BUSINESS INTELLIGENCE COURSEWORK REPORT

Table of contents:

1. INTRODUCTION

This report aims to provide a detailed inside look into the front-end stage of a Business Intelligence (BI) fully integrated system developed for the purposes of a veterinary clinic in support of the veterinary experts' decision-making process. However, before looking into the details it is crucial to set the context and provide a quick description of the system leading up to this point. Finally, this report will go over a synopsis of the entirety of the system, outline its main advantages and disadvantages, discuss about potential future improvements and what could have been done differently.

1.1 Recap

So, as mentioned above, the scope of this project is to develop, implement and deploy a fully integrated BI, decision assistance tool, targeted towards a veterinary clinic. This is achieved by providing an easy access to the clinic's experts to the Food and Drug Administration's (FDA) database regarding drug-related adverse events where animals were involved.

The FDA has been recording cases from as early as 1987, and continuous on doing until today, and within this timeline it has successfully managed to amass a total of more than 1 million reported incidents, which cover hundreds of millions of individual animal, and human, cases with a great level of detail.

The back-end stage of the system is comprised of automated mechanisms for downloading, transforming, regulating, and loading the data into a fully operational, local, database build with PostgreSQL.

Firstly, a python script consumes the FDA's API and stores all of the raw data inside 5 different tables in the staging environment of the database. It follows a complex Extract, Transform and Load (ETL) pipeline which discards unwanted data, regulates data types, and imputes missing entries before inserting it into the temporary stage of the database. From there the data is loaded into the dimensional data warehouse (DW) environment where it is stored into a constellation schema containing a total of 5 fact and 5 dimension tables. In the DW the data is now regulated, normalized, and interconnected in such a manner to allow for complex queries without the need for data marts.

2. DATA VISUALIZATION

Data visualization is crucial for any BI system. All of the previous steps are rendered redundant without a suitable front-end environment to support them.

It is very important to make sure that the visualization satisfies all of the client's requirements. In this scenario especially, where the targeted client is a clinic comprised of basic computer users, it is essential to deliver an efficient, not complicated and visually appealing product. The latter where the requirements for the dashboard delivered.

2.1 The dashboard

The dashboard is the communication interface between the client and the database build in the back-end stage.

2.1.1 Metabase

For the purposes of this project the Metabase tool was chosen and more specifically, Metabase's free and local version. Metabase can be installed locally, with no extra costs, and allows for direct connectivity to a PostgreSQL database. What is more, is that Metabase allows one to create custom queries through an intuitive interface, as seen in figure 1, which can then be visualized to one's desires and displayed on a custom dashboard. Last but not least, Metabase offers the required flexibility and functionality to allow one to host their application on a website and create a shareable link for their dashboards

