Big Data to Enable Global Disruption of the Grapevine-powered Industries

D5.1 - Interactive Visualisation Components

DELIVERABLE NUMBER	D5.1
DELIVERABLE TITLE	Interactive Visualisation Components
RESPONSIBLE AUTHOR	Katrien Verbert (KU Leuven)

GRANT AGREEMENT N.	780751
PROJECT ACRONYM	BigDataGrapes
PROJECT FULL NAME	Big Data to Enable Global Disruption of the Grapevine-powered industries
STARTING DATE (DUR.)	01/01/2018 (36 months)
ENDING DATE	31/12/2020
PROJECT WEBSITE	http://www.bigdatagrapes.eu/
COORDINATOR	Nikos Manouselis
ADDRESS	110 Pentelis Str., Marousi, GR15126, Greece
REPLY TO	nikosm@agroknow.com
PHONE	+30 210 6897 905
EU PROJECT OFFICER	Ms. Annamária Nagy
WORKPACKAGE N. | TITLE	WP5 | Leveraging Data Value
WORKPACKAGE LEADER	KU Leuven
DELIVERABLE N. | TITLE	D5.1 | Interactive Visualisation Components
RESPONSIBLE AUTHOR	Katrien Verbert (KU Leuven)
REPLY TO	katrien.verbert@kuleuven.be
DOCUMENT URL	http://www.bigdatagrapes.eu/
DATE OF DELIVERY (CONTRACTUAL)	30 June 2020 (M30), 31 July 2020 (M31)
DATE OF DELIVERY (SUBMITTED)	31 July 2020 (M31)
VERSION | STATUS	3.0 | Final
NATURE	DEM (Demonstrator)
DISSEMINATION LEVEL	PU (Public)
AUTHORS (PARTNER)	Nyi Nyi Htun (KU Leuven), Francisco Gutierrez (KU Leuven), Diego Rojo (KU Leuven), Robin De Croon (KU Leuven), Jeroen Ooge (KU Leuven), Katrien Verbert (KU Leuven)
CONTRIBUTORS	Ioanna Polychronou (Agroknow)
REVIEWER	Giannis Stoitsis (Agroknow)

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Initial ToC and document structure	15/08/2018	Nyi-Nyi Htun (KU Leuven)
0.2	Visualisation components	22/08/2018	Francisco Gutierrez (KU Leuven)
0.3	First draft	27/08/2018	Nyi-Nyi Htun (KU Leuven), Francisco Gutierrez (KU Leuven)
0.3.1	Edits and updates to all chapters	10/09/2018	Katrien Verbert (KU Leuven)
0.4	Second draft	11/09/2018	Nyi-Nyi Htun (KU Leuven)
0.4.1	Edits and updates to all chapters	13/09/2018	Katrien Verbert (KU Leuven)
0.5	Third draft	14/09/2018	Nyi-Nyi Htun (KU Leuven)
0.6	Internal Review	16/09/2018	Pythagoras Karampiperis (Agroknow)
1.0	Final edits after internal review	28/09/2018	Nyi-Nyi Htun (KU Leuven)
1.1	Revised version of v1 (with EC reviewers' comments)	30/10/2019	Nyi-Nyi Htun (KU Leuven)
1.3	Final draft of the updates	18/11/2019	Nyi-Nyi Htun (KU Leuven), Francisco Gutierrez (KU Leuven) Katrien Verbert (KU Leuven) Diego Rojo (KU Leuven)
1.5	Internal review	19/11/2019	Panagis Katsivelis (Agroknow)
2.0	Final edits after internal review	03/12/2019	Nyi-Nyi Htun (KU Leuven) Katrien Verbert (KU Leuven)
2.1	Draft of v3.0	10/07/2020	Nyi-Nyi Htun (KU Leuven)
2.2	Second draft of v3.0	24/07/2020	Nyi-Nyi Htun (KU Leuven) Francisco Gutierrez (KU Leuven) Diego Rojo (KU Leuven) Ioanna Polychronou (Agroknow)
2.3	Final draft of v3.0	26/07/2020	Nyi-Nyi Htun (KU Leuven) Diego Rojo (KU Leuven)
2.4	Internal review	31/07/2020	Giannis Stoitsis (Agroknow)
3.0	Final edits after internal review	31/07/2020	Nyi-Nyi Htun (KU Leuven)

PARTICIPANTS		CONTACT
Agroknow IKE (Agroknow, Greece)		Nikos Manouselis Email: nikosm@agroknowc.com
Ontotext AD (ONTOTEXT, Bulgaria)		Todor Primov Email: todor.primov@ontotext.com
Consiglio Nazionale Delle Richerche (CNR, Italy)		Raffaele Perego Email: raffaele.perego@isti.cnr.it
Katholieke Universiteit Leuven (KULeuven, Belgium)		Katrien Verbert Email: katrien.verbert@kuleuven.be
Geocledian GmbH (GEOCLEDIAN Germany)		Stefan Scherer Email: stefan.scherer@geocledian.com
Institut National de la Recherché Agronomique (INRA, France)		Pascal Neveu Email: pascal.neveu@inra.fr
Agricultural University of Athens (AUA, Greece)		Katerina Biniari Email: kbiniari@aua.gr
Abaco SpA (ABACO, Italy)		Simone Parisi Email: s.parisi@abacogroup.eu
SYMBEEOSIS EY ZHN S.A. (Symbeeosis, Greece)		Konstantinos Rodopoulos Email: rodopoulos-k@symbeeosis.com

ACRONYMS LIST

PA	Precision agriculture
WP	Work package
NDVI	Normalized difference vegetation index
NDWI	Normalized difference water index
NDRE	Normalized difference red edge index
SAVI	Soil adjusted vegetation index
EVI	Enhanced vegetation index
CIRE	Chlorophyll index
LAI	Leaf area index
API	Application programming interface
HTML	Hypertext markup language
CSV	Comma-Separated Values
URI	Uniform Resource Identifier

EXECUTIVE SUMMARY

Deliverable 5.1 of work package 5 is to develop interactive visualisation components in order to leverage interpretation and understanding of complex data. As a starting point, a systematic review[1] was conducted where we thoroughly analysed a total of 140 research papers and found various decision support systems and visualisation tools that have been proposed in the domain of agriculture. Based on the findings of this review, we developed a total of six interactive visualisation components and delivered them in the first version[2] of deliverable 5.1. In the second version of the deliverable[3], we added new visualisations and also improved the existing ones. We also updated the development technology including the framework and libraries to be in line with the expertise of the development teams at the partner institutions. In this final version of the deliverable, we have added complex dashboards with multiple components that are tailored to meet the requirements of individual pilot partners. For the 5 pilots in this project, we presented a total of 8 dashboards, each designed to support a particular task proposed by the pilots.

This document is structured as follows. Section 1 provides an introduction to the deliverable describing the existing work, previous deliverables and motivations. In section 2, the new platform, including dashboards and individual components, is described in detail together with the development technology. In section 3, a detailed instruction on how to obtain the code and deploy the platform is provided. This document concludes with section 4 where a summary of the deliverable is underlined.

[1] https://doi.org/10.1016/j.compag.2019.05.053
[2] https://doi.org/10.5281/zenodo.1481788
[3] https://doi.org/10.5281/zenodo.3960979

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The volume of data available from farms has rapidly increased as the use of farm sensors, high-tech harvesters and drones is becoming more popular. Many farmers, therefore, feel overloaded by this large volume of data, and need additional tools for understanding and interpreting their data[4]. Visualisation is a powerful technique to mitigate these issues and has demonstrated its usefulness in the precision agriculture (PA) domain[5]. Visualisation is also effective at communicating uncertainty in seasonal climate prediction[6], an important task in agriculture. As Rind et al.[7] stated, visualisation provides a way to utilise "the processing power of modern computers with human cognition and visual abilities to better support analysis tasks".

The benefits of visualising complex data arise from being able to better interact and understand data by aggregating, filtering, searching or scaling down relevant information. As such, a large volume of data with potentially complex information becomes more easily consumable. Due to such benefits, visualisation tools have been widely used in various domains to assist with tasks that might otherwise require significant cognitive effort. For instance, analysis of past volcanic activities using a time series graph can uncover trends and can aid in predicting future eruption of a volcano. In the domain of agriculture and food, visualisation tools are being used in a number of sub-areas, including viticulture, dairy farming, food safety, wheat production, pest management, crop management, irrigation management and fertiliser management.

Under the work package 5 (WP5: leveraging data value), BigDataGrapes developed a number of visualisation components and a trust-aware decision support system for stakeholders in the grapevine-powered industry. As a first step towards accomplishing this, we conducted a systematic review gathering an exhaustive list of decision support systems and visualisation tools that have been proposed in the domain of agriculture. We reviewed a total of 140 research papers thoroughly and presented our findings from 61 relevant papers. Results of this review have been disseminated in the Journal of Computers and Electronics in Agriculture[8].

We found that the majority of the decision support systems in agriculture use at least one interactive visualisation technique which includes: 2D geographical maps, heatmaps (layered over 2D maps), time series, histograms, bar charts, pie charts, radar charts, dashboards and cross section representations of a farm. Thus, various visualisation techniques are being used in different areas of agriculture. Nevertheless, we discovered that the area of agricultural decision support systems still lacks the support from interactive visualisations with modern technology and uncertainty visualisations. For instance, the majority of the interfaces were designed for standalone PCs which lacks the flexibility and cross-platform support unlike web applications. Taking into account these findings, in the first version[9] of deliverable 5.1, we designed a total of six

---

[4] Ruß, G., Kruse, R., Schneider, M. and Wagner, P., 2009. Visualization of agriculture data using self-organizing maps. In *Applications and Innovations in Intelligent Systems XVI* (pp. 47-60). Springer, London.

[5] Wachowiak, M.P., Walters, D.F., Kovacs, J.M., Wachowiak-Smolikova, R. and James, A.L., 2017. Visual analytics and remote sensing imagery to support community-based research for precision agriculture in emerging areas. *Computers and Electronics in Agriculture*, *143*, pp.149-164.

[6] Frías, M.D., Iturbide, M., Manzanas, R., Bedia, J., Fernández, J., Herrera, S., Cofiño, A.S. and Gutiérrez, J.M., 2018. An R package to visualize and communicate uncertainty in seasonal climate prediction. *Environmental Modelling & Software*, *99*, pp.101-110.

[7] Rind, A., Wang, T.D., Aigner, W., Miksch, S., Wongsuphasawat, K., Plaisant, C. and Shneiderman, B., 2013. Interactive information visualization to explore and query electronic health records. *Foundations and Trends® in Human–Computer Interaction*, *5*(3), pp.207-298.

[8] Gutiérrez, F., Htun, N.N., Schlenz, F., Kasimati, A. and Verbert, K., 2019. A review of visualisations in agricultural decision support systems: An HCI perspective. Computers and Electronics in Agriculture, 163, p.104844.

[9] https://doi.org/10.5281/zenodo.1481788

most widely used visualisations to showcase; these visualisations were: 2D map with a heatmap layer, time series, histogram, bar chart, radar chart and scatter plot (see Figure 1).

Figure 1: Six visualisation components that were showcased in the first version of deliverable 5.1

In the second version[10] of deliverable 5.1, we extended the list by adding new visualisation components and also updated the existing ones. The new visualisation components added to the list were: pie chart, parallel coordinates, data table and progress circle. We also extended the use of visualisation libraries from Chart.js to D3.JS and Vega-Lite, together these libraries provided comprehensive visualisation abilities and complement one another. We also migrated from Polymer[11] to React[12] framework because we discovered that the development teams at many of the partner institutions are more familiar with React than Polymer. Thus, choosing React was in line with the expertise of other partners in the project.

In this version of deliverable 5.1, we demonstrate a number of interactive dashboards that are designed to support the requirements of individual pilot partners. These dashboards are described in detail in Section 0. For all 5 pilots in the project, we present a total of 8 newly developed dashboards that are tailored to meet the requirements. These dashboards utilise a combination of the components we have showcased in the previous deliverables. The new dashboards are much more complex and are designed to fulfil particular tasks. The components in the previous version (v2.0)[13] were designed to be more flexible and to work with varying structures of the datasets. These components therefore can still complement the new dashboards when the task at hand is not supported by any of the dashboards. In this regard, we designed a container platform that provides users with the functionalities of both the new dashboards and the flexible components previously delivered, in one place. In addition, we have also added new components such as 3D maps, uncertainty graphs and word clouds that were not present in the previous version (v2.0). Details of all the components are presented in Section 2.1.2.

[10] https://doi.org/10.5281/zenodo.3960979

[11] https://www.polymer-project.org/

[12] https://reactjs.org/

[13] https://doi.org/10.5281/zenodo.3960979

2. PLATFORM AND DEVELOPMENT TECHNOLOGY

2.1. PLATFORM

Exploring large, unorganised, and heterogeneous datasets can be a challenging task. To overcome this, in the previous version[14], we designed a data exploration platform that can accommodate various forms of data. This platform aimed to enable project partners to import any type of data and select visualisation components. By doing so, users can flexibly summon the visualisation components regardless of the varying structures of the datasets.

As described in introduction, we have built a number of interactive dashboards for all 5 pilot partners in the project. We utilise a combination of existing components from the previous deliverables. In addition, we added new components such as 3D maps, uncertainty graphs and word clouds that were not present in the previous version (v2.0)[15]. To accommodate both existing components and newly developed complex dashboards, we have designed a container platform that provides users with the functionalities of both the new dashboards and the flexibilities of existing components.

