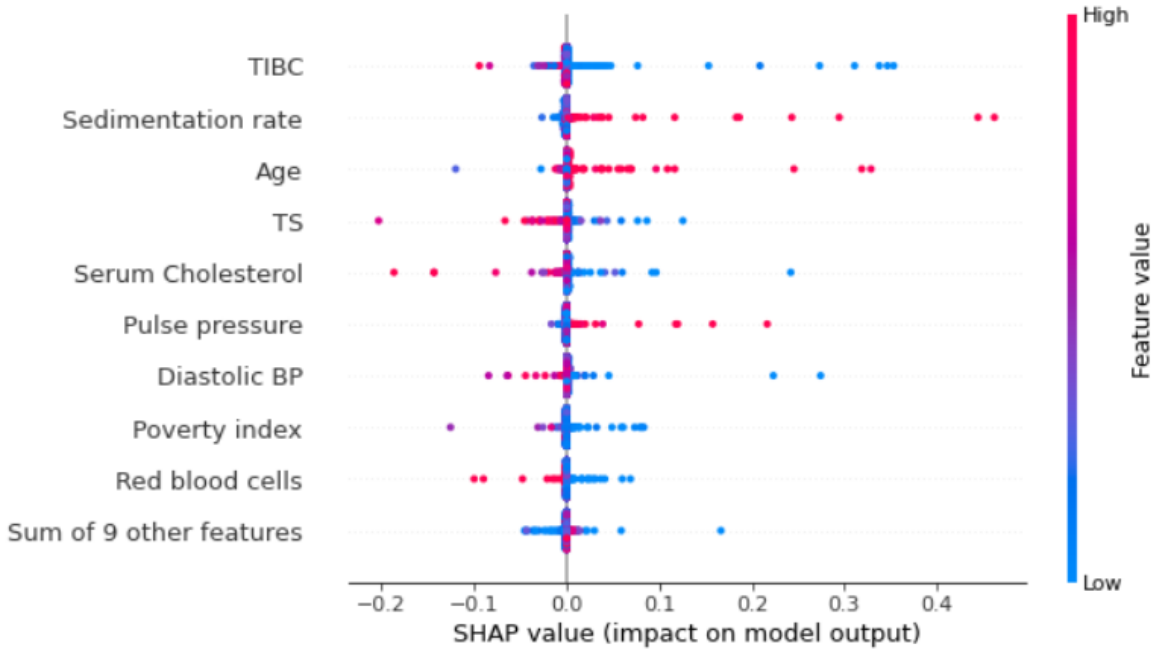

Figure 6: SHAP value (impact on model output)

4.3 Filter methods

Filter feature selection methods use statistical techniques to evaluate the relationship between each input variable and the target variable, and these scores are used as the basis to choose (filter) those input variables that will be used in the model. Filter methods pick up the intrinsic properties of the features measured via univariate statistics instead of cross-validation performance. These methods are faster and less computationally expensive than wrapper methods. When dealing with high-dimensional data, it is computationally cheaper to use filter methods. We have purposely left the feature extraction techniques like Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Linear Discriminant Analysis (LDA), etc., since they appear unsuitable for the context of this work where interpretability is most important for our models. These methods help to reduce the dimensionality of the data or reduce the number of variables while preserving the variance of the data.

There are several filter methods we can exploit and that depends on the type of the input/output features. Before we present the most commonly used below, we briefly introduce the Statistical Hypothesis Testing.

4.3.1 Statistical Hypothesis Testing

In statistics, when we wish to start asking questions about the data and interpret the results, we use statistical methods that provide a confidence or likelihood about the answers. In general, this class of methods is called statistical hypothesis testing, or significance tests. In statistics, a hypothesis test calculates some quantity under a given assumption. The result of the test allows us to interpret whether the assumption holds or whether the assumption has been violated.

The assumption of a statistical test is called the null hypothesis, or hypothesis zero (H0 for short). It is often called the default assumption, or the assumption that nothing has changed. A violation of the test's assumption is often called the first hypothesis, hypothesis one or H1 for short. H1 is really a short hand for some other hypothesis, as all we know is that the evidence suggests that the H0 can be rejected (BRO, 18).

- Hypothesis 0 (H0): Assumption of the test fails to be rejected.
- Hypothesis 1 (H1): Assumption of the test does not hold and is rejected at some level of significance.

Before we can reject or fail to reject the null hypothesis, we must interpret the result of the test.

The results of a statistical hypothesis test must be interpreted for us to start making claims. There are two common forms that a result from a statistical hypothesis test may take, and they must be interpreted in different ways. They are the p-value and critical values. We describe a finding as statistically significant by interpreting the p-value. For example, we may perform a Student's t-test on two data samples and find that it is unlikely that the samples have the same mean. We reject the null hypothesis that the samples have the same mean at a chosen level of statistical significance (or confidence). A statistical hypothesis test may return a value called p or the p-value. This is a quantity that we can use to interpret or quantify the result of the test and either reject or fail to reject the null hypothesis. This is done by comparing the p-value to a threshold value chosen beforehand called the significance level. The significance level is often referred to by the Greek lower case letter alpha (α). A common value used for alpha is 5% or 0.05. A smaller alpha value suggests a more robust interpretation of the result, such as 1% or 0.1%. The p-value is compared to the pre-chosen alpha value. A result is statistically significant when the p-value is less than or equal to alpha. This signifies a change was detected: that the default or null hypothesis can be rejected. The p-value is probabilistic. This means that when we interpret the result of a statistical test, we do not know what is true or false, only what is likely. Rejecting the null hypothesis means that there is sufficient statistical evidence that the null hypothesis does not look likely. Otherwise, it means that there is not sufficient statistical evidence to reject the null hypothesis.

Some tests do not return a p-value. Instead, they might return a test statistic value from a specific data distribution that can be interpreted in the context of critical values. A critical value is a value from the distribution of the test statistic after which point the result is significant and the null hypothesis can be rejected.

- Test Statistic < Critical Value: not significant result, fail to reject null hypothesis (H0).
- Test Statistic ≥ Critical Value: significant result, reject null hypothesis (H1).

It requires that you know the distribution of the test statistic and how to sample the distribution to retrieve the critical value. The p-value is calculated from the critical value. Again, the meaning of the result is similar in that the chosen significance level is a probabilistic decision on the rejection or failure of rejection of the base assumption of the test given the data. Results are presented in the same way as with a p-value, as either significance level or confidence level. For example, if a normality test was calculated and the test statistic was compared to the critical value at the 5% significance level, results could be stated as: The test found that the data sample was normal, failing to reject the null hypothesis at a 5% significance level or the test found that the data was normal, failing to reject the null hypothesis at a 95% confidence level.

Many statistical hypothesis tests return a p-value that is used to interpret the outcome of the test. Some tests do not return a p-value, requiring an alternative method for interpreting the calculated test statistic directly. A statistic calculated by a statistical hypothesis test can be interpreted using critical values from the distribution of the test statistic. Some examples of statistical hypothesis tests and their distributions from which critical values can be calculated are as follows:

- Z-Test: Gaussian distribution.
- Student's t-Test: Student's t-distribution.
- Chi-Squared Test: Chi-Squared distribution.
- ANOVA: F-distribution.

Critical values are also used when defining intervals for expected (or unexpected) observations in distributions. Calculating and using critical values may be appropriate when quantifying the uncertainty of estimated statistics or intervals such as confidence intervals and tolerance intervals. Note, a p-value can be calculated from a test statistic by retrieving the probability from the test statistics cumulative density function (CDF).

Being equipped with the basics on Hypothesis Testing, p-values and critical values, in the next sections we briefly present the most commonly used filter methods and classify them in categories that are based in the input and output type of data.

4.3.2 Numerical Input, Numerical Output

Correlation coefficients are used to measure how strong a relationship is between two variables such as blood pressure and cholesterol level. The rationale behind using correlation for feature selection is that the right variables are highly correlated with the target and at the same time, they should be uncorrelated among themselves (statisticshowto).

For that reason, if two variables are correlated, we can predict one from the other. Therefore, if two features are correlated, the model only needs just one of them (does not matter which one), as the second one does not add additional information and thus it can be discarded. We can also compute multiple correlation coefficients to check whether more than two variables are correlated to each other. This phenomenon is known as multicollinearity. An example is linear regression, where one of the offending correlated variables should be removed to improve the skill of the model. We may also be interested in the correlation between input variables with the output variable in order provide insight into which variables may or may not be relevant as input for developing a model. The structure of the relationship may be known, e.g., it may be linear, or we may have no idea whether a relationship exists between variables or what structure it may take. Depending on what is known about the relationship and the distribution of the variables, different correlation scores can be calculated.

Nevertheless, correlations only describe the relationship, they do not prove cause and effect. Correlation is a necessary, but not a sufficient condition for determining causality. Three requirements to infer a causal relationship are the following:

- A statistically significant relationship between the variables
- The causal variable occurred prior to the other variable
- There are no other factors that could account for the cause

Correlation studies do not meet the last requirement and may not meet the second requirement. However, not having a relationship does mean that one variable did not cause the other (researchbasics).

The Pearson's correlation coefficient can be used to summarize the strength of the linear relationship between two data samples. The Pearson's correlation coefficient is calculated as the covariance of the two

variables divided by the product of the standard deviation of each data sample. It is the normalization of the covariance between the two variables to give an interpretable score.

$$Pearson's\ correlation\ coefficient = cov(x, y)/stdev(x) \times stdev(y)$$

The use of mean and standard deviation in the calculation suggests the need for the two data samples to have a Gaussian or Gaussian-like distribution. The result of the calculation, the correlation coefficient can be interpreted to understand the relationship. The coefficient returns a value between -1 and 1 that represents the limits of correlation from a full negative correlation to a full positive correlation. A value of 0 means no correlation. The value must be interpreted, where often a value below -0.5 or above 0.5 indicates a notable correlation, and values below those values suggests a less notable correlation see for example Figure 7 below.

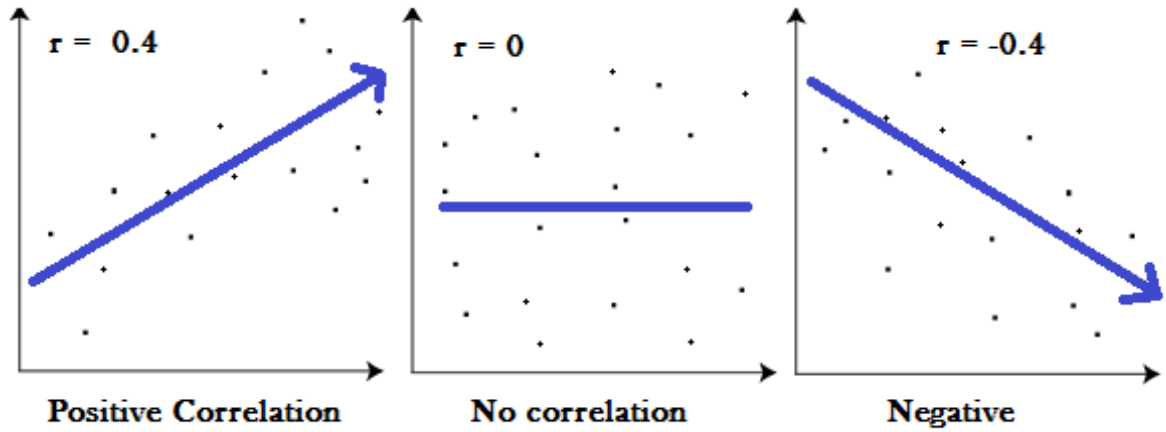

Figure 7: Graphs showing a correlation of -1, 0 and +1

The Pearson's correlation is a statistical hypothesis test that does assume that there is no relationship between the samples (null hypothesis). The p-value can be interpreted as follows:

- *p-value ≤ alpha*: significant result, reject null hypothesis, some relationship (H1).
- *p-value > alpha*: not significant result, fail to reject null hypothesis, no relationship (H0).

The Pearson's correlation coefficient can be used to evaluate the relationship between more than two variables. This can be done by calculating a matrix of the relationships between each pair of variables in the dataset. The result is a symmetric matrix called a correlation matrix with a value of 1.0 along the diagonal as each column always perfectly correlates with itself.

4.3.3 Numerical Input, Categorical Output

Analysis of variance (ANOVA) is an analysis tool used in statistics that splits an observed aggregate variability found inside a data set into two parts: systematic factors and random factors. The systematic factors have a statistical influence on the given data set, while the random factors do not. Analysts use the ANOVA test to determine the influence that independent variables have on the dependent variable in a regression study (ANOVA).

The t- and z-test methods developed in the 20th century were used for statistical analysis until 1918, when Ronald Fisher created the analysis of variance method (M., S., 18).

ANOVA is also called the Fisher analysis of variance, and it is the extension of the t- and z-tests. The term became well-known in 1925, after appearing in Fisher's book, "Statistical Methods for Research Workers" (FIS, 92). It was employed in experimental psychology and later expanded to subjects that were more complex.

The key points of the ANOVA approach are:

- Analysis of variance, or ANOVA, is a statistical method that separates observed variance data into different components to use for additional tests.
- A one-way ANOVA is used for three or more groups of data, to gain information about the relationship between the dependent and independent variables.
- If no true variance exists between the groups, the ANOVA's F-ratio should equal close to 1.

The Formula for ANOVA is:

$$F = \frac{MST}{MSE}$$

Where F=ANOVA coefficient, MST=Mean sum of squares due to treatment, MSE=Mean sum of squares due to error.