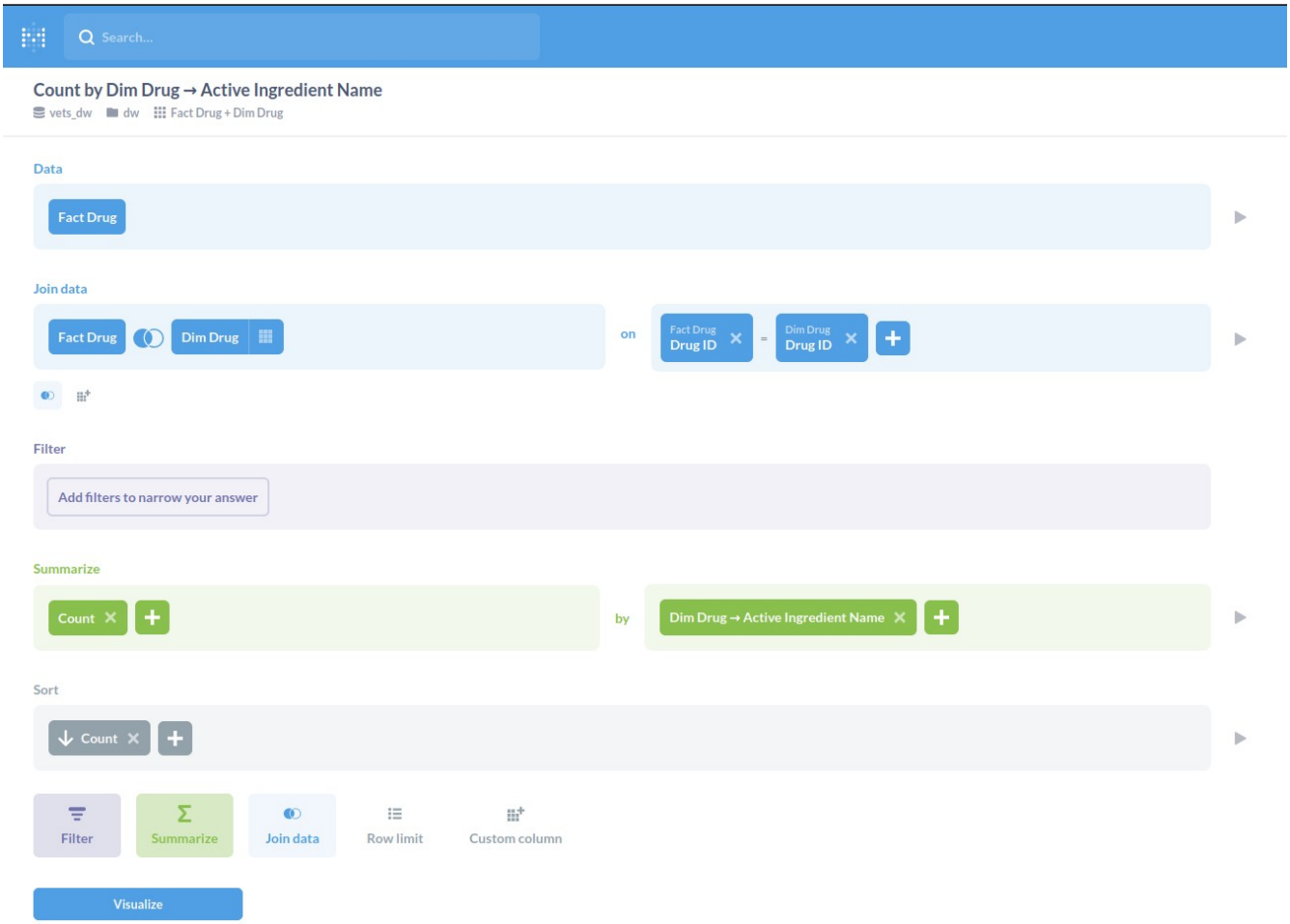

Figure 1: Metabase's custom query tool

2.1.2 Welcome screen

So, upon the first initialization of Metabase (after the initial setup and connection processes) one is provided with a screen similar to the one seen in figure 2.

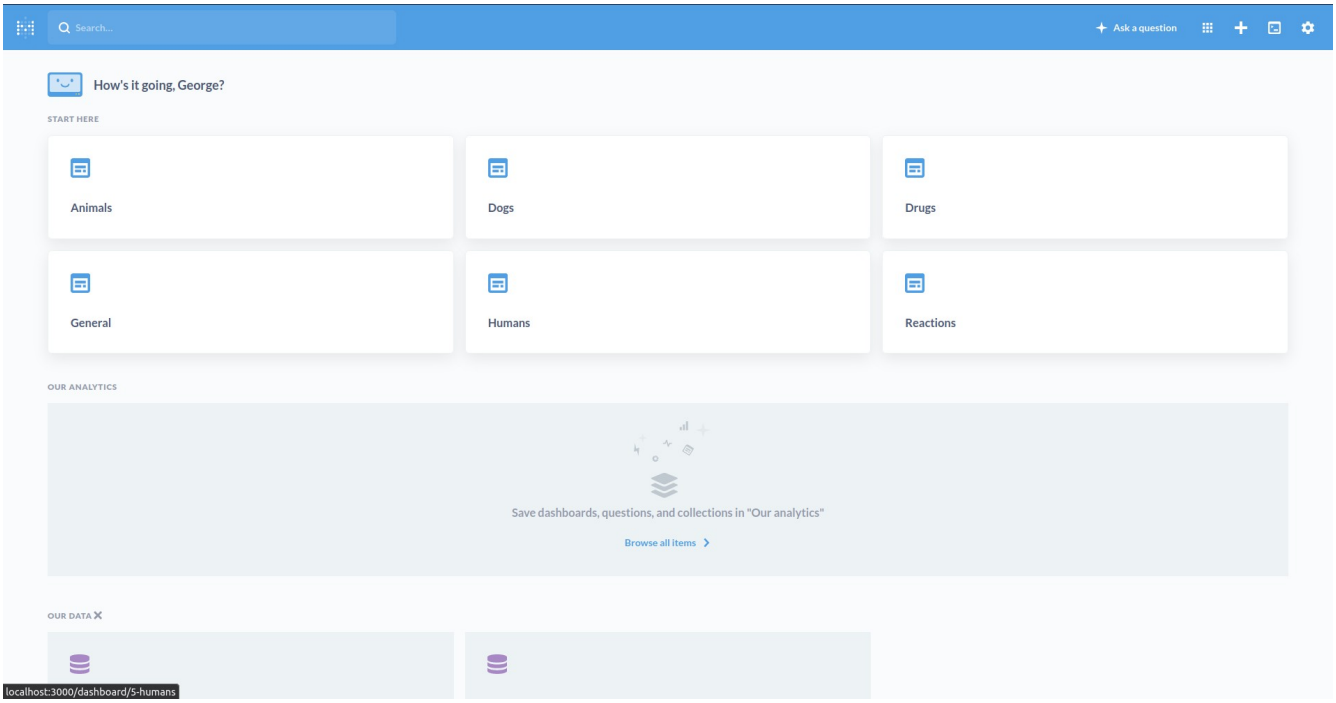

Figure 2: Metabase home screen. This is the first screen the user is welcomed to when entering the application for the first time.