2.1.1. Overview

Figure 2 shows an overview of the platform where we can see 3 distinct areas: *data selector*, *visualisation selector* and *visualisation area*. The *data selector* area allows the user to select a dataset to visualise. The platform currently supports files in a Comma-Separated Values (CSV) format. The CSV format is one of the most broadly used forms to transfer and organise data; we mainly focused on CSV for now because it is also the most commonly used format among the pilot partners. The platform can read any standard CSV file; this file must include a header for each column. Row numbers (if any) are recognised and included as a separate column. In Figure 2, we can see that the climatic dataset[16] (climate.csv), authored by INRA, has been selected. The *visualisation selector* allows the user to select the type(s) of visualisation she/he wishes to use. The new dashboards have been added to the *visualisation selector* under the *Tools* category. Unlike the components which are flexible and work with any standard CSV file, the new dashboards require specific data types and structures. Thus, only the components work with the *data selector,* which complements the lack of flexibility in new dashboards. The user may select as many types of visualisation (both components and tools) as needed from the *visualisation selector* area. The *visualisation area* then displays the visualisations based on the user's selections. In Section 2.1.2, we describe each of the components in the visualisation area. In Section 0, we describe the new dashboards in detail.

[14] https://doi.org/10.5281/zenodo.3960979

[15] https://doi.org/10.5281/zenodo.3960979

[16] https://drive.google.com/drive/folders/1ZyX5LFiOLoI_qJKZiWoyt8oE5pailQSf?usp=sharing

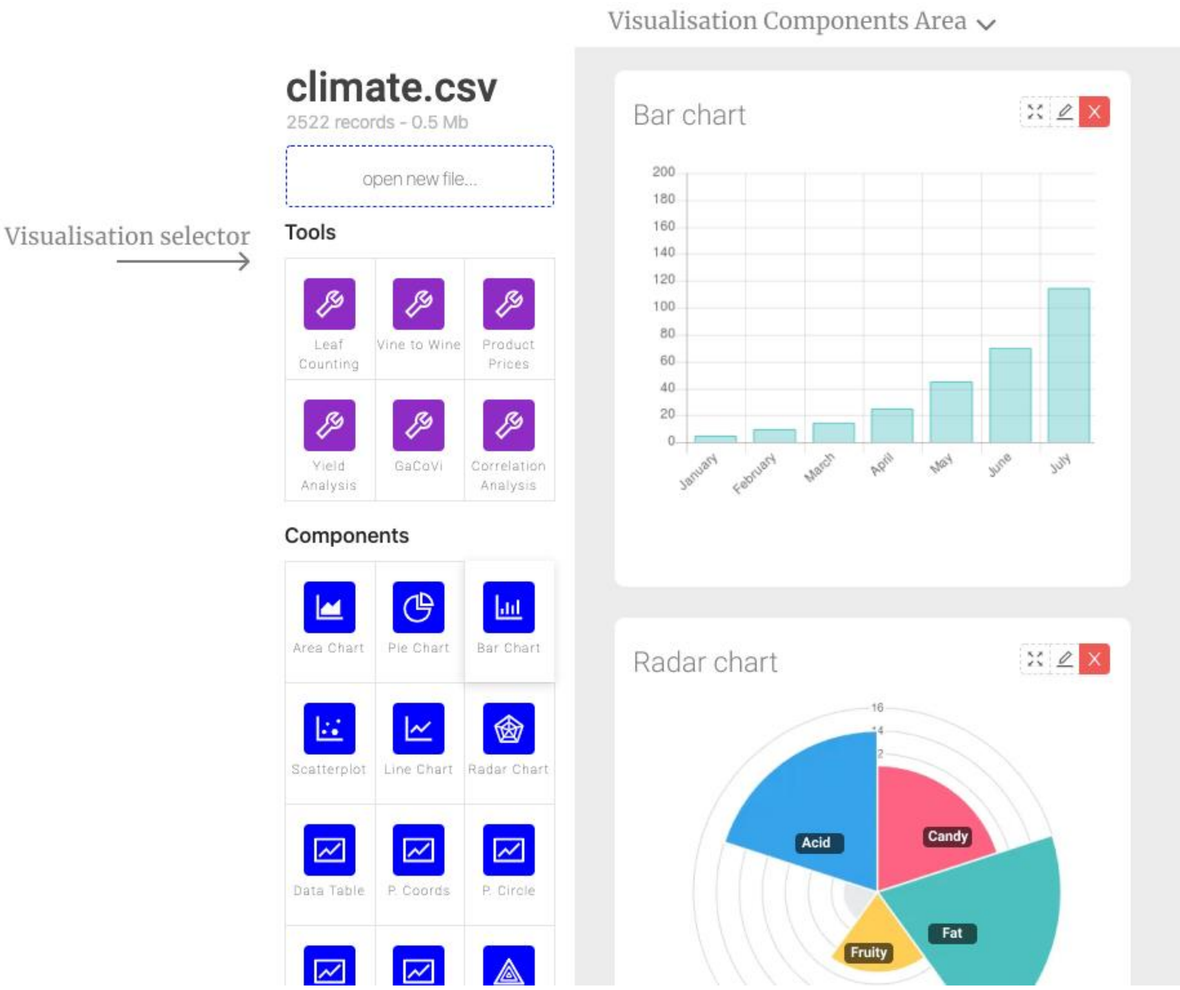

Figure 2: Overview of the platform showing 3 distinct areas: data selector, visualisation selector and visualisation area.

2.1.2. Visualisation Components

In the previous version[17] of deliverable 5.1, we showcased a total of 9 components; these are: bar chart & histogram, time series, scatter plots, radar charts, pie chart, choropleth map (heatmap), parallel coordinates, data table and progress circle. In this version, we have extended the list by adding new visualisation components. In the following, we describe the interaction capabilities of our visualisation components according to the taxonomy of interaction types that has been presented by Yi et al[18]:

Select interaction techniques enable users to mark data items of interest. The technique is used to select items that may be interesting to revisit at a later stage.

Explore interaction techniques enable users to examine a different subset of the data. A typical explore operation is *panning* that enables users to see a different part of a geographic map.

[17] https://doi.org/10.5281/zenodo.3960979

[18] Yi, J.S., ah Kang, Y. and Stasko, J., 2007. Toward a deeper understanding of the role of interaction in information visualization. IEEE transactions on visualization and computer graphics, 13(6), pp.1224-1231.

Reconfigure interaction techniques enable users to change the arrangement of representations to provide them with different perspectives onto the data set. Reconfigure interaction techniques allow users to change the way data items are arranged in order to provide different perspectives on the data set. Examples include the selection of different data elements to represent in parallel coordinates or changing the attributes assigned to the x- and y-axis of scatter plots.

Encode interaction techniques enable users to alter the representation of data, such as changing a pie chart to a histogram. As with other interaction techniques, the purpose of changing the type of representation is to uncover new aspects or patterns. In addition to changing the visualisation techniques, some tools also allow users to change the colour scheme or size.

Abstract / elaborate interaction techniques allow users to adjust the level of abstraction of a data representation. A *tool-tip* interaction technique that provides the user with additional information belongs to this category. Another common abstract/elaborate technique is *zooming*. Through zooming, users can change the scale of a representation so that they can see an overview of a larger data set (using zoom-out) or the detailed view of a smaller data set (using zoom-in).

Filter interaction techniques allow users to change the set of data items based on some specific conditions. In this type of interaction, users specify a range or condition, so that only data items meeting those criteria are presented. Sliders are for instance commonly used to filter the time period that is of interest. Users select ranges by moving sliders to show the data items that meet particular constraints. Many other systems provide a check or input box that enables users to filter the data items.

Connect interaction techniques are used to "*(1) highlight associations and relationships between data items that are already represented and (2) show hidden data items that are relevant to a specified item*". A typical example is a *brushing technique* that highlights a data element that is selected in one view in other related views.

2.1.2.1. Bar Chart & Histogram

A bar chart shows comparisons among discrete categories. One axis of the chart shows the specific categories being compared, and the other axis represents a measured value. Bar charts present bars clustered in groups of more than one, showing the values of more than one measured variable. A histogram is a type of bar chart typically used to represent the distribution of data points in a dataset. Thus, a single component is used to visualise bar chart and histogram. The user can choose to see the distribution of data points by aggregating the columns using the setup page (Figure 3). The following interaction capabilities have been added to the component.

- ***Reconfigure***: Users can select columns from the data set and select a function to aggregate the data; for example. In Figure 3, user can select a column "Fruity" from the dataset and aggregate the value using the mean function. This column will then be added to the bar chart if the checkbox is selected.
- ***Encode:*** The bar chart can be further customised by selecting a colour for the new calculated value.
- **Abstract/elaborate**: The visualisation shows tooltips on top of the bars when the user hovers the mouse over each of the bars in the graph.

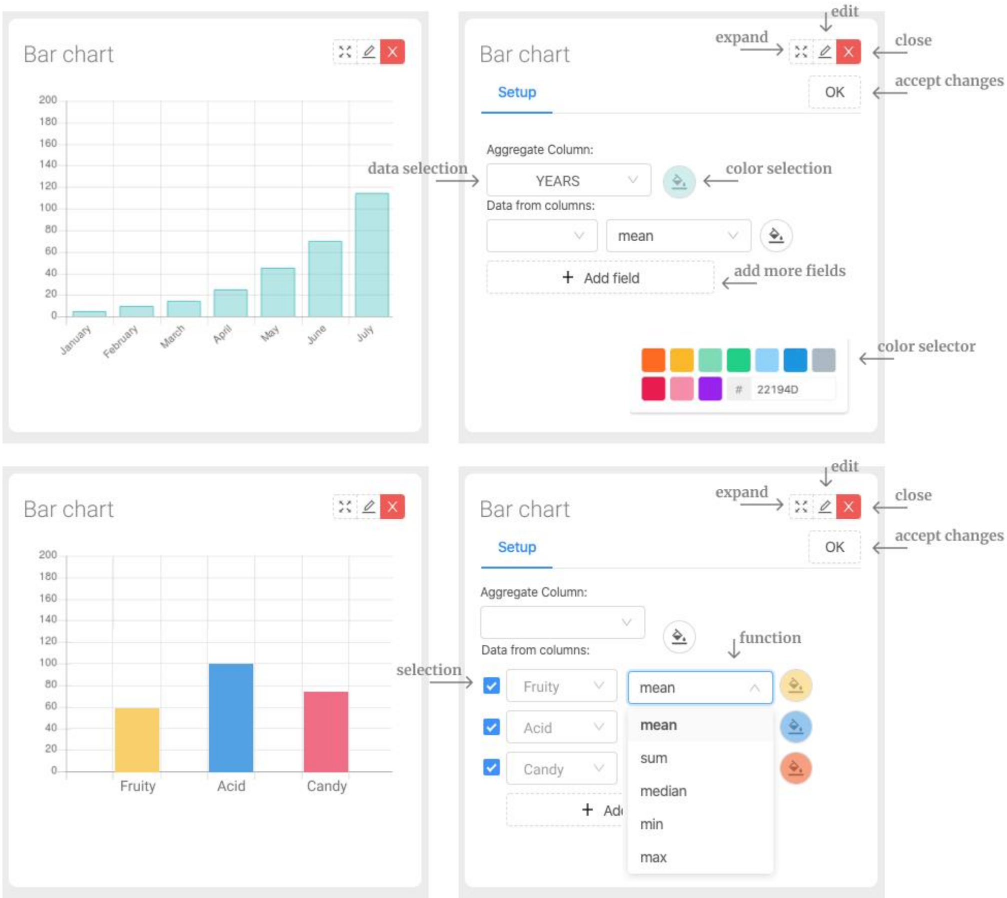

Figure 3: Bar chart and histogram

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Barchart from './VizComponents/Barchart/Barchart.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: [“column 1”, “column 2”, ... ],
    };
  }

render() {
return(<Barchart title="Bar Chart" data = {this.state.Dataset} columns = {this.state.Columns} />);
}
}
```

2.1.2.2. Time Series

A time series is a distribution of data points visualised in time order. It is usually presented as a sequence with successive equally spaced points in time. Figure 4 shows the time series component. The following interaction capabilities have been added to the component:

- **Encode**: The setup page allows users to customise the colour of data series.

- **Reconfigure**: In the setup page, the user can select the fields to display. Then assign a colour to the data selection. For example in Figure 4, the user selected the column "Dat-vrz" and assigns a colour. The checkbox indicates that the selection is active and visible in the time series visualisation.

- **Abstract/elaborate**: The visualisation shows tooltips when the user hovers the mouse over the data points in the graph. A zooming option is available when there are too many data points in the visual component.

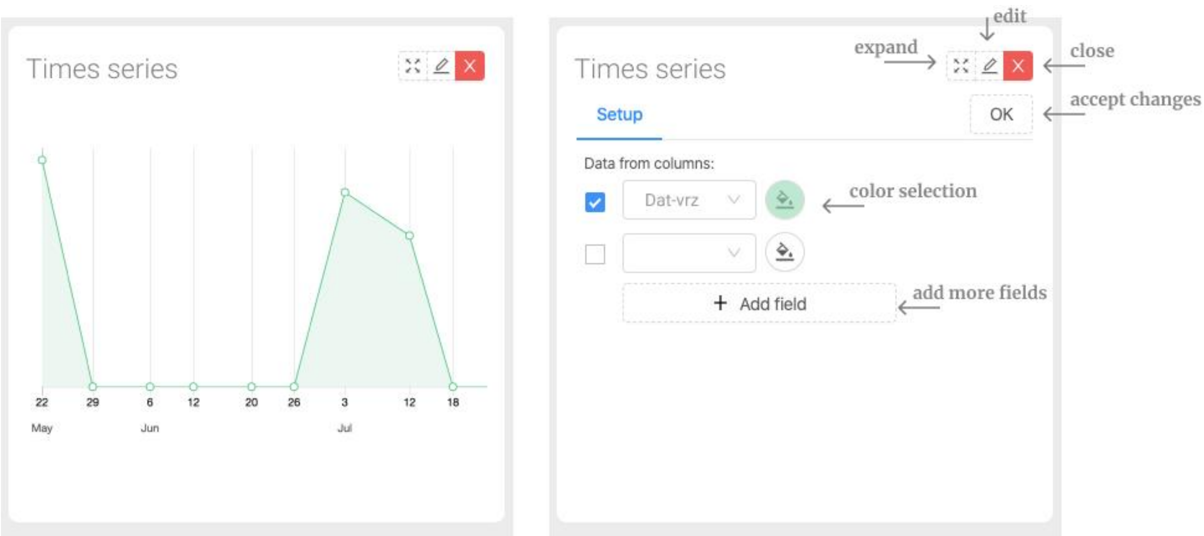

Figure 4: Time series

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Timeseries from './VizComponents/Timeseries/Timeseries.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: [“column 1”, “column 2”, ... ],
    };
  }

render() {
return(<Timeseries title="Time series" data = {this.state.Dataset} columns = {this.state.Columns} />);
}
}
```

2.1.2.3. Scatterplot

A scatter plot is a two-dimensional visualisation that uses dots to represent the values obtained for two different variables - one plotted along the x-axis and the other plotted along the y-axis. The scatterplot visualisation can be seen in Figure 5. The following interaction capabilities have been added to the component:

- **Encode**: The setup page allows users to customise the colour of each x and y axes pair.
- **Reconfigure**: In the setup page, the user can select the data to display, more fields can be added by clicking the "add field" option, datasets can be enabled or disabled by clicking the checkboxes. For example, in Figure 5, the user selected the columns "MTC" and "SSM" from the dataset and assigns a colour. The checkbox indicates that the selection is active and visible in the scatterplot visualisation.
- **Abstract/elaborate**: The visualisation shows tooltips when the user hovers the mouse over the data points in the graph. A zooming option is available when there are too many data points in the visual component.

