Boehringer Ingelheim Animal Health Australia

Deliverable 4 - Project Plan and Requirements,
With Scoping Document
+
MVP
+
Analysis, Design, Testing Documentation
+
User Manual

Project BICI
Group 1

Date: 20/5/2021

Revision History of D3

Under 1.1 Statement of Purpose/Scope/Description
Added: “As one of the world’s largest animal pharmaceutical companies, Boehringer Ingelheim (BI) has access to copious amounts of data; but without effective data analysis the data cannot be harnessed and utilised. BI”

Reasoning: To specify the current problem from the industry’s point of view

Date: 19/4/2021

Under 1.3 Resource Management
Added: “in regard to”

Reasoning: To correct spelling mistake

Date: 19/4/2021

Under 2.1 Tasks/Activities/Phases

For D1
Added: “(COMPLETED)”

“the business problem and the client’s needs. It aims to assess the feasibility of this new project.”

“

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word and Google Docs are used for drafting the documents.
    - MS software and Google Docs are suitable for this project as they are compatible with most operating systems.
- Computers are needed for using the software and writing this deliverable.”

“

- A raw data set was given by BI on the 5th of March for the purpose of analysing the business problem. The set includes: sales to wholesaler, wholesaler to vet and retail sales, retailer to end user and dashboards.”

For D2
Added: “(COMPLETED)”

“

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word and Google Docs are used for drafting the documents.
- Microsoft Project is for drafting the Gantt chart.

    - MS software and Google Docs are suitable for this project as they are compatible with most operating systems.
- Computers are needed for using the software and writing this deliverable."

"

- A new data set was added to the original data set by BI on the 16th of March, for the purpose of analysing the business problem and preparing to construct the prototype. The updated set includes: sales to wholesaler, wholesaler to vet and retail sales, retailer to end user, dashboards and rebate and promotional data."

For D3
Added:
"

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word is used for drafting the documents.
- Microsoft Project is for drafting the Gantt chart.
    - MS software are suitable for this project as they are compatible with most operating systems.
- Jupyter Notebooks (Python) is used for designing the prototype. This software is suitable as it is an open-source application that allows data cleansing and it is a free software.
- GitHub is used for collaboration of the project. It is useful as the code can be pre-loaded into the notebook and it will be ready anytime.
- Computers are needed for using the software, writing the deliverable and constructing the prototype/MVP."

"

- Combined and cleansed data sets, including Western, Eastern and Petwise sales data are needed for prepping the MVP. This was done on the 21st of April.
- The dashboards were removed and more raw data was added to the data set by BI on the 22th of April, for the purpose of analysing the business problem and constructing the prototype. The updated set includes: sales to wholesaler, wholesaler to vet and retail sales, retailer to end user, and rebate and promotional data.
- Feedback on the prototype/MVP is received on the 28th of April from the client for improvement."

For D4
Added:
"

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word is used for drafting the documents.
- Microsoft Project is for drafting the Gantt chart.

    - MS software are suitable for this project as they are compatible with most operating systems.
- Jupyter Notebooks (Python) is used for designing the prototype. This software is suitable as it is an open-source application that allows data cleansing and it is a free software.
- GitHub is used for collaboration of the project. It is useful as the code can be pre-loaded into the notebook and it will be ready anytime.
- Computers are needed for using the software, writing the deliverable and constructing the prototype/MVP."

"

- Raw data, or if there is any updated data set, is needed for updating the scoping document.
- Feedback from the client is needed for improvement."

For D5
Added:
"

- Microsoft Word is used for drafting the documents. It is suitable to use this software as it is compatible with most operating systems.
- Computers are needed for writing the reflective report."

Reasoning: To update the task and revise what software is needed as the project is carrying on, as well as specifying what raw data set was received.

Date: 20/4

<u>Under 2.6 Assumptions</u>
Added:
"

- Assumptions and notes for the combined data set (Western, Eastern and Petwise)
    - Removed sensitive data in columns across all data sets to protect customer privacy (e.g. AccountName, Description).
    - The received western data set for the month of October is incomplete and could not be combined with November and December.
    - EXTERNORDERKEY column in Pet wise consolidated data sets is inconsistent.
    - Barcode represents product name in eastern data sets.
    - Western, eastern and pet wise data sets could not be combined as they have different attributes."

Reasoning: To update the assumption after integrating and practically cleansing the data set

Data: 25/4

Under 2 Project Schedule
Added a new section: “3 Handover Requirements”

Reasoning: This section is for the company to look it after the team leaves the project

Date: 26/4

Under 4.2 Reviews and Audits, Testing
Added: “The data values used for testing is either going to be categorical or continuous, or it can be a mix of both. This will depend on what data sets or values are tested and used, but the data can always be converted if needed.”

Reasoning: Clarify that data values from testing can be categorical or continuous variables.

Date: 26/4

Under 4.4 Tracking/Change Management
Added: “For changes to be made, the following form should be filled out by the group member and signed off by the rest of the other group members. This is to ensure that the other group members have all read, understand and agree to the change. This well also keep track of tracking and managing changes in the project.

Added: Project Change Request Form

Reasoning: To help track and mange changes that need to be made to project.

Date: 26/4

Under 4.6 Conflict Resolution/Negotiation
Added: “ The following form on the next page should be used to help resolve the conflict, as it will allow both sides to put their own input on the situation. After the forms are completed they should be passed on to the rest of the group or other third parties to assist in the four-step conflict resolution process.”

Added: Group Conflict Resolution Form

Reasoning: To help assist the group with resolving conflicts.

Date: 26/4

Under 5.3 Data Preparation
Added: “Regarding the integration of multiple data sets, this is established through simple methods of 1NF/2NF, or otherwise known as normalisation. In simple terms, by establishing similarities between data sets, through clearing of data redundancy methods, we can integrate and create a single file for all data

sets. This is furthered through the method of first normalising and combining the different sets of data sorted by year or by other unique identifiers, which then allows us to create one combined data set."

Reasoning: To clear how transformations are applied to the integrated data and how multiple data sets are transformed into one.

Date: 19/04

Under 5.3 Data Preparation
Added "In some data sets, the process of data preparation is halted due to missing data. There is a range of possible methods that could be used to establish a range of consistency through the data preparation stage. One of those methods relates to regression techniques. Regression techniques allow us to predict what value would fit in the missing data column based on past analysis, and relationships between other columns. Past analysis on past datasets, as well as the relationships created by BI themselves, act as a source of data for these regression techniques. By using regression techniques, instead of simply dropping the rows that have missing values which can cause the loss of valuable information, we are able to consistency fill in missing data values without compromising data validation, or cause data redundancy through strict conditions within data regression."

Reasoning: Regarding data preparation to clear the methods of dealing with missing data.

Date: 21/04/2021

Under 5.4 Modelling
Added "Overall, these modelling techniques are based on the problem of classification rather than regression. Specifically in this project, the classification problem allows us to determine possible whereabouts for the potential of growth for profit margins are, which is based on the feedback from BI as well as the added potential for predictive abilities that were suggested to be included in the project. On the other hand, due to recent feedback from BI, the problem has shifted to include elements of both a classification problem, as with the future prediction capabilities, and a regression problem, as through the feedback it is now evident that regressive analysis is needed to cover the basic needs of the project."

Reasoning: To provide rationale to determine that the problem was both a classification problem and a regression problem.

Date: 22/04/2021

Under 5.4 Modelling
Added: “There is a range of constraints that need to be covered within this model. These constraints include but are not limited to:
Resource Constraints: The model must fit within the constraints of both the project requirements as well as BI’s feedback. For example, one of the constraints is that the project must be based on free to use technology with no cost required.
Technology Constraints: The model must fit within the minimum requirements that we have decided for the model. For example, based on the information that BI has given us, we are able to confirm that Jupiter is one of the minimum requirements used for the model.
Periodic Constraints: The model must fit within the maximum time frame given for the project, including the periodic updates we must complete to ensure the project is consistent throughout the entire plan.
Moreover, more specific examples of constraints include:”

Reasoning: To add details about the constraints we plan to set on the model.

Date: 24/04/2021

Under 5.5 Evaluation
Added: “Moreover, there is a range of evaluation metrics that allow us to determine the levels of performance the model allows. Furthermore, there is certain specific evaluation metrics that evaluate both classification and regression metrics.

Examples of these metrics that could be used include:
MSE or Mean Squared Error evaluation metric: This metric allows us the evaluate the profitability of certain data sets through regression lines, and how close each point is together, to determine what can be done to improve profits.
R Squared evaluation metric: This metric allows us to determine if the model itself fits within the data based on a certain valued range, or to determine based on this value if somewhere in the code we have made an error.”

Reasoning: To cover evaluation metrics.

Date: 28/04/2021 (After Feedback Changes)

Under 5.6 Deployment
Added: “Furthermore, at this stage the final product relates to a Jupyter based solution that covers the problems based on the maximum resource requirements the team has for the model, which is furthered through certain limitations we have due to the constraints given. Moreover, the deployment is based on the interactions between BI and the team, as determination into the final product is solely based on the needs of BI. This in turn covers the final product, but there are certain other procedures that are apparent no matter the final result. For example, through explanation and teaching of our model, we are able to give BI certain constraints in order to maintain and monitor the final product so

that the solution/model that is given works toward their advantage. Furthermore, another example relates to certain scenarios were adaptation is needed for the model, based on the changing needs of BI. In this case, we are able to provide steps within the monitoring stage in order for the BI Staff members to adapt the system without external help, as the system is easy and simple to use."

Reasoning: To cover deployment issues and the final product.

Date: 28/04/2021 (After Feedback Changes)

Specific Feedback Changes:
Regarding deployment issues, the final product, and evaluation metrics, these attributes are heavily dependent on what the company, BI needs in order to have a sustainable model. More specifically, originally the problem was determined as a classification problem, but after further analysis into the feedback given by BI on the 28, we had to change the original classification problem into elements of both classification and regression.

Furthermore, originally, we could not add evaluation metrics as we did not have insight into the certain needs of BI. After the 28/04/2021 meeting, we were successfully able to determine the best possible evaluation methods, which were MSE & R Squared Evaluation Tactics.

Revision History of D4

Under 2.1 Tasks/Activities/Phases
Added: "(COMPLETED)" for D3

Reasoning: To specify that D3 is completed

Date: 5/5/2021

Contents

1. Project Plan

1.1. Statement of Purpose/Scope/Description

As one of the world's largest animal pharmaceutical companies, Boehringer Ingelheim (BI) has access to copious amounts of data; but without effective data analysis the data cannot be harnessed and utilised. BI has initiated a Three-Phase project to deliver integrated data through customisable reports and dashboards. This project is within Phase II and works with a large volume of datasets with mismatching variables. The aim of this project is to upgrade the existing dashboards to allow more data driven insight to be drawn and create a better business model through predictive features (i.e. which products are profitable and can be invested in more). This will create more accessible and easier to understand data for all the stakeholders, in conjunction to increasing efficiency, productivity and reducing costs.

1.2. Risk Management

Risk and Description	Probability / Risk Level	Consequence	Mitigation
1. Technological Failure: With every technology based action, there runs a risk of technological failure	Medium / High	Progress could be lost, creating more work and causing a potential delay in the schedule	Constantly backing up onto an external hard drive or a cloud based software
2. Monetary Costs: Although the project does not intend to involve any costs, both Tableau and Power BI are a paid subscription software	Low / Low	There could be a small cost to this project	Free trials will be utilised for both software and Power BI has a free version that has basic features in the event it is needed for future use
3. Schedule Risk: The members on this project have other commitments and may fail to adhere to the schedule or the work may take longer than expected	Low / Medium	Causing a delay in the deliverables which can cause other unintended consequences	Holding the members accountable by checking in at least once a week and requiring drafts and submissions before the due date

4. Performance Risk: The final deliverable may not be up to standard	Medium / Medium	Further delaying the Three Phase Boehringer Ingelheim project	Using an agile approach with multiple revisions and continuous communication with Boehringer Ingelheim for feedback

Risk Matrix

		Impact		
		Low	Medium	High
Probability	High			
	Medium		4. Performance Risk	1. Technological Failure
	Low	2. Monetary Costs	3. Schedule Risk	

1.3. Resource Management

Project Resources:

- People
    - Although one of the team members dropped out during the project, the rest of the team is ready to cover their share of work.
    - The team is able to efficiently and productively function together with great teamwork
    - The BI team is readily available for any questions or concerns regarding the data and their objectives.
    - Unit convenor Deborah Richards is also available for clarification in regard to the PACE unit
- Hardware
    - Computers are required for this project and each team member is responsible for their own devices and technology.
    - Resources may be shared in the team if a situation that requires it arises.
- Software
    - Microsoft Excel for basic data exploration and preliminary data cleaning

  - Python via Jupyter Notebook is the primary software. It is readily accessible and free, primarily used for further data cleaning, data analysis and machine learning. Libraries used include:
    - Pandas: Data cleaning and making new data frames for analysis
    - Numpy: Sorting through arrays of data
    - GGPlot: Data visualisation and data analysis
    - SKLearn: Primary tool for data analysis, including train-test split and regression modelling
  - Power BI is free for visualisation but there is a 60 day free trial for Power BI Pro which will be used if extras features are required
  - Tableau is a paid subscription visualisation tool, however, there is a 14 day free trial. Resources will be effectively allocated and the trial will only be started when all aspects of the project are ready to be visualised using Tableau
  - Google Docs is utilised to encourage collaboration and constant cross-checking between team members
  - Final reports are written up using Microsoft Word to create easier-to-read reports.
  - Data quality tools such as Cloudingo and Validity may be used in the future for data validation.
- Data
  - Raw data is provided by Boehringer Ingelheim to be cleaned and visualised for the project
  - 2 dashboards which were made using Excel slicer were provided by Boehringer Ingelheim as guidance.
- Money
  - This project does not entail any monetary costs; although, it may be required for paid features of Tableau and Power BI in the future, beyond this project.

1.4. Team Organisation and Structure

The team follows an organic organisation structure where everyone has an equal share of responsibility and are held accountable for everyone else's work. The team members all take on a certain role within the team:

- Chris and Rick: Communication with Boehringer Ingelheim and the PACE team.
- Zoe: Secretary, responsible for meeting minutes and other admin work
- Thomas: Guidance on technical data related work

- Michelle: Writer and editor of deliverables.

2. Project Schedule

2.1. Tasks/Activities/Phases

The following section specifies and justifies what products, documents and diagrams that are appropriate for this project, as well as what will be delivered and the due dates of each deliverable.

Deliverable: D1 - Feasibility Study (COMPLETED)
Due Date: 5pm, Thursday, 11/03/2021
Description: This deliverable is for the team to understand the business problem and the client's needs. It aims to assess the feasibility of this new project.
What product is needed?:

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word and Google Docs are used for drafting the documents.
    - MS software and Google Docs are suitable for this project as they are compatible with most operating systems.
- Computers are needed for using the software and writing this deliverable.

What document is needed?:

- A raw data set was given by BI on the 5th of March for the purpose of analysing the business problem. The set includes: sales to wholesaler, wholesaler to vet and retail sales, retailer to end user and dashboards.

What diagram is needed?:

- No particular diagram is needed.

Deliverable: D2 – Project Plan and Requirements/Scoping Document (COMPLETED)
Due Date: 5pm, Thursday, 1/04/21
Description: This deliverable consists of a project plan, a project schedule, a quality manual and requirements/scoping document.
What product is needed?:

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word and Google Docs are used for drafting the documents.
- Microsoft Project is for drafting the Gantt chart.
    - MS software and Google Docs are suitable for this project as they are compatible with most operating systems.
- Computers are needed for using the software and writing this deliverable.

What document is needed?:

- A new data set was added to the original data set by BI on the 16th of March, for the purpose of analysing the business problem and preparing to construct the prototype. The updated set includes: sales to wholesaler,

wholesaler to vet and retail sales, retailer to end user, dashboards and rebate and promotional data.

What diagram is needed?:

- A Gantt Chart is used for showing the timeline of the project. This graph is appropriate because it provides a clear visualisation of the timeframe for the project

Deliverable: D3 – Updated D2 documents + Design, test cases, prototype/MVP (COMPLETED)
Due Date: 5pm, Thursday, 29/04/21
Description: This deliverable includes an updated version of D2 (Project plan, quality manual and requirements/scoping document), plus a list of assumptions that is relevant, design + testing documents, and prototype/MVP.
What product is needed?:

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word is used for drafting the documents.
- Microsoft Project is for drafting the Gantt chart.
    - MS software are suitable for this project as they are compatible with most operating systems.
- Jupyter Notebooks (Python) is used for designing the prototype. This software is suitable as it is an open-source application that allows data cleansing and it is a free software.
- GitHub is used for collaboration of the project. It is useful as the code can be pre-loaded into the notebook and it will be ready anytime.
- Computers are needed for using the software, writing the deliverable and constructing the prototype/MVP.

What document is needed?:

- Combined and cleansed data sets, including Western, Eastern and Pet-wise sales data are needed for prepping the MVP. This was done on the 21st of April.
- The dashboards were removed and more raw data was added to the data set by BI on the 22th of April, for the purpose of analysing the business problem and constructing the prototype. The updated set includes: sales to wholesaler, wholesaler to vet and retail sales, retailer to end user, and rebate and promotional data.
- Feedback on the prototype/MVP is received on the 28th of April from the client for improvement.

What diagram is needed?:

- A Gantt Chart is needed for showing the edited timeline of the project. This is appropriate because it provides readers a clear visualisation of the timeframe for the project.