It is important to note, that this is not the default home screen of the app, but it has been modified to best fit the veterinary clinic's requirements. A friendly 'Hello' message greets the user in the top left corner, while right below the user has quick access to 6 different dashboards: Animals, Dogs, Drugs, General, Humans and Reactions.

Following is the tab from where the user can access all of the pre-defined requirements individually, in a form of a list, while the final tab shows the user the available databases he or she has access to.

It can be seen that the design is based around simplicity and visual appeal. Specialized tabs, categorize the user interface's features, while bright colors and a block lettered font make elements stick out and easy to find.

This concludes the navigation around the application, which brings one to the report capabilities of it.

2.2 Reports

Reports are communication artifacts which translate raw information into digestible graphic and/or tabular form for the end user. It is common practice for BI systems to provide a number of business reporting functionality, to assist with monitoring the business. They are typically targeted towards providing analytical results and performance indicators for the different business departments, monitoring the fulfillment of business goals and others. However, in this scenario, the system is equipped with the functionality to produce informational reports only.

A business such as a veterinary clinic is built around a relatively simple business model. The overwhelming majority of such businesses provide expert veterinary services in exchange for a time-based payment. Thus, it would be realistic to assume that their current method of business reporting,

excluding informational reporting, is more than capable of satisfying their requirements. It is also safe to assume, that there would be no significant advantages gained by implementing a more complicated and feature-packed reporting system. The time required to update and monitor such a system would greatly outweigh the benefits. Thus, this application is designed for informational reporting.

2.3 Informational reports

Informational reports provide data on an event, or events, without analysis or recommendations. They aim to provide an unbiased view and present the facts. In this project, the reports come in the form of the dashboards presented to the client, accessible through the Metabase interface, as seen earlier. Individual diagrams/queries/tables inside these dashboards will be termed as 'requirements' for the purpose of this document, since they are designed to fulfill veterinary experts' requirements, (in terms of providing answers to questions they were assumed to have).

2.4 Requirements

Requirements answered through the custom dashboards of the Metabase, provided with this product, can be split into two categories. The first category includes all of the pre-defined requirements that come within the dashboards, while the second one includes all of the possible custom answers to requirements that one can come up with through Metabase's custom query interface.

2.4.1 Custom Queries

As seen in section 2.1.1, metabase comes with a pre-installed custom query tool. An interactive interface that offers the functionality to create and visualize queries with great customization and control. Provided that user possesses a complete picture of the DW environment's structure, and basic querying knowledge, one can pick and join tables, filter them, and perform numeric operations. There also exists the option to query the database using SQL commands directly, however it is highly unlikely that the end users of this application will explore this functionality.

2.4.2 Pre-defined requirements

These requirements aim to provide veterinary experts with fast and accurate answers to questions that it was assumed the experts would seek access to. Having them as pre-defined will considerably reduce the users' interaction with the time-consuming querying tool and provide them with a straight-forward access to the data. The questions answered can be further categorized into 6 large groups:

- General
- Animals
- Drugs
- Reactions
- Human
- Dogs

There exists a dashboard for each group, and each dashboard is comprised of a number of graphs and tables fulfilling different requirements.

It is also important to note here that all requirements' graphic visualizations come with accompanying descriptions to help the user understand what data is being presented and how they can interact with the graph. An example of such a description is shown in figure 4.

Other common features of requirements are: The filtering of ‘NaN’ and values that describe something that is unknown (e.g. ‘Other’, ‘Unknown’, ‘Not available’) and the descending ordering of numeric fields.

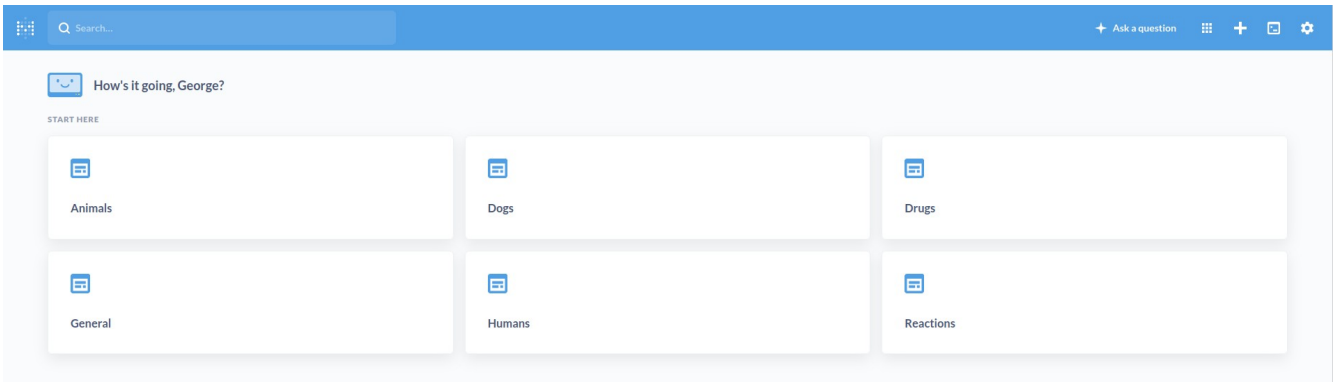

Figure 3: The tab section inside the home screen of the application showing the 6 quick access buttons to the different dashboard for the groups