The ANOVA test is the initial step in analyzing factors that affect a given data set. Once the test is finished, an analyst performs additional testing on the methodical factors that measurably contribute to the data set's inconsistency. The analyst utilizes the ANOVA test results in an f-test to generate additional data that aligns with the proposed regression models.

The ANOVA test allows a comparison of more than two groups at the same time to determine whether a relationship exists between them. The result of the ANOVA formula, the F statistic (also called the F-ratio), allows for the analysis of multiple groups of data to determine the variability between samples and within samples.

If no real difference exists between the tested groups, which is called the null hypothesis, the result of the ANOVA's F-ratio statistic will be close to 1. The distribution of all possible values of the F statistic is the F-distribution. This is actually a group of distribution functions, with two characteristic numbers, called the numerator degrees of freedom and the denominator degrees of freedom.

We may have multiple data samples that are related or dependent in some way. For example, we may repeat the same measurements on a subject at different time periods. In this case, the samples will no longer be independent; instead, we will have multiple paired samples. We could repeat the pairwise Student's t-test multiple times. Alternately, we can use a single test to check if all of the samples have the same mean. A variation of the ANOVA test can be used, modified to test across more than 2 samples. This test is called the repeated measures ANOVA test.

The default assumption or null hypothesis is that all paired samples have the same mean, and therefore the same distribution. If the samples suggest that this is not the case, then the null hypothesis is rejected and one or more of the paired samples have a different mean.

- Fail to Reject H0: All paired sample distributions are equal.
- Reject H0: One or more paired sample distributions are not equal.

4.3.4 Categorical Input, Categorical Output

Information gain (relative entropy, or Kullback-Leibler divergence), in probability theory and information theory, is a measure of the difference between two probability distributions. It evaluates a feature X by measuring the amount of information gained with respect to the class (or group) variable Y, defined as follows:

$$I(X) = H\ (P(Y) - H(P(Y/X))$$

Specifically, it measures the difference the marginal distribution of observable Y assuming that it is independent of feature X(P(Y)) and the conditional distribution of Y assuming that is dependent of X (P(Y/X)). If X is not differentially expressed, Y will be independent of X, thus X will have small information gain value, and vice versa (L., L., V., 11).

Moreover, the **Chi-Squared test** is a statistical hypothesis test that assumes (the null hypothesis) that the observed frequencies for a categorical variable match the expected frequencies for the categorical variable. A categorical variable is a variable that may take on one of a set of labels. An example might be sex, which may be summarized as male or female. The variable is sex and the labels or factors of the variable are male and female in this case. We may wish to look at a summary of a categorical variable as it pertains to another categorical variable. For example, sex and interest, where interest may have the labels science, math, or art. We can collect observations from people collected with regard to these two categorical variables; for example:

```
Sex,     Interest
Male,    Art
Female,  Math
Male,    Science
Male,    Math
...
```

Figure 8: Example of a small categorical dataset

We can summarize the collected observations in a table with one variable corresponding to columns and another variable corresponding to rows. Each cell in the table corresponds to the count or frequency of observations that correspond to the row and column categories. Historically, a table summarization of two categorical variables in this form is called a contingency table, because the intent is to help determine whether one variable is contingent upon or depends upon the other variable. For example, does an interest in math or science depend on gender, or are they independent? This is challenging to determine from the table alone; instead, we can use a statistical method called the **Pearson's Chi-Squared test**. The test calculates a statistic that has a Chi-Squared distribution, named for the Greek lowercase letter chi (χ).

Given the Sex/Interest example above, the number of observations for a category (such as male and female) may or may not be the same. Nevertheless, we can calculate the expected frequency of observations in each Interest group and see whether the partitioning of interests by Sex results in similar or different frequencies. The Chi-Squared test does this for a contingency table, first calculating the expected frequencies for the groups, then determining whether the division of the groups, called the observed frequencies, matches the expected frequencies. The result of the test is a test statistic that has a Chi-Squared distribution and can be interpreted to reject or fail to reject the assumption or null hypothesis that the observed and expected frequencies are the same, that the variables are independent of each other.

The variables are considered independent if the observed and expected frequencies are similar, that the levels of the variables do not interact, are not dependent.

We can interpret the test statistic in the context of the Chi-Squared distribution with the requisite number of degrees of freedom as follows:

We can interpret the test statistic in the context of the Chi-Squared distribution with the requisite number of degrees of freedom as follows:

- Test Statistic ≥ Critical Value: significant result, reject null hypothesis, dependent (H1).
- Test Statistic < Critical Value: not significant result, fail to reject null hypothesis, independent (H0).

The degrees of freedom for the Chi-Squared distribution is calculated based on the size of the contingency table as:

$$degrees\ of\ freedom = (rows - 1) * (cols - 1)$$

In terms of a *p-value* and a chosen significance level (alpha), the test can be interpreted as follows:

- *p-value≤ alpha*: significant result, reject null hypothesis, dependent (H1).
- *p-value > alpha*: not signi_cant result, fail to reject null hypothesis, independent (H0).

For the test to be effective, at least five observations are required in each cell of the contingency table (BRO, 18).

Another approach is the **variance threshold**, which is a simple baseline approach to feature selection. It removes all features which variance does not meet some threshold i.e., features with not much useful information. By default, it removes all zero-variance features, i.e., features that have the same value in all samples. We assume that features with a higher variance may contain more useful information but note that we are not taking the relationship between feature variables or feature and target variables into account, which is one of the drawbacks of filter methods (ANALYTICS).

5 Risk Predictions

One of the main tasks of iHelp is to provide Personalised risk predictions for patients to discern the risk level of developing pancreatic cancer for each person individually. For that purpose, large retrospective datasets with clinical (primary) data will be made available from the pilots to train AI models that make these predictions. In the iHelp project, that EHR data will be combined with secondary data from questionnaires and wearables to form HHR data. In the context of this task we will mainly focus on using EHR, which is technically tabular data, fit models to them, make predictions about the patient's risk level and estimate the most important risk factors. This paragraph discusses the main concepts behind some of these models.

Logistic Regression (LR) is the simplest form of Supervised Learning for Classification. The core concept is that the input variables are multiplied by the weights of the model and the result passes by a nonlinear function, usually Sigmoid function. This function is given by the formula 1/(1+exp(-x)) and is bounded in the range (0,1). In this manner the algorithm maps the inputs to a probability space where values close to 1 for inputs belonging to the relevant class whereas values close to 0 don't. Using a threshold value t, the algorithm estimates that only observations whose output surpasses that value belong to the relevant class. By using a cost function to compare the algorithm's predictions to the ground truth an optimization algorithm (such as gradient descent) can be utilized to train the model by estimating the changes that need to be made to the weights of the model so as to minimize the cost function. Logistic regression is a simple, fast and easily explainable method. Due to its simplicity however, it tends to underfit since it cannot easily capture nonlinear relationships that exist in real world data.

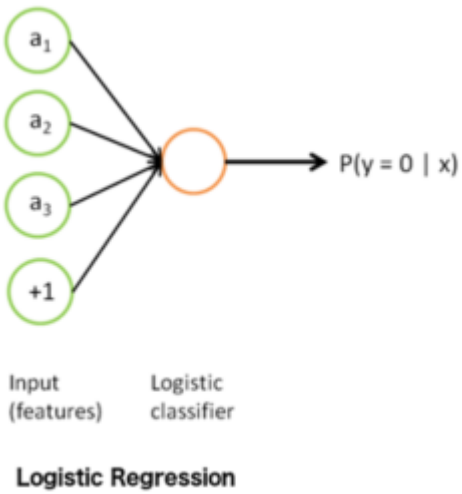

Figure 9: Schematic of a logistic regression model

Perhaps the simplest non-linear classifier is the DT. In this algorithm a tree model is built so that each node tests the value of a feature and the respective branches the outcome of that test. The upper nodes contain tests of the attributes with highest information gain and the tests usually have binary outcomes. The lowest nodes are called leaves and contain the final results with regard to the problem being addressed (e.g classes in a classification problem or values in imputation). Apart from being fast compared to more complex methods this algorithm is also highly explainable since it is quite straightforward. Finally, during the training process, it implements feature selection by evaluating each feature according to the information gain it

provides. However, training must start from scratch in case new data is provided and because of the linear boundaries that split data it is hard to estimate highly nonlinear data without overfitting.

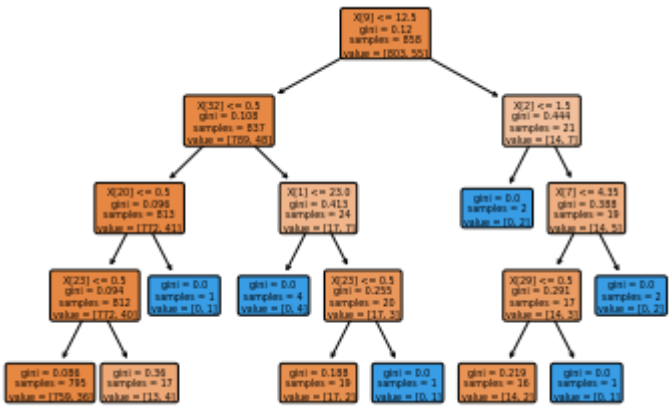

Figure 10: Schematic of a Decision Tree Classifier

Random Forest (RF) is a method based on DT. The core concept is that the algorithm generates many different DTs with various parameters and trains them on different random datasets that are subsets of the original. Such techniques that combine many base models are called ensemble techniques. When training is finished the algorithm has many trained DTs and uses all of them to generate the final answer to the problem. In the case of classification, the model outputs the most selected class whereas in Regression the output is the mean of all the DT's answers. RFs while slower than DTs do overcome it's overfitting tendency while also including the feature selection process that happens in each dt. However, they compromise on explainability and must also be retrained from scratch with all the available data.

Figure 11: Schematics of the first 5 trees of a Random Forest Classifier

Another ensemble technique based on DTs is **Gradient Boosting (GB)**. The difference with Random Forests is that GB generates the next tree by taking into consideration it's previous mistakes instead of randomly generating it. It does this by making a simple initial estimation and then fitting a DT on the residual values, meaning the distance of the estimation from the true values. The first estimation derives from the previous by adding the output value of the DT scaled by a learning rate. Each next DT is trained on the residuals (values in the case of regression or logits in the case of classification) of all of the previous until they don't influence the model anymore or a condition is reached. This way the model bases its final decisions on the initial naive estimation "corrected" by multiple models with each model learning from the mistakes of all of the previous models. This technique makes use of the non-linearity of the DT but also has a way to learn from its own mistakes as it is trained. However, like with Random Forests it compromises on explainability to achieve better performance.

Figure 12: Schematic of a Gradient boosting Classifier's trees

Neural Networks (NNs), and especially feedforward networks aka multilayer perceptrons, is a method that is similar to Logistic Regression. They also estimate a nonlinear classification function by calculating the weight values that minimize the cost function. The main difference with LR is that in the case of neural networks more layers are added between the inputs and the outputs. This way the model has the ability to estimate more complex nonlinear functions and achieve better results. However, as the layers increase so does the complexity of the model and the calculation of the weight values since in this case there is dependence between weights of different layers. For example, if two features have very large numerical differences the next layers may amplify these differences in a way that the model is not able to focus on the smaller features. For this reason, data fed to multilayer perceptrons usually go through a pre-processing step where scaling is applied. In addition, instead of a gradient descent an algorithm called backpropagation is used to calculate the correct changes that need to be made to the weights according to their contribution to the total error.

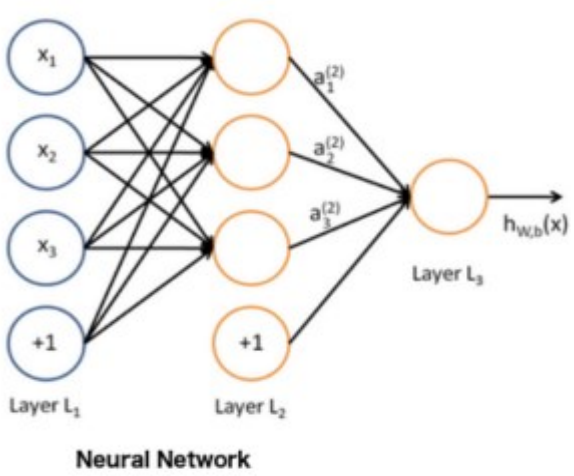

Figure 13: Schematic of a shallow NN

A by-product of the model's increased complexity is the increased potential of overfitting. It is thus very important to utilize techniques that avoid overfitting s.a. cross validation and weight regularization. It is also very important to control the size of the network since if it's too large apart from being prone to overfitting it will also require more computations. However, there is no general guideline to find the correct size of a NN and this is usually estimated during the tuning phase where many different models are trained to estimate the best hyperparameters. Despite their higher computational needs and slower training speeds, neural networks are extremely flexible and can be retrained according to will so long as the data are in the same format, which makes it a great candidate for a federated learning workflow. However, because of the higher complexity of the model it is sometimes hard to completely explain the knowledge it has aggregated.

5.1 SotA algorithms for pancreatic cancer prediction

Pancreatic Cancer is a heavy burden on public health. Patients suffering from pancreatic cancer have very low survival rates due to the difficulty of early identification. The bibliography regarding models that are based on EHR shows that SotA models reach Area Under the Curve (AUC) scores of about 0.7 - 0.85 (M., C., G., 16), (S., K., N., + 18), (M., H., N., + 19) for testing sets under specific circumstances (s.a. having new-onset diabetes). Many implementations use Logistic Regression (B., K., T., + 19), (T., C., 19), (M., R., B., + 21) as the algorithm with the exception of Hsieh et al (H., S., L., + 18) and Muhammad et al (M., H., N., + 19) who utilised artificial neural networks. Muhammad et al claim to have investigated a variety of other cancer types such as lung cancer, prostate cancer, endometrial cancer, prostate cancer and colorectal cancer using Artificial Neural Network (ANN), Support Vector Machines (SVM), DT, Naive Bayes, LDA and LR. Their results indicated that in general ANNs achieve the best performance compared to other algorithms in terms of sensitivity, specificity and AUC. Since we have found no study where a plethora of models have been implemented and compared it is interesting to verify these indications as well as test SotA models such as GB which have been shown to achieve great performance on tabular data.

