DT211C
Interim Progress Report 2018/19

Wine Data Predictive Analysis

Student Name: David O'Neill	Student ID: C15737551	Supervisor Name: Bryan Duggan
Link to Software Repository:	https://github.com/DavidoDIT/FYP-Prototype	

Microsoft Office User

David O'Neill, C15737551

Table of Contents

TABLE OF FIGURES

PROJECT STATEMENT

This project of Wine Data Analysis through Machine Learning is to try and see what the data of wines can tell us and how different features of different wines can effect variables and how reviews can predict what a wines variety rating could be. The project will research and use Natural Language Processing along with Machine Learning algorithms to predict a wines price as well as predict the wines variety and rating from the review given by tasters. This data will be visualised through a cross platform application where wine enthusiasts or people in the hospitality industry can find the information they need and also give their own reviews and keep track of wines they use and taste. The goal for the project is to have completed two main parts which include; the predictive analysis and the cross platform mobile application.

RESEARCH

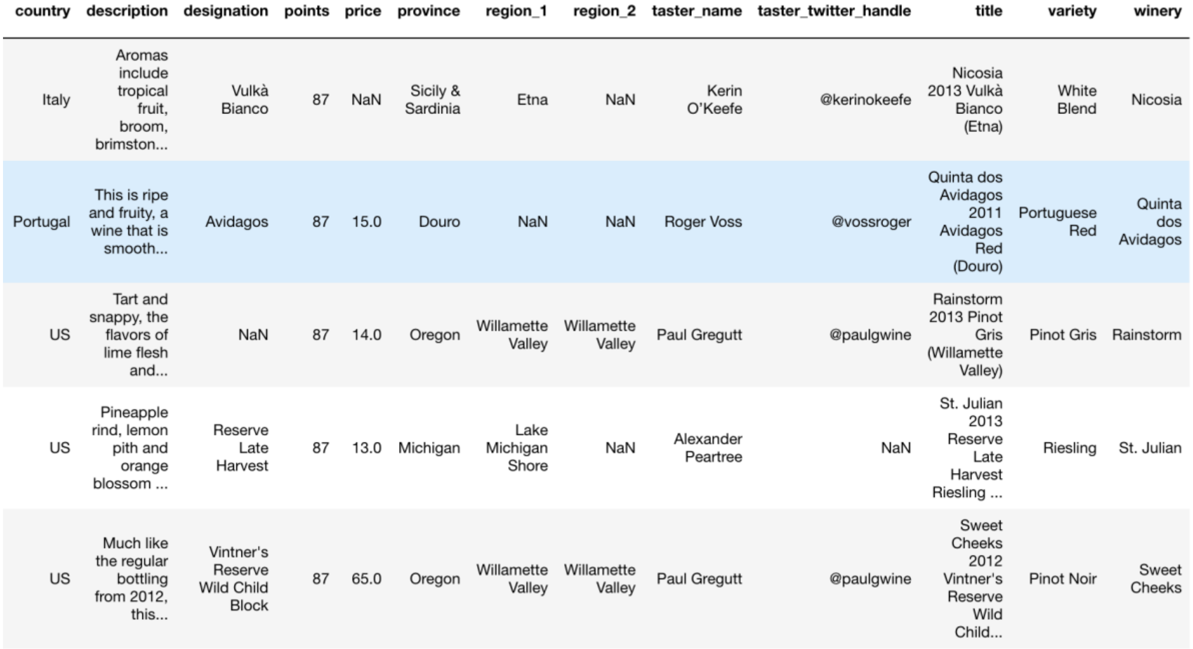
BACKGROUND RESEARCH

This project is a project in which the author plans to predict wine varieties and prices through methods of machine learning and natural language processing and predictive analytics. Predictive Analytics is the art of building using models that make predictions based on patterns extracted from historical data.[1] Just as the analysis of grapes, must and wines is a regular and important part of the wine making process,[2] the analysis of data is the most important part of the data analytics process. In order to go deep into the subject matter and properly understand all of the details and every aspect of the project a lot of research had to be completed. The history of wine, what makes certain wines taste different from other wines? how long is the process? what are the main factors that affect the final product of a wine? All of these questions had to be answered. Along with the interview of a domain expert with extensive knowledge in this field.