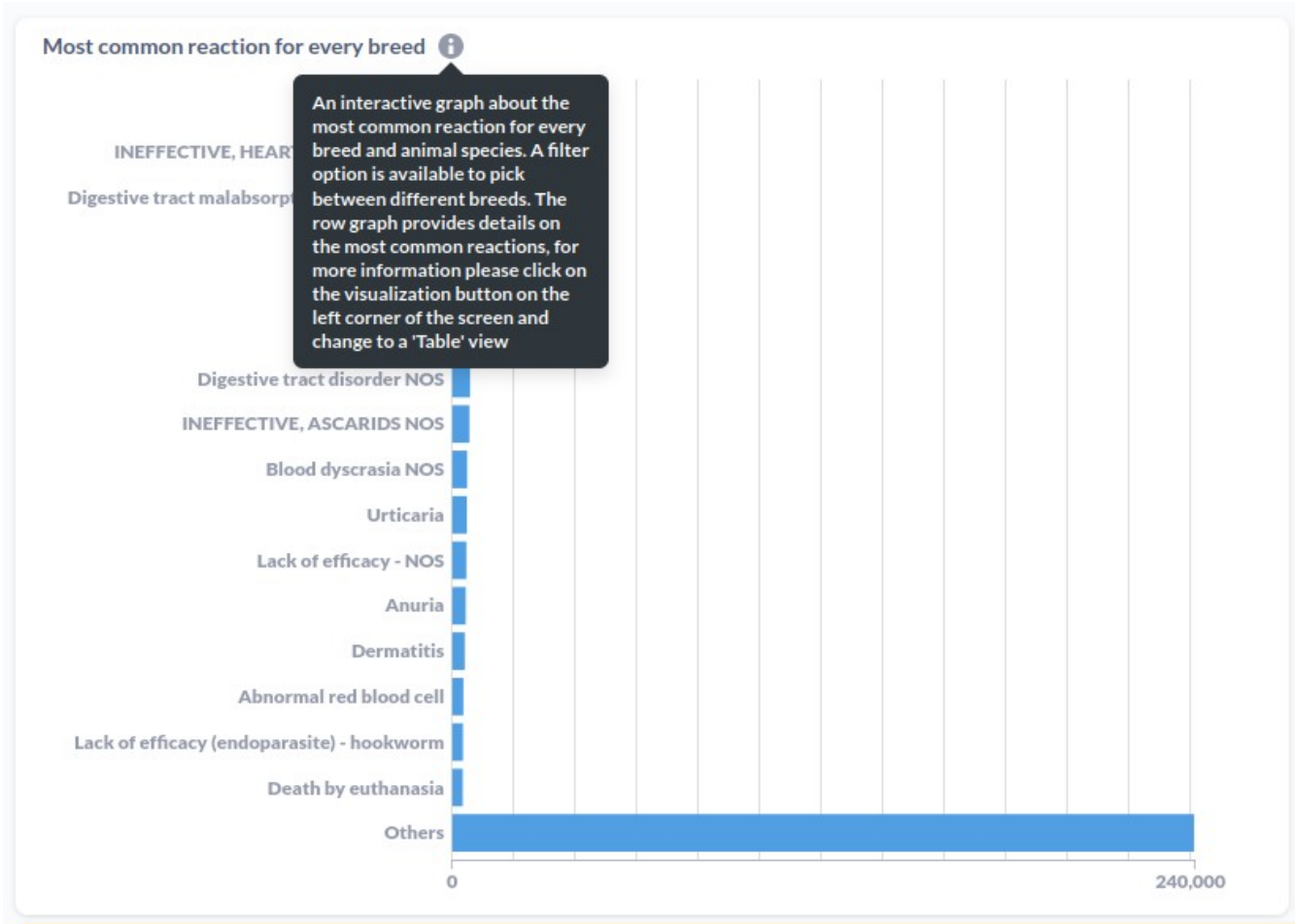

Figure 4: Example of complementary description for the ‘Most common reaction for every breed’ requirement.

2.4.2.1 General

As its name suggests, this group is aimed at general use information that a veterinary expert would want to have quick access to. These include statistical data about the dataset and generalized views of larger categories. The questions answered are:

1. The total reported cases to the FDA
2. The number of individual animal incidents reported to the FDA.
3. The percentage splits of male and female animals within the database.
4. The percentage splits of outcomes reported
5. The total number of species included
6. The list of species included
7. The number of incidents per quarter
8. How many days it takes for an adverse event to appear

Figure 5 is a screenshot of the 'General' dashboard where one can view its design features.

Figure 5: The 'General' dashboard interface.

The minimalistic design of the home page is also discernible here, while data is clearly depicted using vivid and high contrast colors.