Deliverable: D4 - Increment 1 deliverables + user/training manual
Due Date: 5pm, Thursday, 20/05/21
Description: Same as the previous deliverable, with an extra user manual
What product is needed?:

- Microsoft Excel is needed for accessing the raw data sets.
- Microsoft Word is used for drafting the documents.
- Microsoft Project is for drafting the Gantt chart.
    - MS software are suitable for this project as they are compatible with most operating systems.
- Jupyter Notebooks (Python) is used for designing the prototype. This software is suitable as it is an open-source application that allows data cleansing and it is a free software.
- GitHub is used for collaboration of the project. It is useful as the code can be pre-loaded into the notebook and it will be ready anytime.
- Computers are needed for using the software, writing the deliverable and constructing the prototype/MVP.

What document is needed?:

- Raw data, or if there is any updated data set, is needed for updating the scoping document.
- Feedback from the client is needed for improvement

What diagram is needed?:

- A Gantt Chart is needed for showing the edited timeline of the project. This is appropriate because it provides readers a clear visualisation of the timeframe for the project.

Deliverable: D5 - Final Group Reflective Report
Due Date: 5pm, Thursday, 3/06/21
Description: This deliverable is not a resubmission of any documents submitted before. It is a group reflection on what has been done during this whole project.
What product is needed?

- Microsoft Word is used for drafting the documents. It is suitable to use this software as it is compatible with most operating systems.
- Computers are needed for writing the reflective report.

What document is needed?:

- Previous deliverables are needed for self-reflection.

What diagram is needed?:

- No particular diagram is needed for now.

Deliverable: D6 - Project Presentation/Demonstration
Due Date: 5pm, Thursday, 3/06/21
Description: This deliverable is a live demonstration of the project and the solution to the class, sponsors and academics in a lecture theatre.
What product is needed?:

- Google Slides is needed for the presentation. It is best to use this software due to its accessibility.
- The final solution is needed for the presentation.

What document is needed?:

- No further document is needed for now.

What diagram is needed? And why?:

- No diagram is needed for now.

Deliverable: D7 - Final Delivery of the Product to the sponsor
Due Date: After Week 13, up to Thursday of Week 16
Description: This deliverable involves discussion with the client on how to use and install the system, and provide further answers and solutions if the client has other questions, issues or concerns.
What product is needed?:

- The final solution is needed for the presentation.

What document is needed?:

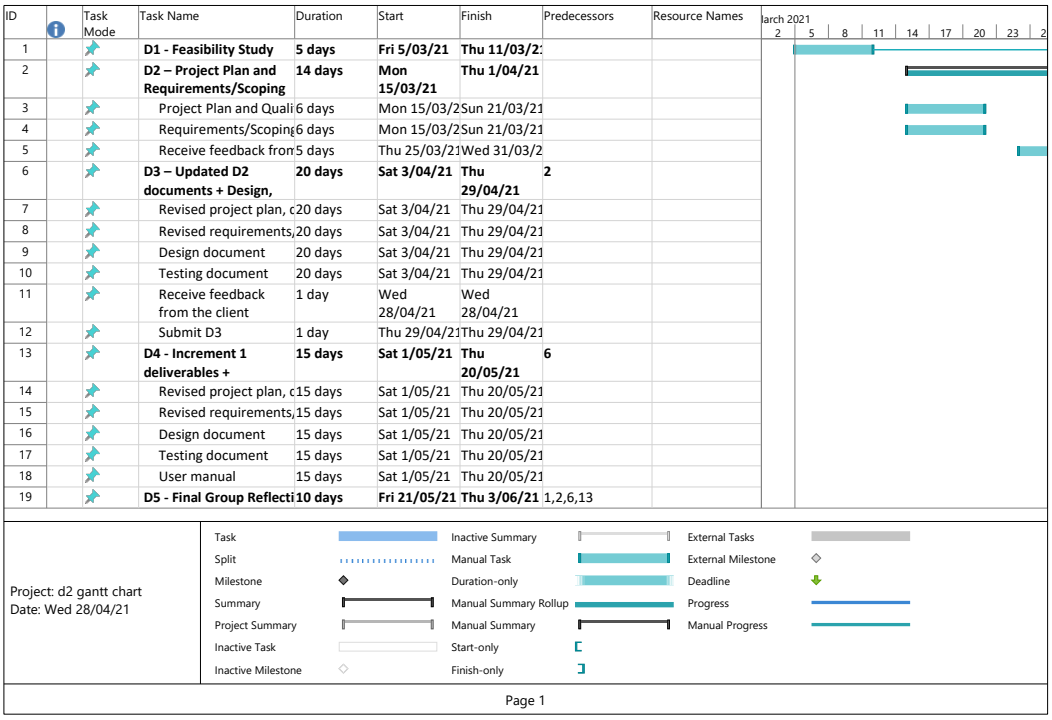
- No document is needed for now.

What diagram is needed?:

- No diagram is needed for now.

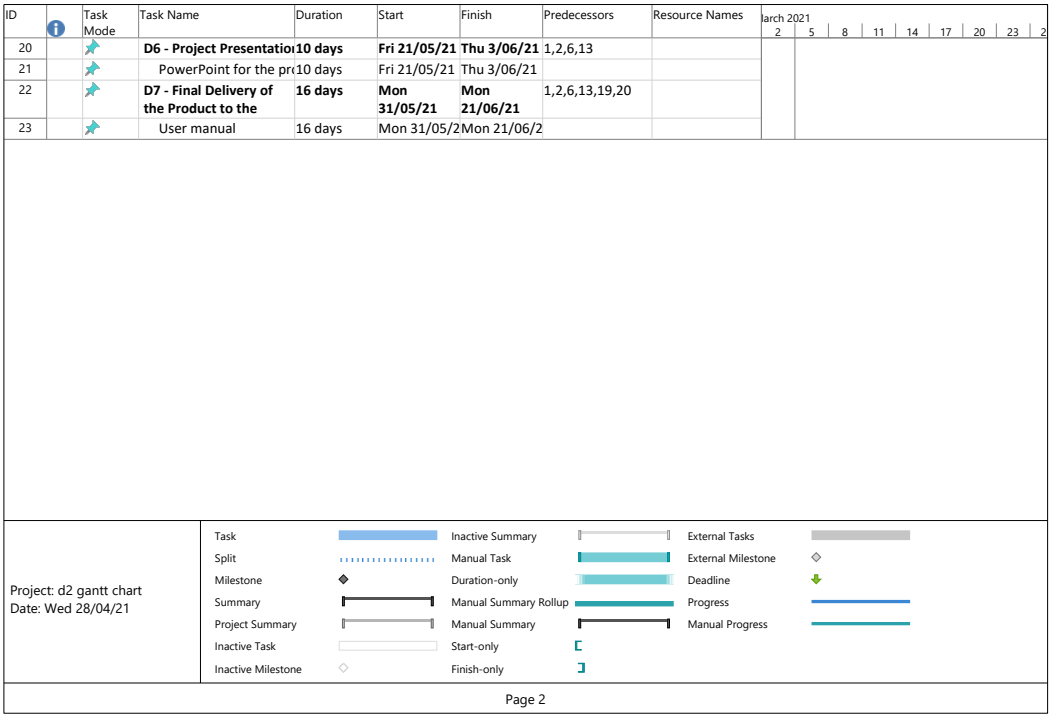
2.2. Timeline

The following Gantt Chart was created with Microsoft Project.



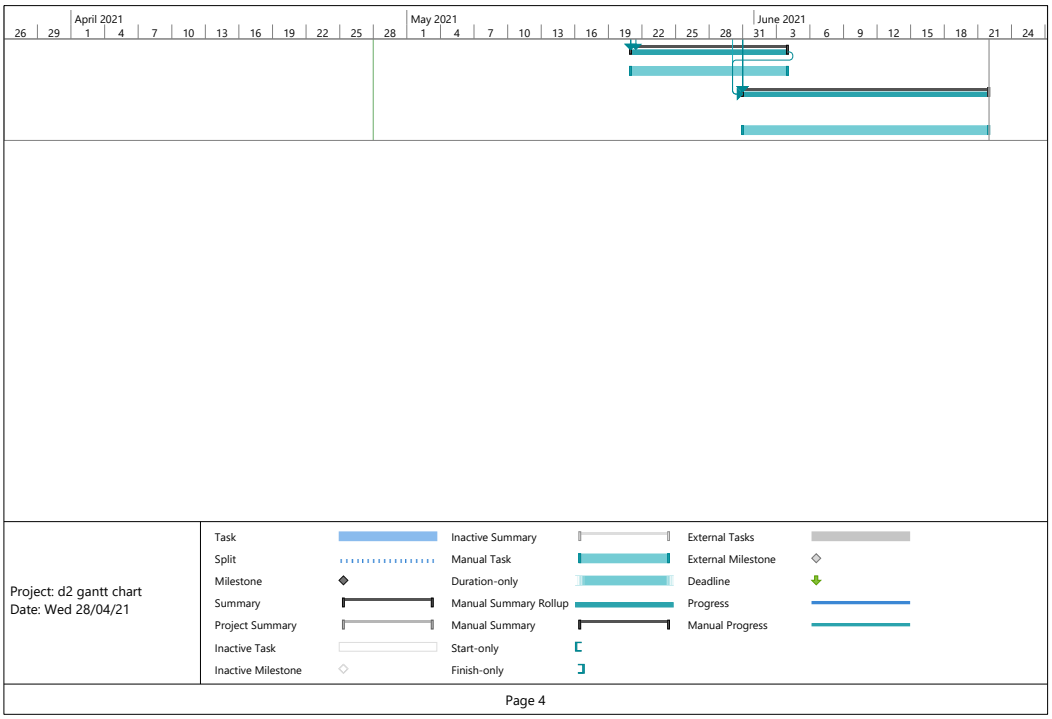
ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		**D1 - Feasibility Study**	**5 days**	**Fri 5/03/21**	**Thu 11/03/2**		
2		**D2 – Project Plan and Requirements/Scoping**	**14 days**	**Mon 15/03/21**	**Thu 1/04/21**		
3		Project Plan and Quali	6 days	Mon 15/03/2	Sun 21/03/21		
4		Requirements/Scoping	6 days	Mon 15/03/2	Sun 21/03/21		
5		Receive feedback fron	5 days	Thu 25/03/21	Wed 31/03/2		
6		**D3 – Updated D2 documents + Design,**	**20 days**	**Sat 3/04/21**	**Thu 29/04/21**	**2**	
7		Revised project plan, c	20 days	Sat 3/04/21	Thu 29/04/21		
8		Revised requirements,	20 days	Sat 3/04/21	Thu 29/04/21		
9		Design document	20 days	Sat 3/04/21	Thu 29/04/21		
10		Testing document	20 days	Sat 3/04/21	Thu 29/04/21		
11		Receive feedback from the client	1 day	Wed 28/04/21	Wed 28/04/21		
12		Submit D3	1 day	Thu 29/04/21	Thu 29/04/21		
13		**D4 - Increment 1 deliverables +**	**15 days**	**Sat 1/05/21**	**Thu 20/05/21**	**6**	
14		Revised project plan, c	15 days	Sat 1/05/21	Thu 20/05/21		
15		Revised requirements,	15 days	Sat 1/05/21	Thu 20/05/21		
16		Design document	15 days	Sat 1/05/21	Thu 20/05/21		
17		Testing document	15 days	Sat 1/05/21	Thu 20/05/21		
18		User manual	15 days	Sat 1/05/21	Thu 20/05/21		
19		**D5 - Final Group Reflecti**	**10 days**	**Fri 21/05/21**	**Thu 3/06/21**	1,2,6,13	

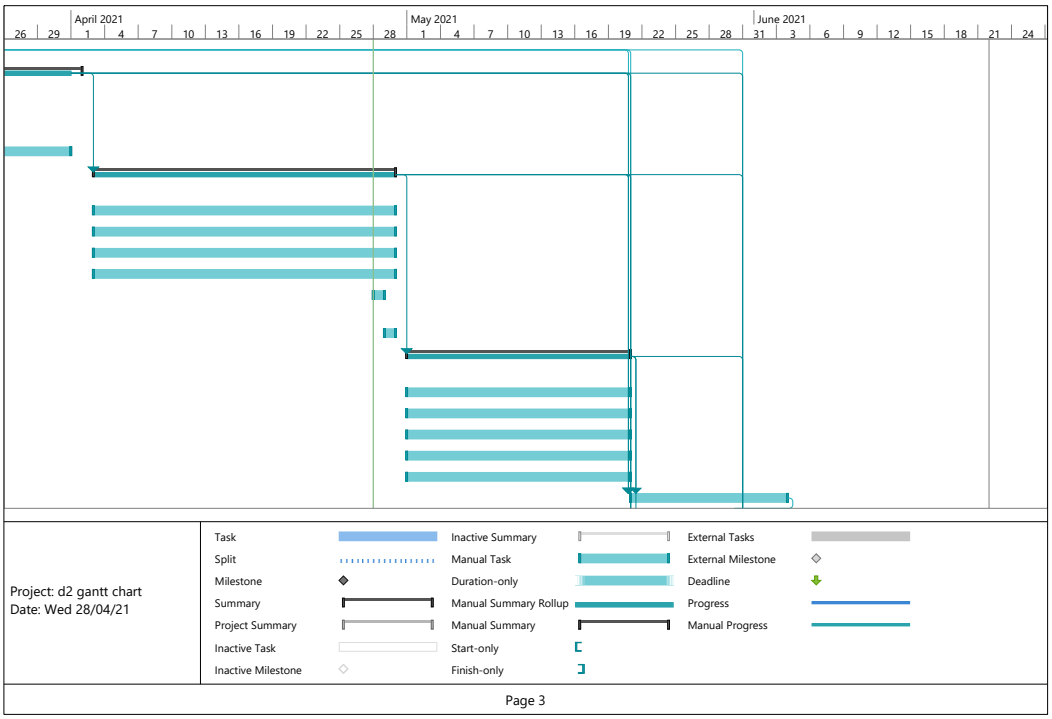
Page 1



ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
20		**D6 - Project Presentatio**	**10 days**	**Fri 21/05/21**	**Thu 3/06/21**	1,2,6,13	
21		PowerPoint for the pr	10 days	Fri 21/05/21	Thu 3/06/21		
22		**D7 - Final Delivery of the Product to the**	**16 days**	**Mon 31/05/21**	**Mon 21/06/21**	1,2,6,13,19,20	
23		User manual	16 days	Mon 31/05/2	Mon 21/06/2		

Page 2


April 2021
May 2021
June 2021
Project: d2 gantt chart
Date: Wed 28/04/21
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 4


April 2021
May 2021
June 2021
Project: d2 gantt chart
Date: Wed 28/04/21
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 3

2.3. Resources Allocated

The following discusses the allocated resources for this project.

- Hardware: each group member has their own computers for this project.
- For software:
    - Microsoft Excel and Word will be used by all members for basic document editing
    - Our team member Thomas will be using Python via Jupyter Notebook for building the prototype/MVP (libraries used: Pandas, GCPlot, SKLearn, Numpy)
    - MS Project will be used by Zoe for making the timeline
- People: each team member has the access to communicate with the client.
- Data: the client has provided a set of raw data to all team members and has been updating it

2.4. Process model discussed/justified

Agile approach is selected as the process model for this project and has been used widely in the industry nowadays. It is a fast-paced and adaptive method to come up with a solution. It breaks the project into different iterations and increments, allowing both developments, testing and reflection to happen at the same time.

The advantages of implementing this approach are:

- Allows communication between the team and the client from an early stage of the project
- Promote teamwork
- Changes can be made throughout the project if the requirement changes
- It has the flexibility that allows errors to be fixed in the middle of the project
- Continuous reflection and multiple revisions create a better project

Although the client has presented the team the business problem, the team is still working through the problem and the requirements may change during the process. This model is most suitable for this project due to the flexibility.

2.5. Documentation Identified/discussed

Deliverable 1

For deliverable 1, the team needs to produce a feasibility study. This report is for the team to understand the current situation and the business problem. It aims to analyse and assess the viability of this new project. This is due at 5pm, on the 11th of March, 2021.

Deliverable 2

For deliverable 2, the team needs to produce a project plan, quality plan and requirements/scoping document. This document aims to communicate the plan and the requirement for this project. This is due at 5pm, on the 1st of April, 2021.

Project Plan and Quality Manual

The first part of the document is the project plan, including:

- the description of the scope,
- information about risk and resource management,
- details of team organisation and structure.

The second part is the project schedule, it consists of:

- details of the tasks and activities, timeline,
- information regarding allocated resources,
- discussion of the selected process model,
- required documentation,
- and a list of assumptions.

The third part is the quality manual, which contains:

- information on quality control and management,
- reviews and audits, testing,
- details of tools for managing quality,
- tracking/change management,
- method of communication,
- and information regarding conflict resolution/negotiation.

Requirements/Scoping Document

This document will be reviewed and edited continuously and it aims to communicate the intended scope of work. This includes:

- the understanding of the data, which has been done in the feasibility study,
- preparation of the data,

- modelling, which is selecting modelling technique, generating test design, building model and assessing the model,
- evaluation,
- and the deployment.

Deliverable 3 & 4

For deliverable 3 and 4, the team needs to review the project plan, quality plan and requirements/scoping document and add design and testing document. User manual is needed for deliverable 4. These updated documents aim to communicate the plan and the requirement for this project well after being reviewed. D3 is due at 5pm, on the 29th of April, and D4 is due the 20th of May, 2021.

The following sections are included in both D3 and D4:

- Revised project plan, quality manual, with changes and revision history
- Revised requirements/scoping document, with changes and revision history
- Design document
- Testing document
- User manual (for D4 only)

Deliverable 5

For deliverable 5, the team needs to produce a final group reflective report. This document is for self-reflection and for the team to review what has been done throughout the project. This is due at 5pm, on 3rd of June, 2021.

