SMIS WEB INTERFACE REPORT

V1.0

TABLE OF CONTENTS

REVISION HISTORY

Version	Description	Date	Author
1.0	Original version of the Web application interface report covering the data access and browsing options in the SMIS Analytics Toolkit.	30/07/2018	Tomasz Kurowski

ABBREVIATIONS

API Application Programming Interface

OS Ordnance Survey

REST Representational State Transfer

SMIS Soil Management Information System

SMIS AT SMIS Analytics Toolkit

1. INTRODUCTION

1.1. PURPOSE

The following document describes the design of the interface of the Web application developed as part of the Soil Management Information System (SMIS) project, referred to as the SMIS Analytics Toolkit (SMIS AT). This Web application is the platform for user interaction with data collected during the SMIS project, and includes features for data browsing, data analysis using machine learning methods, and the visualisation of both the analysis results and summaries of the collected data sets. This Web front-end depends on a database system and a representational state transfer application programming interface back-end (REST API), called the SMIS API, both described in separate documents (the *Database Technical Documentation* and the *Technical documentation and user manual report*), to deliver those functionalities.

The purpose of this document is to provide an overview of the front-end interface design, its structure and navigation options, without focusing on the technical details of the implementation.

It should be noted that certain parts of the interface remain in the final stages of development, particularly regarding the inclusion of additional Established Queries currently being tested and actions based on user feedback received during SMIS presentations. This document will be updated to include any further changes to the interface before the hand-over of the system to ensure the thoroughness and accuracy of the documentation.

1.2. SCOPE

This document focuses on the interface design of the SMIS Analytics Toolkit, particularly the structure and navigation of its Web pages. It describes the available views of collected data and the options for browsing it.

Implementation details, such as how the SMIS AT interacts with other components of the project, are not discussed, and views of the analytics and visualisation functionalities are only presented from the perspective of navigation between views (e.g. accessing details about a specific variable) rather than general usage.

2. DESIGN, STRUCTURE AND NAVIGATION

The SMIS Analytics Toolkit was designed to provide a platform for exploring the data collected as part of the SMIS project, as well as to provide tools used to and interactively visualise patterns related to areas of interest, such as optimal soil management or factors affecting yield, using machine learning methods.

The interface does not depend on the users to provide their own data sets to analyse, but instead relies on a common, curated (and, for grower data, anonymised) database to provide users with a "Big Data" type of wide basis for analysis, which can be then interactively filtered through the interface to explore more specific scenarios of interest to each particular user. One consequence of this approach, with anonymised data shared by all users, is that SMIS maintains no user access control (i.e. persistent user accounts) subsystem as part of its Web interface. While this makes the SMIS AT simpler in operation, any desired restrictions to access need to be provided by the server (e.g. by only hosting the application on a local network or integrating it in a larger system).

While the SMIS AT does not manage user accounts, it does maintain certain persistent settings through session cookies. For example, if a user re-arranges the order of columns in a table or hides certain columns, this new arrangement will be retained in that user's session and will remain until cookies expire or are cleared (or a "Restore defaults" button is pressed; such options are described for individual views in section 3). A standard cookie notification is displayed as required by privacy legislation; if rejected, the settings are no longer persistent for the user.

At several points in this document, "default" behaviour of certain elements of the interface is mentioned. This refers to behaviour which can be modified either individually by each user or by an administrator (affecting all users). As SMIS application administration is not done through the Web interface but depends on the usage of external scripts and configuration files, those administrator-specific means of applying interface changes are not discussed in this document, although the effect of the various options is still described in the relevant sections.

2.1. WEB INTERFACE STRUCTURE

The SMIS AT Web interface follows a simple design with the page subdivided into three sections. The top banner contains logos of SMIS and the Agriculture and Horticulture Development Board (AHDB), both usable as hyperlinks redirecting the user to the starting view of the SMIS AT interface (by default the grower data browser described in section 3.2.1) and the AHDB homepage respectively.

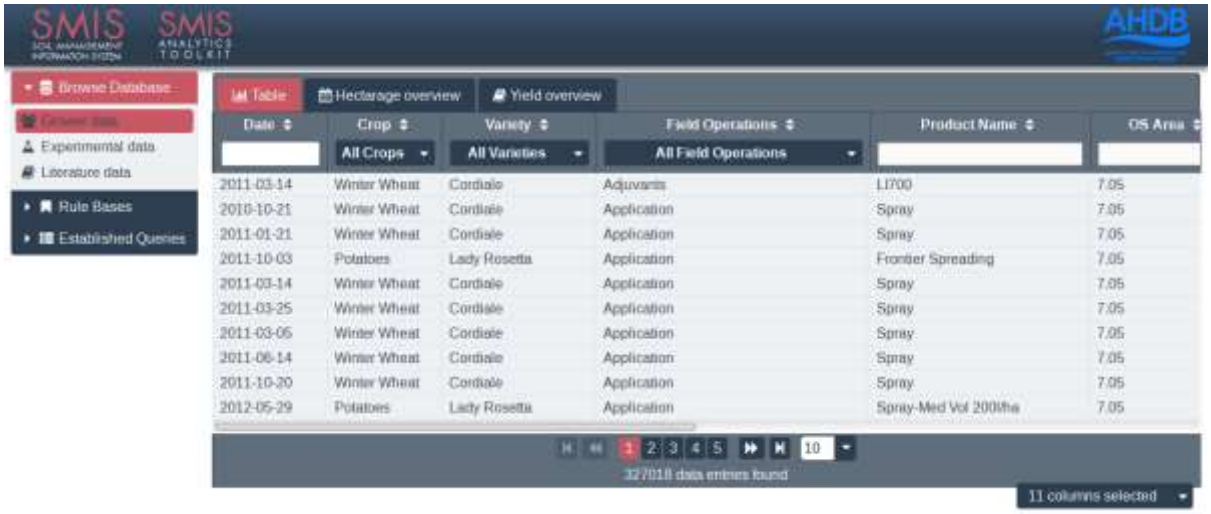


FIGURE 1: GENERAL SMIS INTERFACE OVERVIEW.