5.2 Hyperparameter optimization

Hyperparameter optimization (aka tuning) is the process of choosing a good set of hyperparameters for a learning algorithm. Hyperparameters are quantities or entities such as the learning rate, loss function, optimization algorithm of a LR/NN, max tree depth/leaf nodes of a DT, number of trees of a RF/GB or number/type/amount of neurons of a NN layer. In contrast to the parameters of the models (weights, biases, etc) they do not derive from the training data but are used to influence the learning process.

Simple methods for hyperparameter tuning include grid search where a grid of possible values is created and in each iteration, a combination of them is tried in a specific order, random search where the same process happens but with a random order. Since hyperparameter tuning is a process where the aim is to minimize the cost function of a trained model by changing some hyperparameters it can also be thought of as an optimization problem. Thus, classical optimization algorithms like Bayesian optimization can be applied. For that purpose, algorithms have been specifically designed to perform tuning such as Hyperband, Population based training, Bayesian optimization combined with Hyperband (BOHB).

Different algorithms have different parameters to tune and of course may be implemented in different frameworks. The training processes of different models of the same algorithm are usually irrelevant to one another and can be done in parallel. Therefore, it is desired to have a tuning framework that supports a variety of frameworks (i.e., sklearn, tensorflow, pytorch, xgboost) and can also parallelize the training of different candidate models to speed up the tuning process so as to completely utilize the computer's resources. Ray Tune apart from providing the above also implements SotA algorithms (i.e., Asynchronous Successive Halving (ASHA), Population based training, Baysean Optimisation & Hyperband (BOHB) for choosing hyperparameters that use early stopping in case a model underperforms to search for a more promising model. Also, since it is based on Ray, a distributed computing system it has the ability to be configured to utilize more machines by doing distributed optimization. Finally, it also implements cross validation so that the models are evaluated in a way that handles overfitting.

It should be noted that this final option will not be used in the case of sensitive hospital data except if there is a need for further speedup of the training process. In that case it will be using computational resources of the relevant institution as nodes so that no data leaves the hospital's premises.

5.3 Model Evaluation

When evaluating a ML model, it is important to use a method that does not give out a biased score to be sure that the model will be able to perform just as well in the deployment phase as it did when trained. Therefore, it is important to actually split the available data and hold out a sample to use for an unbiased estimate after the training process is finished. It is also important to be able to evaluate the model while it is being tuned on the training dataset to get an early indication of the model's potential. For the above reasons the datasets are usually split into Train, Validation and Test sets. If there is need for more data in the training set then the train-validation split can occur during training time in a k-fold cross-validation manned where after the train-test split, the test set is split into k groups and training happens by holding out each next group as a temporary validation set in a round robin fashion. That way the model actually trains on all the data but in each iteration it tries to use the remaining k-1 groups to estimate the held out group correctly.

Another thing that is important to take care of when splitting the datasets is to ensure that each dataset should have enough examples for each class to be estimated. This can be easily achieved by splitting the initial dataset according to each class, then splitting the classes amongst themselves and finally merging the respective datasets.

When evaluating datasets with multiple classes or unbalanced classes accuracy can be very misleading as a metric. That is because if the model focuses only on specific classes with high prevalence it may achieve good accuracy even when ignoring all other classes. In such cases it is important to use other evaluation measures instead.

Such measures are sensitivity and specificity which measure the percent of correct positive estimations out of all positive estimations and of correct negative estimations out of all negative estimations for each feature. A good binary classifier will have both high sensitivity and high specificity meaning that it will not only correctly estimate that an observation belongs to a class, but it will also not mistakenly put observations that belong in other classes in that class. To get a better understanding of a model's performance it is common practice to measure the sensitivity against the specificity of the model while varying its decision threshold. By calculating these values for thresholds from 0 to 1 we get the Receiver Operating Characteristic (ROC) curve which is one of the best indicators of the model's performance. Since what we want from a model is to produce more True Positive (TP) than False Positive (FP) classifications, a good model will have an ROC curve that leans towards the top left corner of the plot.

In the figure below (Figure 14) we can see such a curve plotted for three different datasets. It is clearly visible that this model performs better on the train set than on the other sets. Such a thing is an indication of overfitting, and the tuning process should start anew. It is common practice to quantify the performance of the datasets by measuring the AUC and use it as a metric to be able to compare models without having to plot their curves.

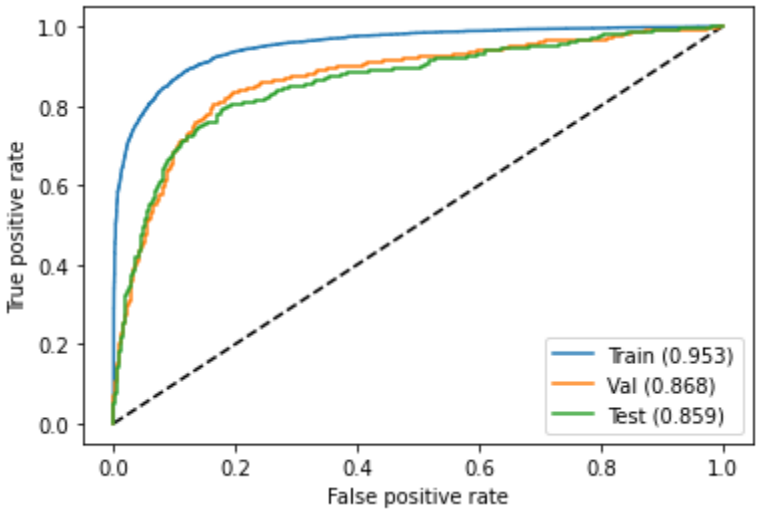

Figure 14: ROC curves of a model for the Train, Validation and Test set[1]

Another useful measure for a clinician would be the **Positive Predictive Value (PPV)** and **Negative Predictive Value (NPV)**. These measure the probability that the model is right if it predicts a class and the probability that the model is right if predicts an observation doesn't belong to a class.

[1] This model has overfit since there is a large gap between the performance on the training set and the validation and test sets.

6 Connection with the Pilots: State of the art

The aim of this Section is to gather all the necessary information about the available datasets that will be used under this Task, for the initial implementation of the AI algorithms. More specifically, each pilot partner gives a description of the already shared datasets and the ones that will be shared in the future. Also, the desirable outcomes described above, aim to shape the methodology to be developed for the implementation of the AI models. Finally, it is of major importance to have a description for possible pre-existing algorithms (out of iHelp project), that could contribute to the building of the basis of the models in iHelp. All these issues are addressed in the following sub-chapters by each partner, separately.

6.1 UNIMAN pilot

6.1.1 Description of available datasets & known risk factors

The University of Manchester (UNIMAN) pilot has access to various datasets (mostly publicly available) as well as newly generated data. This will allow us to employ a range of conventional and emerging approaches to build risk prediction models for pancreatic cancer. Publicly available datasets include

- UK Biobank
- Lifelines
- The National Health and Nutrition Examination Survey (NHANES)
- Alberta Tomorrow Project (ATP)

Based on the previous work of UNIMAN researchers in the area of cancer research, known risk factors for pancreatic cancer are cigarette smoking, heavy alcohol consumption, periodontal disease, increased BMI, low physical activity, increased consumption of red/processed meat and dairy products, VitD (controversial), chronic pancreatitis, hepatitis B infection, SLE, Diabetes, Helicobacter Pylori infection, usage of PPI (Proton-pump inhibitor), chemicals and heavy metals: beta-naphthylamine, benzidine, pesticides, asbestos, benzene, and chlorinated hydrocarbon, etc.

6.1.2 Description of case study & its desirable outcomes

UNIMAN pilot has already developed a risk prediction model using the UK data. The model applied relative risk and prevalence of risk in the general population then we compute lifetime risk for each decade of age group. By using this approach, the UNIMAN pilot is able to demonstrate an individual 10-year risk compared to general population at the same age group. The risk prediction developed using established estimated risk from literature review, so the UNIMAN pilot purpose is to update the list of the risk factors and also its prevalence.

6.1.3 Predictive analytics for risk of developing PC based on genomic and epigenetic markers

The UNIMAN team has developed the Risk Estimation for Lifestyle Enhancement Combined Test (REFLECT) model and Risk Estimation for Additional Cancer Testing (REACT) model for cancer risk prediction. The REFLECT model is a lifestyle-related risk prediction tool via Web-based interface for 11 different cancers, which includes pancreatic cancer. The method is derived from YourDiseaseRisk, which were developed for the US population and which has been adapted. For constructing the algorithms to calculate individual risk, the prevalence of factors related to each cancer type in the UK population and 10-year estimated cancer-specific risk based on UK figures were applied. An individual's cancer risk is compared to the average risk in

the population for someone of the same age and sex. Results are presented in three categories: lower than average, average, and higher than average. In our pilot study, all the consented participants will complete REFLECT assessment initially.

The REACT model is a symptom-based cancer risk assessment tool offered through a Web-based interface in a community setting. REACT is a tool to assist people in deciding whether or not they need to consult their general practitioner (GP) about potentially cancerous symptoms. This tool assesses the symptoms of 5 major cancers affecting people in the UK (i.e., bowel cancer, breast, ovarian, lung, and prostate cancer). Risk estimation in REACT is based on the Risk Assessment Tools (RATs) model, which is utilizing a representative record of symptoms. Each clinical symptom listed in the original RAT model (e.g. constipation or dyspnea, terms easily understood by GPs but necessarily not by a layperson) was translated into layperson language to be used in the REACT questionnaire. To raise awareness of symptoms that may be indicative of pancreatic cancer amongst the public. We are also building the symptom-related risk prediction model for pancreatic cancer based on our previous REACT model.

In the iHelp project, the UNIMAN pilot will collect data on biomarkers including genetic and methylation markers. These markers will be used to assess their incremental risk based on their genetic footprint and risk incur from their lifestyle which impose on their epigenetic markers. These markers can be used to enhance the predictive value for pancreatic cancer.

Genetic markers- Single nucleotide polymorphisms (SNPs) are one of the common types of individuals' genetic variants, which have been used to predict the risk of developing various diseases, including coronary heart disease, diabetes, and cancers. Various susceptible loci for pancreatic cancer have been identified by Genome-Wide Association Study (GWAS). The previous studies have built polygenic risk scores (PRS) from SNPs, and the result revealed that the PRS was associated with pancreatic ductal adenocarcinoma risk.

Methylation markers- DNA methylation (DNAm) is one of the epigenetic changes identified by transferring a methyl group to the C-5 on the cytosine. DNA methylation is generally referring as CpG methylation, which occurs on cytosines followed by guanine residues (CpG). DNAm has emerged to be a surrogate for biological ages; previous researchers have proposed different epigenetic clocks to estimate age-related diseases. Some well-known epigenetic clocks include Hannum's, Horvath's and Levine's clocks. Epigenetic clocks are a potential measurement to predict future morbidity and mortality outcomes. It is to be noted that DNAm was highly correlated to lifestyle factors, including dietary, physical activity, obesity, and smoking. Unlike SNPs, DNAm can be reversed based on lifestyle changes. In sum, genetic predisposition is constituted since birth; in contrast, DNAm could be modulated by lifestyle change.

6.1.3.1 Methodology

The UNIMAN pilot is a community-based pilot. To enable pancreatic cancer risk mitigation, we adopted a 2-step approach. The first step is to identify individuals at above average risk of pancreatic cancer as compared to risk from population at the same age group (5 years) using the REFLECT risk assessment application. The second step is then to assess their genetic and epigenetic risks in these at-risk individuals.

Development of the REFLECT model

To develop a UK version of a cancer risk prediction model for pancreatic cancer, the data required to develop these models are: 1) the identified list of risk factors for inclusion; 2) point estimates of the relative risk for each risk factor; and 3) population prevalence for each of the exposures. To be able to compare

individual risk to the population, further information on cancer incidence by 10-year age bands are required. We used these data to compute risk score. Further details can be found in (L., U., C., + 17).

Development of the REACT web tool

Using the questions and risk estimates from the pre-existing cancer "RAT" (Risk Assessment Tool) and the original research from which this was derived, the web tool was developed and subsequently extensively modified as a result of discussion in the steering group. The web tool is in the form of an on-line questionnaire where users work through the questionnaire which asks questions about specific symptoms, and obtain a risk estimation and explanation and signposting at the end.

Below, we describe the methodology we will use to analyse the biomarkers.

Genomic biomarker:

First, we will perform QC, imputation, and annotation of genotyped data. Next, we will calculate the polygenetic score (PRS) from processed genotyped data by summarising the common variants (risk alleles) to evaluate the risk for genetic predisposition. We will use the classic PRS formula, where βk is the log odds ratio (OR) for SNPk from the previous GWAS, SNPk is the allele dosage for SNPk, and n is the included SNPs number in this study.

$$PRS = \beta_1 \times SNP_1 + \beta_2 \times SNP_2 + \cdots + \beta_k \times SNP_k \ldots + \beta_n \times SNP_n$$

We will assign a threshold of scores from published works to derive score stratification into tertile.