Deliverable 6

For deliverable 6, the team may need to create a PowerPoint for the project presentation. This is still not confirmed. This is needed on the 6th of June, 2021.

Deliverable 7

For deliverable 7, the team is to present the final product to the client, a user manual from D4 will be ready if the client requests one. This is needed between week 13-16.

2.6. Assumptions

- The current data is up to date with what BI has provided
- We can use different software to achieve the machine learning or data cleansing
- More data will be added to the shared drive by BI during this project
- Assumptions and notes for the combined data set (Western, Eastern and Petwise)
    - Removed sensitive data in columns across all data sets to protect customer privacy (e.g. AccountName, Description).
    - The received western data set for the month of October is incomplete and could not be combined with November and December.
    - EXTERNORDERKEY column in Pet wise consolidated data sets is inconsistent.
    - Barcode represents product name in eastern data sets.
    - Western, eastern and pet wise data sets could not be combined as they have different attributes.

3. Handover Requirements

The purpose of the handover requirements is to specifies what BI needs in order to use the solution when this is being handed over, and what they will receive at the end of this project. This is essential as this section is to make sure the deliverable runs smoothly when the team leaves at the end of this project.

Key Stakeholders:

- Alison Barker – Head of CX
- Alexa Bentley – Commercial Policy Manager
- Sarah Bidner – CX and Insight Lead
- Other project team members: Nicholas Curtis (IT), Joe Helo (IT)m Navini Wijeratne (Finance), Shobhana Kambhammettu

Outputs:

- Sales business insight prediction

Software to install:

- Anaconda a. for Jupyter Notebooks
- GitHub Desktop

Needed hardware:

- Computers

Data/Scripts/Test cases/Testing tools:

- Data set needed: sales data, market data, wholesaler to retail
- Testing tools: SKLearn library import for Python

Training manual:

To understand the code, data, graphs and the result, employees should have some knowledge on data science. The following links are pre-provided training instructions that allow employees to understand the model we propose quickly and efficiently.

- Jupyter Notebook: Through extensive research into forming a training manual, the group has decided this external source of information has the best suited instructions for both beginners and experienced Jupyter users to either learn something new or regain basic skill sets. The link as below is from DataQuest : https://www.dataquest.io/blog/jupyter-notebook-tutorial/

- Modelling of Data: Through extensive research as well as personal experience from the group into how to model data, we along with countless others use this training manual to learn the basics of how to model data, with simple data modelling techniques from linear models to decision trees. The link as below is from Scikit Learn: https://scikit-learn.org/stable/user_guide.html
- Data Visualisaion: Through extensive research, we have determined that this user guide created by pandas is best suited toward the basics behind data visualiation, so that BI can understand the information we have given them. The link as below as stated is from Pandas: https://pandas.pydata.org/docs/user_guide/index.html#user-guide
- Graphs: Through extensive research, along with past experience with understanding data graphs, have decided that this link is best suited toward explaining the basics of graphs, along with how to read and understand them. The link is as follows: https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py

Eventually, if recourses allocate it, the group plans to create a personalised training manual, but for now this is the best plan of action to allow BI to understand the system.

A guide on how to use the solution:

- Follow the instructions on GitHub and Jypyter

Log in details:

- No log in details is needed

Access to important documents:

- This actual file

4. Quality Manual

The purpose of this quality manual is to outline the framework of the procedures, responsibilities and processes to achieve quality objectives for the team. The manual will detail the following topics:

1. Quality Control and Management
2. Reviews and Audits, Testing
3. Tools for Managing Quality
4. Tracking/Change Management
5. Communication
6. Conflict Resolution/Negotiation

All these areas aim to improve Boehringer Ingelheim's (BI) goals in delivering integrated data for efficient analysis and decision making while ensuring quality standards are met and minimising the potential of errors.

4.1. Quality Control and Management

Providing quality data through the new system is vital for BI since they will be using the data to create customisable reports and dashboards which will influence their business and operational processes and decisions. Therefore, there will be a strong focus on ensuring that the new system provides quality data. However, quality data is not something that can be fixed or improved upon. It must be quality data that is produced and made upon creation.

In order to sustain and achieve quality data from the beginning, the data created through the system must meet the following criteria:

- Accuracy - The information that the data contains is correct and corresponds to its intended purpose in reality.
- Reliability - Any data found or produced doesn't contract other pieces of data in other data sets or sources of data within the organisation.
- Completeness - All elements of the data are readily available in its entirety.
- Relevance - The data that is collected is appropriate and suitable for its intended purpose.
- Timeliness - The data contains up to date information and represents reality within a reasonable time period.

Low quality data can cause a large amount of problems and increase the probability of errors. Issues such as unreliable information or incomplete data can waste valuable time and money, or can lead to incorrect decision making. Hence it is vital in ensuring that the data and information from the new system is high quality and will allow BI to gather data from different sources and data sets to

make decisions driven by effective analysis. Meeting the criteria of quality data will mean that the system will provide data that will always be fit for its intended purpose and correspond to the realities that the data describes.

4.2. Reviews and Audits, Testing

The raw files and data sets that are sent by BI is through the cloud in the form of .csv files. These files will be imported into Excel and then Python to allow for creation of data frames, datasets and the ability to run audits and tests using these two applications. The use of Excel can allow for review of the raw data, ensuring that the data is clean and there are no errors present. This will involve doing an audit of the data through excel by taking note of any inconsistencies, duplicates, errors and any problems that are found. Here the data must follow the high quality data criteria to be tested successfully without any issues.

Testing the data will involve using SKLearn library import for Python which will provide access to tools used for testing such as train_test_splits, Bayesian Neural Networks, K-Nearest Neighbours and specifically, diagnostic tests from logistic regression. Using these tests will ensure the system is running as intended and whether it is accurate. This will be utilised to reveal patterns and insights into BI's data that would be used for analysis and decision making. The data values used for testing is either going to be categorical or continuous, or it can be a mix of both. This will depend on what data sets or values are tested and used, but the data can always be converted if needed. The model will be tested using SKLearn_Metrics which will calculate an accuracy score to determine where the model is accurate. Further regression diagnostic testing may be done depending on the results.

Pandas is another Python package that will be used for data cleansing. It makes data easier to work with as it has a variety of functions including the ability to handle missing data values, merging data sets, data visualisation, data normalisation and many more. The use of Pandas can correct any errors that are present and allow for more effective analysis with its visualisation functions.

4.3. Tools for Managing Quality

While Excel and Python are the main tools for managing and achieving high data quality data. There are other options to ensure that quality data maintained which are:

- Cloudingo - a Software as a Service tool that scans an entire dataset for duplicate records which can then be merged, converted or deleted without losing vital information. These scans can be done using predefined or custom filters allowing different data sets received from BI to be cleansed

to specific requirements. This function is automated and can be done regularly making it a useful tool to reduce data redundancy.

- Validity - Validity has a range of tools that are available to use to clean data. Specifically, their product DemandTools has automated functions that can manage data through getting rid of duplicates, importing data, standardising and much more. This allows for more efficient data management.

4.4. Tracking/Change Management

For this project, tracking management will involve ensuring that the quality of data and the system is at a reasonable standard. Therefore, when new raw data sets are received, it is important to use both Excel and Python to ensure that the data is cleansed and free of any errors or issues. The use of Pandas in Python will help maintain a high level of data quality.

The cleansing of data must be done regularly to prevent any errors or data redundancy which can have an impact on the new system. Any issues found during the data should be corrected immediately.

Change management will involve changing the processes and/or making organisational changes to achieve desired results or goals. This may happen for a variety of reasons such as changes to roles in the team, use of new applications and technologies, different data sets sent from BI, etc.

The first step in transitioning towards change is defining what the change is and why there is a need for it. Next, a plan must be created for each person's role in the process and what they must learn or do to achieve the new change. Once the plan has been finalised, it should be implemented over stages to make the transition easier for the team with sufficient resources and time to do so. Constant monitoring of each individual's progress should be regularly tracked to ensure that the change is on track to be completed.

For changes to be made, the following form should be filled out by the group member and signed off by the rest of the other group members. This is to ensure that the other group members have all read, understand and agree to the change. This well also keep track of tracking and managing changes in the project.

Project Change Request Form

This form is to request and signify any changes that need to be made to the project. In order to help track and document any change requests, please fill in the following form and have it signed off by the other group members.

Name: Date:

1. **Change description. What is the change?**

2. **Change reason. Why is this change being requested?**

3. **Impact of change. How will this change affect other areas of the project (E.g. scope, resources, timeline)?**

4. **Proposed action. What will happen in order for this change to be put in affect?**

Group member sign off:

Signature	Date

4.5. Communication

Effective communication among the team and sponsor involved in the project is essential as it can greatly impact the success of the project and minimise any misunderstandings and risks. Everyone needs to understand what their roles and responsibilities are and each individual's progress is. Furthermore, communication with the sponsor, BI ensures that both parties are on the same page and the team is on track with the project. These are the following communication methods that are to be used during the project.

- Weekly Zoom meetings with the team - The team has a weekly meeting over the cloud-based video communications application, Zoom, to discuss progress on what needs to be done and what is already completed. Here the team can also bring up any problems or issues that need to be addressed and what queries or documents need to be sent to the BI project team or the unit convenor, Deborah Richards. This is all summarised into the weekly report which is posted on the private group forum on the COMP3850 iLearn.
- Team group chat - The team uses Facebook Messenger to communicate during the week outside of the weekly zoom meetings. Here the team is able to communicate to each other any concerns and update one another on their progress of their part of the project. This should be the main point of communication to set up the date and time of the weekly zoom meetings to allow everyone to be available to attend.
- Email - The main method of communication towards the BI project team and the unit convenor will be through email. Through emails between the BI project team meetings can be scheduled, feedback on deliverables can be received and any questions regarding the project can be sent. Communication with the unit convenor will be mainly used for clarification and enquires on the deliverables and assessments.
- Optional weekly Zoom with the BI project team - Every Wednesday the BI project team has a weekly status meeting where they discuss their own progress on the project. It is optional for the team to participate in the meeting. In this meeting the team is able to discuss any queries or issues with the BI project team.

4.6. Conflict Resolution/Negotiation

At times there may be moments where problems and misunderstandings among the team which can cause conflicts during the project. The following four-step process aims to resolve an issue between two or more members of the team and reach a solution.

1. Define and understand the issue.

The first step is for both sides to agree on what the source of the conflict is, so that both sides recognise that a problem exists. This will involve discussing what both person's needs are not being met and getting as much information on the issue as possible.

2. Mutual agreement to resolve the issue and establish common goals.
Once the issue has been defined from both sides, they must agree to actively attempt to resolve the issue and discuss ways to establish and meet common goals. Establishing common goals can start the path to coming up with an equal outcome.

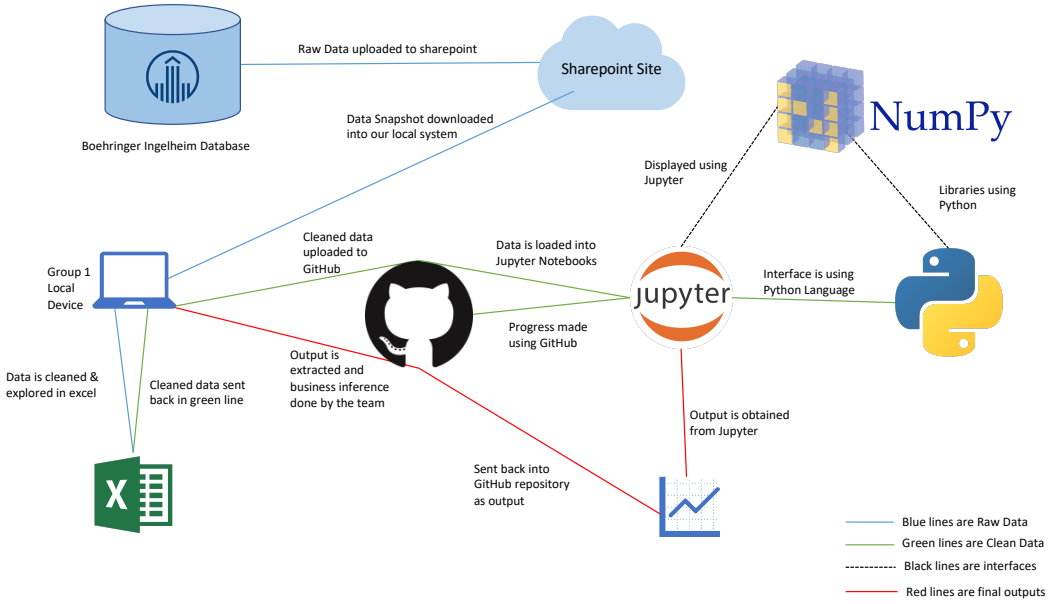
3. Explore ways to reach resolution.
After establishing common goals, both sides must discuss solutions that would enable them to reach these goals that they can both agree on. These solutions must also ensure that the root that caused the issue in the first place will not occur again.

4. Agree on the best solution.
Once multiple solutions have been laid out, both parties must agree on the best solution and state their responsibilities in reaching the resolution. This may involve changes in attitudes, work behaviours or approaches towards the project.

In the rare situations where both sides and the team are unable to agree or find a suitable solution, a third party will be contacted for guidance on how to proceed with finding a solution. The first main point of contact would be the unit convenor, Deborah Richards. For issues that are specific to the project, the BI project team will be contacted.

5. Scoping Document

Context Diagram



5.1. Business Understanding

Boehringer Ingelheim's (BI) current status with their Data uplift programmes is in the second stage and we are now looking at improving their data collection, data processing and implementing machine learning to enhance business outcomes.

We are working on this project with the following assumptions:

- The data provided is the data that BI will receive
- We are able to use different software to achieve the machine learning and cleaning of the data
- There will be more data being input to the shared drive by BI as the project progresses

Overall BI has a range of business objectives that this project plan covers both directly and indirectly. The overarching business objective for BI is to innovate and implement technology to improve both CX operations and to better know the profitability and experience of their product lines. This will also impact the distribution channels with new insight on where the business should have their focus on.

The two main business objectives include:

- Customer Insight based on customer experience
- Customer spending habits in align with financial assessments

Moreover, there are also multiple levels of priority toward objectives. Primary Objectives of BI include:

- Data sets in a central collection: Currently within Boehringer Ingelheim, analysis of data is difficult as most of the information is spread out within both internal and external partners, and includes valuable information toward data mining goals.
- Validation of Data
- Customised Reports based on data mining goals
- Permission allowance for individuals with multiple objectives within a single role
- Overall improvement in existing infrastructure, platforms and business processes
- Improved Potential for Automated Email within updating of business infrastructure

Examples of Secondary Objectives include:

- Automated concession calculations to compliment time reduce
- Market trend analysis based on data mining goals
- Improved Market Models
- Improving cost serving customers to increase profit margins
- Price of Service analysis through data mining goals

Within BI, the current situation the business faces are the outdated forms of information flow within the business. This in turn causes disruption to the effectiveness of employees to complete their job, as previously stated, data is spread across the business both through internal and external systems. This spreading of information with no form of structure is what influences the previous objectives above.

Moreover, there is investigation into the current situation of BI, including hidden issues that cause the situation they currently face.

Examples of this include:

- Outdated flow of information and processes: Within the technological requirements of BI, the current situation is that the lack of a central depository with access ports for information is causing dysfunction to the effectiveness of employees.
- Ineffective Data mining goals: The current situation of BI stipulates that a greater focus on customer information and future customer prediction plans are needed as the current procedures fail to keep up with the dynamic nature of the business itself.

In summary, the current situation of the business is structured toward an update of all overall procedures, mainly including information data flows, automated procedures, and customer analysis goals.

BI has a focus on data mining goals mainly related to customer reports, with a second focus on overall performance analysis. The data mining goals that have been determined are as follows:

- Brand and customer value sales and trade analysis that could transform into reports
- Investment strategies based on data mining of past performance toward rebates and marketing campaigns
- Customer Segmentation analysis with a specific focus into effective selling methods for customers.
- Analysation into future predictive customer marketing trends including product analysis into demands for the future.
- Analysation into competitor trends relating to customers and similar product marketing techniques that could be associated with BI.

Overall, there are also data mining goals that based on analysis on the business suggests importance toward benefiting the business as a whole. Examples of these include:

- Optimization of resources: Within most businesses, data mining analysis into how resources should be spread out across the business based on previous performance is necessary toward improving efficiency among BI.
- Prediction: while prediction data mining goals are apparent within BI, prediction into future competition business plans should also be taken into consideration to give BI an upper edge against its competitors.