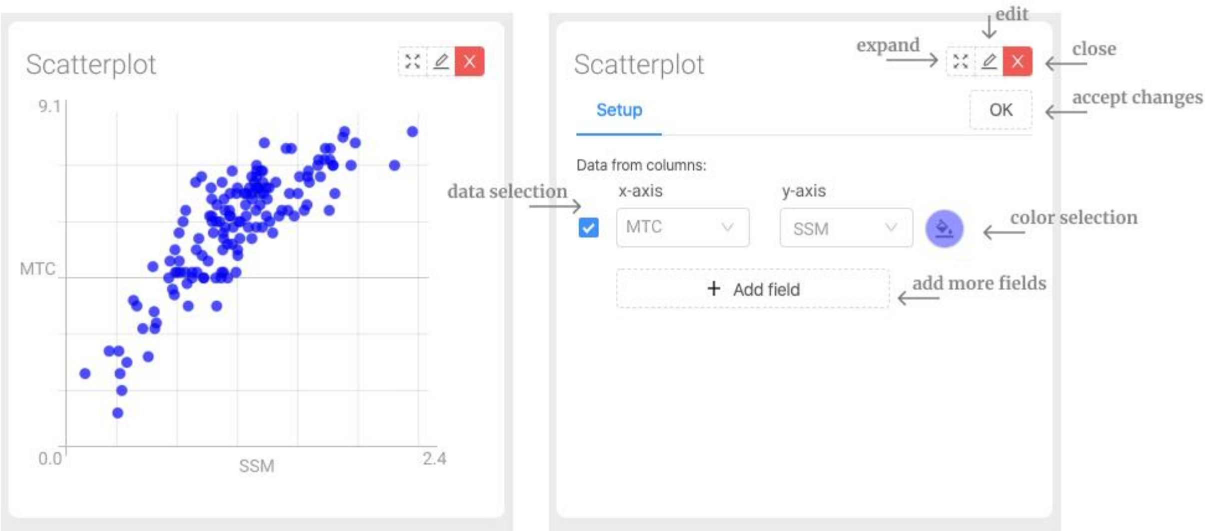

Figure 5: Scatterplot

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Scatterplot from './VizComponents/Scatterplot/Scatterplot.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: ["column 1", "column 2", ... ],
    };
  }

render() {
```

```
return(<Scatterplot title="Scatter Plot" data = {this.state.Dataset} columns = {this.state.Columns}
/>);
}
}
```

2.1.2.4. Radar chart

A radar chart is a representation of multivariate data in the form of a two-dimensional chart of three or more quantitative variables represented on axes starting from the same point. Figure 6 shows the radar chart component. The following interaction capabilities have been added to the component:

- **Encode**: The setup page allows users to customise the colour of each axis.
- **Reconfigure**: In the setup page, the user can select the data to display, more fields can be added by clicking the "add field" option, datasets can be enabled or disabled by clicking the checkboxes. For example in Figure 6, the user selected the columns "Fruity" , "Acid". "Fat" and "Candy" from the dataset and assigns a colour. The checkbox indicates that the selection is active and visible in the radar chart visualisation.
- **Abstract/elaborate**: The visualisation shows tooltips when the user hovers the mouse over the data points in the graph.

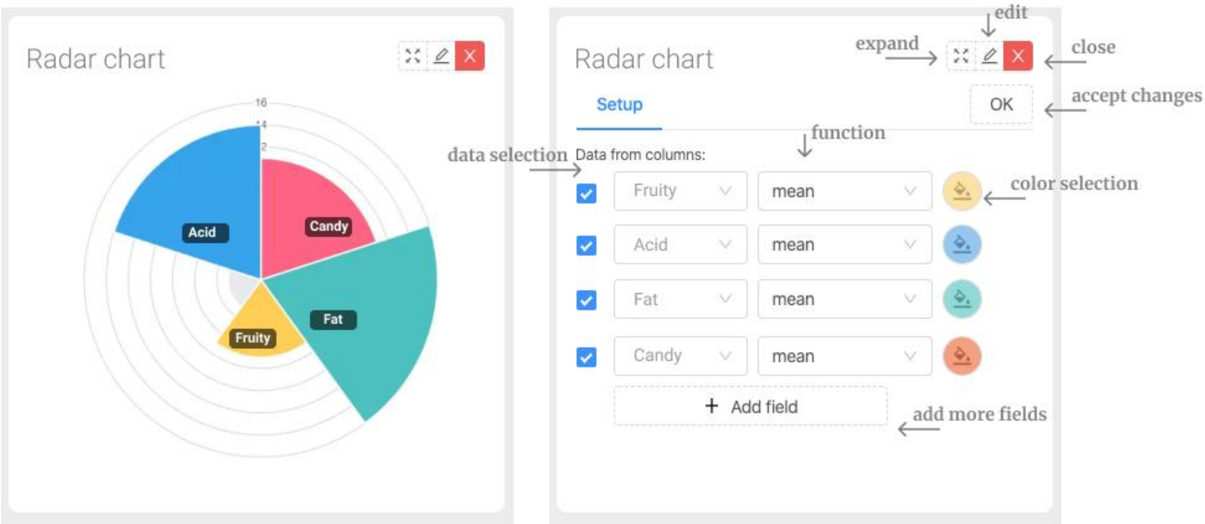

Figure 6: Radar chart

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Radarchart from './VizComponents/Radarchart/Radarchart.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
```

```
      Dataset: /** CSV File **/,
      Columns: ["column 1", "column 2", ... ],
    };
  }

render() {
return(<Radarchart title="Radar Chart" data = {this.state.Dataset} columns = {this.state.Columns} />);
}
}
```

2.1.2.5. Pie chart

A pie chart represents the proportion of a quality using slices in a circular pie. The arc length of each slice is proportional to the quantity it represents. The following interaction capabilities have been added to the component:

- **Encode**: The setup page allows users to customise the colour of each slice.
- **Reconfigure**: In the setup page, the user can select the data to display, more fields can be added by clicking the "add field" option, datasets can be enabled or disabled by clicking the checkboxes. For example in Figure 7, the user selected the columns "Fruity" , "Acid" and "Candy" from the dataset and assigns a colour. The checkbox indicates that the selection is active and visible in the radar chart visualisation.
- **Abstract/elaborate:** The visualisation shows tooltips when the user hovers the mouse over the data points in the graph.

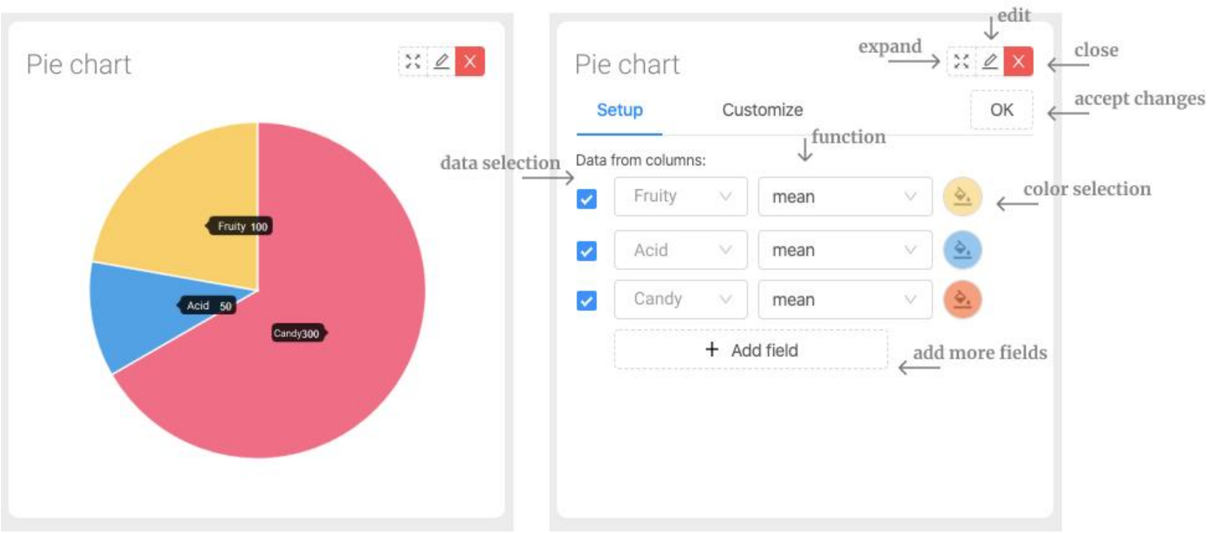

Figure 7: Pie chart

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Piechart from './VizComponents/Piechart/Piechart.js';

export class App extends Component {
```

```
  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: [“column 1”, “column 2”, ... ],
    };
  }

render() {
return(<Piechart title="Pie Chart" data = {this.state.Dataset} columns = {this.state.Columns} />);
}
}
```

2.1.2.6. Choropleth map (Heat map)

Choropleth maps are geographical representations of data where individual values contained in an area of interest are represented with a spectrum of colours. In agricultural visualisations, data such as normalised difference vegetation index (NDVI) is often shown as a layer of heatmap on top of a 2D geographical map. Figure 8 shows food incident records from 2017 to 2018 on a map component with colour scales. Choropleth maps are slightly different from heatmap in that they are often used to show data across larger geographical areas such as states and countries. The following interaction capabilities have been added to the component:

- **Reconfigure**: In the setup page, the user can select the data to display. For example, in Figure 8 the user selected the columns “Location”, and “Year-Month” from the dataset the colours are assigned automatically.
- **Abstract/elaborate**: The visualisation shows tooltips when the user hovers the mouse over the data points in the graph.
- **Filter**: Users can use the brushing technique to select and filter out data in visualisation.

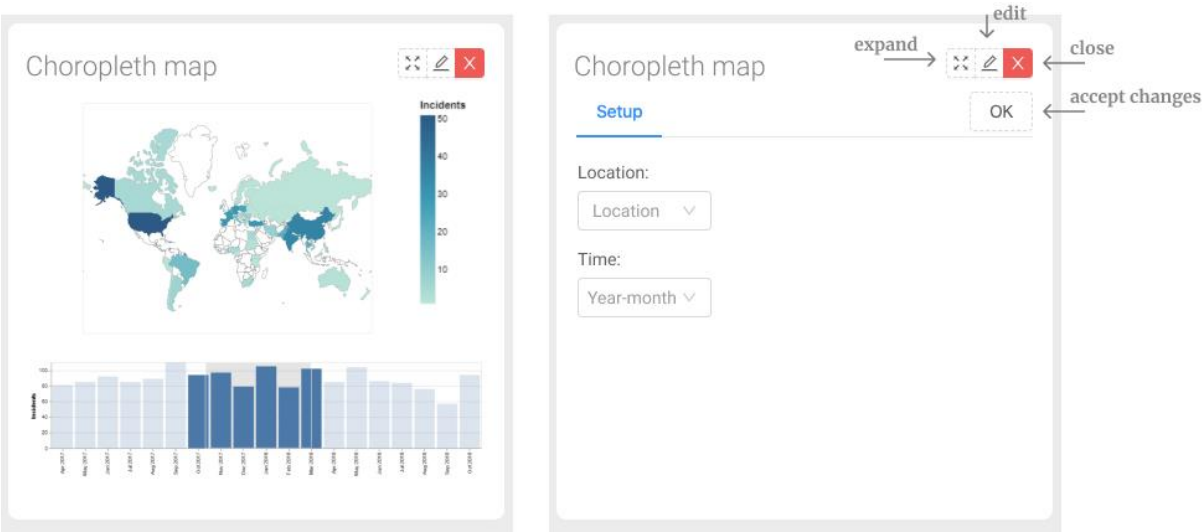

Figure 8: Choropleth map

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Choropleth from './VizComponents/Choropleth/Choropleth.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: [“column 1”, “column 2”, ... ],
    };
  }

render() {
return(<Choropleth title="Choropleth Map" data = {this.state.Dataset} columns = {this.state.Columns}
/>);
}
}
```

2.1.2.7. Parallel Coordinates

Parallel coordinates are used to visualise multidimensional data. Each column (parallel lines) represents a variable and can span up to n parallel lines (see Figure 9). Each horizontal line represents a set of data points across the parallel lines. The following interaction capabilities have been implemented:

- **Reconfigure**: The setup page allows users to select the field to display in each parallel line.
- **Encode**: Users can select a category column; the colours are automatically set for the selection.
- **Filter**: Users can use the brushing technique to select and filter out data in the parallel coordinates’ visualisation.
- **Explore**: A panning option is available when there are too many columns in the visualisation. Users can use the panning option to scroll from left to right, or from top to bottom, depending on the amount of information in the graph.