Epigenomic biomarkers:

In terms of DNAm data analysis, UoM will use methylation QC data to compute Hannum, Horvath and Levine clocks and the available calculator provided by Hovarth's group. Data can be uploaded as anonymised data, and methylation age can then be computed for all three clocks using this platform. We will investigate epigenetic age acceleration by computing the residuals from regressing DNAm age on chronological age and blood cell composition. It is to be noted that epigenetic age acceleration is independent of chronological age and blood cell composition.

6.1.3.2 Results

Here we describe results that can be generated from biomarkers.

Genomic biomarkers:

The PRS score will be stratified into quartile based on non-cancer PRS score's value from the UK population with the lowest quartile will be used as the reference group. At the individual level, we will inform our participants which quartile they belong to and what this means to their risk.

Epigenomic biomarkers:

We will summarise the biological age from the different Epigenetic clocks. At the individual level, we will provide our participants their biological age before and after 6 months' prevention group study.

6.2 FPG pilot

6.2.1 Description of available datasets & known risk factors

The Agostino Gemelli University Policlinic (FPG) pilot has access to a dataset related to patients affected by pancreatic cancer that underwent surgery followed by adjuvant (chemo-)radiotherapy. According to the fact that the real-world data and the patient reported outcomes that will be analysed in the perspective phase are not currently collected in a database, the main role of this retrospective dataset will be the opportunity to have a comparison cohort for the subgroup of patients that will be enrolled with similar clinic-pathological features and in the same setting (adjuvant (chemo-)radiotherapy). Patients of perspective phase will include also other radiotherapy treatment setting; in that case the retrospective analysis will be limited according to the fact that patients are structurally not comparable.

6.2.2 Description of case study & its desirable outcomes

Primary aim of the study is to use AI-based analytic techniques to analyse the role of real-world data (RWD) in prediction of toxicities, quality of life (QoL), and survival outcomes in patients affected by pancreatic cancer. Any further ongoing elaboration of acquired data, realised, and selected also using AI algorithm, will be proposed to clinicians adjunctively to the data currently acquired in good clinical practice.

Secondary aims of the study are:

- Analysis and development of a dedicated RWD infrastructure for patients affected by pancreatic cancer;
- Identify possible recommendation for a better treatment acceptation and tolerability;
- The evaluation of patients and clinicians experience of the IoT and the dedicated application in terms of better understanding, awareness, usefulness, and effectiveness;
- To allow data interoperability;
- Sharing anonymised data or model with iHelp consortium platform.

6.2.3 Previous AI algorithms, relevant outcomes & risk factors identified

To the best of FPG pilot knowledge there are no AI-based models based on RWD able to predict radiotherapy toxicity, but several papers show that acute and late toxicity can occur in intensity modulated radiotherapy (B., G., B., 15), (J., Y., P., + 19).

6.2.4 Development of toxicities risk prediction model

The toxicities risk prediction model aims to address the following toxicity related outcomes:

- Acute upper GI
  - classes: G1, G2, G3, G4, G5
- Late upper GI
  - classes: G1, G2, G3, G4, G5
- Need for hospitalization
  - classes: Yes, No
- Haematological toxicity

- classes: anaemia, neutropenia, thrombocytopenia

6.2.4.1 Pre-processing

The FPG pilot has provided a dataset with sample values. That dataset needs to go through a pre-processing pipeline in order to be transformed as appropriate input in the models that will be described in the next sections. The pre-processing techniques that are applied on the sample dataset provided are the following:

Outlier detection:

An outlier is an individual point of data that is distant from other points in the dataset. It is an anomaly in the dataset that may be caused by a range of errors in capturing, processing or manipulating data.

Therefore, identifying and dealing with outliers is an integral part of working with data, since in the training data they may skew the model, lowering its accuracy and overall effectiveness. There are several approaches for outlier detection. In this dataset we aimed for the detection of both univariate and multivariate outliers. The former is a case with an extreme value that falls outside the expected population values for a single variable while the latter is a combination of unusual scores on at least two variables. For the univariate outliers we used Z-score, one of the most commonly used tools in determining outliers. Z-score is just the number of standard deviations away from the mean that a certain data point is. For the multivariate outliers we have employed the Mahalanobis distance which measures the number of standard deviations that an observation has from the mean of a distribution.

Imputation:

Briefly, this method is the process of replacing missing data with substituted values. We remind the reader that we have thoroughly analysed this technique in Section 3.2. For both the categorical and the numerical missing values in the dataset we have used the KNN algorithm with rounding and without respectively. To this specific dataset the KNN outperformed single value algorithms such as mean, median and most frequent value imputation algorithms (where we substituted all the missing values with the respective mean/median/most frequent value of each specific feature) as well as multiple value imputation algorithms such as iterative and random forest imputation algorithms. Figure 15 summarizes the imputation methods evaluated and the selected ones.

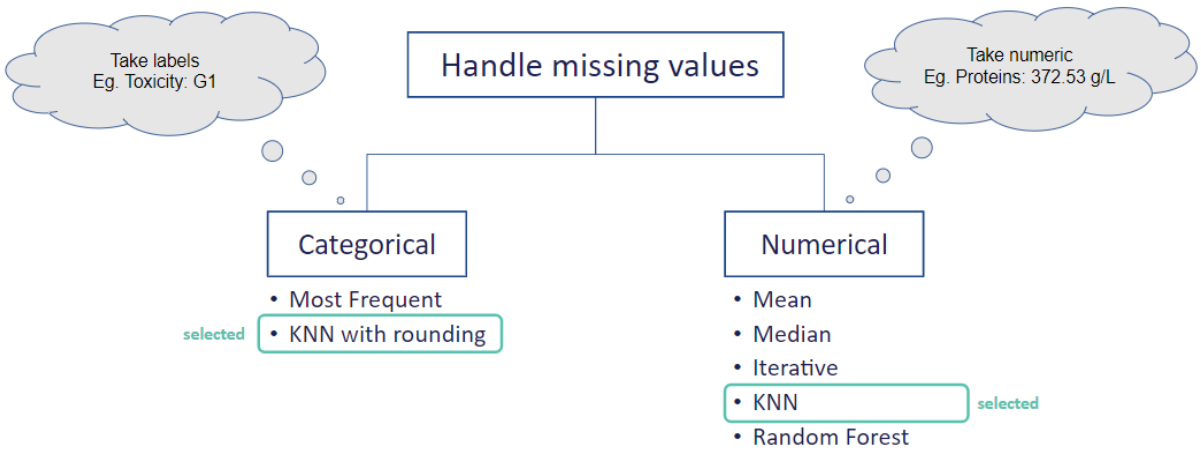

Figure 15: Selected imputation methods

Under/Over sampling:

We plan to apply under sampling since electronic health records (EHR) are sensitive data. Under sampling techniques, remove examples from the training dataset that belong to the majority class in order to better balance the class distribution.

Eliminating unnecessary features:

Another potential pre-processing step that can be applied in order to improve model's performance is to identify the most correlated features and remove them. Moreover, features that we know in advance that they do not provide any additional information with regard to the target prediction are also removed. In addition, features corresponding to measurement units are not taken into consideration.

Creation of embeddings:

Many features in the dataset are categorical and therefore there was the need to apply one hot encoding technique. For instance, the 'overall survival' feature contained two possible values, either 'dead' or 'alive'.

6.2.4.2 Baseline models / Initial approach

The problem of predicting the toxicities of interest is considered a classification problem where the predictor classes refer to the grade of the toxicity in question. For example, both acute upper GI and late upper GI toxicities belong to one of the following five grades: G1, G2, G3, G4 and G5, indicating the severity of the corresponding toxicity.

The retrospective dataset consists of base features and blood exams related features. Specifically, the base features include information about CA19-9 marker diagnosis, TNM staging, radiotherapy related features like settings and techniques applied and more while blood exams include measurements about proteins like haemoglobin, platelets, white_blood_cells, neutrophils, lymphocytes, proteins, urea_nitrogen, c-reactive_protein, creatinine, albumin, alt, ast, ggt, alkaline_phosphatase, lactic_dehydrogenase, direct_bilirubin, indirect_bilirubin, total_bilirubin, inr and ca19_9.

A set of classification algorithms has been implemented in order to predict the toxicities of interest. An overview of the ML models implemented is depicted in the list that follows:

- Logistic Regression
- Decision Tree
- Random Forest
- Gradient Boosting
- LightGBM
- XGBoost
- MLP

Specifically, the Decision Tree classifier is based on the sklearn.tree.DecisionTreeClassifier[2] implementation and is used to perform multi-class classification on the grades (G1, G2, G3, G4 and G5) of the acute upper GI and late upper GI toxicities that denote the severity of the toxicity in question. The idea is to create a

[2] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

model that can predict the grade of toxicity by learning decision rules inferred from the base clinical (phenotypic, disease specific and treatment specific) and the blood exams related features.

More advanced methods are also developed in order to improve the generalizability and robustness of the baseline models. For this purpose, a set of ensemble methods is built and is evaluated in order to find the optimal model, as it is described in section 6.2.4.3. In general, ensemble methods combine the predictions of several base estimators. They are distinguished in averaging methods, where several estimators are built independently and their predictions are then averaged, and the boosting methods where base estimators are built sequentially and then one tries to reduce the bias of the combined estimator. In the context of averaging methods, a Random Forest classifier is developed based on the sklearn.ensemble.RandomForestClassifier[3] implementation, while in the context of boosting methods a Gradient Boosting classifier is developed based on the sklearn.ensemble.GradientBoostingClassifier[4], a XGBoost classifier based on the XGBoost[5] implementation and a LightGBM classifier based on the LightGBM[6] implementation.

Moreover, a multilayer perceptron (MLP) model is developed. An MLP model is actually a fully connected class of feedforward artificial neural network (ANN). A dedicated MLP model is developed to predict the risk of developing each of the toxicities of interest. Specifically, Figure 16 depicts the internal architecture of the MLP for predicting the need for hospitalization.

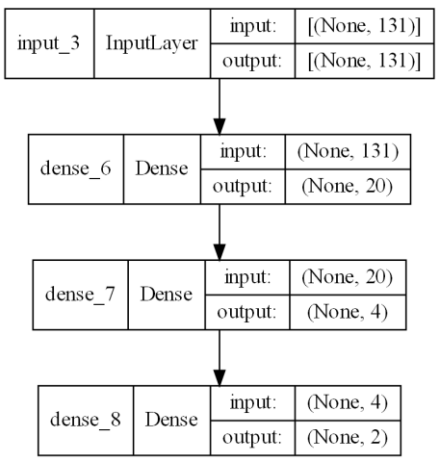

Figure 16: MLP architecture for toxicity: Need for hospitalization

The resulted number of trainable parameters in this MLP model is 2,734. The trainable parameters refer to the weights of the neurons in the model.

Figure 17 depicts the internal architecture of the MLP for predicting the acute upper GI and late upper GI toxicity grades. The number of trainable parameters in this MLP model is 2,749.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html#sklearn.ensemble.GradientBoostingClassifier
[5] https://xgboost.readthedocs.io/en/stable/python/python_intro.html
[6] https://lightgbm.readthedocs.io/en/v3.3.2/Python-Intro.html

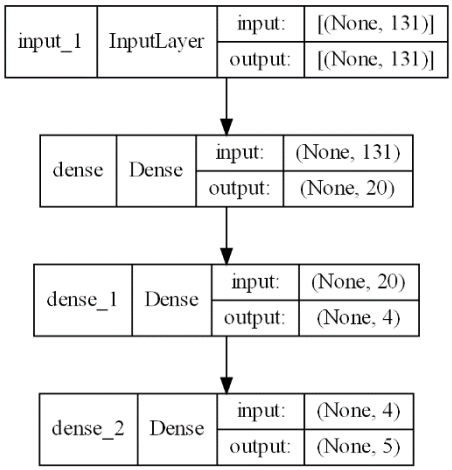

Figure 17: MLP architecture for toxicities: Acute upper GI and Late upper GI

6.2.4.3 Evaluate models and find optimal

The algorithms mentioned in the previous section are evaluated in a simple fashion. The initial dataset is split randomly into two different datasets one used for training and one for the evaluation. During the training process the algorithms are exposed only to the training set. After the training process is over for each of the resulting models and for each desired label, we calculate the macro-average metrics on the separate test dataset (precision, recall, f1-score) and compare them for each model. Currently we consider the model with the highest macro-average F1-score as the best model for the dataset.

6.3 HDM pilot

6.3.1 Description of available datasets & known risk factors

The Hospital de Dénia-MarinaSalud (HDM) pilot has full access to the EMR of Marina Alta region, this dataset contains about 300.000 EMR's from 2009 (when the hospital started) until now. The most important dimensions of information are Person, Encounter, Orders, Clinical Event, Laboratory, Radiology, Diagnosis and Procedures.

Focusing on the Pancreatic Cancer patients, the pilot can provide around 240 patients (from 2009 to 2020) and we can retrieve administrative and clinical information (we expect to have pathological reports with the tumor details of all them).

6.3.2 Description of case study & its desirable outcomes

The case study is based on the remote patient lifestyle monitoring of a group of patients diagnosed with pancreatic cancer together with patients who theoretically have some theoretical risk factor.

An initial assessment of behaviour will be carried out on this group of global patients with respect to the clinical and behavioural variables to be analysed. Once this previous analysis has been carried out, they will be included in the monitoring program that will provide us with information about this behaviour that, together with a review with the type of analytical tests that the clinical experts decide, will create the basis of the HHR that will be used to apply the AI algorithms that they will assess. if there is a correlation between the behaviours and the evolution of the patients.