The area of top bar can also be used to display static content (such as a session cookie notification; this is the only usage enabled by default) or dynamic notifications produced by visualisation tools or other components of the Web interface; the latter functionality is controllable by an administrator and disabled by default.

The central content area taking up most of the interface and flanked by the main menu on the left is used to display the various available views as described in section 3.

The left side of the interface contains a collapsible panel menu which provides the main navigation functionality between the views and is described in section 2.2.

It should be noted that while most technical interface composition and component library choices (not discussed here) were made with adaptability in mind and the interface could potentially be converted to adapt to touch screen and smaller display sizes (e.g. smartphones), the present implementation was designed and optimised for larger monitors as available on laptops or desktop computers through compatible Web browsers (Chrome, Firefox, and Safari were verified to provide full features in testing).

2.2. WEB INTERFACE NAVIGATION

Primary navigation through the SMIS Analytics Toolkit is handled through a collapsible panel menu on the left side of the interface, where the available views are grouped into three categories:

- Browse Database – contains views which allow for browsing and displaying overviews of the collected data.

- Rule Bases – contains views which allow for constructing generic queries of the database and browsing previously generated queries.
- Established Queries – contains views corresponding to SMIS database queries which were verified to provide useful results and were given their own interface and custom visualisation options.

FIGURE 2: COLLAPSIBLE NAVIGATION MENU PANEL.

The views corresponding to the individual entries in the three categories are shown and described in section 3.

3. INTERFACE VIEWS

The following sections describe the individual views available within the SMIS AT Web interface which are displayed in the central part of the Web page. Navigation between these views is handled through the main navigation tab described in section 2.1. The individual view descriptions below are grouped into sets as in the navigation tab. The information and functionalities available through each view are described alongside any options for navigating within the view (e.g. where its contents are subdivided into multiple tabs or alternatives views) or manipulating its contents (e.g. sorting or filtering data displayed in a table). Each section also contains an example screenshot demonstrating its respective view.

3.1. GENERIC VIEW COMPONENTS

Certain interface elements were developed as generic (but customisable) components which can be re-used in multiple views. This has made the building of interfaces simpler and can facilitate the addition of further views (such as new Established Queries) if the system would be developed further using additional analytics options. Implementation details of such components are to be described in a separate technical report. Their general look and use options of a selection of components is also described below to avoid redundant descriptions in each interface which uses those components.

3.1.1. DATA TABLE FILTERING INTERFACE

This is a paginated data table component which supports filtering displayed data and allows the user to select and re-order the displayed columns from a list dependent on the dataset used and certain configuration settings (certain variables may be included in the database but hidden from general users, or merely hidden by default).

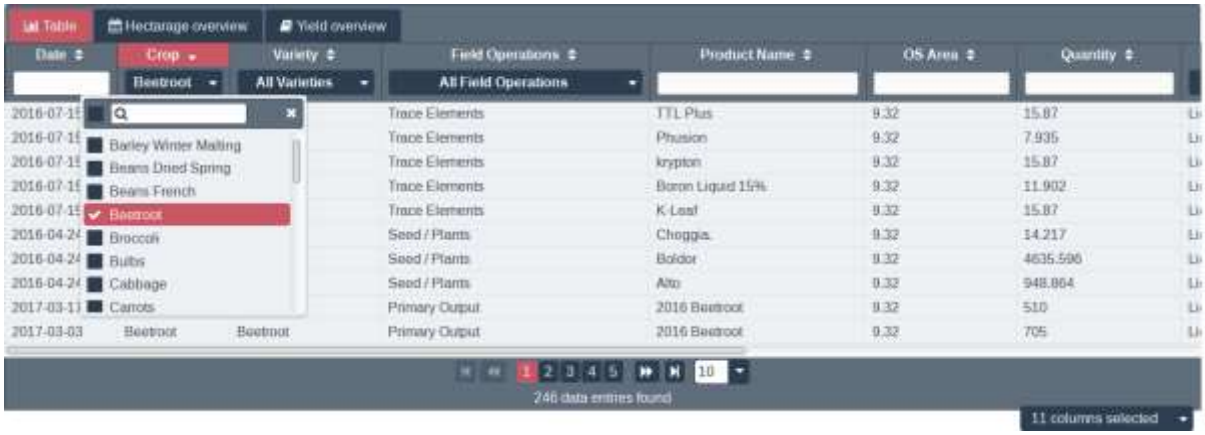


FIGURE 3: EXAMPLE OF A DATA TABLE FILTERING INTERFACE WITH MULTISELECT (CATEGORICAL) FILTERING OPTIONS FOR THE "CROP" VARIABLE SELECTED.