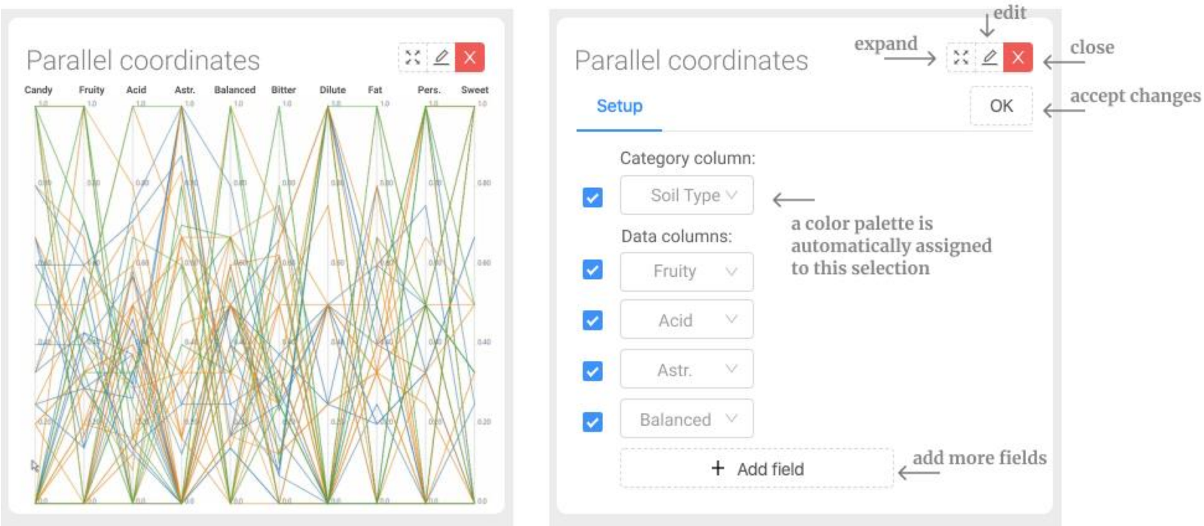

Figure 9: Parallel coordinates

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import PCoordinates from './VizComponents/PCoordinates/PCoordinates.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: [“column 1”, “column 2”, ... ],
    };
  }

render() {
return(<PCoordinates  title="Parallel  Coordinates"  data  =  {this.state.Dataset}  columns  =
{this.state.Columns} />);
}
}
```

2.1.2.8. Data Table

Data tables are useful to show the raw data in an organised manner (see Figure 10). Spreadsheets are the most familiar type of data tables. The following interaction capabilities have been implemented:

- Reconfigure: Users can click on the header of each column to sort the data by column.
- **Explore**: A scroll/pagination option is available when there are too many columns/rows in the data table, users can use the scroll option to scroll from left to right, or from top to bottom, depending on the amount of information in the table.

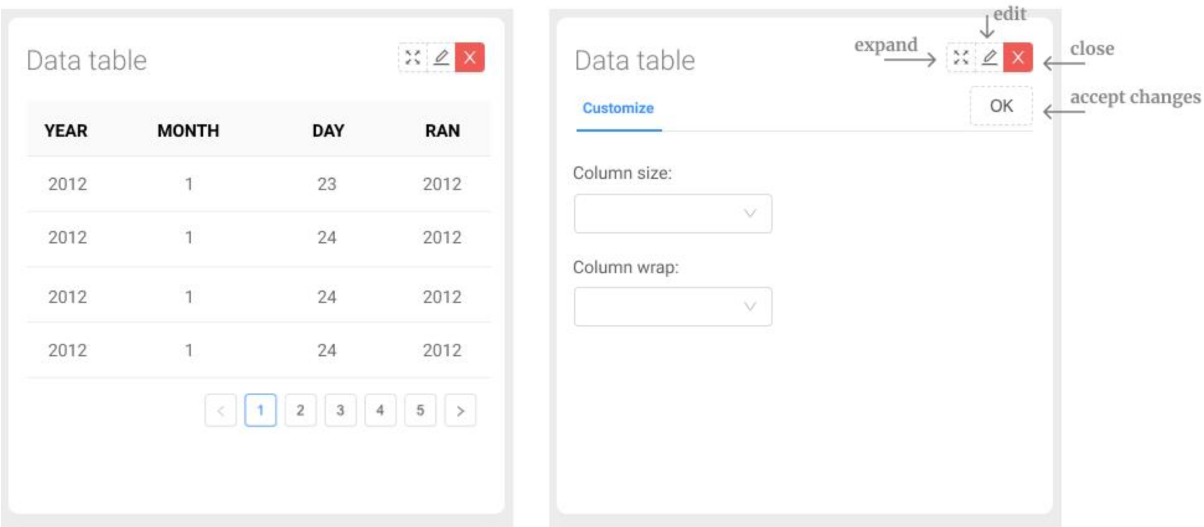

Figure 10: Data table

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Datatable from './VizComponents/Datatable/Datatable.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: ["column 1", "column 2", ... ],
    };
  }

render() {
return(<Datatable title="Data Table" data = {this.state.Dataset} columns = {this.state.Columns} />);
}
}
```

2.1.2.9. Progress Circle

Progress circles show a circular progress bar typically indicating a distribution (like pie charts) or progress (see Figure 11). Progress circles are also sometimes used to design gauge charts. Similar to progress circles, a gauge chart also visualises continuous data on an arc. In fact, popular visualisation libraries such as D3[19], Vega[20] and progressbar.js[21], by default, offer a pie chart or progress circle component to design gauge chart. We did not design gauge chart as a separate component, but it is used in a number of dashboards presented in this deliverable (see Sections 2.1.3.4 and 2.1.3.8).

The following interaction capabilities have been implemented in the progress circle component:

- **Reconfigure**: In the setup page, the user can select the data to display, more fields can be added by clicking the "add field" option, datasets can be enabled or disabled by clicking the checkboxes. For example in Figure 11 the user selected the columns "Fruity" , "Candy", "Balanced" and "Acid" from the dataset, selects a function to apply, in this case the mean function, and assigns a colour. The checkbox indicates that the selection is active and visible in the radar chart visualisation.

- **Encode**: The setup page allows users to customise the colour.

[19] https://d3js.org/
[20] https://vega.github.io/vega/
[21] https://kimmobrunfeldt.github.io/progressbar.js/

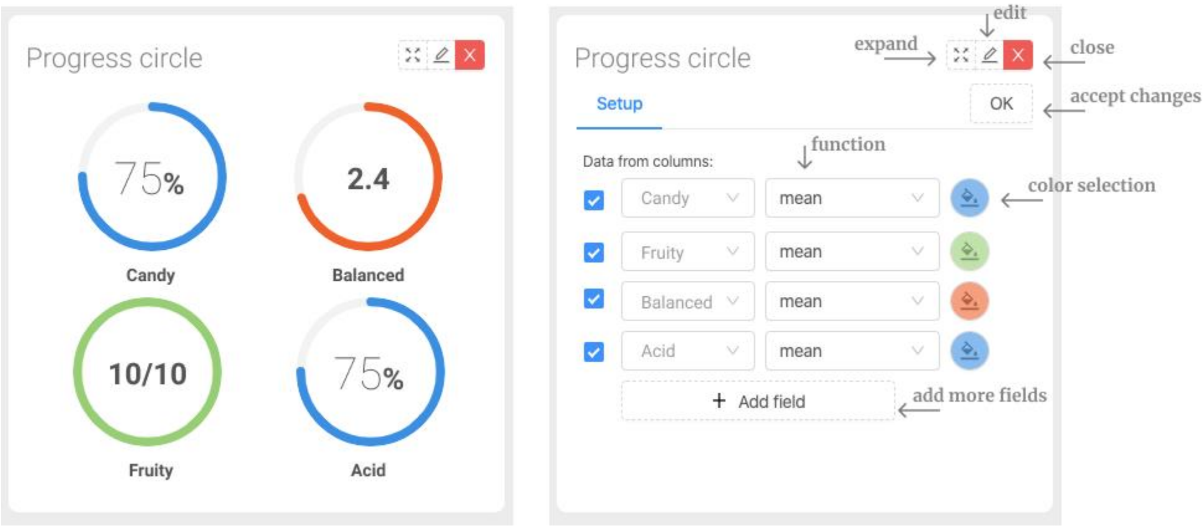

Figure 11: Progress circle

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import ProgressCircle from './VizComponents/ProgressCircle/ProgressCircle.js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: ["column 1", "column 2", ... ],
    };
  }

render() {
return(<ProgressCircle title="Progress Circle" data = {this.state.Dataset} columns =
{this.state.Columns} />);
}
}
```

2.1.2.10. 3D Map

A 3D map is a geographical representation that does not use a top-down orthographic camera view. This component uses a satellite base layer on top of an arbitrary GeoJson data layer. In agricultural visualisation, this allows a close-up view of a particular field showing data such as yield or quality on each subdivision of the field, instead of an aggregated view such as the one shown on the Choropleth Map component. Figure 12 shows values on each of the grid cells of a vineyard. 3D Map is a new addition in this version of the deliverable. The following interaction capabilities have been implemented:

- **Reconfigure**: In the configuration page, the user can select the data to display. For example, in Figure 12 the user selected a column from the dataset.

- **Encode**: The configuration page allows users to customise the colour. Besides, by selecting 3D mode, the user can also encode the data values into the extruded height of each of the GeoJson polygons (which are flat on the 2D mode).

- **Explore**: the user can change the camera view using the following controls:

  - dragPan: Drag to pan
  - dragRotate: Drag while pressing shift/ctrl to rotate
  - scrollZoom: Mouse wheel to zoom
  - doubleClickZoom: Double click to zoom in, with shift/ctrl down to zoom out

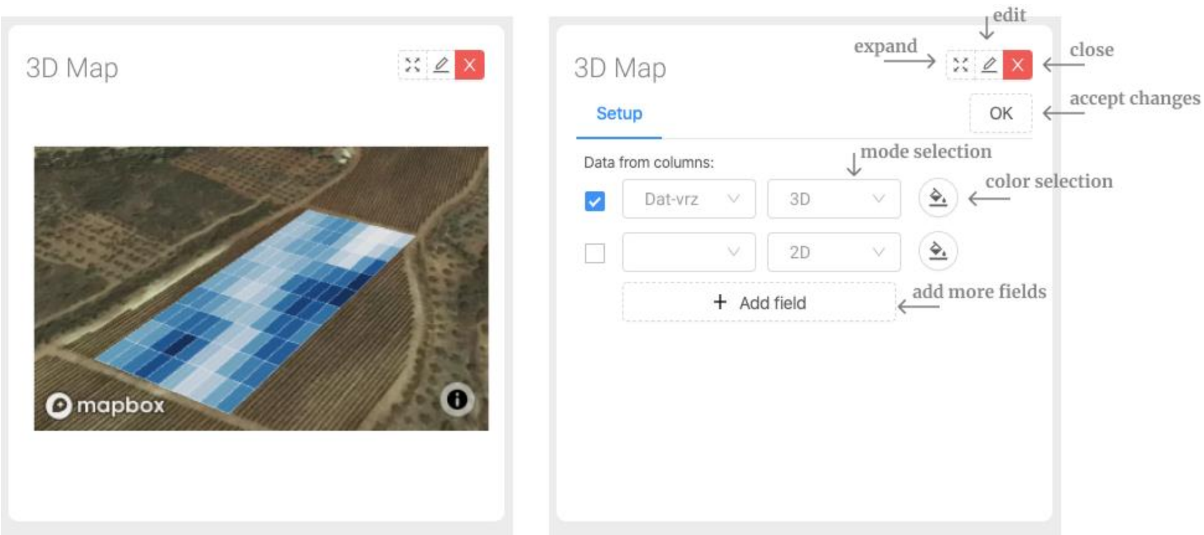

Figure 12: 3D Map

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Map3D  from './VizComponents/Map3D /Map3D .js';

export class App extends Component {

  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: ["column 1", "column 2", ... ],
    };
  }

render() {
return(<Map3D  title="3D Map" data = {this.state.Dataset} columns = {this.state.Columns} />);
}
}
```

2.1.2.11. Uncertainty Graph

The uncertainty graph uses lines to connect individual data points that display quantitative values over a specified time interval (see Figure 11), the uncertainty graph includes a component to depict uncertainty from a prediction of future events given the data points. The uncertainty graph is a new addition in this version of the deliverable. The following interaction capabilities have been implemented:

- **Reconfigure**: In the setup page, the user can select the data to display, more fields can be added by clicking the “add field” option, datasets can be enabled or disabled by clicking the checkboxes. For example in Figure 11 the user selected a data columns from the dataset.

- **Encode**: The setup page allows users to customise the colour of the line graph in the visualisation.