The expected outcomes of the studies are basically three:

- Identify and demonstrate risk factors in pancreatic cancer.
- Identify behaviours that help mitigate risk.
- Identify behaviours that help to improve the disease or suffering of the patient.

6.3.3 Previous AI algorithms, relevant outcomes & risk factors identified

The HDM pilot is not aware of studies that identify how behaviours affect the evolution of the disease, hence the interest in developing the case study.

6.4 MUP pilot

6.4.1 Description of available datasets & known risk factors

The case study related to Medical University Plovdiv (MUP) pilot is a hospital-based case-control study. A total of 899 participants are recruited, including 299 pathologically verified pancreatic cancer cases and 600 controls selected from the same hospital. Cases and control are 1:2 matched by gender and age, in addition, we will use multiple controls. 300 controls will recruit from other cancer patients, 300 controls will recruit from other disease patients except for cancer

In addition, MUP collects various types of data as per the Table 1 below:

Table 1. Various types of data collected by MUP

Symptom data	Comorbidities	Morbidity history	Family history	Physical	Laboratory tests	Gene mutation factor
1. Weakness, helplessness 2. Rapid fatigue 3. Doesn't feel well, feels sick 4. Feeling of bloating in the abdomen 5. Feeling of heat or fever 6. Fever 7. Frequent episodes of constipation 8. Frequent diarrhea 9. Difficulty eating 10. Nausea 11. Loss of appetite 12. Hypersensitivity in the upper abdomen 13. Pain in the upper and / or middle part of the abdomen 14. Abdominal pain that radiates to the back	1. Chronic pancreatitis 2. Acute pancreatitis 3. Cysts of the pancreas 4. Type 1 diabetes mellitus 5. Insulin-resistant diabetes mellitus 6. Choledocholithiasis 7. Chronic cholecystitis 8. Newly diagnosed diabetes mellitus 9. Chronic gastroduodenitis 10. Chronic hepatitis type B, C 11. Liver cirrhosis 12. Gastric ulcer 13. Obesity	1. Acute pancreatitis 2. Acute cholecysitis 3. Acute gastroduodenitis 4. Gastric ulcer 5. Gastric surgery 6. Operations of the pancreas 7. Radiation of the abdominal area 8. Hepatitis type B, C 9. No data on specific past diseases 10. Other past diseases	1. Carcinoma of the pancreas 2. Malignant diseases 3. Lynch syndrome 4. Familial atypical malignant melanoma 5. Genetic mutation of BRCA2 6. No data on specific family morbidity history	1. Pale, dry skin 2. Reduced skin turgor 3. Painless jaundice - skin and visible mucous membranes 4. Palpation pain paraumbilically or in the upper quadrants of the abdomen 5. Abdominal pain, which intensifies on palpation 6. Palpable tumor formation in the abdominal area	1. Amylase 2. Lipase 3. SGOT 4. SGPT 5. GGT 6. Amount of fat in the faecal samples 7. Pancreatic elastase in faecal samples 8. Blood sugar 9. Glycated hemoglobin 10. CA19-9 11. CEA 12. No particular changes into the laboratory tests' data	1. TP53 2. KDMA mutations 3. FOXA2/3 4. PDX1 5. MNX1 6. KRAS 7. NR5A2 8. RBPJL 9. NEUROD1 10. NKX2-2 11. No changes

15. Abdominal pain, which intensifies when lying down and leaning back 16. Abdominal pain, which intensifies after eating 17. Weight loss that is undesirable 18. Changes in stool color - discoloration 19. Darkening of the urine 20. Itchy skin 21. Painless jaundice - skin and visible mucous membranes 22. No complaints 23. Pain in musculoskeletal system 24. Painfull or impaired movements 25. Others	14. No concomitant diseases 15. Morbus Hypertonicus 16. Pyelonephritis chronica 17. Arthrosis 18. Polyneuropathy 19. Other comorbidities			7. Obesity 8. Asictes 9. Edemas 10. No specific physical data 11. Impaired movements 12. Musculoskeletal pains 13. Other clinical symptoms		

6.4.2 Desirable outcomes and model approach

Machine learning algorithms separate into two main tasks, supervised (Classification/Regression) and unsupervised (Clustering). To solve Classification problems labels for each observation (patient) are needed. In our case these labels would be the risk level of the patient. This means that to utilize supervised algorithms we will need an extra column per row in the dataset that represents the risk level per patient (low, medium, high). Since this label is not present we can only use clustering algorithms. Clustering (see Section 2.2 for details) is an unsupervised machine learning method of identifying and grouping similar data points in larger datasets without concern for the specific outcome.

The goal of the MUP pilot is to predict the risk of developing pancreatic cancer in low, medium and high-risk patients. The doctors (based on the bibliography) verified that patients, who appear to have the same symptoms as the cases, belong to the high risk cluster, whereas those who do not have similarities with the cases are considered to be low or medium risk. Initially, we have implemented a k-means algorithm for clustering either on all the features or on combinations of those. The resulting clusters indicated that even the cases do not end up in a specific cluster i.e., the high risk cluster. Therefore, in order to have all the cases in the same cluster we implemented a Constrained K-means algorithm that was introduced by Wagstaff et al. (W., C., R., + 2001). This algorithm creates a graph that links all the cases and forces them to belong to the same cluster. The Constrained K-means results in a cluster containing all the cases including some control group patients with similar profiles as the cases. Then, we label each patient according to which cluster they belong to and we train an XGBoost classifier to explain/interpret with the SHAP method (see Section 4.2 for details) the features of each cluster (F., Y., G., + 2021).

6.4.3 Exploratory Data Analysis

Figure 18 reveals that most cancer patients are in the age range of 50 to 80. On the other hand, we have almost no one over 80 years old, while at least a few cases are in their early 50s. Specifically, the blue bars depict the cases, and the orange bars depict the control groups. Most cases belong to the group age 61-70 years, followed by the group of 51-60 years, and then the group age of 71-80 years comes next. However, very few cases belong below 40 years and 41-50 years and only one case included in the group 81-90 years and above 90 years.

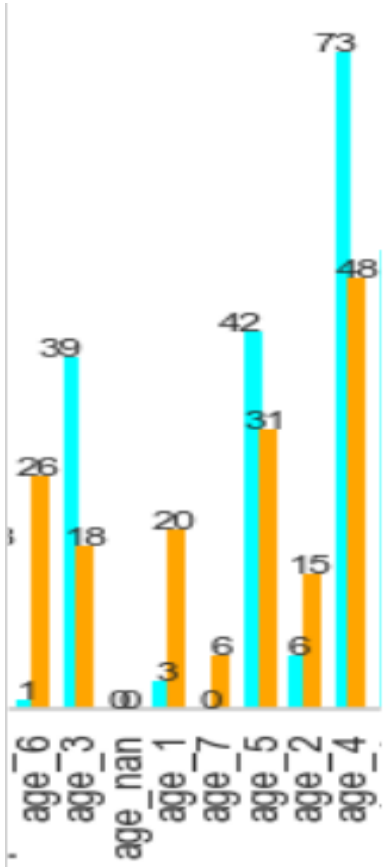

Figure 18: Distribution of patients in age

There are some features, which do not reveal important information, as they only appear in a few number of cases. For instance, there are not many reports on family history for the cases, see Figure 18, and when they contain information it regards only one group of illnesses, malignant diseases. In addition, only 5 cases have morbidity history (see Figure 19), two cases with morbidity history 'acute cholecystitis' and three cases with morbidity history 'Hepatitis type B, C'. Therefore, it is very difficult for a model to learn from so little information. Figure 19 depicts the distribution of patients with regard to the family history feature. Specifically, the blue bars depict the cases and the orange bars depict the control groups. The cases denoted only family history with id 2 that connect with malignant diseases.

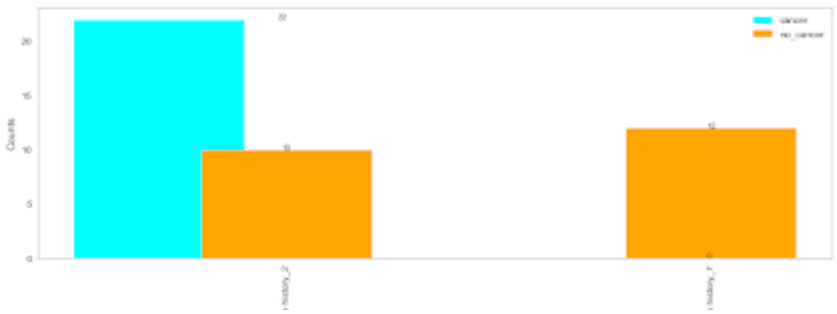

Figure 19: Distribution of patients in family history

In Figure 20 it is shown that two cases have morbidity history 2 (acute cholecystitis) and three cases have morbidity history 8 (Hepatitis type B, C).

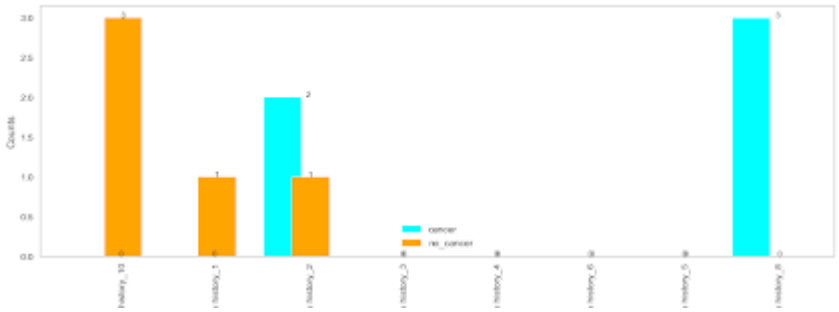

Figure 20: Distribution of patients in morbidity history

Cancer males and females have a similar distribution (see Figure 21), so we do not gain any information on whether cancer affects one gender versus the other. It seems that the women are similar to men in the cases group.

Figure 21: Distribution of patients by gender

6.4.4 Development of the risk predictor models

6.4.4.1 Pre-processing

The dataset described briefly in the previous section needed to be pre-processed in order to be given as input in the models that will be described in the next sections. We used the following methods to get the final dataset:

Undersampling: The dataset was unbalanced with control groups (controls) overcoming the patients with pancreatic cancer (cases), counting 595 and 226 observations, respectively. This can drive the AI model to focus more on the population with the most observations. This problem has already been addressed in (F., H., H., +2020) and the proposed solution is to utilize undersampling or oversampling techniques. We decided to apply undersampling since electronic health records (EHR) are sensitive data and an

oversampling technique would add noise and could lead to incorrect conclusions. Undersampling techniques remove examples from the training dataset that belong to the majority class in order to better balance the class distribution.

Eliminating unnecessary features: The next step was to identify those features i.e., columns that introduced noise and did not add any value in the dataset and therefore there was no need to keep the additional columns that failed to show any variation in the dataset, since they would not help our model learn different patterns. Specifically, the features that were removed are:

- Diagnosis: this feature was provided only for those patients already suffering from pancreatic cancer (cases) and not for the rest (control group). Therefore, we had to remove it from the dataset since it would be of no account for the models to learn a feature that will not appear in the control group in order to make predictions.
- Genotyping: this feature had the same value for all the patients, and thus it would not introduce extra information in the models.

Eliminating unnecessary values: Many features contained values that reflect to the 'No data' category. This characteristic of the dataset made the algorithm put a lot of emphasis on the 'No data' features when clustering the users. This was problematic since the algorithm considered the 'No data' features to be most important in deciding the risk level of the patient. To avoid this issue, we modified the pre-processing process for the data to remove the 'No data' columns after applying one hot encoding method.

Creation of embeddings: A particularity in the MUP dataset is that almost all features correspond to indexes that map to category labels. An AI model cannot understand category labels and therefore it needs numerical values to make it easier for the machine to process the data. To apply any type of algorithm to the data, we need to convert the categorical data to numbers. To achieve this, one-hot encoding, also known as "1-of-N" encoding (meaning that the vector is composed of a single one and a number of zeros), is one way as it converts categorical variables into binary vectors.

6.4.4.2 Baseline models / Initial approach

We used two clustering algorithms, K-means and Hierarchical clustering (HC) and evaluated the generated clusters using the silhouette score metric. The silhouette score algorithm determines the similarity of each object in a cluster with other objects in the same cluster compared to the objects in other clusters. The silhouette coefficient quantifies this. The value of the silhouette coefficient is between [-1, 1] and a score of 1 shows that the data point is very compact within the cluster to which it belongs and far away from the other clusters. On the other hand, a value of -1 has the opposite meaning, while values near 0 denote overlapping clusters. The K-means achieved 0.288 silhouette score, besides the HC, which was 0.214. Therefore, K-means was better than HC, so we analyzed the k-means.

We propose the clustering method to group the patients into 3 clusters, i.e., low, medium and high-risk cancer. Data from 914 patients with non-cancer or cancer were used, and after pre-processing (undersampling and so on), we had 224 of respectively. Table 2 shows the number of patients grouped in the same cluster for each baseline algorithm, K-means and HC. For instance, cluster 0 has 224 patients with cancer and only 52 non-cancer patients (see Table 2.1).