The earliest evidence of wine grape based fermented drinks was found in China as long ago as 7000BC. From there the rest of the world followed and discovered wines. The oldest wine production facility is almost 6100 years old, it is the Areni-1 winery in Armenia. It is still used daily, now more than ever. A bottle of wine can sell from as low as a few cent, to thousands if not millions of euro.[3]

There is a lot of factors that play a part in the pricing of wine and we will look at that in the next section with the interview of the domain expert. First, we will discuss other attempts at similar studies as a part of the research that was completed. One study was Predicting Wine Points using sentiment analysis.[4] (Wine points also mean the rating of the wine, usually from 0 – 100). In this research he authors used sentimental analysis to predict the wines rating. They came to the conclusion that sentimental analysis was a good way to try to predict the rating of a wine, they used logistical regression and the authors said the outcome of the classifier done reasonably well. Another study titled, ‘Wineinformatics: A Quantitative Analysis of Wine Reviewers’.[5] This study uses White-Box classification algorithm: Naïve Bayes and Black-Box classification algorithm: Support Vector Machine and receive almost 87% accuracy when evaluated with the SVM method. There is also studies and experiments completed in Machine Learning In Python: Essential Techniques For Predictive Analysis,[6] that predict how a wine will taste depending on its acidic and sugar levels.

The data used for this project is from Winemag.com[7], A wine review site where there are over 150,000 wine reviews which also include useful features that can be used in the project. The dataset was acquired through Kaggle[5], a data science website. It has everything needed to proceed with the project. A sample of the raw data is:



country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	variety	winery
Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)	White Blend	Nicosia
Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger	Quinta dos Avidagos 2011 Avidagos Red (Douro)	Portuguese Red	Quinta dos Avidagos
US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rainstorm 2013 Pinot Gris (Willamette Valley)	Pinot Gris	Rainstorm
US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN	St. Julian 2013 Reserve Late Harvest Riesling ...	Riesling	St. Julian
US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Sweet Cheeks 2012 Vintner's Reserve Wild Child...	Pinot Noir	Sweet Cheeks

Figure 1: Raw Data Sample

As you may notice from the first few lines of data that the data needs to be cleaned and manipulated to suit the project. A data cleansing process will begin and the data will have to be manipulated so that some columns are changed, new columns added, null values have to be dealt with among other quality issue tasks that have to be completed.

PRIMARY RESEARCH

DOMAIN EXPERT INTERVIEW

Throughout the research process of this report, to get more knowledge on different factors of the project and to find out the details in detail to understand what causes wine prices to fluctuate and what causes the wine to taste the way that it tastes, an interview with a domain expert in the field of study and research had to be carried out. The information below has been gathered from an interview with Lucy Griffin, Lucy is a professional wine taster. Lucy has been in the industry almost ten years and has lived in Bordeaux on a winery as part of her training. She has extensive knowledge in this field and it was a pleasure to interview her. The interview gave an in depth analysis on a lot of things that will help to improve this project on wine review analysis. Below is what was learned from the interview.

There are a lot of questions that you have to ask yourself when researching a predictive model on wine such as what factors affect the price? Why is one wine more expensive than the other? Why do people pay more for grand cru classé Bordeaux than they do for an Italian Pinot grigio?

To start, the following things have to be taken into account. The price or value of the land where the grapes are grown plays an impact on the pricing of different wines. For example, The average price of land for one hectare of grand cru vineyard in Burgundy is over €5 million and can be up to €10 million and more for the really renowned vineyards like Romanée-Conti. To put that into perspective, the max yield defined by the Burgundian appellation is about 35hl/ha – that's a maximum of 5000 bottles of wine.

The cost of labour – varies from country to country. There is a lot of articles about modern day slavery in South African vineyards (There is a documentary about this called Bitter Grapes.[9]) and many Irish and UK retailers now demand fair-trade wines hence prices are higher. How manual the grape growing/harvest is also comes into play. It is cheaper to mechanically harvest but that's not always possible. Some wine regions forbid mechanical processes and insist on a manual harvest. The grand cru calssé vineyards on the left bank in Bordeaux are all harvested by hand. During the interview Lucy stated that when she was at Ch. Mouton Rothschild in the Medoc, she was told that the Chateau employs an amount of 500 people over 4 weeks to harvest their grapes by hand. Some vineyards - along the Mosel in Germany for example are so steep they are not accessible by any machinery so all work is done by hand - ploughing, leaf plucking to encourage ripening, harvest, pruning. There are also wines that have to be hand harvested because of the wine-making style - sweet wines and wines where the grapes are dried or partially dried like Amarone della Valpolicella have to be hand harvested.

Max yields are defined by the appellations laws. Each region has their own max yields. The best winemakers often come in good bit lower than the max but they can't produce any wine over the max for their appellation so even if they have a really great harvest in a good year, they can't sell the extra wine under their appellation. As it's limited quantity, it can affect the cost.