5.2. Data Understanding

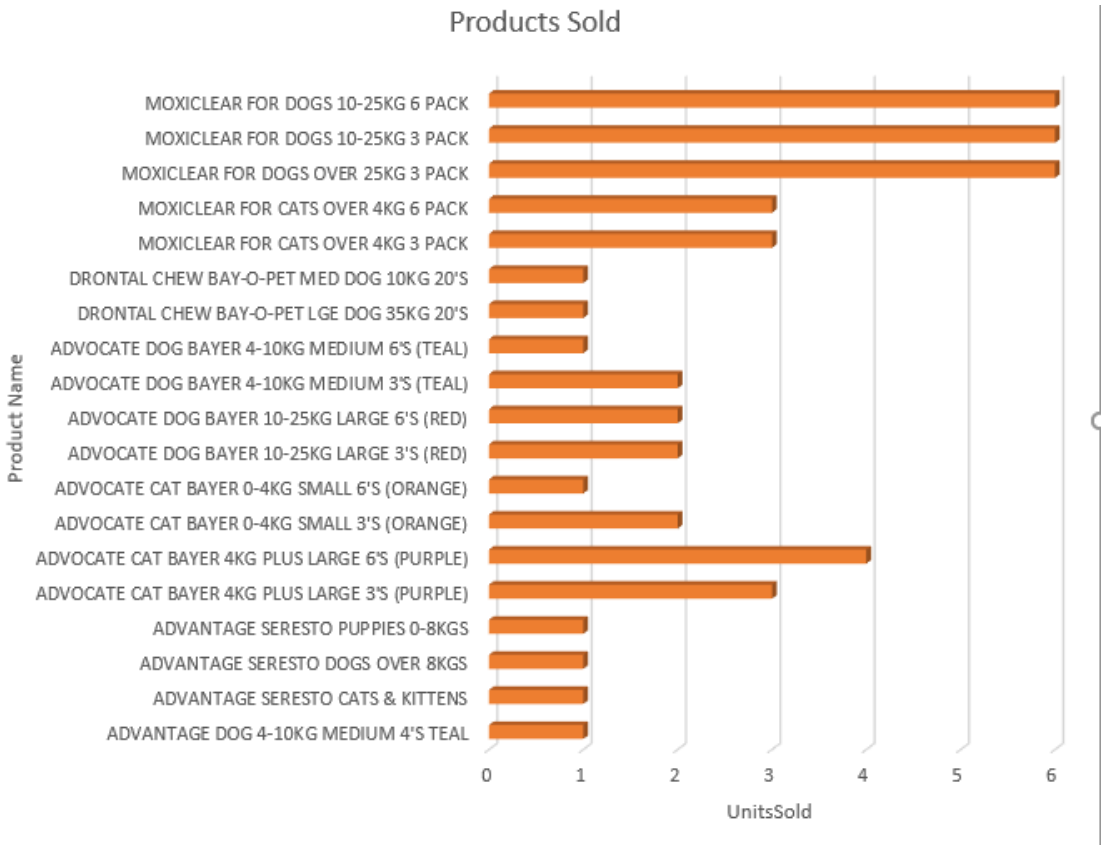
The sources of the data gathered are based on previous data analysis provided through raw sample data sets that we are able to generate reports and queries on. More specifically, this sample data is based on customers, competitors, and marketing procedures. These sources are also direct and indirect according to BI.
These sources specified include:

- Historic customer purchases
- Competitor information
- Customer & Brand Level rebate payments
- Historic marketing strategies and budgets
- Location based customer data
- In depth customer information including items looked at, items returned, customer entitlements, and customer reports.

Since we are receiving .csv files, the data capturing process will be mainly through importing the file into Microsoft Excel and into Python in order to create data frames and to massage the data into a form that is suitable for us to conduct machine learning on.

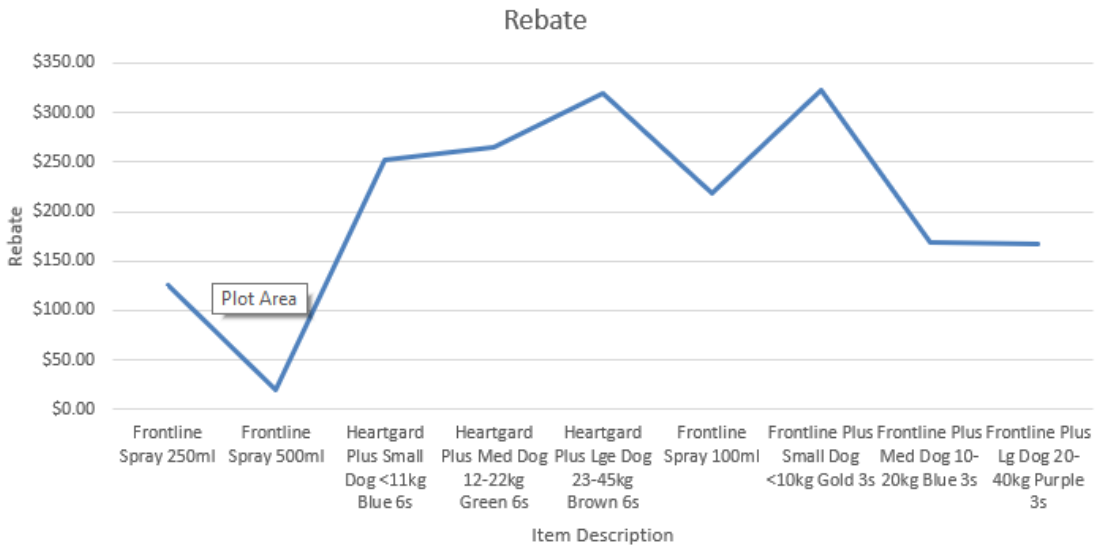
For Data Collection, we have a cloud storage for the raw datasets that are being sent to us, managed by BI, and we will download a snapshot of it when we are in the development stage of our project.

Subsequently, we will monitor and communicate with BI on future uploads of raw datasets & update the Jupyter notebook file and the relevant sources into GitHub for the development process in order to keep the data consistent. Throughout BI, there is a range of data sets that are specific to the department they are formed from. Each report or data set created has its own description and insight into overall capabilities of the business. Furthermore, these are the various pipelines that are evident within the data collection methods we conducted based on the raw sample data sets that were given for this project:



This graph is a representation of the sample data BI has, which is based on a December 2020 Sales Report. On the vertical axis, lies a set of data called Product Name. Product Name is the specific name of the item being sold to a range of customers and companies alike, including specifics like weight, brand, and quantity product. On the horizontal axis, lies a set of data called Units Sold. This data set describes the amount of units the previous entity described has sold, for example, in the table above, Moxiclear for Dogs 10-25KG 6 Pack sold over 6 units.

Another Example comes from the rebate table:



This graph is a representation of the amount of rebates each item receives based on its location, quantity, and popularity. On the horizontal access, lies the rebate data set. The debate data set is described as the discount or percentage off it receives based on the item. This is directly linked to the vertical access title, Item description. Item description is based on the manufacturer's information on what the product is, what it does, and the quantity of the product within its packaging.

Furthermore, the last example relates to data of products relating to its unique identity:

SKU	Name
7383	** Advocate For Cats And Small Kittens Up To 4kg Single Dose
7384	** Advocate For Cats Large Over 4kg Single Dose
7385	** Advocate For Dogs Large Red 10-25kg Single Dose
7386	** Advocate For Dogs Medium Blue 4-10kg Single Dose
7387	** Advocate For Dogs Small Green Up To 4kg Single Dose
7388	** Advocate For Dogs XLarge Over 25kg Single Dose
3285	*** 12 Pack *** Heartgard Plus For Large Dogs 23-45kg Brown 12 Chewables
3286	*** 12 Pack *** Heartgard Plus For Medium Dogs 12-22kg Green 12 Chewables
3284	*** 12 Pack *** Heartgard Plus For Small Dogs Up To 11kg Blue 12 Chewables
3653	*** Combo *** Milbemax Allwormer For Cats 0.5 - 2kg 2 Tablets x 2
4255	*** Combo *** Milbemax Allwormer For Cats 0.5 - 2kg 2 Tablets x 4

In this table, there are two categories of information that directly lead to each other. On the left, this data set is called SKU. The SKU is described as the stock taking unit, in order to help stock takers to effectively categorise and record the SKU of certain products. In the second column, is the simple name of each product that SKU is associated with.

The sample data BI has provided allows in depth insight into exploration and describing of data, through data collection methods and past historic reports.

Additionally, one of the key fields that would should be considered in the data understanding phase is the relation of the customer data such as the “Wholesale customers” and the “BI customers” and these different fields affect the data as mentioned above as this gives another dimension of analysis that could be done based on the customer type.

This would need to be further explored in the further section during data preparation and during the modelling phase to ensure that the predictions done would be in line with the business understanding and the outcomes we would like to achieve.

Overall, one of the important steps toward the implementation of the new system, is ensuring data quality.

To ensure data quality throughout the implementation of the proposed system, as well as the current situation BI has, there are basic elements of data quality that must be followed. They include:

- Accuracy: This element relates to how correct the given data sets, as well as data sources are that ensure data quality
- Reliability: This element refers to the validity of the data set and sources of data, compared to other sources of data within the organisation
- Completeness: This element refers to how detailed the data sets given are toward the purpose of its creation.
- Relevance: This element is closely aligned with completeness, as it refers to how accurate the given sources and data sets are toward its intended purpose.
- Timeliness: This element refers to the date data sets were created, and whether they are still useful toward the present future.

Overall, these elements are the basis toward which we created criteria to ensure that data loading is successful, based on the initial dataset. These include:

- All similar attributes across the csv files are the same type
- There are no empty cells or incompatible tuples within the dataset
- The values that are to be merged have proper Primary or Composite Primary keys to identify them from the other sources in the same timeframe.
- All the datasets contain the same product keys etc.

Based on these criteria, there are a range of data quality tools that could be used toward addressing data quality, selecting data, cleaning data, constructing data, integrating data, and formatting data.

Two examples of these include:

- Cloudingo: Cloudingo is a popular data quality enforcer as it reduces data redundancy and improves data integrity, which for BI could be useful toward finding data inconsistencies throughout their various data sets. This program also has a 10-day free trial in order for the company to

have detailed use of the program to see if it is viable for the company's goals. The program also offers easy to use drop down menus and graphs, which especially for a company like BI where customers are prioritised, the company can spend less time ensuring data quality, and more time on ensuring customer satisfaction.

- Validity: Validity has a range of tools readily available for BI to use, which were specifically made for customer relationship management meaning it is closely aligned toward ensuring data quality specifically within BI. This program also has a free trial meaning that it can be used as a trial with the given sample data sets to ensure its accuracy and relevance in the company based on customer and employer needs.

Overall, through the methods above, BI has a range of methods readily available to ensure data quality.

5.3. Data Preparation

Firstly, from the main data sets, we will include the attributes: { Unit Price, Units Sold, Manufacturer Name, Product Name, SKU, Customer Number, PostCode and Distributors Name } from the sales report.

Additionally, for the B2B sales reports, we will include {ShippedQTY, SKU, Company}

Now that we have these consolidated fields, we will now need to combine them into various entities and sort them into their types depending on the availability of the information.

For example, with PetWise Data, we do not have a listing price within the dataset so we need to find another dataset source that contains the same SKU and price to ensure completeness of the data.

Our Consolidated Dataset needs to be processed and broken down into 3NF/BCNF such that we can efficiently utilise these tables. We need to check the tuples for every attribute to ensure that the types are the same and that there are no outliers within the data.

Regarding the integration of multiple data sets, this is established through simple methods of 1NF/2NF, or otherwise known as normalisation. In simple terms, by establishing similarities between data sets, through clearing of data redundancy methods, we can integrate and create a single file for all data sets. This is furthered through the method of first normalising and combining the different sets of data sorted by year or by other unique identifiers, which then allows us to create one combined data set.

This can be done through the use of aggregate functions on python with the

Pandas library and supplemented with checks using Microsoft Excel as we can construct simple histograms or scatter plots quickly depending on the column selected. This will give us a quick overview of the data within that attribute.

Now, given that we have clean and normalised data, we can create attributes that will allow us to get a better insight into the performance of the products. For example, we can utilise the time data to give us the average change in sales per month and that is a simple way to determine what is the trend in performance for a particular product which can then be broken down into dashboards for business insights.

Combining the datasets into Product, Sales, Customers and the location tables from all the data provided and subsequently removing duplicates based on their primary keys will allow us to gain granularity on what transformation we can apply onto the respective attributes.

Based on the initial data exploration, we can generally assign the following key attributes to their data types to load into GitHub and subsequently Jupyter Notebooks to act as a cloud-based data storage mechanism. These would be the initially loaded fields and should there be any transformations required, they will be added into this table below.

Attribute Name	**Description**	**Data Type**
Period	Time Period	Date
Qtr	Time Period in Quarter	Date
CalYr	Calendar Year	Date
Manufacturer	Manufacturer Name	VarChar
SubCategory2	What type of product is sold	VarChar
SubCategory3	Based on SubCat2, what category of product it is	VarChar
Brand	Brand of the product sold	VarChar
Product	Name of the product sold	VarChar

ProductCode	Internal Product Code used	Integer
Species	What species of animal that the product is for	VarChar
Region	Which state in Australia the sales are made to	String
Units	How many units were sold	Integer/Double/Long
Value	Price per Unit	Double/Long/Float
BI Business Unit	Which BI Business Unit made the sale	VarChar
Loyalty	What level of loyalty the customers are at	VarChar
Rebate	How much Rebate was given to the customers	Double/Long/Float

In some data sets, the process of data preparation is halted due to missing data. There is a range of possible methods that could be used to establish a range of consistency through the data preparation stage. One of those methods relates to regression techniques. Regression techniques allow us to predict what value would fit in the missing data column based on past analysis, and relationships between other columns. Past analysis on past datasets, as well as the relationships created by BI themselves, act as a source of data for these regression techniques. By using regression techniques, instead of simply dropping the rows that have missing values which can cause the loss of valuable information, we are able to consistency fill in missing data values without compromising data validation, or cause data redundancy through strict conditions within data regression.

5.4. Modelling

Since we are using the SKLearn library import for Python, we have access to packages such as linear regression, logistical regression, train_test_splits, Bayesian Neural Networks and K-Nearest Neighbours. There are multiple applications with the Naive Bayes such as word dictionaries to process keywords and probabilities to determine if a sale would be made in a certain state which would give the business better insight on where to focus their operations efforts in.

These various modelling types will allow us to test out various ways we can utilise the data to bring business insights for BI. Some specific examples include using logistic regression in order to determine which product would be the most

profitable in the next month.

This section of the document will change as the data is transformed and if the required modelling techniques needed to produce the outcome is insufficient, we are required to do more research to find out what other tools are available for us to use.

Depending on the model being built and tested, the data inputs that are given may need to be continuous or categorical in order to achieve a more accurate result. This will mean embedding certain value or type constraints into the dataset as we create the data frame in Python.

There is a range of constraints that need to be covered within this model. These constraints include but are not limited to:

- Resource Constraints: The model must fit within the constraints of both the project requirements as well as BI's feedback. For example, one of the constraints is that the project must be based on free to use technology with no cost required.
- Technology Constraints: The model must fit within the minimum requirements that we have decided for the model. For example, based on the information that BI has given us, we are able to confirm that Jupiter is one of the minimum requirements used for the model.
- Periodic Constraints: The model must fit within the maximum time frame given for the project, including the periodic updates we must complete to ensure the project is consistent throughout the entire plan.

Moreover, more specific examples of constraints include:

We should constrain "delivered items" to a binary result of Yes or No before carrying out the logistical regression as that will allow us to better predict if the sale would carry through or not. Conversely, we would want to ensure that the price value or the total value transacted is a continuous variable in order to obtain more accurate predictions of the total amount sold for a particular product.

These constraints are bound to change as we decide to include a different number of variables and if the model evaluation does not show us a significant result.

After building all the models, we would be required to conduct diagnostics tests based on the predicted and residual amounts from the regression. These diagnostic values will give us some insight into the model's viability and accuracy in being used and what potential changes could be made to said models to make them more valid.

We can utilise part of the SKLearn import for model testing by importing the Accuracy Score libraries, this will allow us to have a nominal score of the model that has been trained and how accurate the prediction is based on the train test split. This will be compared to the other diagnostics tests and will convey a

simple explanation for the model results.

In order to build these models we need to ensure that the sales data has been prepared as in section 3 and we are given the data across various months in order to conduct predictive analysis.

The Data Quality standards have been set up as above in section 2 but to reiterate, for the most optimal set up of the project, the dataset should be Accurate, Reliable, Complete, Relevant and Timely. This would mean the data frames loaded should include time variables for us to do predictive modelling and Analysis of Variance (ANOVA) analysis for the models. The Normalisation of the complete data set will be converted into data frames for use within Jupiter Notebook. Once the data frames are set up, only then can we conduct the regression needed and build the model.

Overall, these modelling techniques are based on the problem of classification rather than regression. Specifically in this project, the classification problem allows us to determine possible whereabouts for the potential of growth for profit margins are, which is based on the feedback from BI as well as the added potential for predictive abilities that were suggested to be included in the project. On the other hand, due to recent feedback from BI, the problem has shifted to include elements of both a classification problem, as with the future prediction capabilities, and a regression problem, as through the feedback it is now evident that regressive analysis is needed to cover the basic needs of the project.

5.5. Evaluation

The process to evaluate the project is a simple four step process that allows us to overall re-assess the project plan, along with any requirements we have missed during the overall implementation of the project itself for BI.
The four-step process is as follows:

1. Planning

Through analysis into our project plan, including cross referencing with the original feasibility report, we are able to verify and consolidate important objectives that we strive to include. Moreover, through this planning stage of evaluation, it is important to also take into account the primary & secondary stakeholders of BI, which are crucial to their success. The primary stakeholders include:

- Customer Experience
- Commercial Policy
- Animal Health Finance & Controlling
- Key Account Managers/Sales Managers
- Brand/Marketing Managers
- The secondary stakeholders include:
- Head of Pet/Equine and Production Animal Business Segments

- Head of AH
- Through evaluation of the project plan to meet the needs of stakeholders, including effects into stakeholders, step one of evaluation can be completed.

2. Implementation

Evaluation during the Implementation of our proposed plan is crucial to the success of the project. Through evaluation during implementation, we are able to determine if the goals of the project are being met, and also develop the minimum requirements for the project. Other goals of evaluating during implementation include:

- Ethical and Social Implications
- Legal Implications
- Resource conserving
- Environmental implications

Within BI, these four goals of successful evaluation are crucial to how society views BI overall, as public image, especially within the implementation of a new system is important toward BI system standards.

3. End-Of-Project Evaluation

Evaluation at the end of the implementation plan is crucial to the success of the new system. Through evaluation at the end of the implementation phase, analysis into if the new system meets the desired needs of BI are possible. Moreover, during this phase, it is possible to also see if the minimum requirements of the project in comparison to the needs of BI have a direct impact short term and long term.