Table 2. Distribution of all the patient per cluster

Table 2.1. k-means

Cluster	Label	Number of patients
0	Cancer	224
	No cancer	52
1	Cancer	0
	No cancer	99
2	Cancer	0
	No cancer	73

Table 2.2 Hierarchical Clustering

Cluster	Label	Number of patients
0	Cancer	224
	No cancer	150
1	Cancer	0
	No cancer	47
2	Cancer	0
	No cancer	27

We interpret from Table 2.1 that the clusters are separated as follows:

- Potential high-risk patients→ Cluster 0
- Potential medium-risk patients → Cluster 1
- Potential low-risk patients → Cluster 2

We assume that a new patient A with similar characteristics (e.g., symptoms) to patient B, who has already been diagnosed with cancer, will be at high risk of developing pancreatic cancer. Figure 22 to Figure 24 visualize the most significant features per cluster that characterize the patients belonging to it.

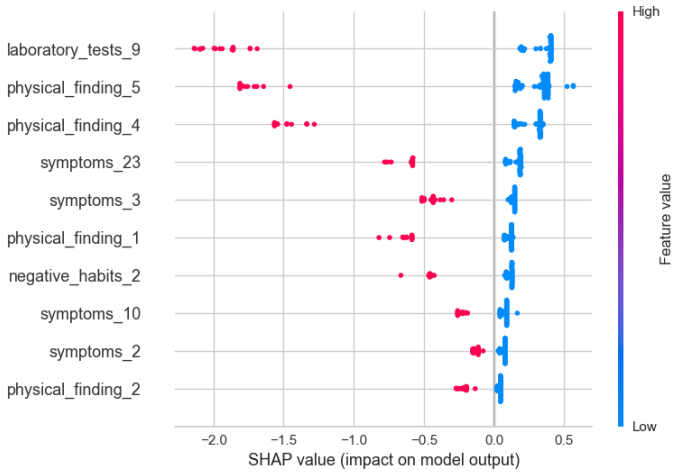

Figure 22: Cluster 0 (Potential high-risk cluster)

The patients are placed in cluster 0 because of the absence of the following features: laboratory_tests_9 (Glycated hemoglobin), physical_finding_5 (Abdominal pain, which intensifies on palpation), physical_finding_4 (Palpation pain paraumbilically or in the upper quadrants of the abdomen).

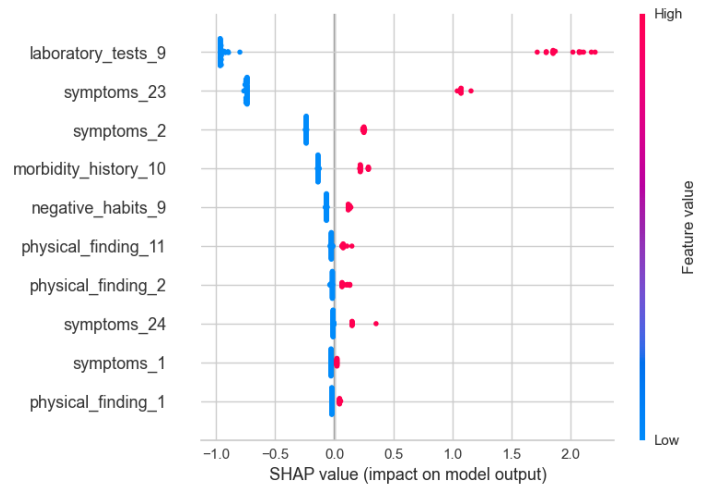

Figure 23: Cluster 1 (Potential medium risk)

The patients placed in cluster 1 appear to have in common the laboratory_tests_9 (Glycated Hemoglobin) and symptoms 23 (Pain in musculoskeletal system), symptoms_2 (Rapid fatigue) and morbidity_history_10 (Other past diseases).

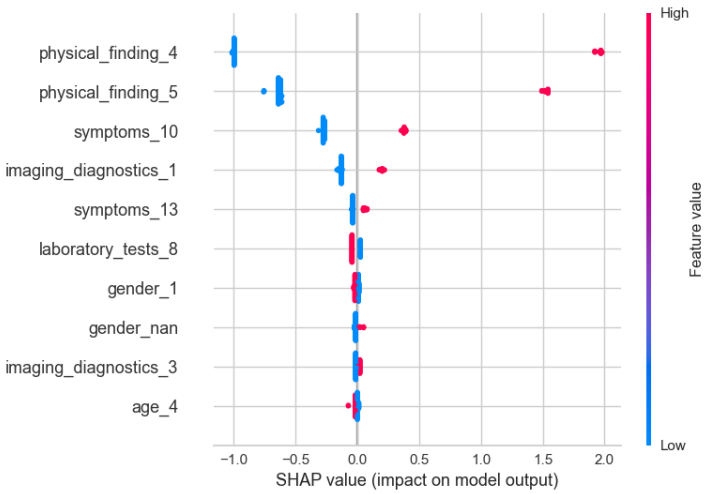

Figure 24: Cluster 2 (Potential low risk)

The patients placed in cluster 2 appear to have in common the physical_findings_4 (Palpation pain paraumbilically or in the upper quadrants of the abdomen) and physical_findings_5 (Abdominal pain, which intensifies on palpation) and symptoms_10 (Nausea).

We summarize all the information of the above figures in Table 3.

Table 3. Significant features per cluster

Not laboratory_tests_9 (Glycated hemoglobin)	Laboratory_tests_9 (Glycated hemoglobin)	physical_findings_4 (Palpation pain paraumbilically or in the upper quadrants of the abdomen)
Not physical_finding_5 (Abdominal pain, which intensifies on palpation)	symptoms 23 (Pain in musculoskeletal system)	Physical_findings_5 (Abdominal pain, which intensifies on palpation)

Not physical_finding_4 (Palpation pain paraumbilically or in the upper quadrants of the abdomen)	Symptoms_2 (Rapid fatigue)	Symptoms_10 (Nausea)
	Morbidity_history_10 (Other past diseases)	Imaging diagnostics_1 (Transabdominal ultrasound)

6.4.4.3 All features model with constrained clustering

After we applied the undersampling technique and having in mind the small number of patients, we observed that all the cases were in the same cluster, and thus we annotated them as high-risk. However, the question that arises is, what happens if we have a massive number of patients? For instance, before applying pre-processing, the cases were separated into two clusters (cluster 1 and 2) on baseline models, as seen in Table 4. As it turned out after the discussion with the clinicians, it was not right.

Table 4. Clustering before undersampling

Cluster	Label	Number of patients
0	Cancer	0
	No cancer	34
1	Cancer	51
	No cancer	27
2	Cancer	37
	No cancer	17

Therefore, we decided to use Constrained K-means clustering based on the Python implementation[7]. Constrained K-means achieved grouping all the cases in the same cluster by setting it as a constraint. Specifically, we produced a graph connecting all cases together and set this as a constraint in the clustering procedure to ensure that all cases are grouped in the same cluster. Table 5 shows how the clusters are separated with Constrained K-means.

Table 5. Patient distribution per cluster based on Constrained K-means

Cluster	Label	Total per label
0	Cancer	0
	No cancer	74
1	Cancer	224
	No cancer	52

[7] https://github.com/Behrouz-Babaki/COP-Kmeans

2	Cancer	0
	No cancer	98

Based on Table 5, the high-risk patients are in cluster 1. In Figure 25 to Figure 27 the important features of each cluster are depicted using SHAP plots.

It is difficult for the cluster 1 (see Figure 26) to highlight essential characteristics, as it mainly highlights the features that are not included in this cluster. Therefore, we focused on feature-based models to overcome this limitation.

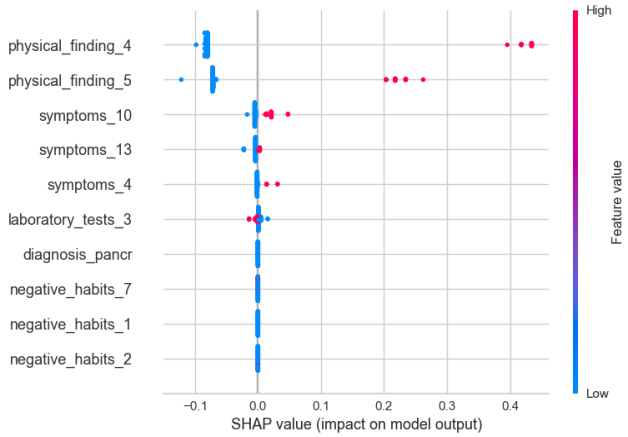

Figure 25: Cluster 0

The patients placed in cluster 0 appear to have in common the physical_findings_4 (Palpation pain paraumbilically or in the upper quadrants of the abdomen) and physical_findings_5 (Abdominal pain, which intensifies on palpation) and symptoms_10 (Nausea). On the other hand, On the other hand, cluster 1 reveals only what features are not included in the cluster.

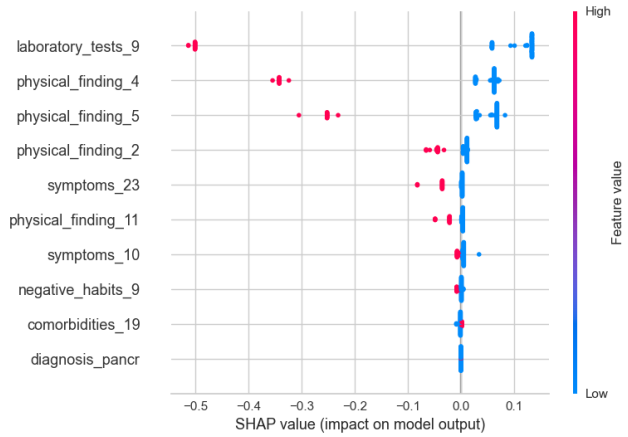

Figure 26: Cluster 1

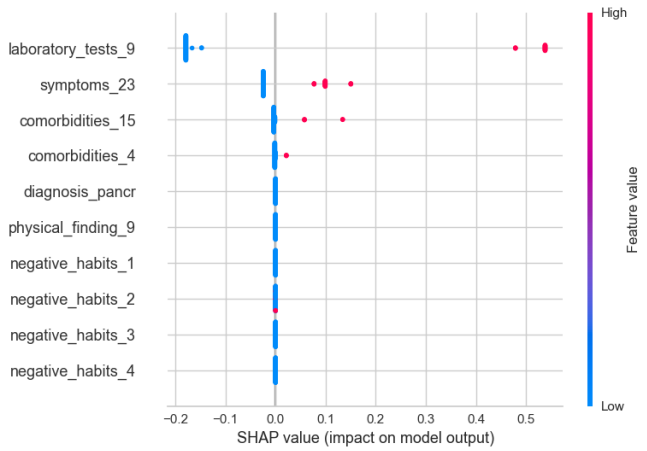

Figure 27: Cluster 2

The patients placed in cluster 2 appear to have in common the laboratory_tests_9 (Glycated hemoglobin), symptoms_23 (Pain in musculoskeletal system) and symptoms_15 (Abdominal pain, which intensifies after eating).

6.4.4.4 Symptom base model with constrained clustering

The approach for predicting the risk of the patients in the sequel, is to apply constrained clustering for each feature (symptoms, comorbidity etc.) separately, for several reasons. First and foremost, each feature's range of appearance varies so if the analysis for each feature was not carried out separately, the model would focus on the feature with the most prevalence. To make it clear, if control groups have a lot of family history disease, the clustering algorithm identifies this feature as significant and ignores all the other features. Therefore, it is preferred to study each feature separately.

Consequently, this section describes the results of clustering the data based only on the symptom features. In Figure 28 the number of cases and control groups that appear for each symptom are depicted.

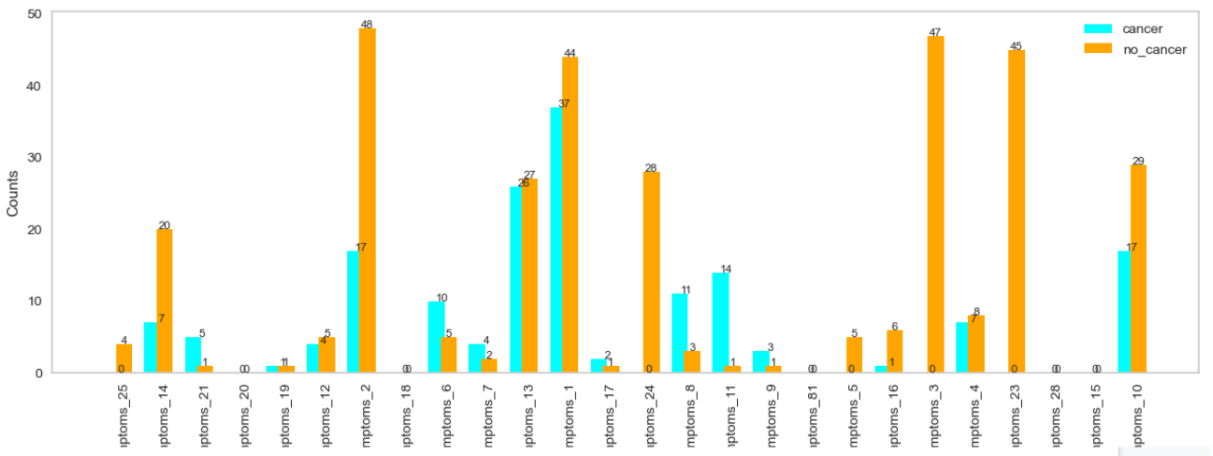

Figure 28: Distribution of cases and control groups per symptom

Constrained K-means clustering was applied to solve the problem with the cases belonging to multiple clusters. In Table 6 the corresponding number of patients per cluster can be found and in Figure 29 to Figure 34 the clusters are illustrated along with the corresponding SHAP plots.