Another factor to consider is Trends - One of the best examples of this is Chateauneuf-du-pape. It always sells for a premium and people in Ireland know it well and love it. There are smaller lesser known areas that are right next to Chateauneuf-du-pape (like Lirac or Vacqueyras) that have identical terroir and weather and grow the same grapes and make wine in the same style but you can get it at a fraction the price. Along with this is, How easy or difficult the grapes are to grow - Pinot Noir is a fussy thin skinned grape. It is susceptible to rot and disease. It needs near perfect conditions to grow - thus is usually more expensive although Chile have started growing Pinot Noir and can make it relatively inexpensively thanks to their dry moderate climate in the central valley areas.

Another factor to consider is a big one. It is the Weather/Vintage - Great Vintages cost a premium. It is most prevalent in Bordeaux where good vintages can be aged for longer and increase in price with time. Weather is one of the biggest factors on how a wine can vary from year to year - it is mainly down to how ripe the grapes get or if there's an event which causes a low harvest. The frosts across France last year and this year would have pushed up prices a lot especially across Burgundy and Chablis. The hot summer and drought this year meant grapes were getting “sugar ripe” without phenolic ripeness - this is where the grape produces lots of sugar but the phenolics in the skin which contributes most of the flavour isn’t ripe. It results in high alcohol wines without good flavour.

Next factor to discuss is the Aging - not all wine is aged, some is kept in stainless steel tanks until filtration and bottling. Other is aged in stainless steel and oak chips can be added to give it some flavour. Other wine is aged in either French or American oak barrels. The size and origin of the barrel depends on the wine and what complexities the winemaker wants to add to their wine. A bigger barrel is used to add more oxidative flavour to the wine and little oak influence. A smaller barrel is used to add oak flavours to the wine. A French barrel is usually 225L and costs upward of €700 per barrel. Some winemakers opt for second hand barrels from neighbouring chateaux or age a portion of their wine in one or two year old barrels. Along with Aging, Irrigation is forbidden in most of Europe but widely used in the New World.

The trade position or how many middle men are involved plays a big part in pricing. Burgundy is really fragmented - a grower might own 2 rows of vines and sell their grapes to a winemaker (possibly through a 3rd party). These negotiates (importer, exported of wines) blend, bottle, age and sell the wine under their own name. Whereas in other regions, the grape grower is the winemaker and bottler as well. Along with, legislation around use of pesticides/fungicides and if the wine is organic etc - grapes are susceptible to rot/disease. The age of the vineyard also comes into play, older vines generally make better quality wine which is more concentrated, but are also lower yielding.

Supply and demand is another factor to consider, Champagne is a good example of how supply and demand affects the price. Alongside this is where the wine is bottled – not all wine travels in bottles. Wine is increasingly bulk shipped in massive bladders in freight containers and bottled in the UK and in Germany for the Irish market. The Winemaking process is a good pricing factor too - sweet wine like Sauternes/Ice Wine is made by concentrating the grape juice so that there is so much sugar in the juice that the yeast dies during fermentation and the wine is left with some residual sweetness. It is done with two different methods but the end result is that more grapes makes less wine which make it more expensive. The same can be said for Amarone wine in Italy. It is a dry style but the bunches of grapes are left out to dry before fermentation. It makes a really concentrated wine but the volumes are much less than if the grapes weren’t allowed to raisin first. Also, what’s rare is expensive. There is a couple of vineyards in Europe which can claim they are pre-phylloxera (they are mostly in the sandy soils of Northern Spain). The most the oldest vines in Europe.

When it comes to predicting wine prices on let’s say, a yearly basis, you have to consider why are some vintages more expensive than other and what commercial factors make the wine price go up and down? The following should be taken into account when studying why prices may fluctuate:

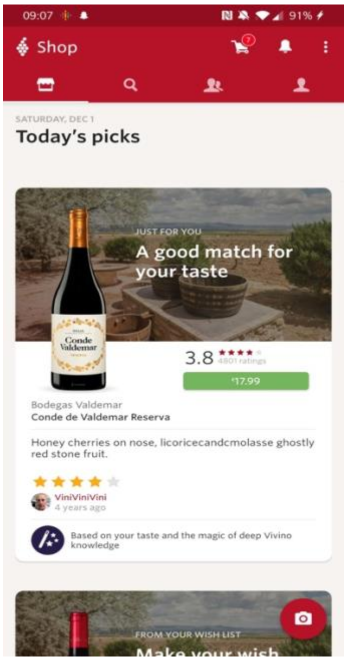
1. Weather – is the biggest variable from year to year. Wine seems really ‘glam’ but most are regular fruit farmers for 360 days of the year and winemakers for five and just like the poor grain harvest is affecting every agricultural commodity in Ireland at the minute and driving up prices from butter to deadweight chickens, the poor grape harvest this Autumn in the Northern Hemisphere is going to put wine prices through the roof.
2. Cost of Diesel and Crude Oil prices - the shipping has to be taken in to account. This will also have a bearing on the cost at cellar door as the farmer will use diesel to plough, tend the land etc.
3. Cost of Glass (Bottle), Cork/ Metal (Closure), Paper (Label)
4. Changes in cost of labour

All of the above is what plays a factor in wine prices changing. This interview was invaluable to the research process of this project and it really shows the level of detail you have to consider when working with wine features.