The columns can be hidden or shown using a checkbox interface on the lower right (showing the number of selected columns by default). The columns can also be re-arranged by dragging. The column order and selection for a user are remembered via session cookies. The default column arrangement can be restored using a "Restore defaults" button in the column selector on the lower right.

Each column has an associated type (set by an administrator) which controls the filtering options available. The four column types are:

- Text (default type) – after clicking the column header the user can enter a (case insensitive) search query which is used to filter the data.
- Date (used for dates) – after clicking the column header the user can select a date range from a calendar.
- Numerical (used for quantitative data) – after clicking the column header the user can select a number range between the lowest and highest values available for that column.
- Multiselect (used for categorical data) – after clicking the column header the user can select any number of values from a searchable list by clicking individual check boxes.

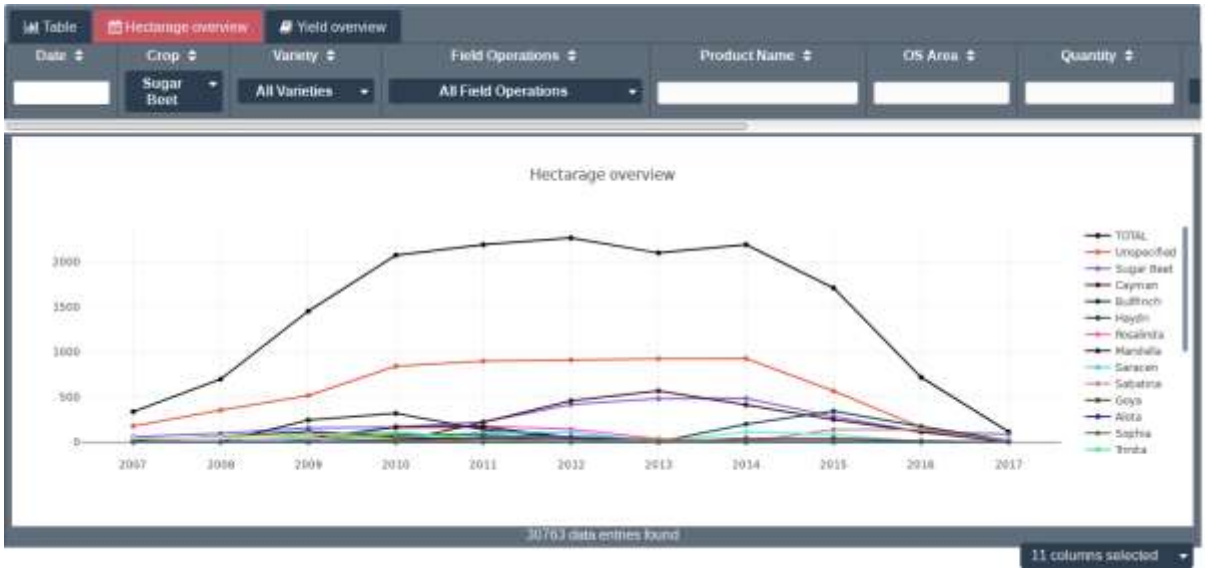


FIGURE 4: EXAMPLE OF A DATA TABLE FILTERING INTERFACE SWITCHED TO AN ALTERNATIVE SUB-VIEW (COLUMN HEADERS AND FILTERING OPTIONS REMAIN VISIBLE).

Any combination of search options can be entered into any combination of columns, which impacts the amount of data displayed in the table and the hectarage/yield overviews. The amount of data is also displayed at the bottom of the view, either in terms of the number of "data entries" (rows) or by some custom measure derived from the data (e.g. the hectarage represented by the filtered entries for grower data).

Additional sub-views can be connected to the filterable table and are generally accessible through tabs on top of the interface. The filtered state of the data is preserved between those sub-views and the table headers and filtering options remain visible, allowing for adjustment of the filtering options. The sub-views are generally specialized visualizations of the selected data, dynamically generated and updated through filtering.

3.1.2. QUERY EDITOR INTERFACE

The query editor interface allows users to build queries to be used in machine learning analytics. A user can select the dependent variable being modelled and a list of variables to be included in the modelling (including both collected data variables viewable in data browsing views and derivative variables such as previous crops in a rotational context). It should be noted that variables go through a feature selection step as part of the analytics, so any (or potentially all, resulting in a failed query) of the selected variables might ultimately be removed from the resulting analysis.

The user can also add filters to the query, limiting the data used to particular datasets, e.g. ones pertaining to a specific crop or soil type.

An “unrestricted” query builder interface is used in the rule base part of the application (see section 3.3.1) where it can be used for exploratory interrogation of the collected data sets, allowing for modelling scenarios other than the ones implemented as separate Established Queries at the cost of requiring the user to manually edit a more complex query.

Many queries which are possible to construct may not be theoretically meaningful and some of those which are may not give significant results; meaningful queries with enough collected data to provide significant results were used for developing Established Query views, which can provide custom visualisations while taking the burden of constructing a complicated query off the user.

Established Query interfaces use customised versions of the query editor, with the modelled variable (such as yield or compaction) and relevant independent variables to be used being pre-selected by the developers and unavailable for user edits. Filtering options (on the pre-selected set of variables) remain available to users and are the primary means of user interaction next to the custom interactive visualisations developed for each specific Established Query. Indeed, some filtering options may be mandatory (e.g. Crop selection for yield modelling), though this depends on the Established Query.