2.4.2.2 Animals

Here lie animal-related requirements. This group depicts animal-related analytics from the database. Despite these, there also exist a number of more complicated queries:

1. Most common reaction for every breed

2. How size is correlated with adverse events
3. Reproductive status of animals and adverse events
4. Most common active ingredient that causes adverse events for every breed
5. Most common reaction for every breed, the active ingredient consumed and the animal's health assessment prior to exposure to exposure to the active ingredient
6. Most common active ingredient for every breed and the outcome of the incident

The figure below shows the query screen for the fifth requirement:

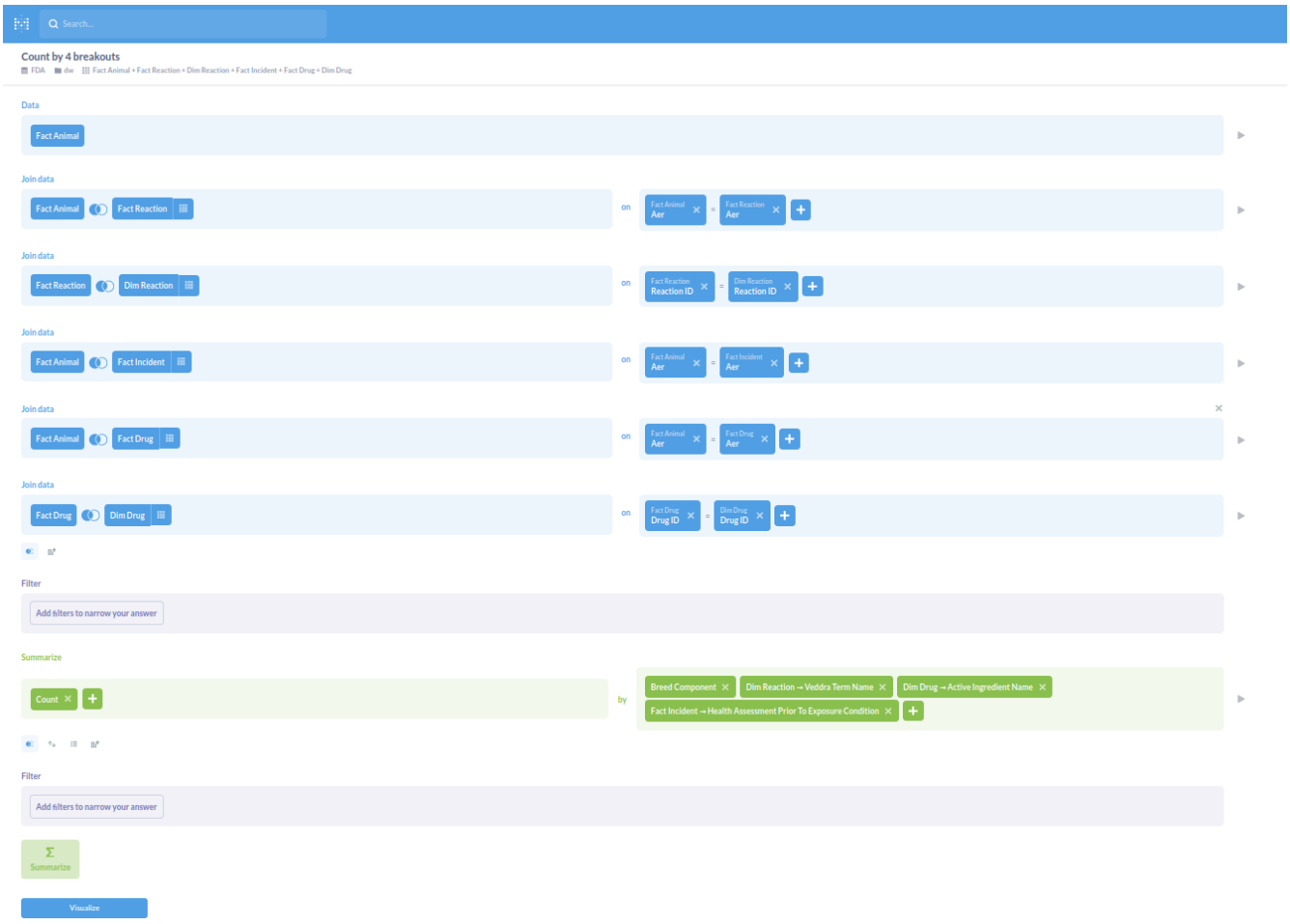

Figure 6: The query tool parameters to answer the 5th requirement from the animal group.

This figure is a clear indication of the flexibility that this tool allows when it comes to custom queries while it also depicts how simplified a complex query can become through the interactive interface. To reproduce the same results through plain text SQL commands would be a complicated task that requires a great level of knowledge and skills.