Moreover, there is a range of evaluation metrics that allow us to determine the levels of performance the model allows. Furthermore, there is certain specific evaluation metrics that evaluate both classification and regression metrics. Examples of these metrics that could be used include:

- MSE or Mean Squared Error evaluation metric: This metric allows us the evaluate the profitability of certain data sets through regression lines, and how close each point is together, to determine what can be done to improve profits.
- R Squared evaluation metric: This metric allows us to determine if the model itself fits within the data based on a certain valued range, or to determine based on this value if somewhere in the code we have made an error.

4. Diffusion Evaluation

Along with analysis into data quality including the sources of data, it is important to analyse what type of information the new system creates, but more specifically who receives this new information at BI. Moreover, this can be established through test cases, analysis and research into relevancy of data based on which department is receiving/requesting the data.

Thus, this four-step evaluation plan where every stage of the project is evaluated is crucial toward the success of the new system for BI.
The overall minimum requirements of the project are to rework the existing system into a more user friendly, interface friendly, and faster system that allows all sources of data to input information into one central depository, for faster work among all departments.

More specifically, the main requirements of the project are to assist for the future of:

- Automated Rebate Calculations
- Value Chain stock hold
- Market and Competitor Trend Analysis
- Future planning for investment strategies
- Customer Interactions and Costing
- Customer P&Ls based on Customer Communication Managers
- Validation of Data
- Analysation into dashboards

Upon completion of the project, the review process is similar toward the evaluation process, in terms of comparing our goal toward our work. More specifically, the review process specifically useful for this project relies on Post-Implementation Reviews. Post-Implementation reviews focus on the thoughts of the stakeholders of the business, which especially within BI is crucial to its success, as they have supported the project throughout the course of its implementation. Within Post-Implementation reviews, there are various activities that must be conducted during this phase, based on research into the company, and online research. These include but are not limited to:

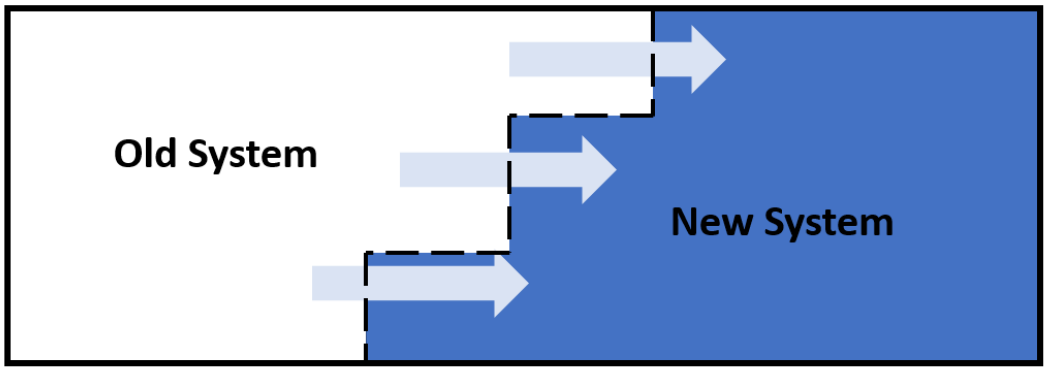
- Surveys for Stakeholders, prioritizing those stakeholders who received biggest system changes
- Holding Review meetings, where stakeholders of BI can ask questions, suggest further adaptations, or overall question certain aspects of the new system.
- Interview with each stakeholder, in order to hear any unheard thoughts during the review meetings
- Create a PIR report, which is a collection of all thoughts and queries of stakeholders in order to eventually and possibly adapt the system further to their given needs.

After the review phase, determining the next steps for BI is as simple as evaluating what the needed adaptations for the newly implemented system entail. Moreover, specifically they include:

- Needed aesthetical changes
- Revised goals for stakeholders
- Bug and visual problem fixes
- Re-establish the current situation, and create another project plan toward fixing and adapting changes into a revised system.

Determining the next steps is completely dependent on the success of the new system to meet the current requirements of BI, as well as the stakeholders of BI effectively using the new system as intended.

5.6. Deployment



Through analysis into the system, as well as the stakeholders' needs of BI, the chosen implementation plan refers to a term called phased implementation. Phased implementation refers to the process of individual testing of each added component toward the finished plan. In this process, there are certain advantages that relate to BI. These include:

- Ease of Training: Participants and stakeholders within the new system can slowly and gradually learn the new system bit by bit as the phased implementation is only one adaption at a time, until gradually the whole system is adapted
- System Delay: Within the testing of each individual component, the next stages of the new system can be delayed if new problems or bug/visual glitches are found.
- Comparison: Stakeholders can see head on the difference between the new system and old system, meaning that these stakeholders for BI are able to see if the system is beneficial toward their needs.

The individual components that we are going to convert into our new solution are the visual aspect of the dashboards and the predictive analysis component. BI's old system largely refers to their excel operations and we are now aiming to convert it into a more robust and systematic platform of Jupyter Notebooks. There are initially 3 steps toward completing the implementation of the new system. This includes:

1. Currently, our project is at the project plan phase and we will have our Minimum Viable Product (MVP) by the end of April. This would include the MVP prototype itself and a presentation with BI to showcase the prototype against some test cases.

2. After the initial presentation of the prototype, adjustments and subsequent versions of the prototype will be communicated with BI as we have a clearer idea of what would suit the organisation's needs best and we can produce a comprehensive user manual and make adjustments to the scoping document should there be a need to. Once the final version is done, we will have a demonstration of the product to show the prototype live and the results to the

client, academics and the rest of our cohort.

3. Finally, we will deliver the completed product with the full instructions on how to install our solution and to answer any further questions from BI regarding this project. We will include any potential adjustments that can be made in the future to the project and the completed user manual as documentation.

Furthermore, there is an important need to determine methods of training employees in order to use the new system. Moreover, there is a five-step system that covers the overall training of employees through a piecemeal approach. They include:

- Modality: Within BI, one of our main priorities is ensuring that resources are used appropriately. This step refers to if training requires new technology, or devices for employees, or if they can be done on existing devices. Moreover, the new system requires an approach toward training that involves a trainer and the trainee. In this case, the approach that would train employees would be instructor led training. This involves a traditional classroom approach where the trainer demonstrates and teaches individuals the new system, as well as answering any questions they might have
- Microlearning: Within BI, our main approach is to address every problem one step at a time. Similarly, while training employees, the main approach is to slowly allow employees to understand the new system, instead of providing everything at once. This in turn allows for more effective means of training.
- Marketing: Within BI, the term marketing refers to the marketing of products, or strategies that could be used to increase profit. Within training terms, marketing refers to explaining how the new system benefits the users, as they have to use personal time in order to learn the new system. Small in-company marketing strategies should be used to make aware to employees that a new system is approaching, as well as displaying its benefits for employees themselves.
- Maintenance: Maintenance of the new system is crucial to success. Similarly, maintenance of training strategies, procedures and programs is also crucial to the success of the training system. This includes supporting the training program, such as a weekly overview of employee progress to test its effectiveness and if it needs tweaking.
- Hypercare: Hypercare refers to supporting the employees while the new system is implemented. Strategies that could be useful toward providing hypercare include:
    - Conducting interviews to test if employees are learning, to answer questions, and understand the overall mood of employees toward the new system
    - Guidelines and tools to help employees that suffer from over working, stress, and zoom fatigue.
    - Providing emotional support, including small events such as providing free meditation sessions or longer lunch breaks

Furthermore, at this stage the final product relates to a Jupyter based solution that covers the problems based on the maximum resource requirements the team has for the model, which is furthered through certain limitations we have due to the constraints given. Moreover, the deployment is based on the interactions between BI and the team, as determination into the final product is solely based on the needs of BI. This in turn covers the final product, but there are certain other procedures that are apparent no matter the final result. For example, through explanation and teaching of our model, we are able to give BI certain constraints in order to maintain and monitor the final product so that the solution/model that is given works toward their advantage. Furthermore, another example relates to certain scenarios where adaptation is needed for the model, based on the changing needs of BI. In this case, we are able to provide steps within the monitoring stage in order for the BI Staff members to adapt the system without external help, as the system is easy and simple to use.

Overall, while training is important, timeliness of the project is also crucial toward the success of the system. This in turn relates to having a communication plan. While overseeing the system, a communication plan can have a range of benefits, including providing insight into how the team operates, and what time frame the new system is split into. Moreover, a communication plan specifically clarifies information flows between individuals, such as meeting times, who attends meetings, and what specific topic is discussed. In addition to this, a communication plan also allows BI to understand the budget of the system, while not referring to physical cash flow, but rather to the time used to understand and potentially implement this cash free system.

5.7. Feedback from BI

BI Meeting with Group 1. 31st March 2021, 3pm

Feedback Received

1. **Business Understanding:** you have really captured the nature of the project and the needs of the business in improving our information flows for better customer and performance analysis.

2. **Data Understanding:** I realise I've shared a lot of different data sets to give you some insight into the challenges we face and the different types of data we are trying to integrate. Your data understanding section goes through some of these different data sets and I'm wondering if I've made it too complex and whether you should just focus on Petwise/Eastern/Western data (as an example) or just the market data. I'll leave it up to you. The other element that you've captured Data Preparation, but I can't see it in data understanding is the customer attribute (often there is both a Wholesaler Customer ID and a BI Customer ID – we have mapping files in the global data warehouse to link these). In addition, you've mentioned manufacturer name as an

attribute, but to expand on that – these are competitor sales, which is the holy grail of sales data. So, there is a big opportunity to do more detailed and predictive analysis on customer buying behaviour of both ours and competitors' products in the future, once the foundation has been set with the information flows and data processes that you're proposing. I've also attached the templates our IT team are asking us to use to upload data into the global data warehouse.

3. **Data Preparation:** No issue. Happy that we recognised that Data Prep is the biggest component. Currently can approach Alexa about data learning.

4. **Data Exploration/Modelling:** This also looks good. Please let me know if you need me to share more data than the sets I've already shared – I've only provided one quarter of data, so happy to share a larger timeframe if needed.

5. **Evaluation:** If you would like to talk to any of the other stakeholders (primary or secondary) to understand their needs, let me know and I can set it up. Stakeholder management is critical in any project, so I'm glad you've captured that.

6. **Deployment:** Should I set up some time towards the end of April for you to demo the prototype/work achieved? I'll have a look in calendars and send through some options. You've mentioned training, it might be worth expanding on this as to how users will be trained, and what kind of hypercare will be offered. There are always teething issues, so how can the new solution be supported. And finally, you could mention recommending that BI have a communication plan to prepare, then announce and then review the change/solution.

Team action taken:

1. Adjusted deployment section accordingly to expand on the training manual content.
2. Requested for more time series data for modelling of the project.
3. Presented the scoping document to BI Team. (Alison, Nicholas, Alexa and Sarah)
4. Set up meeting for End April to next presentation.
5. Added in a more in-depth contextual diagram for BI to understand our data flow.
6. Conveyed intentions to work on Eastern, Western and Petwise data as part of our data understanding effort.

6. Prototype/MVP

Introduction:
This Jupyter notebook acts as the MVP for the COMP3850 Computing Industry Project and we will include any code or research done in here.

Research:
Due to the nature of the project, we will be exploring using python code to try determine some business insights for BI with their given sales data.

Aims:
MAIN GOALS - Where should Boehringer Ingelheim focus their sales efforts to. From the main goal, we can break it into smaller components.
SUB-GOALS

1. Which business unit has been performing so far (Can be done graphically)
2. Modelling to be done for predicted level of sales

Our Approach
To answer our hypothesis, we are using the following approach.

1. Data preparation: load and transform the data as preparation for the analysis
2. Answering sub-hypothesis 1 through exploration and visualisation of the dataset
3. Answering sub-hypothesis 2 through creating models in order to try to predict sales levels

Packages

```
]: import seaborn as sns
   import pandas as pd
   import numpy as np
   from datetime import datetime
   import matplotlib.pyplot as plt
   %matplotlib inline
   from sklearn.linear_model import LogisticRegression
   from sklearn.linear_model import LinearRegression
   from sklearn.model_selection import train_test_split
   from sklearn.feature_selection import RFE
   from sklearn.metrics import confusion_matrix, accuracy_score
   from sklearn.metrics import mean_squared_error
   from sklearn.metrics import mean_absolute_error
   from sklearn.metrics import r2_score
   from scipy import stats
   from scipy.stats import pearsonr
   from numpy.polynomial.polynomial import polyfit
   from sklearn.neural_network import MLPClassifier
   from sklearn.feature_selection import RFE
   from sklearn.tree import DecisionTreeClassifier
   from sklearn.tree import DecisionTreeRegressor
   from sklearn.cluster import KMeans
   from matplotlib.pyplot import imread
   from sklearn import preprocessing


   import pmdarima
   import datetime
   #To remove the red box errors just to make the notebook look neater
   import warnings
   warnings.filterwarnings("ignore")
```

6.1. Dataset and Data Frame Manipulation

Market Data

We have combined and cleaned the data from excel and are going to load the current data set as of April 6th into various data frames.

We will conduct our exploration and analysis on this dataset. We have made some revisions to the initial dataset we are loading in order to facilitate the modelling and to make the analysis more in line with the business objectives.

Data Cleaning and Manipulation

In [129]:
```
# Import the Distribution/Sales Data into Jupyter
eastern = pd.read_csv("Data Extract/CSV/EASTERN-DATA-COMBINED.csv")
western = pd.read_csv("Data Extract/CSV/WESTERN-COMBINED-DATA.csv")
combined = pd.read_csv("Data Extract/CSV/RENAME-CONSOLIDATED-DATA-COMBINED.csv")
market = pd.read_csv("Data Extract/CSV/PACE students - market data (confidential).csv")
wholesaler_to_retail = pd.read_csv("Data Extract/CSV/PACE students - wholesaler to retail.csv",encoding='cp1252')
```

In [5]:
```
wholesaler_to_retail.columns
```

Out[5]:
```
Index(['Year', 'Month', 'Distributors Name', 'CUSTOMER ID', 'Postcode',
       'WHS_SKU_ID', 'Suppliercode', 'Product', 'Quantity', 'Price'],
      dtype='object')
```

In [6]:
```
txt = " 'VENDOR','SHIPDATE','ORDERKEY','CONSIGNEEKEY','C_COMPANY','C_ZIP','EXTERNORDERKEY','SKU','ORIGINALQTY','SHIPPEDQTY' "
print(txt.lower())
```

```
 'vendor','shipdate','orderkey','consigneekey','c_company','c_zip','externorderkey','sku','originalqty','shippedqty'
```

Aggregating data

Market Data

In [7]:
```
market_drop = market.drop(['CalYr','MAT', 'YTD',
       'Brand', 'Product', 'ProductCode',
       'ProductionCompanion', 'Region', 'Units', 'Units ADJ', 'Doses',
       'Manufacturer ADJ', 'Brand ADJ', 'Product ADJ',
       'SubCategory3.1', 'Species ADJ', 'Doses ADJ',
       'Doses / Unit', 'Monthly Doses',
       'BI Market Only FLG'],axis = 1).dropna()
```

In [134]:
```
market_drop
```

Out[134]:

	Period	Qtr	Manufacturer	SubCategory2	SubCategory3	Species	Value	YEAR	BI Business Unit
0	2020-10-01	Dec-20	Elanco	Parasites - External	Cattle Tick & Worm Pour-On	Cattle	103385.00	2020	Cattle & Sheep
1	2020-10-01	Dec-20	Elanco	Parasites - External	Cattle Tick & Worm Pour-On	Cattle	95091.00	2020	Cattle & Sheep
2	2020-11-01	Dec-20	Elanco	Parasites - External	Cattle Tick & Worm Pour-On	Cattle	29946.00	2020	Cattle & Sheep
3	2020-11-01	Dec-20	Elanco	Parasites - External	Cattle Tick & Worm Pour-On	Cattle	200.00	2020	Cattle & Sheep
4	2020-11-01	Dec-20	Elanco	Parasites - External	Cattle Tick & Worm Pour-On	Cattle	32790.00	2020	Cattle & Sheep
...	...	...	...	...	...	...	...	...	...
199394	2018-03-01	Mar-18	Vetoquinol	Nutrition & Metabolism	Vitamin & Mineral - Oral	Cat & Dog	3273.14	2018	Pets
199395	2018-03-01	Mar-18	Vetoquinol	Nutrition & Metabolism	Vitamin & Mineral - Oral	Cat & Dog	1435.84	2018	Pets
199396	2018-03-01	Mar-18	Vetoquinol	Nutrition & Metabolism	Vitamin & Mineral - Oral	Cat & Dog	3130.32	2018	Pets
199397	2018-03-01	Mar-18	Vetoquinol	Nutrition & Metabolism	Vitamin & Mineral - Oral	Cat & Dog	1341.17	2018	Pets
199398	2018-03-01	Mar-18	Vetoquinol	Nutrition & Metabolism	Vitamin & Mineral - Oral	Cat & Dog	1037.10	2018	Pets

199091 rows × 9 columns

In [135]:
```
# Merge data to Period with Units sold
market_drop_unit = market_drop.groupby(['Period','Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3', 'Species',
       'YEAR', 'BI Business Unit']).sum().reset_index()
market_drop_unit
```

Out[135]:

	Period	Qtr	Manufacturer	SubCategory2	SubCategory3	Species	YEAR	BI Business Unit	Value
0	2018-01-01	Mar-18	Bayer	Alimentary System	Antidiarrhoeals	Cattle & Cat & Dog & Horse	2018	No Bus	25510.00
1	2018-01-01	Mar-18	Bayer	Anaesthetics	Anaesthetics Local & General	Cattle & Sheep	2018	Cattle & Sheep	74344.00
2	2018-01-01	Mar-18	Bayer	Anaesthetics	Anaesthetics Local & General	Sheep	2018	Cattle & Sheep	2120.00
3	2018-01-01	Mar-18	Bayer	Antibiotics	Antibiotics Oral - Companion	Cat	2018	Pets	1050.00
4	2018-01-01	Mar-18	Bayer	Antibiotics	Antibiotics Oral - Companion	Cat & Dog	2018	Pets	31783.45
...	...	...	...	...	...	...	...	...	...
15859	2020-12-01	Dec-20	Zoetis	Vaccines & Antisera	Vaccines Other	Other	2020	No Bus	241116.48
15860	2020-12-01	Dec-20	Zoetis	Vaccines & Antisera	Vaccines Pig	Pig	2020	Swine	1290803.56
15861	2020-12-01	Dec-20	Zoetis	Vaccines & Antisera	Vaccines Sheep, Goat, Cattle	Cattle	2020	Cattle & Sheep	4156440.92
15862	2020-12-01	Dec-20	Zoetis	Vaccines & Antisera	Vaccines Sheep, Goat, Cattle	Pig	2020	Swine	508284.39
15863	2020-12-01	Dec-20	Zoetis	Vaccines & Antisera	Vaccines Sheep, Goat, Cattle	Sheep	2020	Cattle & Sheep	3455646.53

15864 rows × 9 columns

Wholesaler to Retail

In [10]:
```
wholesaler_to_retail.head()
```

Out[10]:

	Year	Month	Distributors Name	CUSTOMER ID	Postcode	WHS_SKU_ID	Suppliercode	Product	Quantity	Price
0	2020.0	June	Eastern Distributors	AU924404	3079	WSLDB3	CA472075WAL	SENTINEL LARGE DOG 22 - 45KG BLUE 3 CHEWS	6.0	56.401100
1	2020.0	June	Eastern Distributors	AU924404	3079	WSLDB6	CA472083WALZ1	SENTINEL LARGE DOG 22 - 45KG BLUE 6 CHEWS	2.0	99.507053
2	2020.0	June	Eastern Distributors	AU924404	3079	WSMDY6	CA472583WALZ1	SENTINEL MEDIUM DOG 11 - 22KG YELLOW 6 CHEWS	3.0	91.334903
3	2020.0	June	Eastern Distributors	AU924404	3079	WSSDG6	CA473083WAL	SENTINEL SMALL DOG 4 - 11KG GREEN 6 CHEWS	6.0	84.136057
4	2020.0	June	Eastern Distributors	AU924404	3079	WACS6	04506700	ADVANTAGE CAT 0-4KG SMALL 6'S ORANGE	1.0	50.102048

In [11]:
```
wholesaler_to_retail.columns
```

Out[11]:
```
Index(['Year', 'Month', 'Distributors Name', 'CUSTOMER ID', 'Postcode',
       'WHS_SKU_ID', 'Suppliercode', 'Product', 'Quantity', 'Price'],
      dtype='object')
```

In [12]:
```
wholesaler_to_retail_drop = wholesaler_to_retail.drop(['WHS_SKU_ID', 'Suppliercode', 'Product'],axis =1)
wholesaler_to_retail_drop = wholesaler_to_retail_drop.dropna()
wholesaler_to_retail_drop
```

Out[12]:

	Year	Month	Distributors Name	CUSTOMER ID	Postcode	Quantity	Price
0	2020.0	June	Eastern Distributors	AU924404	3079	6.0	56.401100
1	2020.0	June	Eastern Distributors	AU924404	3079	2.0	99.507053
2	2020.0	June	Eastern Distributors	AU924404	3079	3.0	91.334903
3	2020.0	June	Eastern Distributors	AU924404	3079	6.0	84.136057
4	2020.0	June	Eastern Distributors	AU924404	3079	1.0	50.102048
...	...	...	...	...	...	...	...
144505	2020.0	May	Western Distributors	AU923112	3840	12.0	10.600000
144511	2020.0	May	Western Distributors	In Google	2756	6.0	69.360000
144512	2020.0	May	Western Distributors	In Google	2756	6.0	69.360000
144513	2020.0	May	Western Distributors	In Google	2756	5.0	69.360000
144514	2020.0	May	Western Distributors	AU919284	3216	1.0	25.150000

51178 rows × 7 columns

6.2. Data Visualisation/Exploration

According to For Dummies, a correlation coefficient can be interpreted this way:

- A positive (uphill) relationship: the variables move together.
- Exactly 1 is a perfect relationship Between 0.70 and 1 is a strong relationship Between 0.50 and 0.70 is a moderate relationship Between 0.30 and 0.50 is a weak relationship A negative (downhill) relationship: the variables move opposite to each other.
- Exactly -1 is a perfect relationship Between -0.70 and -1 is a strong relationship Between -0.50 and -0.70 is a moderate relationship Between -0.30 and -0.50 is a weak relationship A correlation score between -0.30 and +0.30 shows no relationship.

The following graphs are using the values of “Units ADJ” on the y-axis.

2.1. Business Units

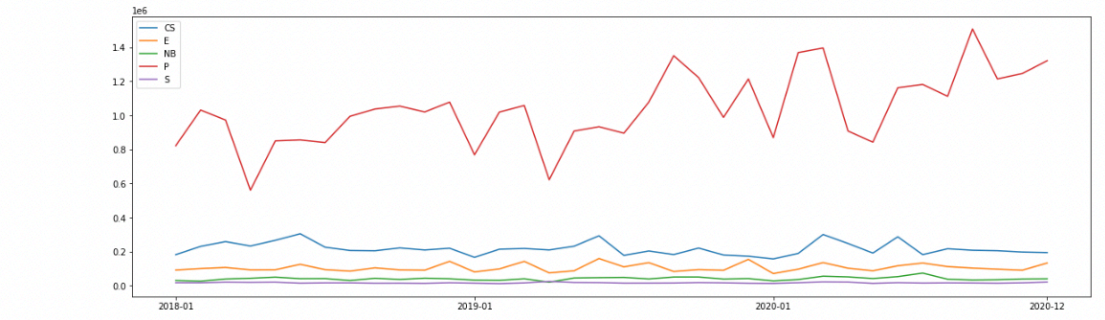
In [13]:
```
market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Cattle & Sheep'].plot(x='Period',
y='Units ADJ',figsize=(20,5),c = "red")
market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Equine'].plot(x='Period',y='Units
ADJ',figsize=(20,5),c = "blue")
market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'No Bus'].plot(x='Period',y='Units
ADJ',figsize=(20,5),c = "green")
market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Pets'].plot(x='Period',y='Units A
DJ',figsize=(20,5),c = "black")
market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Swine'].plot(x='Period',y='Units
ADJ',figsize=(20,5),c = "orange")
```

```
Out[13]: <AxesSubplot:xlabel='Period'>
```

In [14]:
```
x1 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Cattle & Sheep']['Period']
y1 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Cattle & Sheep']['Units ADJ'
]
x2 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Equine']['Period']
y2 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Equine']['Units ADJ']
x3 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'No Bus']['Period']
y3 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'No Bus']['Units ADJ']
x4 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Pets']['Period']
y4 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Pets']['Units ADJ']
x5 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Swine']['Period']
y5 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Swine']['Units ADJ']
x_tick = ['2018-01', '2019-01', '2020-01', '2020-12']
plt.figure(figsize=(20, 6))
plt.plot(x1, y1,label ="CS")
plt.plot(x2, y2,label ="E")
plt.plot(x3, y3,label ="NB")
plt.plot(x4, y4,label ="P")
plt.plot(x5, y5,label ="S")
plt.xticks(x_tick)

plt.legend()
```

Out[14]: <matplotlib.legend.Legend at 0x1b0059ffeb0>

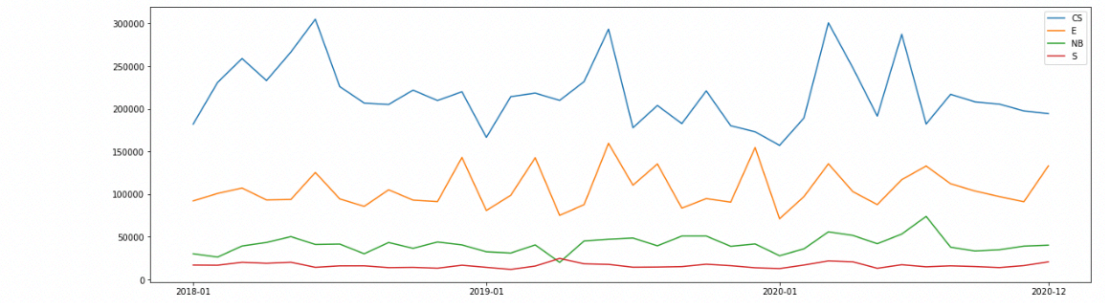


From the combined sales graph, we can observe that Pets are overwhelmingly larger in terms of sales of products. However, we need to also understand that this plot does not have the distinction between products and their associated usage. But let's observe the trends in the sales for the other units except pets.

In [15]:
```
x1 = .loc[market_drop_unit['BI Business Unit'] == 'Cattle & Sheep']['Period']
y1 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Cattle & Sheep']['Units ADJ'
]
x2 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Equine']['Period']
y2 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Equine']['Units ADJ']
x3 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'No Bus']['Period']
y3 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'No Bus']['Units ADJ']
x5 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Swine']['Period']
y5 = market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Swine']['Units ADJ']
x_tick = ['2018-01', '2019-01', '2020-01', '2020-12']
plt.figure(figsize=(20, 6))
plt.plot(x1, y1,label ="CS")
plt.plot(x2, y2,label ="E")
plt.plot(x3, y3,label ="NB")
plt.plot(x5, y5,label ="S")
plt.xticks(x_tick)

plt.legend()
```

Out[15]: <matplotlib.legend.Legend at 0x1b006ebdf70>

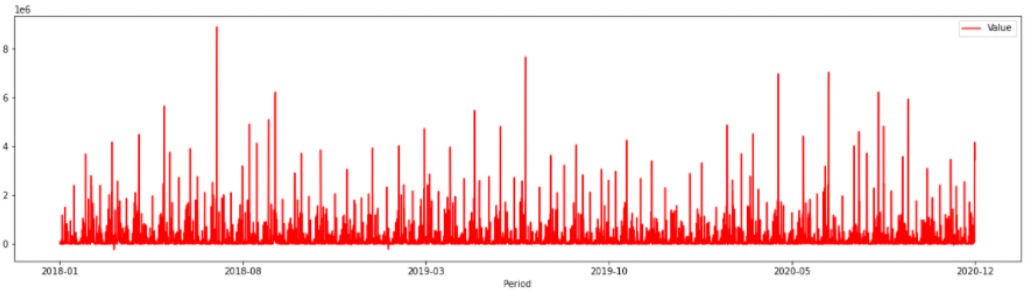


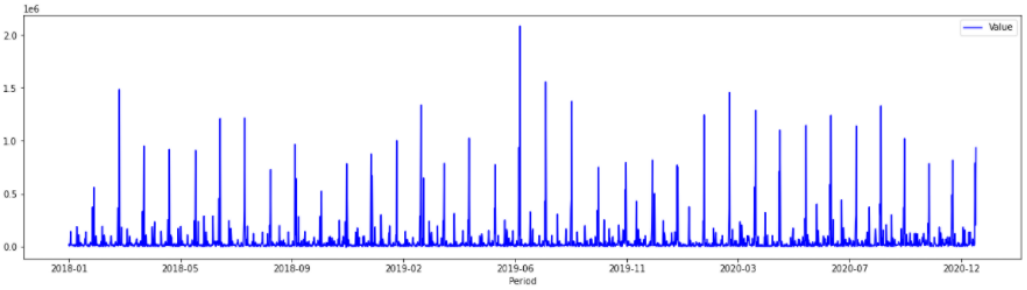
After more work on the MVP, we have decided to also test out the graphs utilising values on the y-axis.

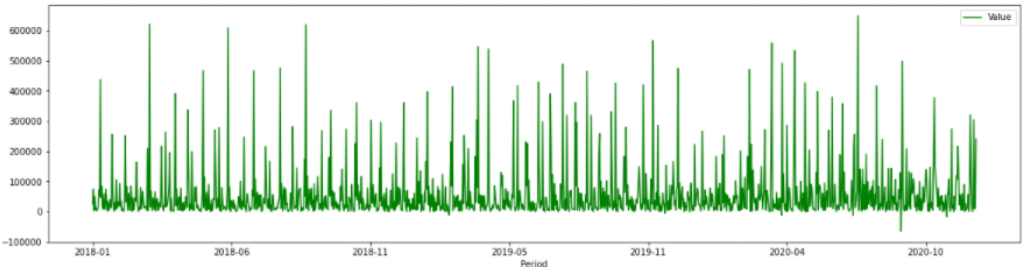
2.1. Business Units

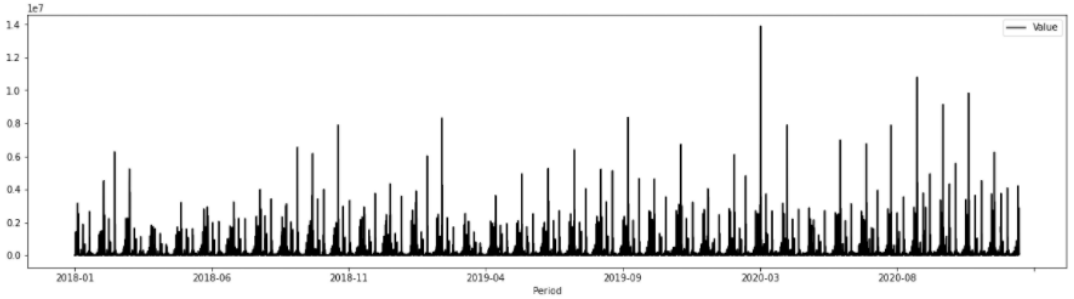
```
In [13]: market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Cattle & Sheep'].plot(x='Period',y='Value',figsize=(20,5),c = "red'
         market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Equine'].plot(x='Period',y='Value',figsize=(20,5),c = "blue")
         market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'No Bus'].plot(x='Period',y='Value',figsize=(20,5),c = "green")
         market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Pets'].plot(x='Period',y='Value',figsize=(20,5),c = "black")
         market_drop_unit.loc[market_drop_unit['BI Business Unit'] == 'Swine'].plot(x='Period',y='Value',figsize=(20,5),c = "orange")
```

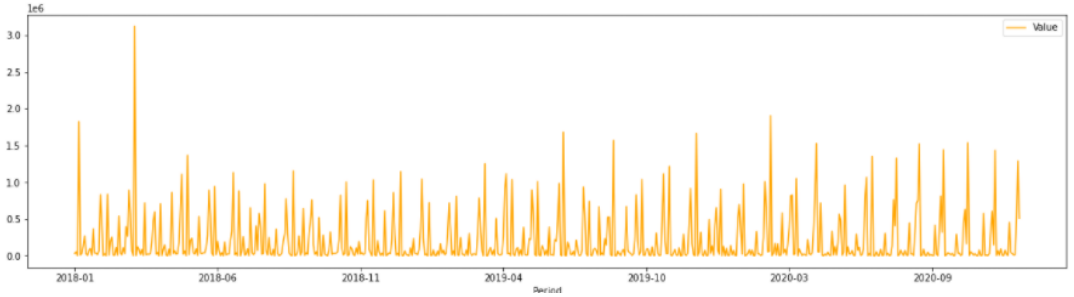
Out[13]: <AxesSubplot:xlabel='Period'>



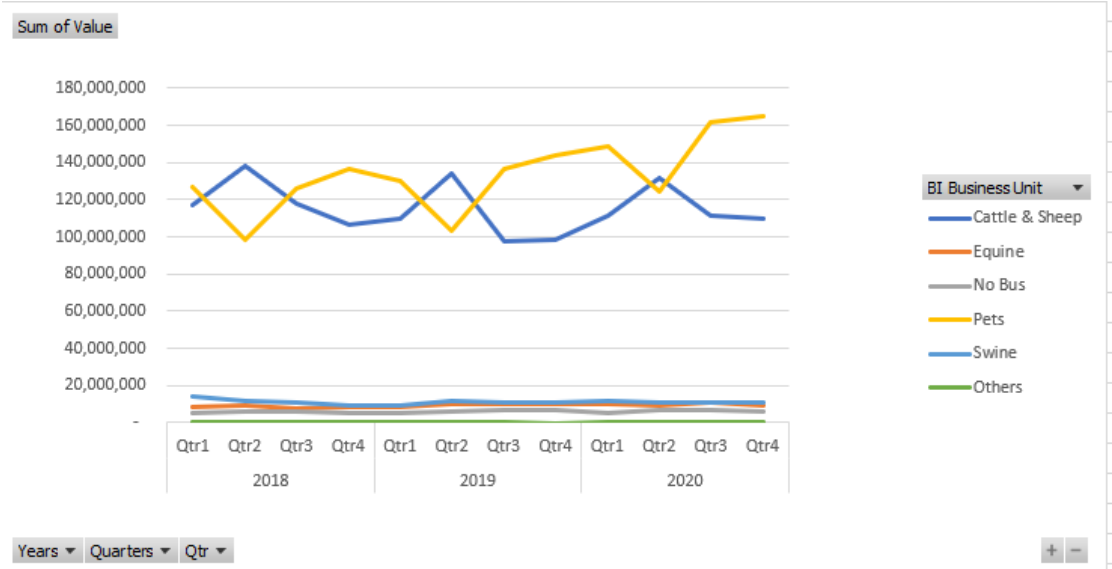








Based on the graphs shown above, we can't infer much about the various values per business unit.