3.2. DATABASE BROWSER VIEWS

3.2.1. GROWER DATA BROWSER

Purpose: Browsing collected grower data (including filtering it by selected columns) and displaying overviews. The primary function is to inform the user about the volume of data available for specific scenarios without needing to manually verify their viability in a query constructor. Additionally, overviews of yield data (e.g. grouped by year, variety, or soil type) present in collected grower data may be informative on its own.

Navigation and user interaction: By default, the user sees an overview of all the collected grower data in a paginated tabular form (as described in section 3.1.1). There are three available sub-views (the default tabular view, a hectarage overview, and a yield overview), although all three are controlled by a single interface dependent on the tabular display of data. Each data row (representing a single operation on a field) is described by a set of columns containing variables captured by the SMIS grower datasets. There are 44 variables currently stored for SMIS grower data, though not all are available for each row and only a subset is used for analytics. The number of variables is not hard-coded and will

freely expand as more extensive data is added to the system. The following selection of variables is displayed by default; it is controllable by SMIS administrators:

- Date
- Crop
- Variety
- Field Operations
- Product Name
- OS Area
- Quantity
- Texture
- Units
- Yield
- Yield Units

At the bottom of the table there is a number representing the amount of data included in the currently displayed (i.e. filtered) dataset in terms of rows or hectarage. The former option is the default, but it can be changed by an administrator.

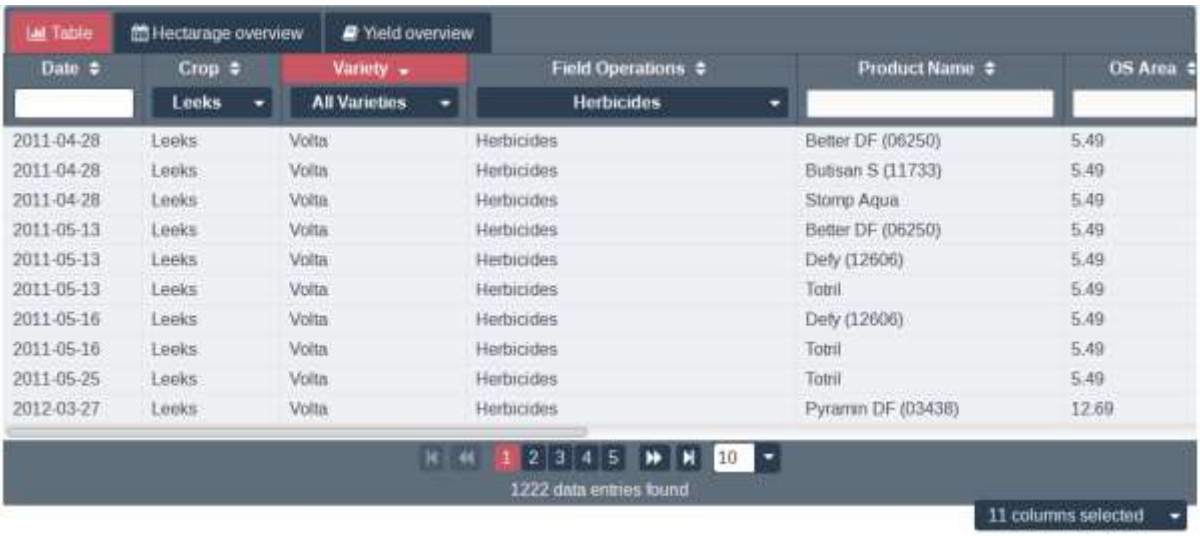


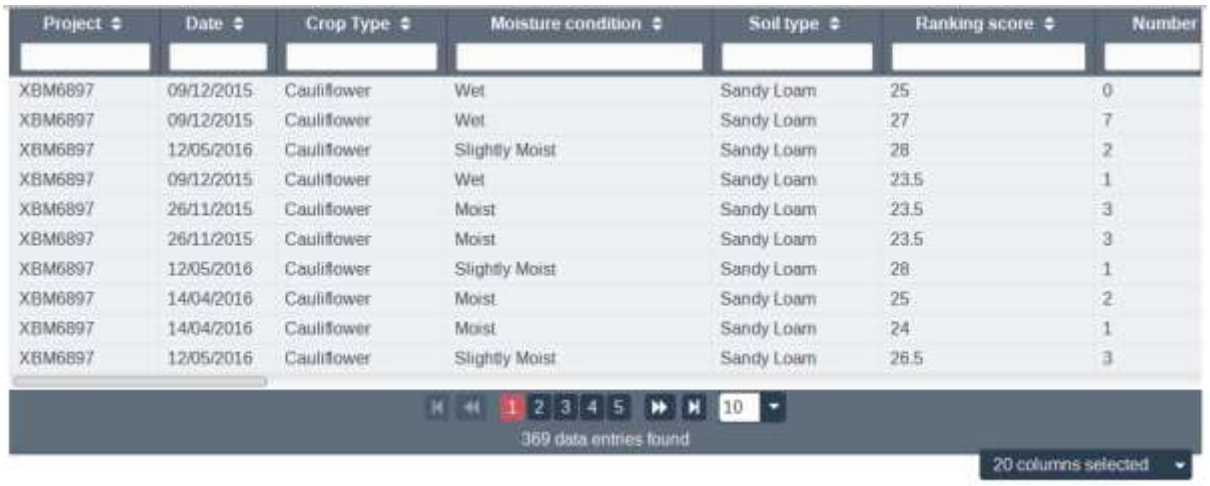
FIGURE 5: GROWER DATA BROWSER TABLE VIEW.