Table 6. Constrained K-means clustering for symptoms

Cluster	Label	Total per label
0	Cancer	88
	No cancer	33
1	Cancer	0
	No cancer	44
2	Cancer	0
	No cancer	11

We observe that cluster 0 contains all the cases but also some control groups with the same symptoms and thus, it is considered as the high-risk cluster. The potential medium and low clusters are clusters 1 and 2 respectively (see Figure 29 to Figure 31).

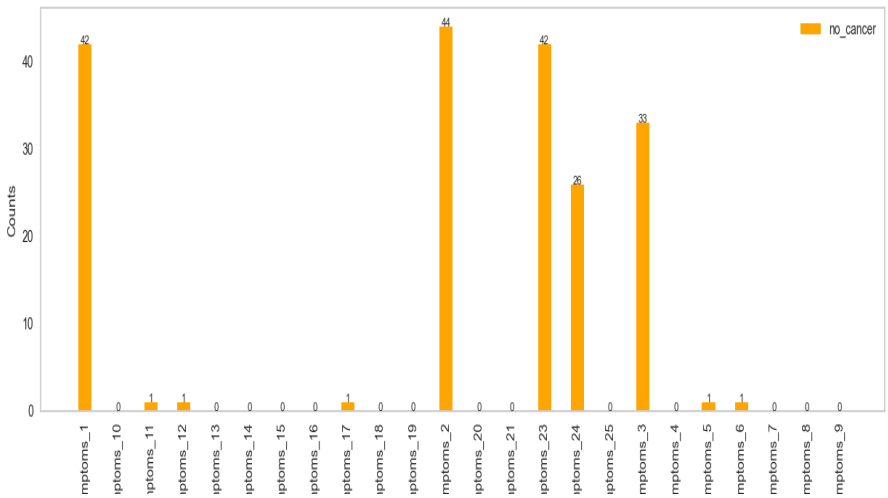

Figure 29: Cluster 1 Distribution plot

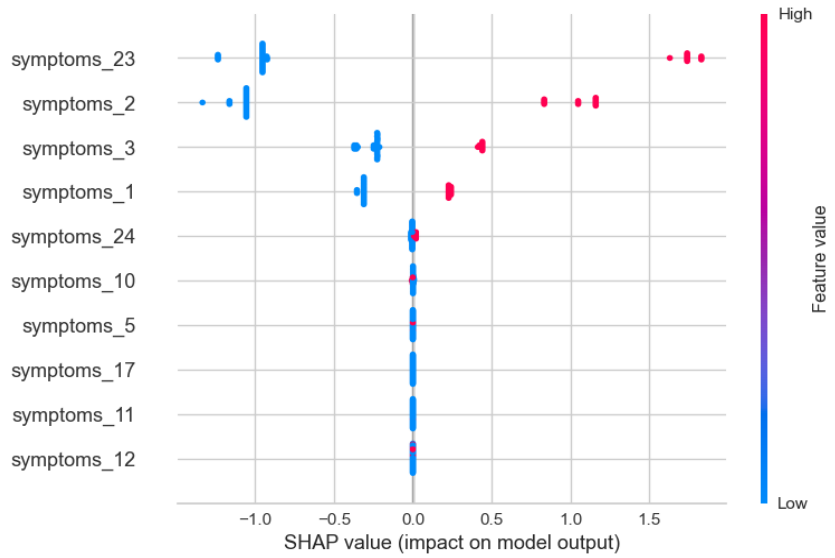

Figure 30: Cluster 1 is the potential medium risk cluster

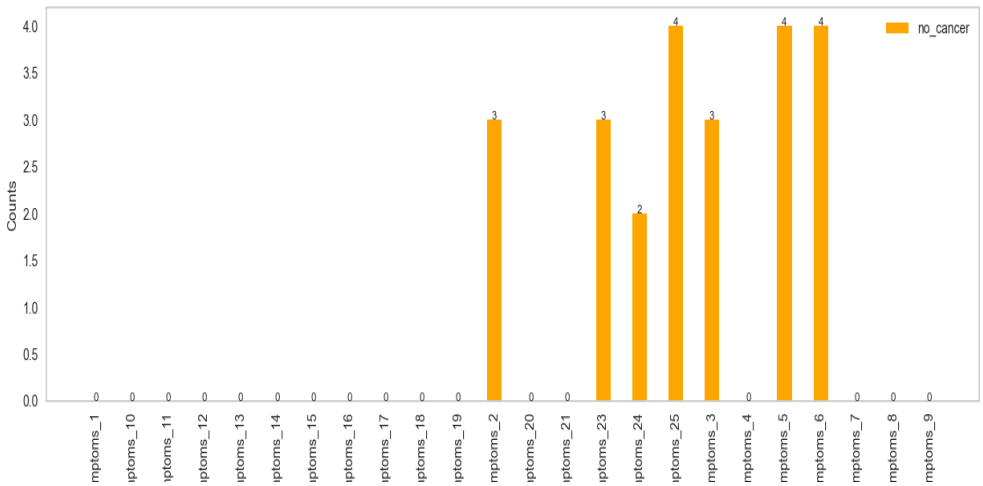

Figure 31: Cluster 2 Distribution plot

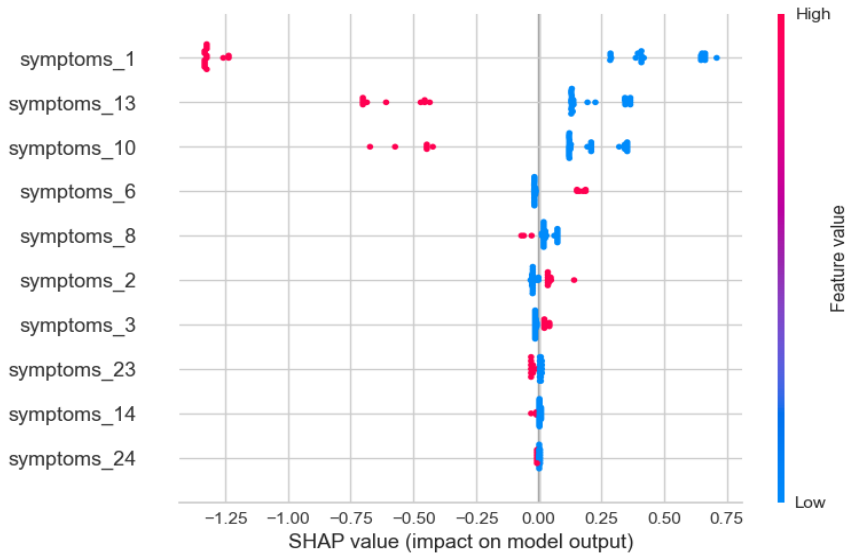

Figure 32: Cluster 2 is the potential low risk cluster

The most significant symptoms in high risk patients based on Figure 30 are those including "Pain in the musculoskeletal system" (symptom 23), "Rapid fatigue" (symptom 2) and "Does not feel well/sick" (symptom 3), whereas they include the "Pain in the upper and/or middle part of the abdomen" (symptom 13), "Nausea" (symptom 10) and "Weakness/helplessness" (symptom 1). The results appear to be meaningful from a clustering perspective, as 37/88 cases contain "Weakness/helplessness" (symptom 1). It is justifiable that the "Weakness/helplessness" symptom is not a dominant characteristic of high risk patients (sixty positions in SHAP with a small effect value), as 44 / 88 = 50% of the control groups have it, otherwise half of the patients without cancer would be annotated as high risk which is not valid based on the clinicians feedback. Hence, when describing a cluster, we should consider a combination of symptom values jointly and not independently.

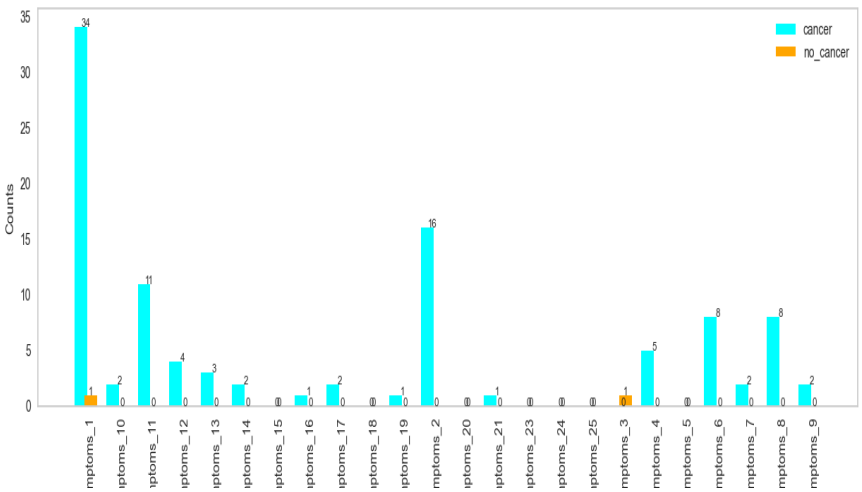

Figure 33: Cluster 0 Distribution plot

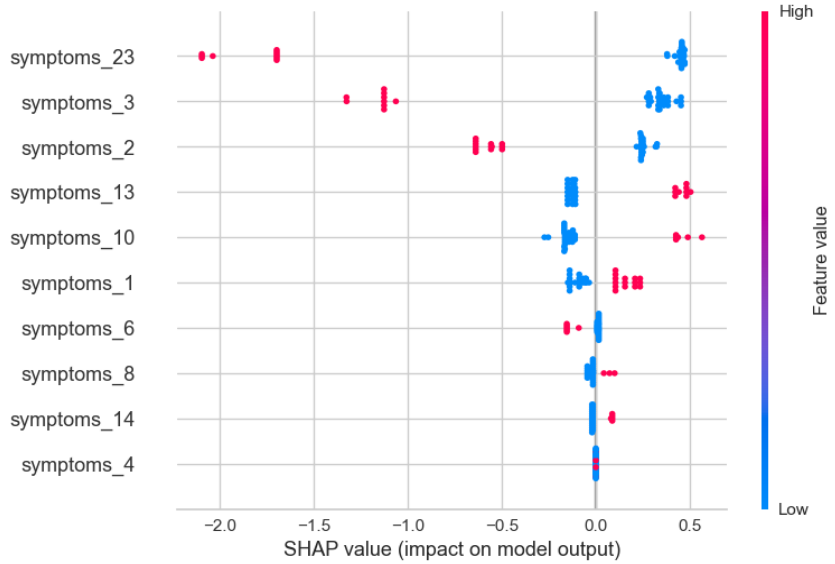

Figure 34: Cluster 0 is the potential high risk cluster

6.4.4.5 Experimental results

When we extracted the labels for the patient, we trained an XG Boost, an AI classification algorithm that learns to predict the risk. We took 90 patients as a test set to evaluate the models we have seen above. All the accuracies are very high, as shown in Table 7. For the labels extracted from the symptom-based models of constrained clustering, the results were equally good, see Table 8. The evaluation of the best model included the following metrics: f1-score, accuracy, precision, recall, macro average and weighted average.

- **Accuracy** is described as a combination of both types of observational error and is the sum of True Positives and True Negatives, divided by the sum of True Positives, True Negatives, False Positives and False Negatives. $Accuracy = \frac{tp+tn}{tp+tn+fp+fn}$, where tp are the True Positives, tn are the True Negatives, fp are the False Positives and fn are the False Negatives.
- **Precision** (also called positive predictive value) is the fraction of relevant instances among the retrieved instances. $Precision = \frac{tp}{tp+fp}$.
- **Macro Averaged Precision** is used for models with multiple classes. When macro-averaging, all classes contribute equally regardless of how often they appear in the dataset. Macro-precision measures the average precision per class.
- **Recall** in an imbalanced classification problem with more than two classes, is calculated as the sum of true positives across all classes divided by the sum of true positives and false negatives across all classes. $Recall = (tp)/(tp + fn)$
- **F1-score,** the traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall: $F1 = \frac{tp}{tp + \frac{(fp+fn)}{2}}$
- **Macro Averaged Recall** is used for models with multiple classes. When macro-averaging, all classes contribute equally regardless of how often they appear in the dataset. Macro-recall measures the average recall per class.
- **Macro-F1 averaging** is used for models with multiple classes. When macro-averaging, all classes contribute equally regardless of how often they appear in the dataset. Macro F1-averaging is performed by first computing the F1-score per class and then averaging it.

Table 7. Without constrained k-means clustering for symptoms

#	F1-score		
	k-means	Hierarchical Clustering	Constrained Clustering
0	0.98	0.99	0.97
1	1.00	0.95	0.98
2	0.93	1.00	0.97
Accuracy	0.98	0.99	0.97

For the best model, symptom-based with constrained clustering, we have the follow evaluations score:

Table 8. Constrained k-means clustering for symptoms

Clusters	Precision	Recall	f1-score	Support
0	0.96	1.00	0.98	25
1	1.00	1.00	1.00	9
2	1.00	0.50	0.67	2
accuracy			0.97	36
macro avg	0.99	0.83	0.88	36
weighted avg	0.97	0.97	0.97	36

The constrained clustering appears to work well for this specific dataset and the results seems to be meaningful. Furthermore, the constrained clustering converted the unsupervised problem to a semi-supervised one. The best clustering approach for predicting the risk of the patients is the constrained clustering for each feature (symptoms, comorbidity etc.) separately for several reasons. First and foremost, each feature's range of appearance varies, so if you do not analyze each feature individually, the model may focus on the feature with the most prevalence. To show what we mean, if control groups have a lot of family history of the disease, the clustering algorithm raises this feature as significant and ignores all the other features. Therefore, it is preferred to study each feature individually.