Looking at the graph per quarter, we can actually see some quarterly trends with Cattle & Sheep being very close to the value of pets.

6.3. Modelling

Based on the market data, this is the predictions on future sales. We are using various types of linear regression in order to see if we are able to predict future revenue based on the criteria we set. We have decided to use the variable of "value" as this is a continuous variable that will allow us to obtain the best business insights as to the performance of BI. The initial selection of "Units ADJ" showed a difference in the volume of products sold by BI but based on their feedback, Pets and Cattle & Sheep should be on the same level.

This feedback has been investigated and we have determined value to be a better predicted variable as this is what BI is more interested in and can also allow us to verify that the predicted variable is in-line with BI's expectations.

```
In [28]: market_drop.columns

Out[28]: Index(['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3',
                'Species', 'Value', 'YEAR', 'BI Business Unit'],
               dtype='object')
```

```
In [29]: Period = market_drop['Period']
         Period_Unique = Period.unique()

         Qtr = market_drop['Qtr']
         Qtr_Unique = Qtr.unique()

         Manufacturer = market_drop['Manufacturer']
         Manufacturer_Unique = Manufacturer.unique()

         SubCategory2 = market_drop['SubCategory2']
         SubCategory2_Unique = SubCategory2.unique()

         SubCategory3 = market_drop['SubCategory3']
         SubCategory3_Unique = SubCategory3.unique()

         Species = market_drop['Species']
         Species_Unique = Species.unique()

         BI_Business_Unit = market_drop['BI Business Unit']
         BI_Business_Unit_Unique = BI_Business_Unit.unique()
```

```
In [30]: le = preprocessing.LabelEncoder()

         le.fit(Period_Unique)
         arr1 = le.transform(Period)

         le.fit(Qtr_Unique)
         arr2 = le.transform(Qtr)

         le.fit(Manufacturer_Unique)
         arr3 = le.transform(Manufacturer)

         le.fit(SubCategory2_Unique)
         arr4 = le.transform(SubCategory2)

         le.fit(SubCategory3_Unique)
         arr5 = le.transform(SubCategory3)

         le.fit(Species_Unique)
         arr6 = le.transform(Species)

         le.fit(BI_Business_Unit_Unique)
         arr7 = le.transform(BI_Business_Unit)
```

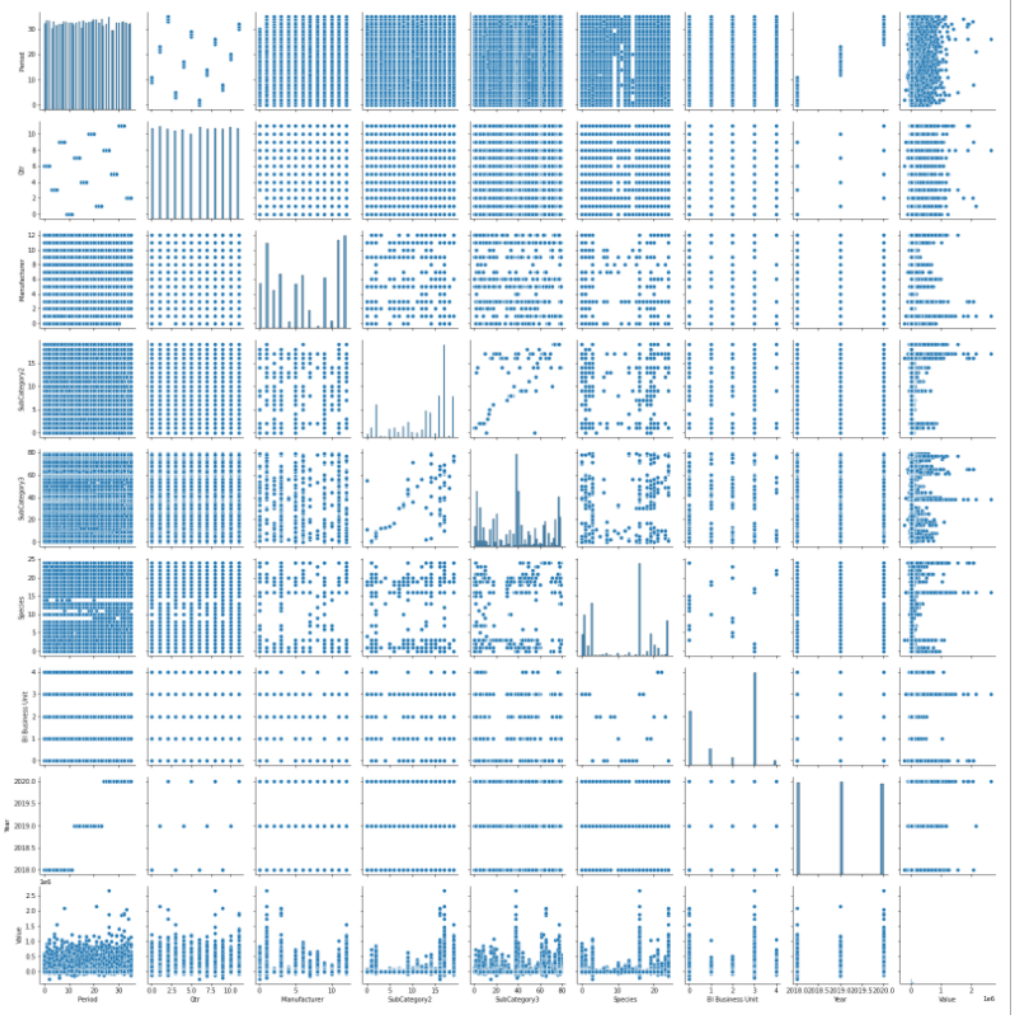
```
In [31]: market_drop_model = pd.DataFrame({'Period':arr1,'Qtr':arr2, 'Manufacturer':arr3, 'SubCategory2':arr4,'SubCategory3':arr5,'Specie
```

```
In [32]: market_drop_model.columns

Out[32]: Index(['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3',
                'Species', 'BI Business Unit', 'Year', 'Value'],
               dtype='object')
```

After encoding the string types to be readable integers for the model to run, we want to observe the relationships between the datapoints in the overall dataset.

```
In [33]: sns.pairplot(market_drop_model)
Out[33]: <seaborn.axisgrid.PairGrid at 0x25fff61f7c0>
```



Since most of the data is categorical, it would make sense that the data points do not have any clear linearity in the relationship to value in the bottom row.

Now we want to create our baseline model using the linear regression model from the sklearn import.

```
In [48]: model = LinearRegression()
         X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3',
                'Species', 'BI Business Unit', 'Year']]
         y = market_drop_model[['Value']]
         X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
         y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


         model.fit(X_train,y_train)
         # Obtain MSE and r2 for testing data
         y_test_pred = model.predict(X_test)
         print("Checking testing values")
         print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
         print("MAE:" , mean_absolute_error(y_test, y_test_pred))
         print("R2:" , r2_score(y_test, y_test_pred))

         Checking testing values
         RMSE:  52797.31913489867
         MAE: 20413.367034649193
         R2: 0.0142817936330214
```

From this, we can see that 1.4% of the variance in the model can be explained using all the variables listed in the “market_drop_model” variables. We would now want to test out the individual variables to determine if there is any

significant change in the R^2 value. More detailed explanation on the testing will be included in the testing and evaluation document.

Now we are individually testing the linear regression with the individual variables to see which variables are the best predictors.

In [49]:
```
X = market_drop_model[['Period']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  53174.05650488723
MAE: 20852.524178342755
R2: 0.00016434196573988924
```

In [50]:
```
X = market_drop_model[['Qtr']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  53178.79257503653
MAE: 20860.77070222976
R2: -1.3771309994137226e-05
```

In [51]:
```
X = market_drop_model[['Manufacturer']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  53179.98425601378
MAE: 20862.00282761879
R2: -5.8590332122632205e-05
```

In [52]:
```
X = market_drop_model[['SubCategory2']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  52822.199568428616
MAE: 20442.45687686711
R2: 0.013352546642070418
```

In [53]:
```
X = market_drop_model[['SubCategory3']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  52971.65320424556
MAE: 20579.917820666527
R2: 0.007761463453755835
```

In [54]:
```
X = market_drop_model[['Species']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  53124.07403309945
MAE: 20809.483114838727
R2: 0.002043106667499628
```

In [55]:
```
X = market_drop_model[['BI Business Unit']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  53150.28907518666
MAE: 20818.408658782017
R2: 0.0010579436367595951
```

In [56]:
```
X = market_drop_model[['Year']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)
model = LinearRegression()

model.fit(X_train,y_train)
# Obtain MSE and r2 for testing data
y_test_pred = model.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  53174.335096922405
MAE: 20853.08934830393
R2: 0.00015386516563076214
```

Based on the R2 value, we can see a non-material change even with all the variables tested independently and observing the RMSE and MAE shows that the overall model provided the best results.

However, the linear regression model still does not have a satisfactory result and we should test other models.

KNN

The second model we are testing now is the K-Nearest Neighbours model. In order to maintain the integrity of the testing, we will set up the code to be the same as the linear regression with the same train test split but we will change the model into the KNN

```
X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3',
       'Species', 'BI Business Unit', 'Year']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)

from sklearn.neighbors import KNeighborsClassifier
K_Neighbour = KNeighborsClassifier(n_neighbors=1)

K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
#print('The accuracy of the training with all the variables KNearestNeighbour model is : ', accuracy_score(y_train, y_trainpred))
#print('The accuracy of the testing with all the variables KNearestNeighbour model is : ', accuracy_score(y_test, y_testpred))

print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  10116.361706840718
MAE: 22539.089644260832
R2: 0.9638109040633203
```

Looking at the R^2, the model is performing much better than the initial baseline model and we can use the KNN model as good model. However, we still would like to test the various variables to check for any significant changes.

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['Period']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  10251.87296914955
MAE: 33219.22047540716
R2: 0.9628348860864643
```

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['Qtr']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  10250.513848250856
MAE: 23190.516355462914
R2: 0.9628447396098
```

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['Manufacturer']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  9217.290489529392
MAE: 16878.68645227721
R2: 0.9699575335057995
```

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['SubCategory2']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  9767.64196844382
MAE: 20735.807715006842
R2: 0.9662628415916469
```

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['SubCategory3']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  8727.098683060329
MAE: 18376.209161570627
R2: 0.9730679873636048
```

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['Species']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  16349.088709223139
MAE: 21045.795183143513
R2: 0.9054816489386929
```

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['BI Business Unit']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  14210.114190005368
MAE: 20636.069214058793
R2: 0.9285957319617795
```

```
#X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3','Species', 'BI Business Unit', 'Year']]
X = market_drop_model[['Year']]
y = market_drop_model[['Value']]
X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)


K_Neighbour = KNeighborsClassifier(n_neighbors=1)
K_Neighbour.fit(X_train,y_train)
y_train_pred = K_Neighbour.predict(X_train)
y_test_pred = K_Neighbour.predict(X_test)
print("Checking testing values")
print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
print("MAE:" , mean_absolute_error(y_test, y_test_pred))
print("R2:" , r2_score(y_test, y_test_pred))
```

```
Checking testing values
RMSE:  nan
MAE: 71161.1908283838
R2: 1.1607976475026427
```

We can see that the variable for year should be reconsidered based on the RMSE and R^2 value. Overall, we can also take the model that considers all the variables but if we were to remove a variable it would be removing year.

ARIMA

We have also tried implementing the Autoregressive Integrated Moving Average model and determined that the results are unfeasible at the current stage.

In [114]:
```
market_drop['Period'] = pd.to_datetime(market_drop['Period'])
market_drop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 199091 entries, 0 to 199398
Data columns (total 9 columns):
 #   Column            Non-Null Count   Dtype         
---  ------            --------------   -----         
 0   Period            199091 non-null  datetime64[ns]
 1   Qtr               199091 non-null  object        
 2   Manufacturer      199091 non-null  object        
 3   SubCategory2      199091 non-null  object        
 4   SubCategory3      199091 non-null  object        
 5   Species           199091 non-null  object        
 6   Value             199091 non-null  float64       
 7   YEAR              199091 non-null  int64         
 8   BI Business Unit  199091 non-null  object        
dtypes: datetime64[ns](1), float64(1), int64(1), object(6)
memory usage: 15.2+ MB
```

In [121]:
```
market_drop_pets = market_drop[market_drop['BI Business Unit'] == 'Pets']
market_time_series_model = pd.DataFrame({'Period':market_drop_pets['Period'],'Value':market_drop_pets['Value'].astype(int)})
market_time_series_model=market_time_series_model.set_index('Period')
```

In [122]:
```
market_time_series_model.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 105359 entries, 2020-10-01 to 2018-03-01
Data columns (total 1 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Value   105359 non-null  int32
dtypes: int32(1)
memory usage: 1.2 MB
```

```
import pmdarima as pm
def arimamodel(timeseries):
    automodel = pm.auto_arima(timeseries,
                              start_p=1,
                              start_q=1,
                              test="adf",
                              seasonal=False,
                              trace=True)
    return automodel
```

```
def plotarima(n_periods, timeseries, automodel):
    # Forecast
    fc, confint = automodel.predict(n_periods=n_periods,
                                    return_conf_int=True)
    # Weekly index
    fc_ind = pd.date_range(timeseries.index[timeseries.shape[0]-1],
                           periods=n_periods, freq="W")
    # Forecast series
    fc_series = pd.Series(fc, index=fc_ind)
    # Upper and lower confidence bounds
    lower_series = pd.Series(confint[:, 0], index=fc_ind)
    upper_series = pd.Series(confint[:, 1], index=fc_ind)
    # Create plot
    plt.figure(figsize=(10, 6))
    plt.plot(timeseries)
    plt.plot(fc_series, color="red")
    plt.xlabel("date")
    plt.ylabel(timeseries.name)
    plt.fill_between(lower_series.index,
                     lower_series,
                     upper_series,
                     color="k",
                     alpha=0.25)
    plt.legend(("past", "forecast", "95% confidence interval"),
               loc="upper left")
    plt.show()
```

The above sections of code is to prepare the data and the model to build the ARIMA analysis.

```
In [125]: automodel = arimamodel(market_time_series_model["Value"])
          plotarima(36, market_time_series_model["Value"], automodel)

Performing stepwise search to minimize aic
 ARIMA(1,0,1)(0,0,0)[0]             : AIC=2560894.285, Time=14.44 sec
 ARIMA(0,0,0)(0,0,0)[0]             : AIC=2591678.176, Time=1.29 sec
 ARIMA(1,0,0)(0,0,0)[0]             : AIC=2581941.915, Time=1.74 sec
 ARIMA(0,0,1)(0,0,0)[0]             : AIC=2584767.592, Time=4.16 sec
 ARIMA(2,0,1)(0,0,0)[0]             : AIC=2560847.167, Time=32.73 sec
 ARIMA(2,0,0)(0,0,0)[0]             : AIC=2577531.679, Time=2.42 sec
 ARIMA(3,0,1)(0,0,0)[0]             : AIC=2560834.453, Time=41.58 sec
 ARIMA(3,0,0)(0,0,0)[0]             : AIC=2574217.538, Time=3.04 sec
 ARIMA(4,0,1)(0,0,0)[0]             : AIC=2560595.197, Time=52.62 sec
 ARIMA(4,0,0)(0,0,0)[0]             : AIC=2572195.769, Time=3.49 sec
 ARIMA(5,0,1)(0,0,0)[0]             : AIC=2560377.547, Time=59.14 sec
 ARIMA(5,0,0)(0,0,0)[0]             : AIC=2570344.799, Time=4.33 sec
 ARIMA(5,0,2)(0,0,0)[0]             : AIC=2560189.788, Time=65.86 sec
 ARIMA(4,0,2)(0,0,0)[0]             : AIC=2560198.143, Time=52.30 sec
 ARIMA(5,0,3)(0,0,0)[0]             : AIC=2559937.166, Time=149.26 sec
 ARIMA(4,0,3)(0,0,0)[0]             : AIC=2559939.325, Time=129.50 sec
```

Based on the Akaike Information Criterion, since the values are so large for the various ARIMA parameters, we can consider this model as unsuitable for the data input.

DTR

Another potential model we tested is the Decision Tree algorithm.

DTR

```
In [127]: X = market_drop_model[['Period', 'Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3',
                'Species', 'BI Business Unit', 'Year']]
          y = market_drop_model[['Value']]
          X_train, X_test = train_test_split(X, test_size=0.4, random_state=7)
          y_train, y_test = train_test_split(y, test_size=0.4, random_state=7)

          regressor = DecisionTreeRegressor()
          regressor.fit(X_train, y_train)

          y_train_pred = regressor.predict(X_train)
          y_test_pred = regressor.predict(X_test)

          print("Checking testing values")
          print("RMSE: ", mean_squared_error(y_test,y_test_pred,squared=False))
          print("MAE:" , mean_absolute_error(y_test, y_test_pred))
          print("R2:" , r2_score(y_test, y_test_pred))