2.4.2.3 Drugs

This group focuses on the substances administered to the animals and how they can be correlated to the adverse events. The requirements satisfied are:

1. How the individual administrating of the substance is connected to the adverse events
2. Which are the most common off-label uses related to adverse events
3. Which are the most lethal active ingredients

Figure 7: Drugs dashboard (Note that the browser was zoomed out to 60% to include all three graphs. Actual text size on the screen at regular zoom will differ)

2.4.2.4 Reactions

Taking a closer look on the reactions related to the adverse events:

1. What are the most common reactions appearing
2. What reactions are connected to which active ingredients
3. Most common reactions on a quarter basis
4. Which are the most lethal reactions

Figure 8: Reactions' group dashboard

2.4.2.5 Humans

The FDA's database does possess a considerable amount of human related adverse events. This group categorizes these incidents and provides analytical statistics on them to help veterinary experts with their human patients. The requirements covered are:

1. The total number of human-related incidents
2. The most common reactions for humans and the drugs related to these incidents.

2.4.2.6 Dogs

Dog-related cases account for more than half of the total incidents recorded into the database with more than 700000 incidents reported. That is a direct consequence of the fact that the dog is the most common pet animal. Thus, it was found important to separate dogs from the other animals so that experts specializing in treating this species can have quick access to information they need.

The dog dashboard covers the following requirements:

1. The total number of dog cases
2. The most common reaction for every dog breed
3. The list of different dog breeds included into the databases
4. The most lethal active ingredients for dogs
5. The most lethal reactions for dogs
6. How many crossbred dogs are prone to adverse events when compared to non-crossbred ones

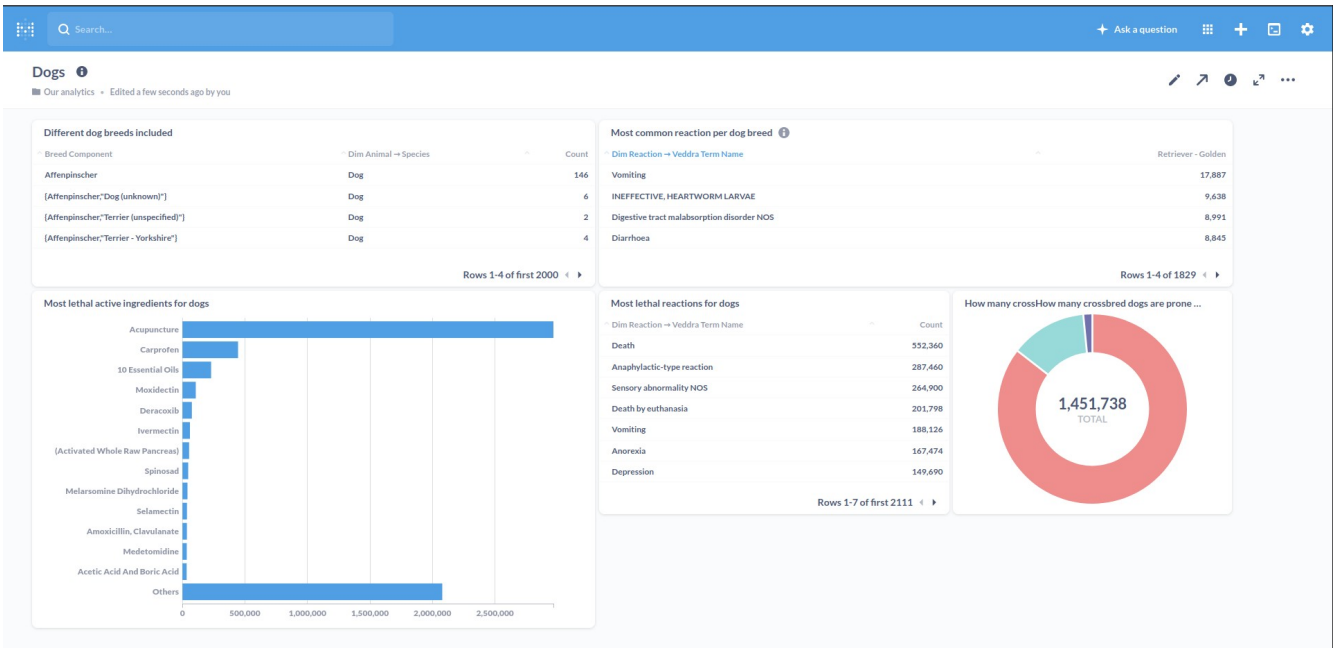

Figure 9: Dogs dashboard

3 DISCUSSION

Let us now explore the whole scope of the capabilities of the system and take a deeper look into why would a clinic want such a BI system installed on their local machines. What does it offer, what kind of advantage is gained when compared to existing infrastructure and how will it accelerate the workflow of the business. Lastly, a short section regarding future improvements is also included

3.1 Advantages

So what is there to be gained, what does this system offers that is different from its counterparts.

3.1.1 Automation

To start with, the system is fully integrated and automated from the downloading phase all the way to the front-end environment. With an initial setup and execution it delivers a product that is ready to use with no manual interaction required by the user, apart from the interface.

3.1.2 Information available

The system successfully delivers an in-depth look on numerous drug-related adverse events where animals were involved, with access to more than 1 million incidents and hundreds of millions of individual animal cases accompanied by great detail describing the event. Such a number is not attainable through traditional means for most, if not all, veterinary clinics.