The two visual overviews of the data ("Hectarage overview" and "Yield overview") can be accessed by clicking the respective tabs on the top of the page. The tabular column headers remain visible in those overviews and the filter settings persist between the three available sub-views. The dynamically generated hectarage/yield overviews display data from the filtered datasets, broken down by crop or (if only one crop is selected) by variety. Details about each data point can be viewed by hovering the mouse over it.

3.2.2. EXPERIMENTAL DATA BROWSER

Purpose: Browsing collected experimental data (including filtering it by selected columns). The primary function is to inform the user about the types and volume of data available in the collected experimental datasets. Unlike with grower data, no special visualisation options are available for experimental data; it is included in rule base displays where relevant automatically.

Navigation and user interaction: The user can see an overview of all the collected experimental data in a paginated tabular form. Each data row (representing a single set of variables) is described by a set of columns containing variables captured by the SMIS experimental datasets. The columns used by different experimental datasets may be different and unused columns are simply left blank.



Project	Date	Crop Type	Moisture condition	Soil type	Ranking score	Number
XBM6897	09/12/2015	Cauliflower	Wet	Sandy Loam	25	0
XBM6897	09/12/2015	Cauliflower	Wet	Sandy Loam	27	7
XBM6897	12/05/2016	Cauliflower	Slightly Moist	Sandy Loam	28	2
XBM6897	09/12/2015	Cauliflower	Wet	Sandy Loam	23.5	1
XBM6897	26/11/2015	Cauliflower	Moist	Sandy Loam	23.5	3
XBM6897	26/11/2015	Cauliflower	Moist	Sandy Loam	23.5	3
XBM6897	12/05/2016	Cauliflower	Slightly Moist	Sandy Loam	28	1
XBM6897	14/04/2016	Cauliflower	Moist	Sandy Loam	25	2
XBM6897	14/04/2016	Cauliflower	Moist	Sandy Loam	24	1
XBM6897	12/05/2016	Cauliflower	Slightly Moist	Sandy Loam	26.5	3

1 2 3 4 5 10
369 data entries found
20 columns selected

FIGURE 6: EXPERIMENTAL DATA BROWSER TABLE VIEW.

3.2.3. LITERATURE DATA BROWSER

Purpose: Browsing the curated literature data (including filtering it by selected columns). The primary function is to inform the user about the types and volume of data available among the collected literature. Hyperlinks to the specific sources are also accessible for each entry.

Navigation and user interaction: The user can see an overview of all the collected literature data in a paginated tabular form. Each data row (representing a single captured conclusion identified in the literature) is described by a set of columns summarising the relevant information on the captured relationship contained in the paper. Each of the papers included in the table can be accessed by clicking the reference included in the "Citation" column, which opens a new page containing (at least) the relevant abstract on the publisher's page, if the paper is not accessible in its entirety.

The following selection of variables is displayed by default; it is controllable by SMIS administrators:

- Category
- Title
- Citation
- Country
- Crop
- Inherent factor
- Management solution
- Reference type

For some of the collected sources, summaries dealing with the contents of the paper directly related to the information of interest are available as (by default) hidden columns. Other than by enabling those (very large) columns, these summaries can also be viewed by hovering over the “Citation” value for the row.

FIGURE 7: LITERATURE DATA TABLE VIEW.

3.3. RULE BASE VIEWS

3.3.1. QUERY CONSTRUCTION INTERFACE

Purpose: Constructing generic queries to be run on the entirety of the collected data. The user can select any number of dependent variables to be modelled from among all the variables collected in the system. For each of the dependent variables, any number of independent variables can be selected. Once a query is successfully executed, the results are displayed as a table. The main purpose of the interface is to ‘prototype’ and test queries which can be then used to create an Established Query, which has a pre-defined dependent variable, a list of allowable independent variables, and additional, custom visualisation settings tailored for a specific application.