          Checking testing values
          RMSE:  50798.51933128566
          MAE: 18281.694122644476
          R2: 0.08750363458171517
```

However, since we have the KNN being much more accurate, we can ignore this model.

Based on our testing, we can focus on the KNN in order to optimise the model and provide a proper analysis of the market data.

6.4. Prototype/MVP Feedback from BI

BI Meeting with Group 1. 28th April 2021, 2pm

Feedback Received

- Check for outliers Heteroscdasiticy Assumption Testing 0's in units Shape of data Transformations
- Check C&S Vs Pets cause C&S is meant to be equal to pets. (Data Visualisation)
- Forecasting future market changes. Look at changes that happen and tell them how we think we can sell in the future. Long term forecast.
- Forecast down to category levels.
- What's recovery look like for Cattle & Sheep.
- Previous 3 years of data with drought. How will that affect the data.

- 2020 remarkable year for pets. What is going to look like after.
- Vet visits went up. Cause people noticed more issues with pets.
- Up to 3 million new pets in the business

The convenor Deborah meeting with Group 1. 6th May, 2021, 5pm

Reason for this meeting: BI couldn't help with the coding issue when the data isn't loading

Feedback Received

- Seek another tutor Sonit for further assistant

Sonit meeting with Group 1. 13th May, 2021, 11am

Reason for this meeting: Deborah couldn't help with the MVP issue

Feedback Received

- Seek another tutor Sonit for further assistant

BI Meeting with Group 1. 19th May 2021, 3pm

Feedback Received

- Reduce the variable and categories for better result from the MVP

7. Analysis, Design + Testing Documentation

Communicate different aspects of the project implementation

- Data
    - The data we get comes in a variety of forms from Boehringer Ingelheim (BI). This is shared through dropbox and access is given to us by BI using our student emails.
    - The datasets that come through have varying attributes and require some cleaning and transformation in order for us to manipulate them into useable model data.
        - This is done initially from excel and done again in Jupyter.
        - Excel => Initial Merge/Integration of data from multiple months
        - Jupyter => Dropping columns and dropping rows with NA, sectioning out the data into different data frames, aggregation of the values.

- Goals
    - Our main goal is to utilise data science theory in order to extract valuable business insights from the sales data given to us. This can be split into smaller goals/hypothesis to make the structure of our project more manageable.
    - Stretch goal is to be able to predict the performance of the industry in the coming periods.

- Options
    - In terms of software options, Jupyter was the most suitable for us as we can split up the code into different sections and have the outputs being split based on the code sections. The alternative in order to still use Python packages could be to use Spyder but the outputs interface is less user friendly than Jupyter and they come from the same anaconda installer.
    - In terms of Data options, BI sent over multiple large datasets with various time ranges. We had market data, wholesale data, retailer sales data, dashboards and distribution data. We are currently looking at market data and the distribution data as we are more familiar with the business aspects of sales and can value-add to the analysis by leveraging on our expertise.

- Choice of Models
    - We are limited in terms of model choice based on the data given to us. I.e market data does not have a price associated with it and only the volume of doses/units sold. The other attributes are also in String type which require us to convert them into readable data by the fitting code.
    - There are 2 main choices in consideration at the moment for market data to develop our baseline.

Linear Regression and Logistic regression as we want to know what variables are the key to affecting the number of units sold and we can predict for the next years, what the expected values should be for the future predictions.
    - Other models we can consider are the K-Nearest Neighbours model, Autoregressive Integrated Moving Average and Decision Tree model.

- Starting Configurations
    - No specific starting configuration for the models. For our KNN model, we have the initial K set to 1. This value could change depending on future testing and feedback from BI.
    - At the moment the starting configurations would be how we set up our dataframes to be fed into the model to be trained. This includes the attribute data type, which attributes are being fed, how we split the X and Ys for the model.
    - For the ARIMA model, we have set the number of time periods to be the unique time periods in our data.

- Resource available
    - All available models and syntax can be found online as they are open-source development packages. More information can be found in the testing document.

Develop Baseline

- How did we develop this
    - We just used all the historical data given to us by BI and fed it into a linear regression model.
    - We then tested the train test splits using RMSE, MAE and R^2 metrics.
    - This model will be our baseline for further development and we can compare the predicted values based on the test data.

- How do we beat this later
    - In future models, we will use other types of models such as logistical regression and or more complex machine learning algorithms to train the model.
    - We can then compare the predicted outputs of the 2 different models with the same test data to give us a comparison of which model returns a more “accurate” result.
    - Additional commentary will be given in the form of qualitative analysis in order to determine if it is truly a better model or is it a coincidence that the model performed better.

Data Manipulation

- How to turn raw data into input for modelling
    - Make sure all input files are in the CSV format
    - Store them into one folder on the desktop
    - Using Pandas import package, load the CSVs as dataframe

    - After the dataframe is loaded, conduct basic exploration.
        - Df.head()
        - Df.describe()
        - Df.columns
    - Drop and append dataframes as required
        - Drop NA rows
        - Drop irrelevant columns
        - Append rows/combine datasets on a column
        - Aggregate the values in the dataset

- Data sharing
    - Currently we have our GitHub set up for the project itself but we do not load data into GitHub due to privacy concerns.

Results of Modelling

- Outline different models used
    - Currently we have tested out Linear Regression, KNN, ARIMA and DTR models.
    - Linear Regression with the data inputs to test given an input, output mapping. Is there a linear relationship between the units sold and the variables.
    - Various different models used to find the best model for us to run a prediction on the value.

- Results observed
    - Linear regression has issues with the MSE but the R^2 values look accurate. Further testing of assumptions required to determine if we should make any adjustments to the model since the MSE is not normal.
    - KNN has the best R^2 value and we can move forward with developing the model based on the KNN parameters by testing different combinations of parameters and changing the K value.
    - ARIMA results were poor based on the AIC values and DTR showed a poorer R^2 value than the KNN but better than the baseline linear regression.

- Interpret results and discuss implications for business questions
    - For business question, refer to main goal above.
        - Assuming that the R^2 values are correct and the RMSE is due to the LabelEncoder, we can tell that there is a relationship between the Business Unit, the time period and the value.
        - We can then determine for a specific month, roughly how much units will be sold for a particular business unit.

Feature engineering

- From data, what trends/ranges to look for?

  - We are looking at sales and distribution data across multiple time periods for all customers of the business.
  - Key thing to look for is any change in the sales or distribution data. This is because we then need to explain if there is something wrong with the data there or is it an area that the business should focus on.
  - Additionally, look for comparisons within the data to each other and compare with the understanding of the business to see if it passes a sense check since the business will know what is their biggest earner.
  - We should also be keeping an eye on the year 2020 since that is when COVID-19 emerged. The pandemic changed affected the world by forcing everyone into a lockdown in order to prevent the spread of the pandemic and thus there might have been changes to the business that could be allocated to that. There could also have been a potential improvement in one part of the business due to this.

- Describe what data characteristics are being looked for
  - We are looking for trends in the data so the first thing that we should be doing is plotting the data in order to view the trends of the data we are looking at and also the relationships of the variables within our data.
  - This could include a scatterplot matrix to determine all the variables' relation to each other to see if there are any trends. For our data, when we did the scatterplot matrix, we realised that this data is not suitable for a linear regression and we should be doing a logistic regression for our baseline model.

- How our data pipeline works to produce these features
  - The data flow starts with us collecting the data from BI and then doing a high-level cleaning of the snapshot of the data as at extraction time. This includes type changes for data from string to date, the data type for values string to integer/float values.
  - Once we are comfortable with the changes in excel, we will then save the file as a CSV and store it in our data folder.
  - Load the CSV files into Jupyter and then create them as dataframes using the Pandas package.
  - Once the dataframes are loaded, we will do analysis on the attribute types and the column names. Subsequently, we will drop the irrelevant columns that are not relevant to the actual study being done. If there is a need for the other columns later in the project, we can change the drop code to not drop as many attributes and change the structure of the subsequent code.
  - The finalised dataframe can now be analysed and passed through the various imports to produce graphs and statistical analysis.
  - If there is a need to make any changes to the data itself such as doing a log transform or a aggregation of the values, we can make the changes on the already cleaned dataframe and easily assign that to a different variable for future use.

- Give each feature/characteristic a name and then attribute some form of data ranges/stats definition
    - Specifically to our data, we have the “Value” as our main response variable for market data since that is what we are looking for as a measure of performance.
    - We also have the variable “BI Business Unit” as part of the predictor variable. This ranges of categorical data determines which part of BI’s business the sales were coming from and can be split further into the specific product type.
    - For time, we have the “Period” that allows us to structure the data and we can utilise to conduct time-series analysis on.
    - Based on the performances of the models using the individual variables on their own, we can determine which variables are most significant to the model. The testing results can be seen from the MVP section above and this will be updated as the MVP evolves.
    - In our final model presented, we will include the best model parameter and variable inputs.

Solution Architecture

- Explain data in and out of each stage.
    - From the first initial project brief, we first looked at the data that we have and did some high-level views on excel on what are the limits and constraints with the data. For example, what fields were available to us, what are the limits of the data, what type key metrics were in, what kinds of business insights BI would want to receive and which metric is the most likely for them to be able to achieve their business goals. We then created our hypothesis on how to achieve better business insights and started the implementation of the project.
    - We can generally split our project into 4 stages.
    - First, we have the initial loading of the data. This is a combination of both the excel cleaning and the Jupyter verification of the data. The inputs for this stage are the raw datasets provided by BI through the dropbox system. We are extracting the data based on a snapshot at a certain date and will specify which date is being run on our GitHub.
    - Second, we have the transformations for the data. This will include any of the statistical transforms that we need and the relevant transformations into the modelling data frames. The final output here should be data frames with the correct types of data and the correct attributes for us to pass into the functions.
    - Thirdly, we have our data visualisation and exploration phase where we will graph out the data in order to located or identify any trends that are expected or unexpected. For our current project, the graphs of the individual “Value” per business unit shows us how much volume has been transacted month to month. In addition to this, additional data analysis has been done on the “Units ADJ” based on the feedback from BI and we have made the decision to shift the focus from “Units ADJ” to “Value”

  - In tandem to the previous step, we have also done a high level view of the data provided and after creation of scatter plots to look for relations within the data. The selection of the variables used for modelling is based on business requirements and from the scatter plot. i.e date and period are effectively the same so there will definitely be a covariance between the two metrics. Hence we select one of them to be part of the dataset.
  - Lastly, we have our modelling phase where we will require to test assumptions made for the models and subsequently change the data itself to suit the model inputs. We used a label encoder to change the string types of the data into a actual value. This should not have been done since we were planning to use a linear regression and instead we should have used a logistic regression since we have categorical data.
  - Once we have our output, we will conduct some qualitative analysis and testing on the results. Then this result will be stored in the jupyter notebook for reference in the future.

- Include resource available/constraints to each section
  - For each of the stages, we have access to the inputted data and the subsequent transformation/output of it.
  - The data given from BI have multiple formats and sources so time is taken to ensure that the data loaded is accurate and fit for purpose.
  - One computer system needs to run through all the code and the inputs should be in the same folder as the set up for the project. Then, the outputs will be able to be saved even without running the project again.
  - We are using the official documentation/user manuals for the various imports we have used.

Algorithms/Model Methods

- What are the models used/initial conditions/configuration settings?
  - We are thinking of using a regression model and a time-series model in order to achieve the sub goal of having insights into BI's sales data.
  - Currently, there is a linear regression model as a baseline model to compare the other models such as the KNN, ARIMA and DTR to.
  - The parameters used for the models have been set up in the MVP in Jupyter and the specific details are in the user manual. On a high level, the set up parameters are K =1 for the KNN model since we are experimenting with various models at the current development stage and ARIMA's x = 36 based on the number of distinct time periods in the data.
  - The data is set to take in encoded data that have been transformed from String data. The encoding labels are in the MVP as arrays and can be re-converted into the String types on demand.

- Any settings needed?
    - Due to the nature of the dataset, we need to change some of the dataset in order to suit the regression models better.
    - Label encoding is one of the ways we can do that and we have implemented it.
    - Transformation of the dataset is not applied as the variables are categorical. The predicted value of “Value” is the only continuous value that we have and hence we are using the regression models.
    - The time series model is included as there is the availability of time-data based on the inputs given from BI.

Detailed Data Descriptions

- Details on data being used
    - Eastern
        - Data associated with the eastern distributors
        - Contains the following attributes: ['DistributorsName','SellingPostcode','WholesalerSKU','CustomerNumber','DebtorName','ManufacturerSKU','BARCODE','UnitsSold','UnitPrice','ManufacturerName','InvoiceDate']
    - Western
        - Data associated with the western distributors
        - Contains the following attributes: ['Stockcode','Suppliercode','Barcode','Quantity','Price','AccNo','Postcode','Date']
    - Combined
        - Combined dataset for both eastern and western distributors
        - Contains the following attributes: ['Vendor','Shipdate','Orderkey','Consigneekey','C_company','C_zip','Externorderkey','Sku','Originalqty','Shippedqty']
    - Market
        - Market data relating to BI’s various business units. [Confidential Data]
        - Contains the following attributes: ['Period','MAT','YTD','Qtr','CalYr','Manufacturer','SubCategory2','SubCategory3','Brand','Product','ProductCode','ProductionCompanion','Species','Region','Units','Value','Doses','YEAR','ManufacturerADJ','BrandADJ','ProductADJ','SubCategory3.1','SpeciesADJ','UnitsADJ','DosesADJ','Doses/Unit','MonthlyDoses','BIBusinessUnit','BIMarketOnlyFLG']
        - Largest dataset provided by BI and is currently the main dataset we are working on.
    - Wholesaler_to_retail
        - Latest addition to data inputs of the total sales from the wholesaler to retail shops.
        - Contains the following attributes: ['Year','Month','DistributorsName','CUSTOMERID','Postcode','WHS_SKU_ID','Suppliercode','Product','Quantity','Price']

- Data being generated
    - Market_drop
        - Dropped the unnecessary columns for our usage. Only contains ['Period', 'Qtr', 'Manufacturer', 'SubCategory2', SubCategory3', 'Species', 'Value', 'YEAR', 'BI Business Unit']
        - At the moment only a filtering of the attributes has been done and Market_drop will be transformed into other data frames.
    - Market_drop_unit
        - First, did a group by function on Market_drop with the keys ['Period','Qtr', 'Manufacturer', 'SubCategory2', 'SubCategory3', 'Species', 'YEAR', 'BI Business Unit'].
        - Summed the “Value” data in order to collate the performance of the actual sales per business unit per period.
        - Reset the index after in order to maintain the table integrity so that we can subsequently use the data set as we want to keep the data in their individual rows.

- Data being stored
    - Market_drop_model
        - This is the largest transformed dataframe that we have in order to fit it into the initial regression test.
        - Label encoded all the String variables into integer labels
        - Transformations are stored in arrays for us to convert it back into strings.
        - Various sub arrays go into this dataframe for the encoding and for transformations into other suitable formats to be passed into the data.

- Summaries/reports
    - Graphs that are drawn are labelled in the Jupyter Notebook based on their inputs and commentary for the respective headers.
        - Line graphs
        - Scatter plot Matrix
        - Matrices for analysis
    - Modelling results are also kept in the Jupyter Notebook.
    - Commentary on the project is in the notebook in the form of markdown cells or commentary in the in-line code.
    - Excel graphs are kept within the data that is on the local system. Refer to data flow diagram above.

TEST SPECIFICATIONS

Model Evaluation

- How will data/outputs be compared/tested/evaluated for correctness and accuracy
    - Quantitative
        - R^2 Metric

- Can be seen as the % of the model that is explained by the model.

- Root Mean Squared Error
  - $RMSE = \frac{\sqrt{\sum(Predicted_i - Actual_i)^2}}{N}, where\ i = 1,2,3, \dots, N$
  - This will allow us to have a reference of the total error of the predicted value from the model and we can compare this value to the other models that we are testing
- Mean Absolute Error
  - $MAE = \frac{\sum|Predicted_i - Actual_i|}{N}, where\ i = 1,2,3, \dots, N$
  - Although very similar to RMSE, the 2 have different scaling in how the errors are tabulated.
- Based on the 3 metrics listed above, we are able to obtain an initial "evaluation" of the models that we are testing with the various parameters. The comparison needs to be done for like-for-like models and thus needs to always be compared to their baseline models that include all variables during RFE or on a high level when comparing across models.
- The inclusion of RMSE and MAE is in order to reduce the bias of any significant differences in the predicted value across the models and allows us to have another view on the accuracy of the prediction the model has done.

- Qualitative
  - Based on the overall look of the model and the results, does it make sense
  - Are the outputs of the models in line with BI's Business expectations
    - Initial meeting, comment on Cattle & Sheep as that's meant to be on par with the Pets sales.
    - Our Pets sales is vastly ahead of all other business units.
    - We can tell that something is wrong and need to verify.
      - The verification has been done and we have determined that a more accurate predictor to be used is the variable of "Value".

- If choosing between models, how do we compare the difference
  - First, we need to decide which model is the simpler one based on the parameter inputs and the complexity of the theory.
  - Since this project is to be handed over to BI, if the quantitative metrics difference is immaterial, the model with the easiest to understand back end will be the most accepted as there is the least resistance to change.
  - Additionally, due to the size of the datasets, there is a need to choose between the number of parameters being included for the analysis

- Some models are more fit to purpose than others and requires us to confirm with BI and the initial business requirements if they are able to accept the modelling at all.

Performance Evaluation results

- What tests have been run so far
  - We have run
    - Scatter plot matrix to test for linear relationship between variables of market_drop_model.
    - Linear regression, KNN, ARIMA and DTR.
      - Results that have been shown above in the MVP Section shows us that by-far, the KNN model outperforms the other models in terms of the variance for a dependent variable that is explained in the regression model.
      - For feature selection, we have decided to not remove any of the variables yet as BI would like to make some considerations as to which variables, they find the most important.
      - However, running the regression models on the various features individually shows an immaterial change in the R^2 value. We can infer from this that there would be no significant impact onto the model's accuracy should we choose to remove them.
      - Removing the features would help with Parsimony of the model and can make the model perform more efficiently.
- What are the future planned tests for future iterations?
  - Changing the initial parameters for the KNN model to improve model accuracy and efficiency
  - Removing variables from the overall model based on BI's business insights to streamline the model and potentially reduce noise.
  - Cutting data to determine the specific impacts of COVID-19.