3.1.3 Interactivity

The front-end design of the system makes it more approachable and acceptable to basic software users when compared to methods such as excel spreadsheets and/or a hand written filling system. Pre-defined

static elements provide solutions to queries without the need for manual interaction while the high-contrast color scheme and readable font make information easy to find at a glance. It also provides great flexibility with interactive controls allowing for users to customize their view, add filters to the graphical elements and come up with their own unique queries, fitted perfect to their needs.

3.1.4 Flexibility

The constellation schema in which the data warehouse tables are arranged allow one to access all of the available data and extract what they need, almost instantaneously. Given that an individual does possess a complete view of the schema and available fields, the possibilities and capabilities of the questions the system is able to answer are only limited by ones imagination.

3.1.5 Cost

Perhaps the most enticing characteristic of this BI system - from a business perspective - is the amount of resources one needs to spend to implement it into their business. At its current state this product is built entirely on free and open source software and does not require any changes to the business' IT infrastructure (provided that there already exists at least one local machine with a reliable connection to the internet). The product can directly be installed, on demand, on the existing machines of a clinic, without any costs (excluding labor).

3.2 Disadvantages and limitations

However, despite all of its advantages the system does not come without its limitations

3.2.1 Efficiency

Perhaps the most apparent limitation of the BI system is its back-end efficiency. An average of 30 hours required to have a fully operating product may have significant effects on the operation of a business. Due to the large amount of raw-data it receives when it consumes the FDA's API, loading the data into the local database is a task which can take an average of 25 hours, depending on one's internet connection and hard drive speeds. The main source of 'delay' in this situation is the use of data frames during the staging phase of the system. Downloading and splitting the data into 5 pandas dataframes before loading it into the staging environment is the most time consuming task due to the inefficiency of dataframes when used with large volumes of data, which is the case here.

This pattern is also apparent in all steps leading up to the final load of the data into the dimensional data warehouse environment.

A python script is responsible for loading the data from the staging to the temporary environment. The script initially fetches the entirety of the data from every table of the staging environment and then proceeds to perform the necessary cleaning, transforming and imputing on it. Many of the fields are discarded and all operations are performed utilizing minimum nested if statements to improve the efficiency of the process. However, due to the large volume of data that needs to be handled, the process lasted an average of 3 hours during testing. It is important to note here that this time can shift upwards or downwards depending on one's system specifications, namely the amount of available RAM and processing power.

The final load of the data into the dimensional data warehouse does not follow this trend however. With the data now being transformed to perfectly fit into the corresponding fields and the normalization

processes taking place - that significantly reduce the number of entries inside tables from a few million to a few thousand or even less than one hundred - the process is noticeably faster than the previous steps with an average of less than a minute elapsing when loading dimension tables. Loading fact tables despite being slower than loading dimension tables, due to a total number of rows exceeding 2 million for each table, still remains a fast process lasting a maximum of 5 minutes during testing.

Efficiency is also an issue when it comes to the front-end of this product. Complicated queries, such as the one seen in figure 5, can take several minutes to produce results while there is also noticeable delay when it comes to interacting with the corresponding graphs.

3.2.2 Data sources

Despite making the system inefficient, the FDA's database is also quite limited when it comes to its diversity. Due to the fact that the organization collecting the data is situated in the United States, the data collected is heavily closely related to that country. In fact, there exist no entries inside the database with incidents that originate from outside of the US (It is not possible to reach a different conclusion from the available data). Thus, although the available data covers a wide range of animal species, the database lacks data on region specific animals, and animal breeds, from outside the US. This, limits the system's adaptability and efficiency when it comes to business situated in different parts of the world.

3.2.3 Structure

Moving on to the core of system. The system's environment is built around the JSON files it receives from consuming the FDA's API. This architecture design choice does make the system quite efficient, however it can also lead to many sources of errors in the future.

Any major alteration on the JSON files which will affect the fields used by the system can potentially harm the results and / or the entirety of the product. The FDA clearly state in their documentation that such changes do occur of a frequent basis ,thus it would be a requirement to constantly check for updates on a regular basis and adjust the system accordingly.

It has been observed, by inspecting and comparing old JSON files with newer ones, that these changes mainly involve the addition of new fields or the removal of some others. Since the staging environment is loaded using pandas dataframes that look for position of nested information within the JSON files and not the exact position of a field (which would be the case if a python queries where used to extract individual data, see the figure below for more) the system would remain unaffected from such changes. However, there still exists the possibility of some major alteration to the structure of the JSON file which could ultimately brake the pipeline while at the same time the system would not take advantage of the information provided with the addition of new fields in the future.

3.2.4 Installation, remote updates and fixes

Last but not least, one could be concerned about the inefficiency with which the software is currently distributed. To install the system on a computer one would either need to possess the necessary knowledge and follow the instructions provided or it would require the physical presence of a technical expert. Upon finishing the initial setup, the issue would affect the longevity of the product. Updates, upgrades and bug fixes would require a technical expert to remotely connect to every machine the software is installed on and manually perform all of the necessary changes. This, although feasible for small quantities of the distributed software, would make for a large burden when scaling things up.