6.4.5 Implementation and integration of results

The MUP risk predictor pretrained model is serialized using the Pickle8 Python library. Pickle is a generic object serialization module that can be used for serializing and deserializing objects. It is a format that is compatible with the runtime execution environment of the Analytic Workbench component that hosts and exposes all pretrained models.

[8] https://docs.python.org/3/library/pickle.html

Once the MUP risk predictor model is successfully trained and properly evaluated, it gets serialized and packaged in a .zip file along with its metadata. Specifically, the metadata refers to:

- the full set of features
- the set of features that should be excluded when an inference is requested
- possible values per feature
- the possible predicted values that refers to the possible risk level prediction outcomes (low, moderate, high)
- the algorithm that the model implements, an XG Boost Classifier
- a model container file that references the model.pickle file
- a description of the model

Apart from the model metadata, the .zip file also contains:

- the predictor model itself
- the dataset which the model has been created with
- the evaluation metrics of the trained model
- the predictions for the validation dataset

Apart from the main integration path that relies on the Analytical Workbench, the personalized predictor exposes a REST API that is mostly used to continuously retrain the predictor model with different configuration parameters. The predictor REST API consists of methods that allow to trigger a retraining process as well as a method for making inferences based on a given model and a method for retrieving the optimal performance model, as it is depicted in Figure 35.

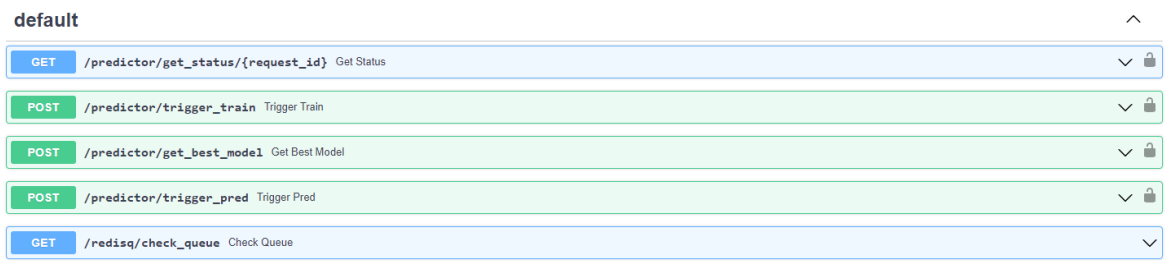

Figure 35: Predictor REST API.

6.5 TMU pilot

6.5.1 Description of available datasets & Known Risk factors

In the Taipei Medical University (TMU) pilot, data is stored in TMU-Clinical Data Repository (TMU-CDR) database that includes, historic patient data. The clinical database includes data from three affiliated hospitals, namely TMU Hospital, Wanfang Hospital and Shuang Ho Hospital. EHR data includes age, sex, diagnostic codes, laboratory test reports, medications, comorbidities, family history. Data is stored in a secure storage facility at TMU and must stay on premise and can only be accessed externally to the platform in Taiwan, after acquiring required permissions from the authorities.

For pancreatic cancer, the modifiable risk factors include obesity, dietary factors, alcohol and smoking whereas non-modifiable factors include age, genetic risk factors, familial pancreatic cancer, chronic pancreatitis and diabetes mellitus (S. S. K., + 16), (C. X. M., + 21), (J., Y. S., + 19).

For liver cancer, the modifiable risk factors include health behaviors and lifestyle factors (tobacco, alcohol use, smoking, obesity) (S., L., L., 21) whereas non-modifiable cancer risk factors include genetics (genetic mutations), family history, age, gender, race and ethnicity, and infections (hepatitis B virus, hepatitis C virus) (S., C., Y., + 19), (K., S., C., + 18), (A., A., A., + 17).

6.5.2 Description of case study & its desirable outcomes

The TMU pilot objective is to predict high risk individuals towards pancreatic and liver cancer for early-stage management of the disease. For this purpose, we will develop AI based machine learning models that will be applied to the data from TMU-CDR (Clinical Data Repository) to predict high-risk individuals for pancreatic and liver cancer so that modifiable risk factors (lifestyle, behavior) can be addressed early. This data will include variables such as clinical visits, diagnoses, medications, comorbidities, pancreatic and liver cancer diagnostic tests, etc.

6.5.3 Previous AI algorithms, relevant outcomes & Risk factors identified

Previously, the TMU pilot has worked on projects that allowed them to develop AI algorithms for patients with hematologic malignancies (S., F., C., + 20) and those on haemodialysis (T., A., C., + 18). Although they have not carried out risk prediction in the previous studies, however, with the assistance of iHelp technical partners, the TMU pilot aim to develop AI-based machine learning models to generate risk predictions (for pancreatic and liver cancer) based on the TMU-CDR database.

7 Conclusions

AI technologies hold immense promise to further improve the care of people with cancer and to facilitate cancer research. In this context, this document summarizes the actions performed under T4.1 - “Personalized Health Modelling and Predictions” as it provides an extended description of the mechanisms and AI models that will be implemented for the realisation of personalised health and risk prediction models. After an initial analysis of the available description of the datasets, specific methods and algorithms were selected for the training models. During the second phase of the project, since some real data became available, the mechanisms and the AI models were further specified and/or modified, in order to exploit the provided data for identifying the pre-mentioned risk factors. The primary data that become available were provided by two out of five pilots, namely UNIMAN and MUP. Regarding the MUP dataset, the problem of predicting the toxicities of interest has been faced as a classification problem and an overview of the ML models implemented, including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, LightGBM, XGBoost, MLP models. Currently we consider the model with the highest macro-average F1-score as the best model for the MUP dataset. Moreover, regarding the FPG pilot, after the training, we consider the model with the highest macro-average F1-score as the best model for the dataset.Regarding the UNIMAN pilot, the results that can be generated from biomarkers and their analysis, include the PRS score that will be stratified into quartile based on non-cancer PRS score’s value from the UK population with the lowest quartile will be used as the reference group. Regarding the Epigenomic biomarkers, the biological age from the different Epigenetic clocks can be summarized.

Bibliography

T. Akinyemiju et al, “The burden of primary liver cancer and underlying etiologies from 1990 to 2015 at the global, regional, and national level: results from the global burden of disease study 2015”. *JAMA oncology*, vol. 3, no. 12, pp. 1683-1691, 2017.

A. Baecker et al, “Do changes in health reveal the possibility of undiagnosed pancreatic cancer? Development of a risk-prediction model based on healthcare claims data”. *PloS one*, vol. 14, no. 6, p. e0218580, 2019.

M. I. Bittner, A. L. Grosu, and T. B. Brunner, “Comparison of toxicity after IMRT and 3D-conformal radiotherapy for patients with pancreatic cancer–a systematic review”. *Radiotherapy and Oncology*, vol. 114, no. 1, pp. 117-121, 2015.

B. Boursi et al, “A clinical prediction model to assess risk for pancreatic cancer among patients with new-onset diabetes”. *Gastroenterology*, vol. 152, no. 4, pp. 840-850, 2017.

J. Brownlee, *Statistical methods for machine learning: Discover how to transform data into knowledge with Python*. Machine Learning Mastery, 2018.

R. A. Fisher, “Statistical methods for research workers”. *In Breakthroughs in statistics (pp. 66-70)*. Springer, New York, NY, 1992.

M. H. Hsieh et al, “Development of a prediction model for pancreatic cancer in patients with type 2 diabetes using logistic regression and artificial neural network models”. *Cancer management and research*, vol. 10, p. 6317, 2018.

J. Jung et al, “Stereotactic body radiation therapy for locally advanced pancreatic cancer”. *PLoS One*, vol. 14, no. 4, p. e0214970, 2019.

B. Kenner et al, “Artificial Intelligence and Early Detection of Pancreatic Cancer: 2020 Summative Review”. *Pancreas*, vol. 50, no. 3, p. 251, 2021.

K. P. Ko, A. Shin, S. Cho, S. K. Park, and K. Y. Yoo, “Environmental contributions to gastrointestinal and liver cancer in the Asia–Pacific region”. *Journal of Gastroenterology and Hepatology*, vol. 33, no. 1, pp. 111-120, 2018.

C. T. C. Lee, J. X. Hu, and C. M. Liu, “Exploring prior diseases associated with pancreatic cancer”. *Current Problems in Cancer*, p. 100707, 2021.

I. Lee, G. H. Lushington, and M. Visvanathan, “A filter-based feature selection approach for identifying potential biomarkers for lung cancer”. *Journal of clinical Bioinformatics*, vol. 1, no. 11, pp. 1-8, 2011.

A. Malhotra et al, “Can we screen for pancreatic cancer? Identifying a sub-population of patients at high risk of subsequent diagnosis using machine learning techniques applied to primary care data”. *PloS one*, vol. 16, no. 6, p. e0251876, 2021.

S. Midha, S. Chawla, and P. K. Garg, “Modifiable and non-modifiable risk factors for pancreatic cancer: A review”. *Cancer letters*, vol. 381, no. 1, pp. 269-277, 2016.

https://dnamage.genetics.ucla.edu/

P. A. P. Moran, and C. A. B. Smith, "The correlation between relatives on the supposition of mendelian inheritance". *Transactions of the Royal Society of Edinburgh*, vol. 52, pp. 899-438, 1918.

W. Muhammad et al, "Pancreatic cancer prediction through an artificial neural network". *Frontiers in Artificial Intelligence*, vol. 2, p. 2, 2019.

M. Zhang, E. Wei, and R. Berry, "Faithful edge federated learning: Scalability and privacy". *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3790-3804, 2021.

S. Y. Su, C. J. Chiang, Y. W. Yang, and W. C. Lee, "Secular trends in liver cancer incidence from 1997 to 2014 in Taiwan and projection to 2035: An age-period-cohort analysis". *Journal of the Formosan Medical Association*, vol. 118, no. 1, pp. 444-449, 2019.

S. Y. Su, L. T. Lee, and W. C. Lee, "Mortality trends in chronic liver disease and cirrhosis from 1981 to 2015 in Taiwan". *Population health metrics*, vol. 19, no. 1, pp. 1-9, 2021.

S. Syed-Abdul et al, "Artificial Intelligence based Models for Screening of Hematologic Malignancies using Cell Population Data". *Scientific Reports*, vol. 10, no. 1, pp. 1-8, 2020.

A. Sharma et al, "Model to determine risk of pancreatic cancer in patients with new-onset diabetes". *Gastroenterology*, vol. 155, no. 3, pp. 730-739, 2018.

S. S. Thakur et al, "Artificial-intelligence-based prediction of clinical events among hemodialysis patients using non-contact sensor data". *Sensors*, vol. 18, no. 9, p. 2833, 2018.

H. J. Tsai, and J. S. Chang, "Environmental risk factors of pancreatic cancer". *Journal of clinical medicine*, vol. 8, no. 9, p. 1427, 2019.

Q. Z. C. Yang, "Combining Machine Learning Predictive Algorithms With Behavioral Nudges to Increase Rates of Serious Illness Conversations in Patients With Cancer". *JAMA oncology*, vol. 7, no. 5, pp. 781-782, 2021.

A. Yapp et al, "Communication-efficient and Scalable Decentralized Federated Edge Learning", 2021.

"Analysis of Variance (ANOVA) Definition & Formula", INVESTOPEDIA, 2021. [Online]. Available: https://www.investopedia.com/terms/a/anova.asp . [Accessed: 07-Dec-2021].

"Correlation Coefficient: Simple Definition, Formula, Easy Calculation Steps", Statistics How To, 2021. [Online]. Available: https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/ . [Accessed: 07-Dec-2021].

"Feature Selection Techniques in Machine Learning", AnalyticsVidhya, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/ . [Accessed: 07-Dec-2021].

"Introduction to Correlation Research", University of Connecticut, 2021. [Online]. Available: https://researchbasics.education.uconn.edu/correlation/ . [Accessed: 08-Dec-2021].

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html#sklearn.ensemble.GradientBoostingClassifier

https://xgboost.readthedocs.io/en/stable/python/python_intro.html

https://lightgbm.readthedocs.io/en/v3.3.2/Python-Intro.html

List of Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
ANOVA	Analysis of variance
ASHA	Asynchronous Successive Halving
ATC	Athens Technology Centre
ATP	Alberta Tomorrow Project
AUC	Area Under the Curve
BOHB	Bayesian optimization combined with Hyperband
CA	Consortium Agreement
CDR	Clinical Data Repository
D	Deliverable
DoA	Description of Action
DL	Deep Learning
DT	Decision Trees
EHRs	Electronic Health Records
EU	European Union
FP	False Positive
FL	Federated Learning
FPG	Agostino Gemelli University Policlinic
GB	Gradient Boosting
HDM	Hospital de Dénia-MarinaSalud
HHRs	Holistic Health Records
ICE	Information Catalyst for Enterprise
LDA	Linear Discriminant Analysis
LR	Logistic Regression
MAR	Missing at Random
MCAR	Missing completely at random
ML	Machine Learning
MNAR	Missing not at Random
MUP	Medical University Plovdiv
NNs	Neural Networks
NHANES	National Health and Nutrition Examination Survey
NPV	Negative Predictive Value
PCA	Principal Component Analysis
PHC	Personalising Health and Care
PHRs	Patient Health Records
PPI	Proton-Pump Inhibitor
PPV	Positive Predictive Value
QoL	Quality of Life
RF	Random Forest
ROC	Receiver Operating Characteristic
RWD	Real-World Data
SHAP	SHapley Additive exPlanations
SVD	Singular Value Decomposition

SVM	Support Vector Machines
T	Task
TP	True Positive
UK	United Kingdom
UNIMAN	University of Manchester
UPRC	University of Piraeus Research Center
XAI	eXplainable AI