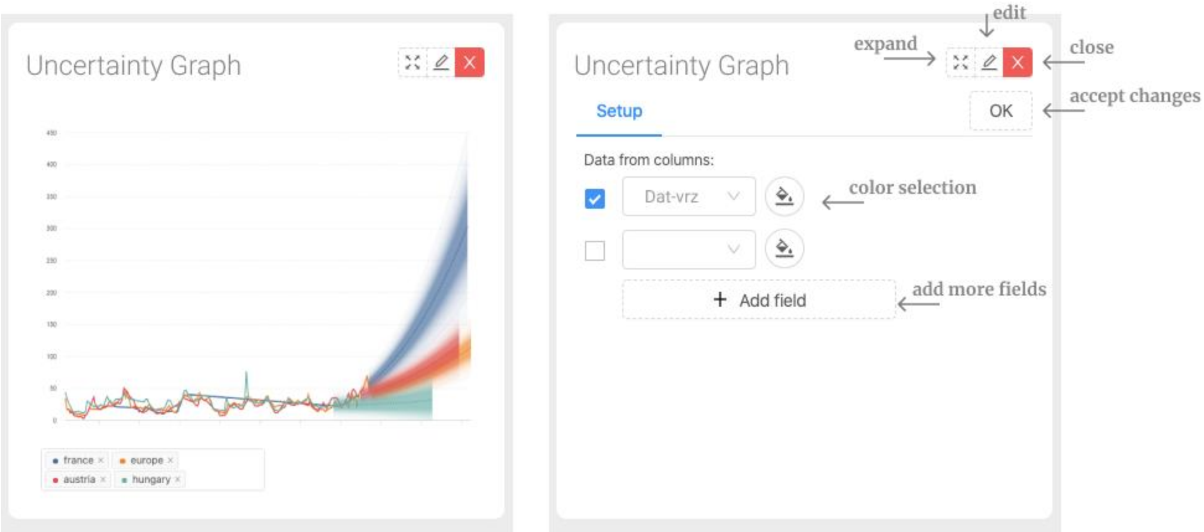

Figure 13: Uncertainty Graph

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import UncertaintyGraph  from './VizComponents/UncertaintyGraph  /UncertaintyGraph .js';

export class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: [“column 1”, “column 2”, ... ],
    };
  }

render() {
return(<UncertaintyGraph     title=”UncertaintyGraph"   data   =   {this.state.Dataset}   columns   =
{this.state.Columns} />);
}
}
```

2.1.2.12. Word Cloud

A word cloud is a visualisation that represents text data, typically used to depict tags on websites to visualise the relevance of tags or terms. Tags or terms are usually single words, the importance of the word is highlighted by its size, color can be used to depict importance or cluster (see Figure 11). The Word Cloud component is a new addition in this version of the deliverable. The following interaction capabilities have been implemented:

- **Reconfigure**: In the setup page, the user can select the data to display, more fields can be added by clicking the "add field" option, datasets can be enabled or disabled by clicking the checkboxes. For example in Figure 11 the user selected a column from the dataset, and assigns a colour.

- **Encode**: The setup page allows users to customise the colour or a theme.

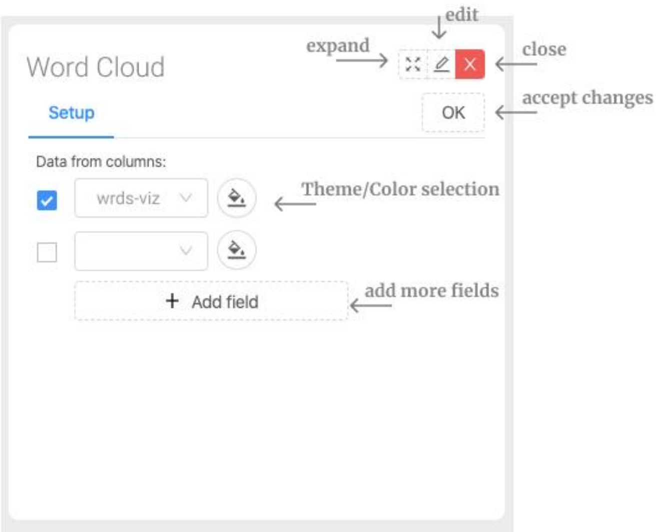

Figure 14: Word Cloud

Reusing the Component

The following example code shows how the component can be imported and configured in the React framework.

```
import React, { Component } from 'react';
import './VizComponents/Components.css';
import Wordcloud  from './VizComponents/Wordcloud /Wordcloud.js';

export class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      Dataset: /** CSV File **/,
      Columns: ["column 1", "column 2", ... ],
    };
  }