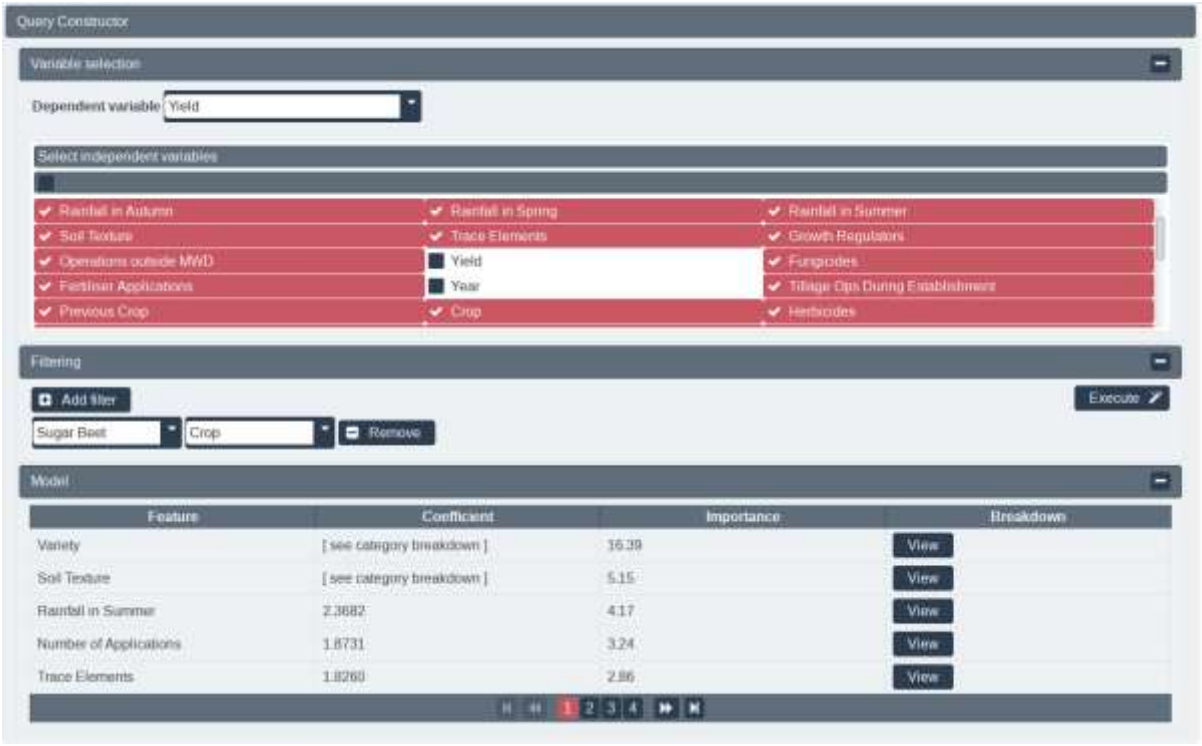


FIGURE 8: FULL-FEATURED QUERY CONSTRUCTOR INTERFACE.

Navigation and user interaction: The user must select a dependent variable from of a list of all variables using a drop-down menu and a list of independent variables to be used in modelling (all except the selected dependent variable are checked by default) from a box containing all variables contained in the database.

The user can also add filters by clicking the “Add” button. This creates a pair of drop-down menus which allow the user to select a variable and value to be used, creating a more specific query. It should be noted that only filters corresponding to selected modelling variables will have any impact on the results.

By clicking the “Execute” button, a model for the selected variables and filters is generated. It should be noted that this can take much longer than for Established Queries, which depend on pre-generated collections of variables.

The results are displayed in a simple table with variables sorted by importance in the model. Unlike with Established Queries, no variable importance threshold is applied, so all selected variables which passed the initial variable selection stage in the modelling will be displayed, even if the significance is very low.

3.3.2. RULE BASE BROWSER

Purpose: Displaying the successfully generated models as a filterable graph containing all the identified connections between collected variables. Connections also reflected in the collected Literature or Experimental data are annotated with those datasets (which can be viewed when the relevant edge is clicked). By default, only the results of Established Queries are displayed in the unfiltered graph. An administrator can also manually add the results of generic queries created by the query construction interface to this main view.

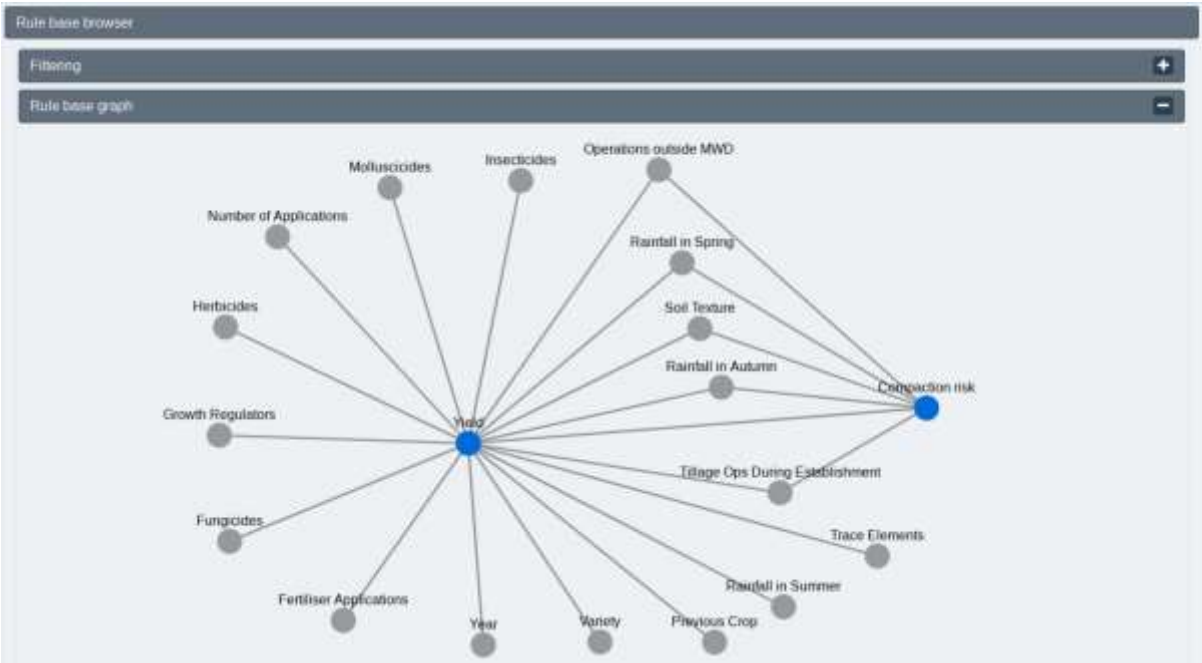


FIGURE 9: FILTERED RULE BROWSER GRAPH.