3.3 Implementation, product management and workflow

Let us now explore how the aforementioned will affect the clinic's workflow and how the system will be setup and upgraded.

3.3.1 Implementation and product management

To start with, the initial setup of the product will require the physical presence of at least one software expert, who will make sure that:

- The directory, where the local copy of the application will be installed, is organized correctly
- The local environment is properly configured to execute the necessary python scripts
- The local database is properly configured and that a reliable connection can be achieved with the application
- Everything in the back-end phase executes correctly
- The Metabase tool functions properly
- The dashboards are organized in the intended way and
- That all pre-defined requirements are satisfied

That would be the only time the system would require the physical presence of a technical expert to the clinic, provided no unforeseen circumstances. All later interactions, including: bug fixing, upgrades and updates, can take place remotely utilizing remote computer remote access tools and software.

3.3.2 Workflow

The features and characteristics described above have the potential to positively affect the workflow of a veterinary clinic. Experts will now possess access to large volumes of data providing them with an undistributed view of the unbiased conclusions drawn from them, thus helping them reach to decisions in a more efficient and scientific manner. They would be able to avoid substances that are prone to causing adverse events and focus on more effective solutions. Experts would also have the ability to gain a clear view of the medical analytics for a species, that they have not examined before, in great detail and without having to leave the application's interface and navigate to different archives.

This BI system has the potential to improve the decision-making process of all veterinary experts within the cleaning, in terms of accuracy and speed. The clinic would overall benefit from faster solutions and more satisfied patients and pet-owners, which would ultimately result in the growth of the business in other aspects apart from the informational one.

3.4 Future improvements

The approach followed in this project as a whole, proved successful in creating a capable decision-making assistance tool for use in a veterinary clinic environment. Building upon the latter, the system can be further expanded with the adaptation of new data sources which would cover a wider range of incidents, both in terms of the geography limitations and the species limitations. The product will have to adapt to accept these new flow of information and implement it in such a manner that it will co-exist and connect with the existing FDA dataset to produce a larger database.

The implementation of system is another area that improvements could be made. All stakeholders would benefit if the system could be delivered packaged inside a single file that is easily distributed in executable form. The end-use could then execute that file and have everything being taken care off,

while developers could remotely expand upon the application and deploy updates and upgrades remotely without the need to interact with each local machine on which the software is installed.

Efficiency is also an area offering great room for improvement. The current estimation for a total of 25 hours, for the execution of all back-end processes, is a considerable amount of time which could potentially harm the experience of a user. More testing is required to discover new means of loading the data into the staging environment while the same has to follow into the ETL pipeline processes taking place afterwards.

It should also be mentioned here that this 25 hours process has to take place on a quarter basis - since the FDA update their database on a quarter basis and it is recommended to update the whole dataset with - and although it could potentially fit within the working schedule of the clinic, it is an inconvenience that everyone would benefit if it was limited even further. A potential solution to this, would be to create a new database during the updating process. It would then follow than once the update is complete, the new database would replace the old one. Thus, there would be no down-time required during the updating process.

On the topic of the quarterly updates, one could not simply arrange a scheduler to perform the updates. It is not uncommon for online services, such as the FDA, to experience down-times and/or seize to function for small periods of time. In the, not so unlikely, event that the update process coincides with one of these events, the whole process could be harmed. Thus, it is recommended to add a script that would execute before the update process would take place and which would arrange whether or not the update process would take place. In the event that something is wrong the script would reschedule the update process. A fail-safe mechanism should also be implemented during the initial phase of the back-end of the system, to catch any unexpected events during the download process.

Last but not least, it is recommended that the system would be further expanded upon to offer clinic experts the means to share their own experience and incidents through the application’s interface. In this way they could contribute back to the scientific community and help to further expand the database.

4 SYNOPSIS

The product developed and implemented for the purposes of the Business Intelligence course is a fully integrated end-to-end BI system which provides a centralized view of historical data to veterinary experts and acts as a decision-making assistance tool.

This is achieved by utilizing the established FDA’s database with drug-related adverse events on animals. Through profiling, cleaning, transforming, regulating, and imputing the dataset, the system delivers all extracted information into a digestible form for the end user.

The system is centered around a fully operational, local, and free to host database complete with a staging, a temporary and relational data warehouse environments, while an ETL pipeline moves the data throughout them.

Veterinary experts have an in-depth access to more than 1 million reported incidents, including hundreds of millions of individual animal cases.

The system also provides a free front-end tool for veterinary experts to interact with, complete with six different dashboards specifically designed to group all of the available data and present it using pre-determined queries and corresponding visual graphics. The user also has the ability to customize their dashboards and design their custom queries through an interactive querying tool.

The system succeeds in meeting a veterinary clinic's requirements in that it provides answers to a plethora of questions while being flexible enough to accept specialized queries. Last but not least, the BI system is fully deploy-able and operational using free tools. However, that is not to say that the system is perfect.

Please find the GitHub repository for the individual part of the project: https://github.com/georgios-antoniadis/Business_Intelligence_Coursework_Individual