render() {
return(<Wordcloud title="Word Cloud" data = {this.state.Dataset} columns = {this.state.Columns} />);
}
}
```

2.1.3. Task-based Visualisation Tools

In this section, we showcase a number of visualisation tools (dashboards) that are designed to address specific analytical tasks, which emerged from the requirements proposed by each pilot partner. As a reminder, the five pilots are: 1) table and wine grapes pilot, 2) wine making pilot, 3) farm management pilot, 4) natural cosmetics pilot and 5) food protection pilot. Visualisation requirements for most pilots varied from each other due to the differing requirements in analytical tasks. An overview of the tasks being addressed by each pilot is as follows:

1) Table and wine grapes pilot (pilot 1, partner = AUA)
    a. Correlation analysis of grapevine responses to terroir
2) Wine making pilot (pilot 2, partner = INRAE)
    a. Visualising from vine to wine - parameters influencing wine quality
    b. Counting grapevine leaves,
3) Farm management pilot (pilot 3, partner = Abaco)
    a. Water availability and irrigation recommendations
4) Natural cosmetics pilot (pilot 4, partner = Symbeeosis)
    a. Grapevine by-products biological efficacy predictor
5) Food protection pilot (pilot 5, partner = Agroknow)
    a. Price prediction,
    b. Recall prediction,
    c. Risk assessment

When designing the tools to address these tasks, we followed an iterative design approach and utilised existing visualisation components that were proposed in the previous version[22] of deliverable 5.1. Bi-weekly update and feedback sessions were organised with individual pilot partners from November 2019. During the first few sessions, we gathered the requirements and refined the tasks. Later, we presented updates on the visualisations and the pilot partners provided us with feedback. In the following sub-sections, we present the visualisation tools for each of the tasks. Datasets were provided by the respective pilots in the CSV format. Thus, in the current versions, offline datasets were used except for pilot 3 (Abaco) where data was already available through an API call from Abaco. In a future iteration following the first end-user evaluation, we hope to integrate with the BigDataGrapes platform and receive data directly from APIs for the remaining pilots.

[22] https://doi.org/10.5281/zenodo.3960979

2.1.3.1. Pilot 1: correlation Analysis of Grapevine Responses to Terroir

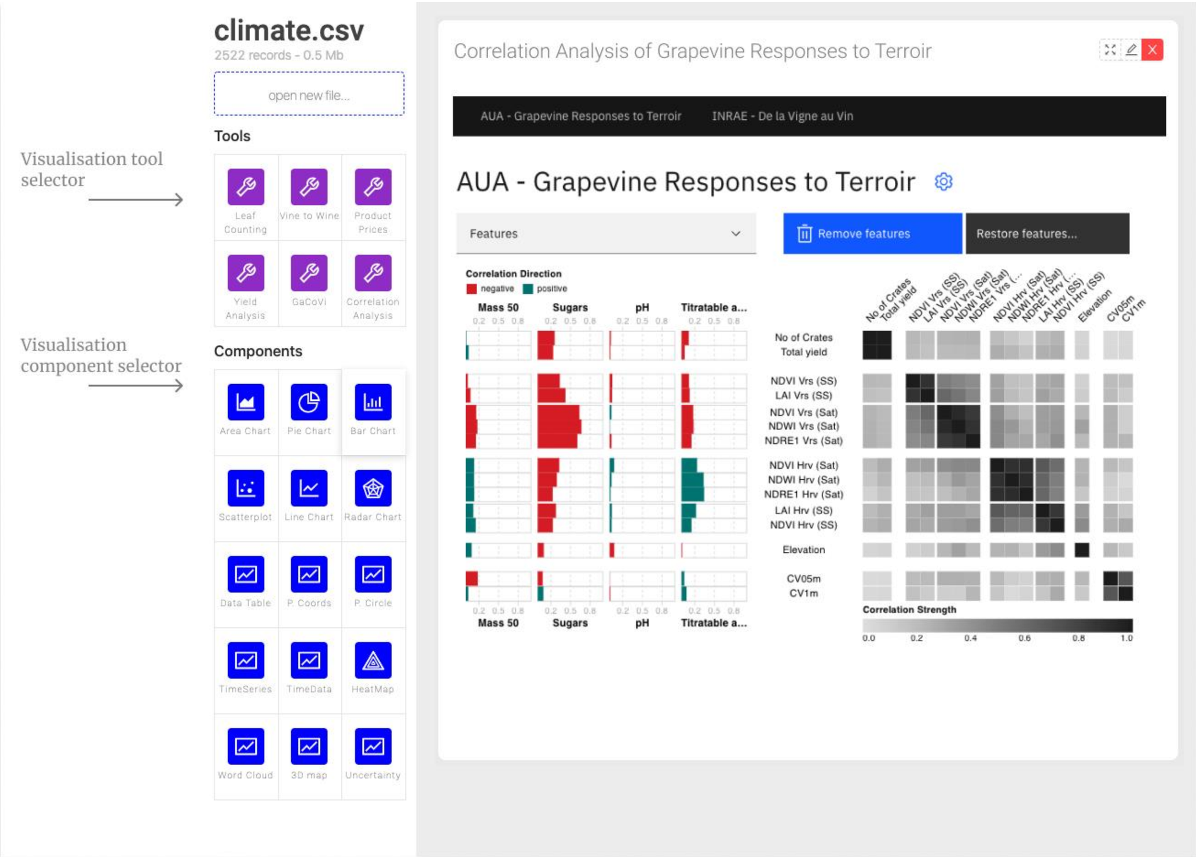

Figure 15: Correlation analysis visualisation for pilot 1 and pilot 2

Description

The purpose of this tool is to address the requirement proposed by pilot 1 (AUA): Grapevine responses to terroir. Based on sensors, farming, phenological data, and satellite data, users can visualise different correlations between them and explain how they affect the grapevine properties. This tool adapts GaCoVi (Gapped Correlation Visualisation) which was published in our previous paper[23]. For evaluation purposes, a stand-alone version of the tool has temporarily been hosted at one of KU Leuven servers: http://picasso.experiments.cs.kuleuven.be:3603/.

Although this tool was particularly designed for pilot 1, we later discovered that pilot 2 (INRAE) also had similar requirements that would benefit from this tool but focusing on the effect of different vineyard characteristics on the properties of the wine. Therefore, we also fitted GaCoVi for INRAE. For evaluation, we hosted a stand-alone version of the tool for INRAE at one of KU Leuven servers: http://picasso.experiments.cs.kuleuven.be:3602/.

The dashboard is integrated by the following components:

- GaCoVi (visualisation): it shows the correlations between the variables in this dataset. To do this, the variables are divided into dependant variables or features, in our case vineyard variables, and independent or target variables (grapevine variables in the case of AUA and wine variables in the case of INRAE).

[23] Rojo Garcia, D., Htun, N.N. and Verbert, K., 2020. GaCoVi: a Correlation Visualization to Support Interpretability-Aware Feature Selection. Proceedings of EuroVis 2020 Short Papers.

- On the left side of the visualisation, one can see the correlation of each feature or vineyard variable with each of the target variables (grapevine quality or wine quality variables). The longer the bar, the higher the correlation is. If the bar is red, the correlation is negative, and if it is green, positive.
- On the right side of the visualisation, we can see the correlation between the different features or vineyard variables. The darker the grey, the higher the correlation strength. The features are ordered so that closer rows have a higher correlation. Since this is, in most datasets, not entirely possible, we minimise the weighted error. Sometimes, even though two rows are next to each other, they are not highly correlated. To make this easy to identify, there is a gap between rows that is bigger the lower the correlation. If they are not correlated at all, their correlation is 0, the gap will be one square size. In summary, the closer the rows are, the more correlated they are.

- Remove Features Inputs: to remove features, users can select them from the dropdown menu or click on the name in the visualisation. By clicking on the Remove Features button, the selected features get removed. To bring back some of the deleted features, click on the Restore Features button, select them, and click Restore.
- Settings: the user can choose which target variables to focus on and, in the INRAE scenario, select a subset of the dataset (red or white wines).

Data

We used the data from both pilot 1 and pilot 2 for GaCoVi. Datasets were received in CSV format. Both datasets we used in the current version of GaCoVi include the following variables:

- **AUA version:** the satellite data, coloured blue in the followings, are provided by Geocledian. AUA has supplied all the other data. The variables are as follows (target variables are in bold):
  - **Mass 50: The weighted mass of 50 berries, in g.**
  - **Sugars: Determination of sugar content in must, in Brix.**
  - **pH: Determination of pH in must.**
  - **Titratable acidity: Determination of titratable acidity in must, in g of tartaric acid/L.**
  - No of Crates per cell: No of Grape Crates per cell.
  - Total yield (Kg): Mass of Harvested Product per cell (Kg).
  - CV1m: Soil Electrical Conductivity 1m depth.
  - CV05m: Soil Electrical Conductivity 0.5m depth.
  - Elevation (m): Elevation (m).
  - NDVI Veraison (SS): Normalized Difference Vegetation Index derived from a SpectroSense sensor during Veraison.
  - LAI Veraison (SS): Leaf Area Index derived from a SpectroSense sensor during Veraison.
  - NDVI Harvest (SS): Normalized Difference Vegetation Index derived from a SpectroSense sensor during Harvesting.
  - LAI Harvest (SS): Leaf Area Index derived from a SpectroSense sensor during Harvesting.
  - NDVI Veraison (Sat): Normalized Difference Vegetation Index derived from the satellite during Veraison.
  - NDRE1 Veraison (Sat): Normalized Difference Red Edge Index (v1) derived from the satellite during Veraison.
  - NDWI Veraison (Sat): Normalized Difference Water Index derived from the satellite during Veraison.

    - NDVI Harvest (Sat): Normalized Difference Vegetation Index derived from the satellite before Harvesting.
    - NDRE1 Harvest (Sat): Normalized Difference Red Edge Index (v1) derived from the satellite before Harvesting.
    - NDWI Harvest (Sat): Normalized Difference Water Index derived from the satellite before Harvesting.
- **INRAE version:** the satellite data, coloured blue in the followings, are provided by Geocledian. INRAE has supplied all the other data. The variables are as follows (target variables are in bold):
    - **dureeFermAlc (Alcoholic Fermentation Length): In day number**
    - **sucres (Sugars): Laboratory analysis carried out on must before alcoholic fermentation, in g/L**
    - **pH: Laboratory analysis carried out on must before alcoholic fermentation**
    - **aciditeTotale (Total Acidity): Laboratory analysis carried out on must before alcoholic fermentation, in gH2SO4/L**
    - ETP1: Cumulated Penman evapotranspiration in mm during april and may.
    - ETP2: Cumulated Penman evapotranspiration in mm during june and July.
    - ETP3: Cumulated Penman evapotranspiration in mm during august and September.
    - vent: Number of days with a wind speed > 30km/h from April to September.
    - sableux: Sandy soil texture (present 1 or absent 0).
    - argileux: Clay soil texture (present 1 or absent 0).
    - dateVendange (Harvest Date): Day of the year (number between 1 and 366).
    - rendement: Yield in kg/ha at harvest.
    - NDVI: Normalized Difference Vegetation Index.
    - NDRE1: Normalized Difference Red Edge Index (v1).
    - NDRE2: Normalized Difference Red Edge Index (v2).
    - NDWI: Normalized Difference Water Index.
    - SAVI: Soil Adjusted Vegetation Index.
    - EVI: Enhanced Vegetation Index 2.
    - CIRE: Chlorophyll Index - Red Edg

2.1.3.2. Pilot 2: visualising from vine to wine - parameters influencing wine quality

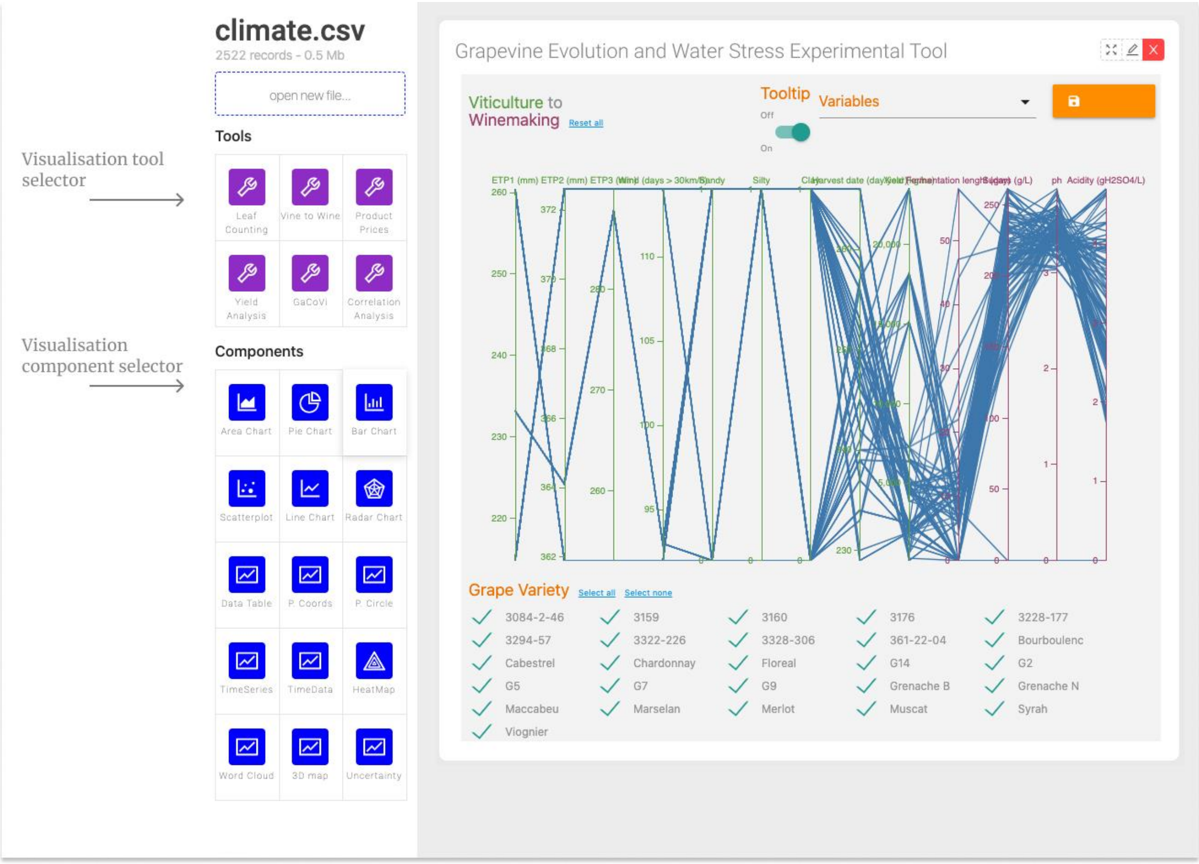

Figure 16: Vine to wine visualisation for pilot 2

Description

The purpose of this tool is to address a requirement proposed by pilot 2 (INRAE): visualising from vine to wine - parameters influencing wine quality. Based on existing large and heterogeneous data provided by different data sources gathered by diverse teams at each step of wine production, including climatic, soil, harvest and winemaking data, sensory and lab analysis, users can visualise the influence of all of these parameters on the wine quality in response to a changing environment. Users are able to explore their dataset by selecting any number of variables of their interest, and by filtering the records based on parameters such as grape variety, year, wine colour and aroma. For evaluation purposes, a stand-alone version of the tool has temporarily been hosted at one of KU Leuven servers: http://picasso.experiments.cs.kuleuven.be:3329/.

Data

All of the data used in this tool has been supplied by INRAE. The variables are as follows:

- viticulture:
  - ETP 1 : penman evapotranspiration in mm during April and May
  - ETP 2 : penman evapotranspiration in mm during June and July
  - ETP 3 : penman evapotranspiration in mm during August and September
  - wind : number of days with a wind speed > 30km/h from April to September,
  - sandy : soil texture is sandy (yes = 1, no = 0)
  - silty : soil texture is silty (yes = 1, no = 0)
  - clay : soil texture is clay (yes = 1, no = 0)

- harvestDate : day of the year where harvest was made (number between 1 and 366)
  - yield : yield in kg/ha
- winemaking:
  - fermentationLength: the length of fermentation in number of days
  - sugars: sugar content in the laboratory analysis carried out on must before fermentation, in g/L
  - ph: ph content in laboratory analysis carried out on must before fermentation
  - total acidity: acid content in laboratory analysis carried out on must before fermentation, in gH2SO4/L
- aroma:
  - name: the name of the aroma (e.g. baking, balanced, biting, bitterness, brownRed, candy, etc.). A total of 85 aromas are present in the dataset.
  - percentage: the rate of detection by experts for each aroma during sensory analysis (0: none, 100: everyone detected the aroma).

2.1.3.3. Pilot 2: counting grapevine leaves

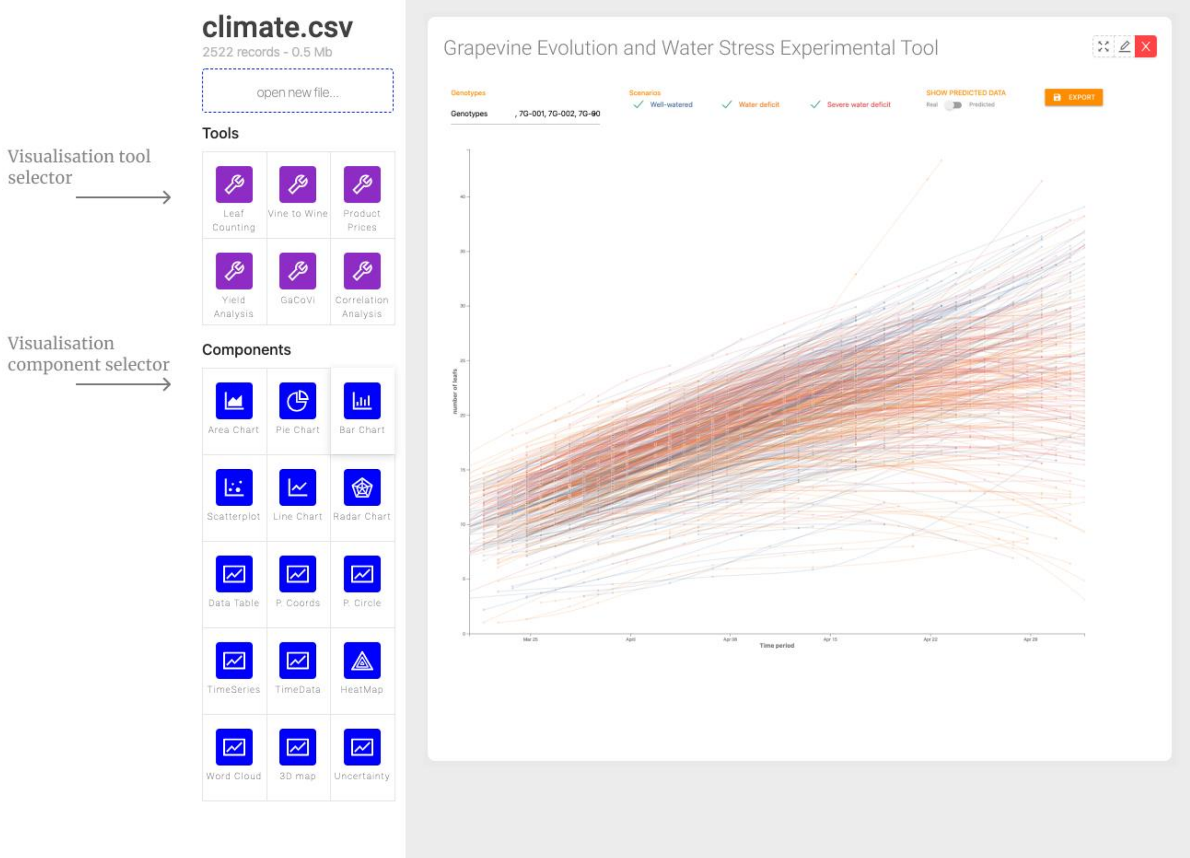

Figure 17: Leaf counting visualisation for pilot 2

Description

The purpose of this tool is to address a requirement proposed by pilot 2 (INRAE): counting grapevine leaves. It visualises the evolution of existing grapevine leaves on a plant over time. Based on an existing large number of grapevine images derived from the PhenoArch platform, a machine-learning pipeline was developed by CNR (a tech partner), aiming at counting leaves from side-view grapevine images. Deep learning techniques based on artificial neural networks were exploited to infer the number of leaves from each grapevine image. This output is then visualised as shown in

the tool. Users are able to filter the plants based on their genotype and experimental scenarios (e.g. well-watered, water deficit and severe water deficit). For each plant (represented with a line), users can click on any of the data points to inspect the plant further. Once a data point is clicked, the user is presented with the actual image of the plant captured on the given day where they can also add or update a comment. For evaluation purposes, a stand-alone version of the tool has temporarily been hosted at one of KU Leuven servers:
http://picasso.experiments.cs.kuleuven.be:3327/.

Data

All of the data used in this tool has been supplied by INRAE and CNR. The variables are as follows:

- uri: the unique URI of an image on the PhenoArch platform.
- real label: the number of leaves in an image, counted by humans.
- predicted label: the number of leaves in an image, counted by a machine learning approach (developed by CNR).
- date: the date on which the image of the plant was captured
- plant_alias: a unique identifier of the plant. It is a composition of several indicators, including genotype, experiment ID, plant ID and car number).
- genotype: the genotype of the plant (e.g. "TP-GRE" for the variety GRENACHE and TP-SYR for SYRAH).
- scenario: the type of experimental scenario (e.g. well-watered, water deficit and severe water deficit).

2.1.3.4. Pilot 3: water availability and irrigation recommendations

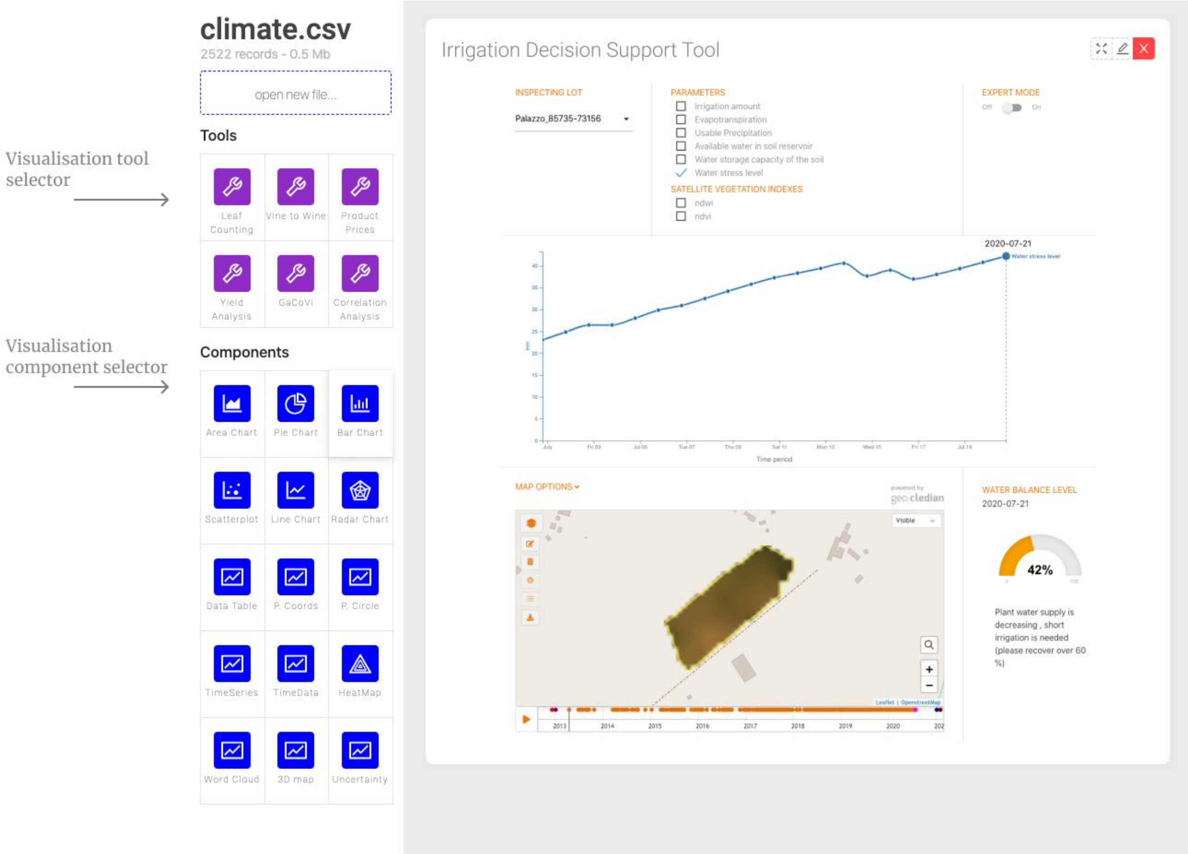

Figure 18: Water availability and irrigation recommendations visualisation for pilot 3

Description

This tool is designed to address a requirement proposed by pilot 3 (Abaco): water availability and irrigation recommendations. It aims to provide decision supports in optimisation of irrigation requirements, one of the best practices that has a high impact economically on food chain industry. In fact, the water consumption in irrigation is connected to the knowledge of the real needs of the crops also in relation to final production quality and quantity. In this tool, we visualise a number of weather parameters received from the APIs of Abaco and provide irrigation recommendations. Users are able to select a plot to inspect and select a variety of weather and satellite parameters to visualise. For any given day, users can inspect the water balance level (on a gauge chart) and suggested irrigation amount by hovering on any data point. The map widget, developed by Geocledian, also visualises the selected plot on a map where satellite parameters (e.g. NDVI, NDWI, NDRI, etc.) are overlaid. For evaluation purposes, a stand-alone version of the tool has temporarily been hosted at one of KU Leuven servers: http://picasso.experiments.cs.kuleuven.be:3328/.

Data

All of the data used in this tool has been supplied by Abaco and Geocledian. The variables are as follows (satellite parameters received from Geocledian are coloured blue):

- AWC: available water content, measured in mm, represents the dimension of the water reservoir, the maximum capacity.