Navigation and user interaction: The user can view a graph containing all the variables included in generated modelling results as well as experimental and literature data as nodes connected by edges representing the identified relationships. A (filtering-only) query editor interface (see section 3.1.2) is available and can be used to restrict the view to connections of interest. Clicking graph edges shows the user a pop-up box identifying the modelling results or literature/experimental dataset the edge was based on. Clicking graph nodes shows the user the list of edges for that node.

3.4. ESTABLISHED QUERY VIEWS

3.4.1. FACTORS AFFECTING YIELD

Purpose: Allowing a user to generate a machine learning model of factors influencing yield for a specified crop (and, optionally, specific conditions such as soil type) and visualising the results, with

variables influencing yield for a specific crop and scenario (set of conditions) being ranked and displayed as a bar plot, with a more detailed breakdown available for categorical variables.

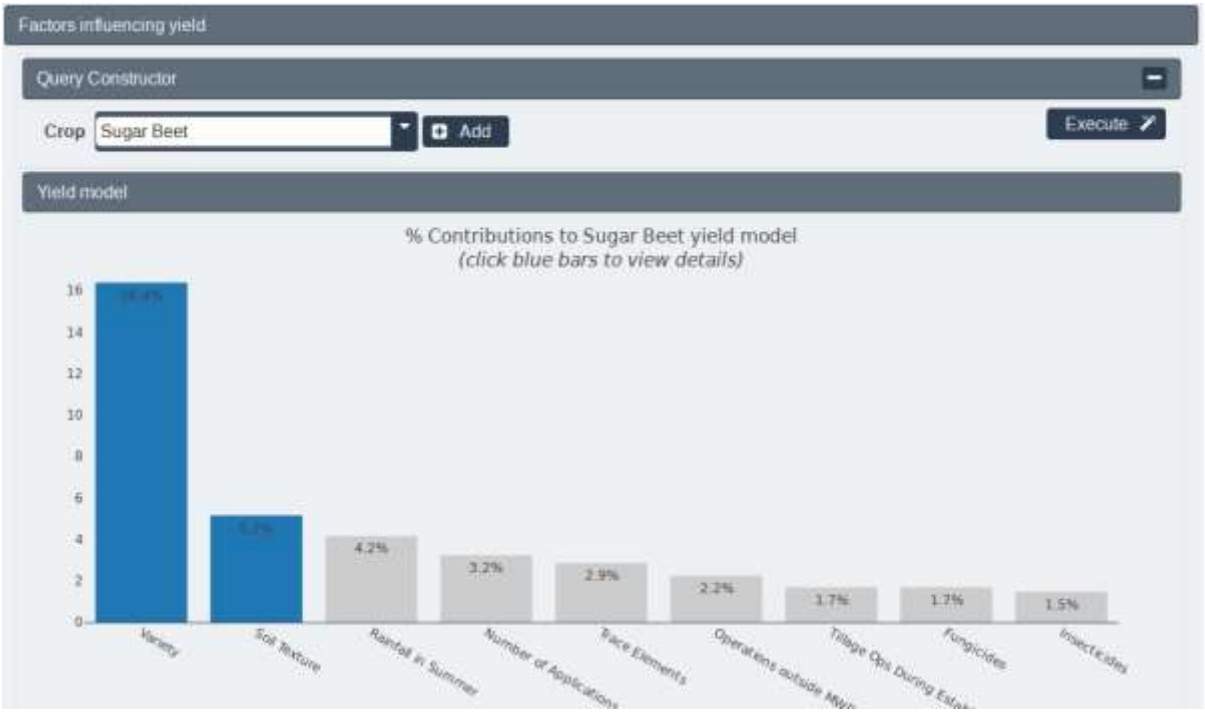


FIGURE 10: FACTORS AFFECTING YIELD VIEW, INCLUDING QUERY CONSTRUCTOR AND EXAMPLE MODEL.

Navigation and user interaction: A (filtering-only) query editor interface (see section 3.1.2) is available and can be used to filter the data used in the generation of a model. The "Crop" variable is mandatory as yield models can be selected only for a specific plot. The user selects the crop of interest from a drop-down menu and clicks the "Execute" button to generate a model and display its visualisation as a bar plot. Optionally, the user can add an additional filtering variable by clicking the "Add" button. This creates a pair of drop-down menus which allow the user to select a variable and value to be used, creating a more specific query.

Hovering the mouse pointer over a bar in a generated visualisation shows a window with a distribution (for numerical variables) or pie chart (for categorical variables) of the variable in the filtered dataset used for yield modelling. Bars representing categorical variables can be clicked to view a breakdown showing the relative impact (sorted from best to worst) of each category (e.g. crop variety) on yield. The user can return to the main view by clicking the "Back" button.

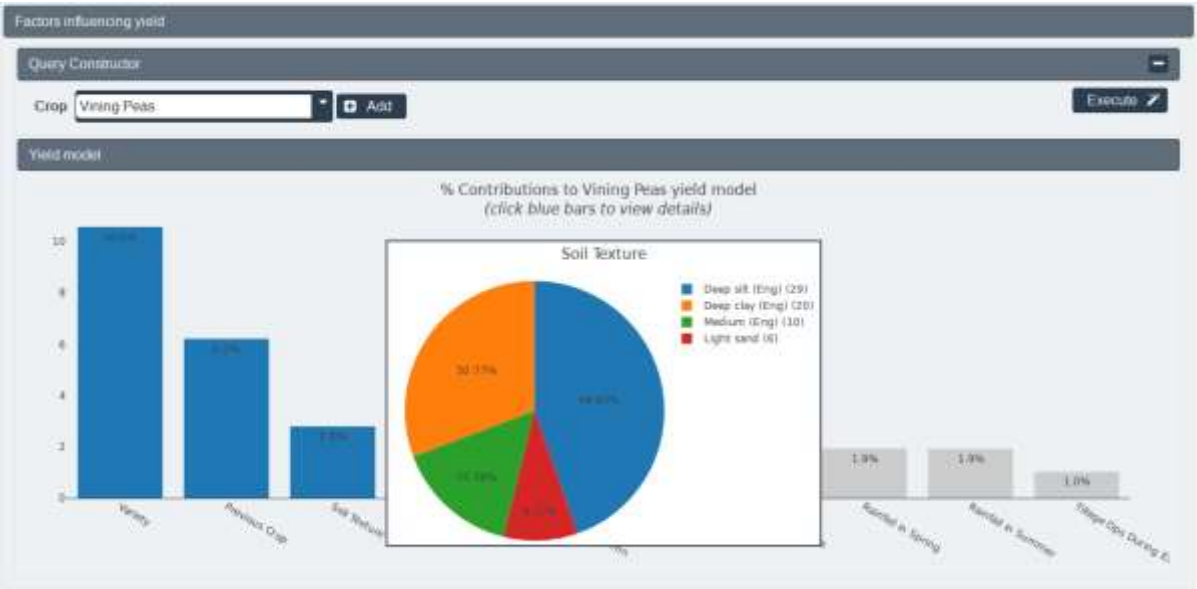


FIGURE 11: EXAMPLE OF A CATEGORICAL VARIABLE DISTRIBUTION PIE CHART.

3.4.2. COMPACTION RISK

Purpose: Allowing a user to generate a machine learning model of factors influencing compaction risk (optionally limiting the data to only specific scenarios, e.g. a single soil type) and visualising the results, with variables being ranked and displayed as a bar plot, with a more detailed breakdown available for categorical variables.

Navigation and user interaction: A (filtering-only) query editor interface (see section 3.1.2) is available and can be used to filter the data used in the generation of a model. There are no mandatory fields, so the modelling is run on all available data by default once the user clicks the "Execute" button, prompting the system to generate a model and display its visualisation as a bar plot. Optionally, the user can add an additional filtering variable by clicking the "Add" button. This creates a pair of drop-down menus which allow the user to select a variable and value to be used, creating a more specific query.

Hovering the mouse pointer over a bar in a generated visualisation shows a window with a distribution (for numerical variables) or pie chart (for categorical variables) of the variable in the filtered dataset used for compaction risk modelling. Bars representing categorical variables can be clicked to view a breakdown showing the relative impact (sorted from best to worst) of each category (e.g. soil type) on compaction risk. The user can return to the main view by clicking the "Back" button.

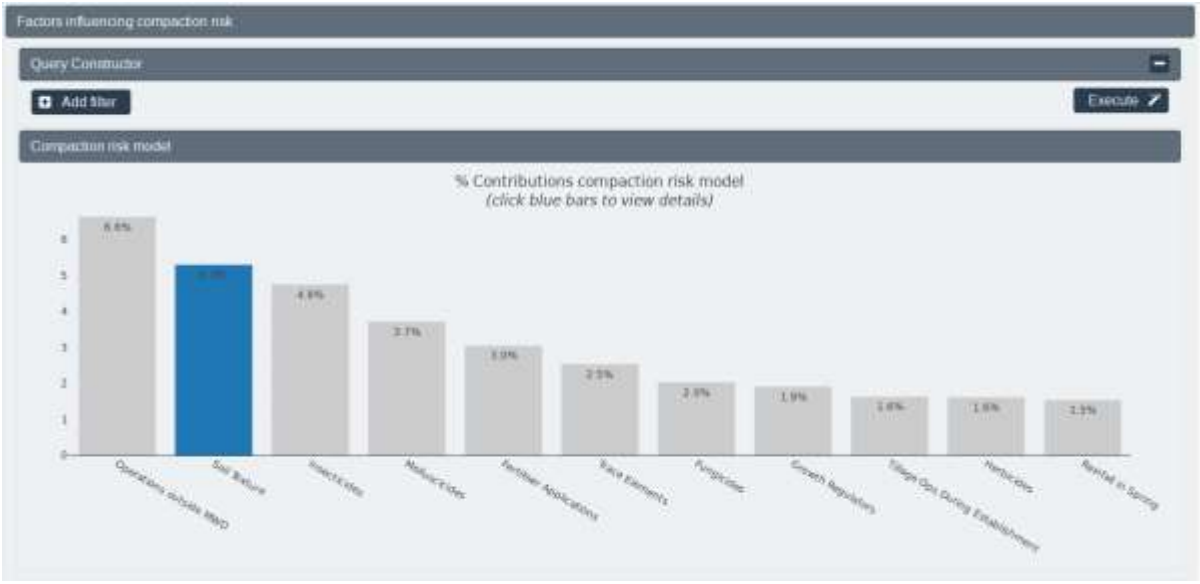


FIGURE 12: FACTORS INFLUENCING COMPACTION RISK VIEW (QUERY CONSTRUCTOR HIDDEN).