- RAW: Ready available water, measured in mm, is the usable reserve of the water that the plant can immediately use with the roots.
- PU: usable participation,measured in mm, is the rainfall that enter into the ground.
- Prec: daily precipitation value measured in mm.
- Et0: evapotranspiration from reference culture measured in mm.
- ETR: daily evapotranspiration measured in mm.
- H2DispStart: the available water at the start of the day measured in mm.
- H2DispEnd: the available water at the end of the day measured in mm.
- Irr: Irrigation amount to be supplied measured in mm
- WC = Water content, measured in mm, is calculated as H2DispEnd + PU + Irr - ETR
- Water stress level: water stress level, measured in %, is calculated as (1 - (H2DispEnd / WC)) * 100
- Tmin: minimum temperature measured in °C
- Tmax: maximum temperature measured in °C
- Wmax: maximum wind speed measured in $m*s^{-1}$
- Wmed: medium wind speed measured in $m*s^{-1}$
- URmax: maximum relative humidity measured in %
- URmin: minimum relative humidity measured in %
- NDVI: Normalized difference vegetation index
- NDWI: Normalized difference water index

2.1.3.5. Pilot 4: grapevine By-Products Biological Efficacy Predictor

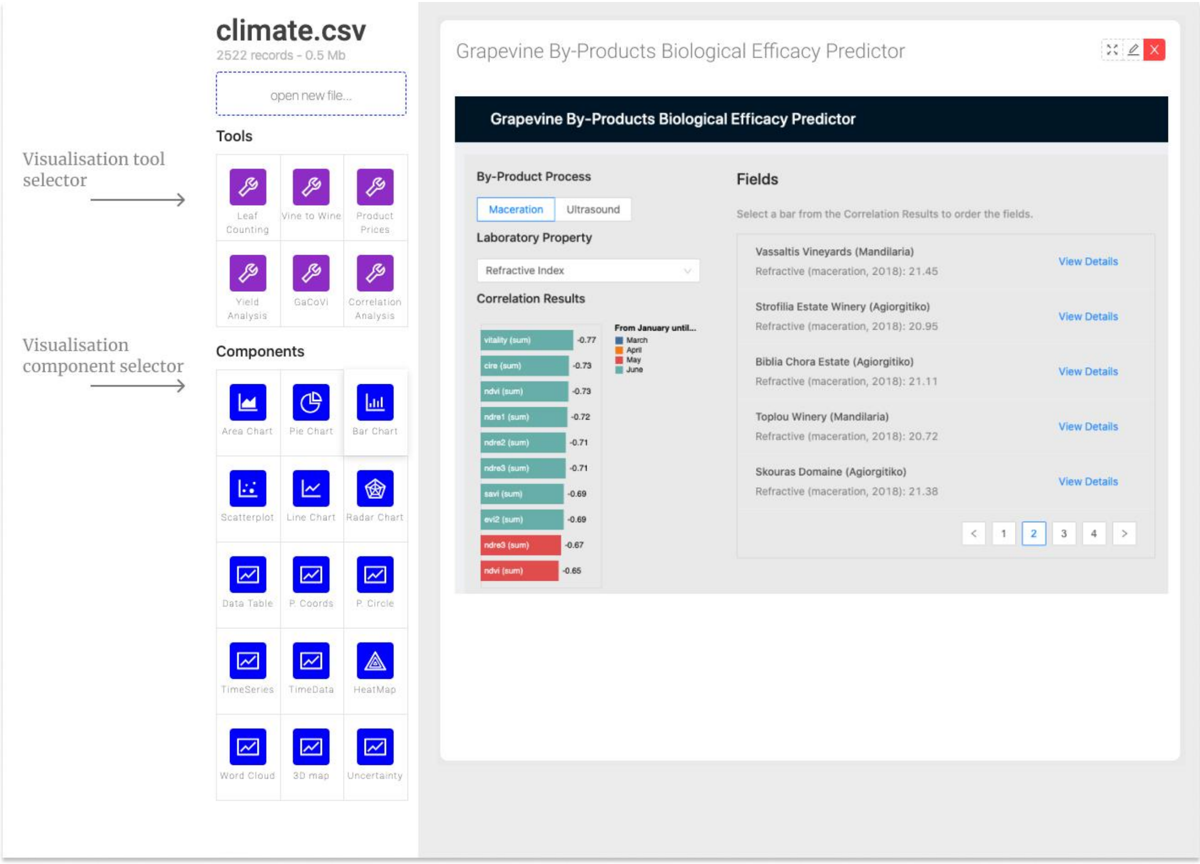

Figure 19: Grapevine by-product biological efficacy predictor for pilot 4

Description

This tool is designed to address the requirement proposed by pilot 4 (Symbeeosis): Grapevine by-products biological efficacy predictor. It aims to predict the origin of vineyards that could supply the best quality leaves for next year's natural cosmetics production, prior to performing the laboratory analyses of the grapevine by-products. It leverages the correlations founds between the laboratory results and satellite indexes to gain insight on the next year's grapevine by-products. It is designed for grapevine practitioners and cosmetic industry end-users. For evaluation purposes, a stand-alone version of the tool has temporarily been hosted at one of KU Leuven servers: http://picasso.experiments.cs.kuleuven.be:3620/.

The dashboard is integrated by the following components:

- Inputs: allow users to select a laboratory property (e.g. pH, reflective index, antioxidant activity etc.) and an extraction method (after maceration or ultrasound assisted extraction) that they want to focus on.
- Bar Chart: shows the ten satellite parameters that correlate with the selected laboratory property. Each satellite parameter measurements are aggregated on different time periods that go from January until March, April, May, or June; coloured blue, orange, red and green, respectively. The used aggregation function (minimum, maximum, mean, or sum) appears next to the satellite index name. The bars can be selected to order the Field List based on the value of the satellite parameter using the corresponding time-period and aggregation function.

- Fields List: shows the available Fields with their last available value for the selected laboratory property. If a bar from the Bar Chart is selected, the Fields list will be ordered based on its value on ascending order if the correlation is negative or descending order if the correlation is positive.
- Field Detail Menu
    - Map: shows the vineyard location.
    - Correlation Table: shows the value of the selected laboratory property, and the value of the satellite index for the previous years and current year.
    - Laboratory Analysis Table: shows the results of all the available properties from the laboratory analysis carried out in previous years.

Data

The data includes:

- Two years results of the laboratory analysis on the biological activity of two different extracts from leaves (after maceration or ultrasound assisted extraction) for 16 Greek vineyards (private/industry data) provided by one of the pilot partners, Symbeeosis.
- Satellite vegetation indices data (public/open data) supplied by another project partner, Geocledian.
- The correlations between the laboratory analysis results and the satellite vegetation indices have been provided by CNR, another technical partner in the project.

In particular, this includes the following variables from the laboratory analysis:

- pH
- Refractive Index
- Total microbial count (CFU/g)
- Yeasts and moulds (CFU/g)
- Antioxidant activity DPPH (Mg/mL trolox)
- Antioxidant activity ABTS (Mg/mL trolox)
- Total phenolic content, TPC (Mg/mL gallic acid)
- Total flavonoid content, TFC (Mg/mL quercetin)

and the following satellite indexes:

- ndvi - Normalized Difference Vegetation Index
- ndre1 - Normalized Difference Red Edge Index (v1)
- ndre2 - Normalized Difference Red Edge Index (v2)
- ndwi - Normalized Difference Water Index
- savi - Soil Adjusted Vegetation Index
- evi2 - Enhanced Vegetation Index 2
- cire - Chlorophyll Index - Red Edge
- npcri - Normalized Pigment Chlorophyll Ratio Index

2.1.3.6. Pilot 5: price prediction

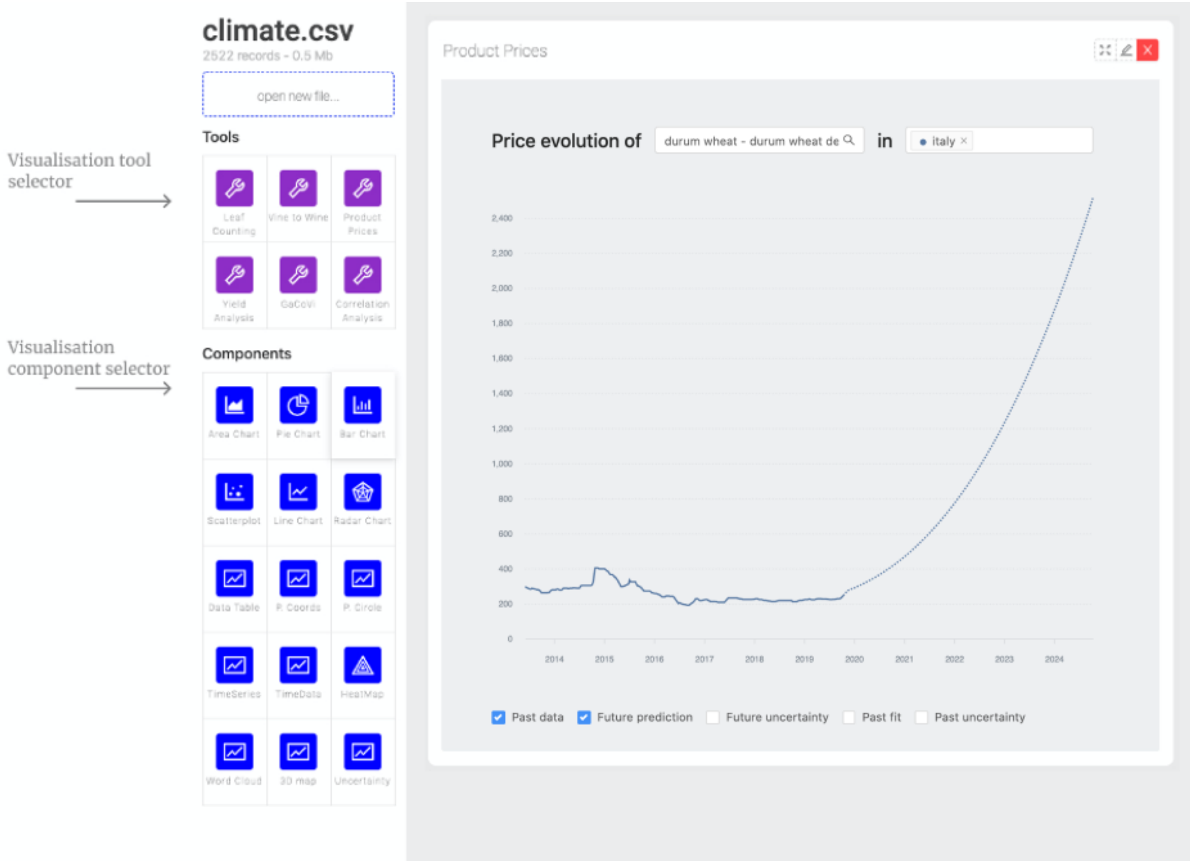

Figure 20: Price prediction visualisation for pilot 5

Description

This tool is designed to address a requirement proposed by pilot 5 (Agroknow): price prediction. Selling price of food products can be affected by various factors and can change over time depending on the country of origin. The selling price variations can also provide an indication for the potential fraud and adulteration of an agricultural product. This tool was designed to answer the questions such as "how likely will the price of this product increase or decrease over time?", "how did the price of this product evolve over time?" and "how does the price of this product vary between different countries/regions?". Users can select a product and a country (or countries) of their interest from the lists. They can also select the "Future prediction" checkbox to observe the prediction for the price trend for the next 5 years. Uncertainties in this prediction can also be seen by selecting the "Future uncertainty" checkbox. The remaining checkboxes, "Past fit" and "Past uncertainty", allow users to also visualise mean and variations in the historical data. For evaluation purposes, a stand-alone version of the tool has temporarily been hosted at one of KU Leuven servers: http://picasso.experiments.cs.kuleuven.be:3541/.

We used a simple linear regression model in this version due to its simplicity. In the future iteration, we plan to integrate with a more advanced model, LSTM (long short-term memory) model which performs significantly better than linear regression and is currently being developed by CNR for the analytics layer of the BigDataGrapes platform.

Data

All of the data used in this tool has been supplied by Agroknow. The dataset contains a collection of product prices from various European countries over the past two decades. In total, the dataset contains 352,590 entries for over 400 products in various European countries. The dataset includes the following variables:

- country: the country in which the given product price was obtained.
- dataSource: the source from which the price data was obtained.
- price: the price of the product,
- priceDate: contains a date object from the moment the price data was collected
    - contents of priceDate object are: {millisSinceEpoch,daysSinceEpoch, month, year, day}
- priceStringDate: contains a string with the date of data collection.
- price_id: a unique id for the data entry.
- product: the name of the product.
- url: the web address from which the data point was obtained.

2.1.3.7. Pilot 5: recall prediction

There are 2 additional tools contributed by the pilot 5 and project leader, Agroknow, for their FOODAKAI platform . These are: 1) recall prediction and 2) risk assessment tools for food products. These tools are designed to support an increasing requirement for recall prediction and product risk assessment tasks in the food supply chain. These tools compliment the price prediction tool we have developed, and together they contribute towards pilot 5. We present the first tool (recall prediction) in this sub-section and the second tool (risk assessment) in the next sub-section. Please note that because both recall prediction and risk assessment tools are developed for the FOODAKAI platform, they are not available within the container platform we have developed; they are integrated and available within the FOODAKAI platform: https://www.foodakai.com/.

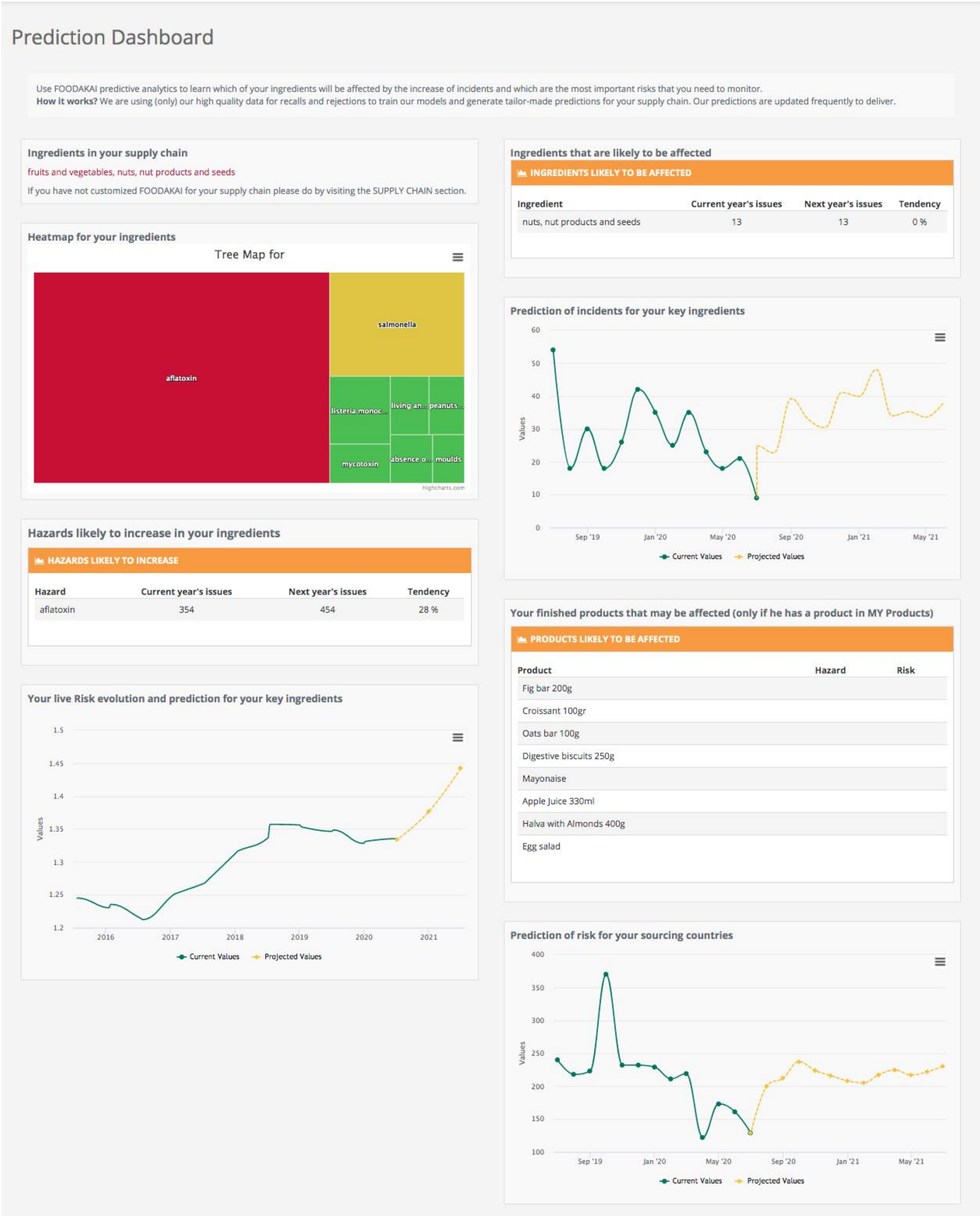

Figure 21: Recall prediction visualisation for pilot 5

Description

This tool is designed to address the recall prediction task proposed by pilot 5 (Agroknow). It is designed to help users understand which are the most important risks of food products that they need to monitor. Reliable data for recalls and rejections are used to train the prediction models and to generate tailor-made predictions for the supply chain. The dashboard includes blocks that provide:

- Prediction of the incidents trend for the key ingredients used in the supply chain. The predictions were computed using prediction models that are trend with tailor-made datasets for the specific ingredients. The actual and predicted values are displayed in a line chart.
- Prediction of the ingredients risk that will be increased due to the increase of the number of incidents for the ingredients. The actual and predicted values are displayed in a line chart.
- A table with ingredients for which the incidents are likely to be increased within the next months. The table presents both the current year incidents and the anticipated incidents for the next year.
- A table with hazards that are likely to be increased in the ingredients that are being used by the user's company. The table presents both the current year incidents and the anticipated incidents for the next year.
- The risk treemap for the ingredients highlighting the top risk hazards based on the current and predicted values.
- Finished products that will be affected due to the increase of the risk in ingredients used in the products
- Prediction for the risk of the countries or region from which the user's company is sourcing ingredients. The actual and predicted values are displayed in a line chart.

The predictions for the incidents and for the specific hazards are produced using the recalls prediction algorithms that were developed in the context of the Big Data Grapes project by Agroknow. This tool is available in the FOODAKAI platform and will be evaluated with end-users during the evaluation sessions that are planned as part of the project.

Data

We extracted data from textual information that includes mainly announcements about food recalls and border rejections (FDA, RASAFF, FSSA, AUSTRALIAN FOOD SAFETY AUTHORITY). We also used numeric data such as lab test results and prices. The main source of our datasets is the open data published by the governments.

2.1.3.8. Pilot 5: risk assessment

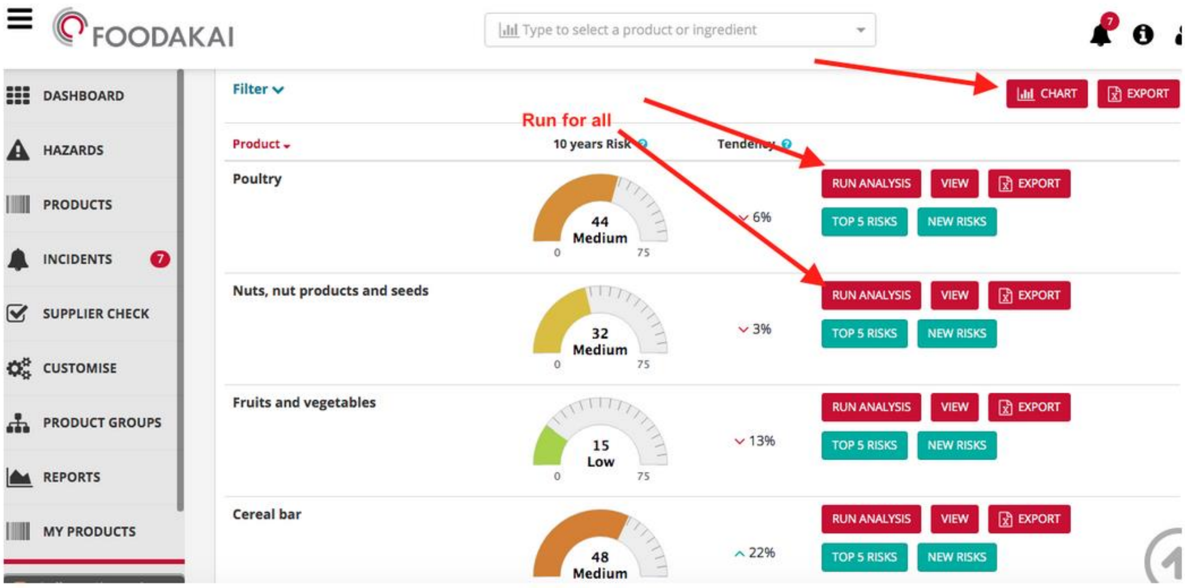

Figure 22: Risk assessment visualisation for pilot 5

Description
This tool is designed to address the risk assessment task proposed by pilot 5 (Agroknow). It aims to help users perform a risk assessment for products based on their ingredients. One can focus on specific regions by selecting the corresponding data source. The main goal of the tool is to help food safety experts to identify the ingredients with unacceptable hazard risk. The risk for each ingredient is estimated in relation to a specific hazard for the ingredient (see the gauge charts), e.g. risk for aflatoxins in peanuts, risk for pesticides in peanuts, risk for salmonella in peanuts, etc.

The risk assessment module focuses both on the reputation of companies and on the impact of the specific hazard on the public health. Regarding the reputation, the risk assessment model includes a severity factor which is based on how bad a specific type of notification is for the reputation of the company. The most severe is the recall of a product from the market. As regards the impact on the public health, we are using standard classification of the severity defined and used by agencies and authorities like the Food and Drug Administration (FDA) of the United States.

With this tool, users can see the emerging (new) and increasing (tendency) risks for raw materials like figs. The risk estimation approach being used is ranking the risk for raw materials, products groups and individual products. This tool is available in the FOODAKAI platform and will be evaluated with end-users during the evaluation sessions that are planned as part of the project.

Data
The risk estimation is based on the frequency of the food safety incidents that Agroknow has collected and processed. It should be noted that the risk is continuously being updated (live risk) every time new food safety incidents are collected from announcements. Sources include FDA, RASAFF, FSSA and AUSTRALIAN FOOD SAFETY AUTHORITY, and the data are processed by Agroknow.

2.2. DEVELOPMENT TECHNOLOGY

The technologies used to build the container platform and its contents are Meteor, React[24], Vega-Lite[25], Chart.js[26] and D3.js[27]. Meteor[28] is a full stack framework to build Web Applications in JavaScript. It integrates with MongoDB and uses a Distributed Data Protocol together with a publish–subscribe pattern to spread data changes from server to clients without requiring writing complex synchronisation code.

React is a JavaScript library maintained by Facebook for building user interfaces and has recently become extremely popular. React can be used to build Web Applications; it is optimal for fetching reactive data in modern complex applications. The use of React is in line with the expertise of many project partner institutions. The development team at Agroknow, for example, has already been using React for the FOODAKAI application.

We reused the Chart.js library from the previous versions in order to rebuild some of the existing visualisation components. In addition to Chart.js, we also started using a number of other visualisation libraries such as D3.js and Vega-Lite. D3.js is a popular JavaScript library that contains a collection of well-known visualisation components that are freely customisable. Vega-Lite is a high-level grammar of interactive graphics that enables the agile specification of data visualisations. Vega-Lite combines a traditional grammar of graphics, providing visual encoding rules and logic composition for a layered display of information, following a well-defined syntax of interaction. Both D3.js and Vega-Lite libraries offer a number of additional visualisations that compliment Chart.js.

[24] https://reactjs.org/
[25] https://vega.github.io/vega-lite/
[26] https://www.chartjs.org/
[27] https://d3js.org/
[28] https://www.meteor.com/

3. USER MANUAL

3.1. DOWNLOADING THE CODE

The code for this version of deliverable 5.1 has been uploaded to the BigDataGrapes GitHub repository: https://github.com/BigDataGrapes-EU/Deliverable-5.1-v3.

One can follow the following steps to download and utilise the tool.

1. Download meteor framework from https://www.meteor.com/install

2. Install the framework following the instructions provided.

3. Download or clone the project:

```
$ git clone https://github.com/BigDataGrapes-EU/Deliverable-5.1-v3.git
```

3.2. RUNNING THE APPLICATION

1. Navigate to the cloned/downloaded folder:

```
$ cd Deliverable-5.1-v3
```

2. Run the project using the meteor command, this will automatically install all the dependencies needed and run the project.

```
$ meteor
```

3. In case of missing dependencies simply run:

```
$ meteor npm install
```

4. Open the application in browser http://localhost:3000/

4. CONCLUSIONS

In this document, we presented a new version of deliverable 5.1: Interactive Visualisation Components, produced under work package 5. In the previous versions, we showcased a number of individual visualisation components, iteratively improving the existing ones while adding new components. We now have reached to a point of building complex dashboards with multiple components that are tailored to meet the requirements of individual pilot partners. For the 5 pilots in this project, we presented a total of 8 dashboards, each designed to support a particular task proposed by the pilots. All of the codes have been published at the GitHub repository of BigDataGrapes (https://github.com/BigDataGrapes-EU/Deliverable-5.1-v3)

All of the new dashboards are currently being evaluated with real end-users recruited by each pilot partner. In the next months, the dashboards will be improved further using the feedback received from evaluation. We will also migrate from using local datasets to using the APIs to receive data from the BigDataGrapes platform which are currently being contributed by both pilot and technology partners. In addition to data, since the analytics layer will also be within the BigDataGrapes platform, we will have a chance to better streamline the way our dashboards work. After these improvements, the dashboards will be re-evaluated the second time with end-users.