ALTERNATE EXISTING SOLUTIONS

Vivino

Throughout the research stage, two applications came to light that were found to reflect similarly to what this project plans to achieve. The first is ‘Vivino’. [10] Vivino is an application on the google play store that holds a database of wines and their prices and reviews and ratings. The application lets users view these reviews and also make their own review as well as creating their own ‘wine cellar’ in the application. This application is similar to what this projects final product will hope to look like plus the added sections that include the machine learning algorithms that will run against user input data. Below is what the application looks like.



SATURDAY, DEC 1
Today’s picks

JUST FOR YOU
A good match for your taste

3.8 ★★★★ 4801 ratings
$17.99

Bodegas Valdemar
Conde de Valdemar Reserva

Honey cherries on nose, licoricecandcmolasse ghostly red stone fruit.

★★★★
ViniViniVini
4 years ago

Based on your taste and the magic of deep Vivino knowledge

FROM YOUR WISH LIST
Make your wish

Figure 2: Vivino Homepage

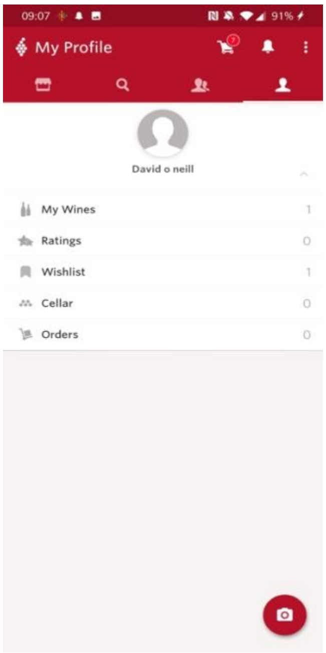


Figure 3: Vivino Userpage

MINTEC

The second existing technology is a tool called MINTEC. [11] MINTEC is an advanced analytical tool that is widely used in the wine industry. The tool tracks wine prices for the year ahead and each year following depending on the cost of different commodities. This tool isn't just used in the wine industry, it is used in all types of businesses where analytics is needed. It follows a similar structure to how this project is structed, regarding the input and output.

TECHNOLOGIES RESEARCHED

Python

- Python is a simple language that people pick up easily enough quite often as a hobby or for work to automate tasks. It has great libraries for building web apps. It's great for handling HTTP, especially with its libraries such as Flask and Django. It scales well and is used on websites such as Nasa and Reddit. It is heavily used for scientific computing. It's libraries NumPy and SciPy are great examples of this. It has libraries that support sound, mouse and keyboard interactions such as PyGame, which is great for Creating games.
- Python is great for data analysis and machine learning. It comes with libraries such as Pandas, SciKitLearn, Seaborn and MatPlotLib. These libraries are great for implementing numerous machine learning models, plotting and visualizing data, as well as manipulating datasets for running data analysis. It is also excellent for Natural Language Processing, it uses the library NLTK.

R

- R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible.

- One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control. [12]

Java

- The Java™ Programming Language is a general-purpose, concurrent, strongly typed, class-based object-oriented language. It is normally compiled to the bytecode instruction set and binary format defined in the Java Virtual Machine Specification. [13]
- Java can be used for machine learning, although it is not as popular as Python. Some libraries used for Machine Learning with Java include ELKI, DeepLearning4j, JSAT and MALLET.

JavaScript

- JavaScript is one of the three core languages of websites, It adds behaviour and interactivity to a website. JavaScript is classed as a scripting language so it doesn't have the same features such as C++ or Java.

React Native

- React Native is a framework which uses a mixture of React and JavaScript which gives you a native iOS and Android application. It was created by Facebook and instead of targeting the browser it targets mobile platforms.

MongoDB

- MongoDB is a NoSQL database. MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time. MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use.[14]

MySQL

- MySQL is an open-sourced database management system developed and supported by Oracle. MySQL databases are relational. It stores data is separate tables rather than one large table. It is very fast, reliable, scalable and easy to use.
- The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). [15]

NodeJs

- NodeJS is an open-source, cross platform environment for executing JavaScript code outside of a browser. It is used to call APIs and to talk to the backend of a web interface. It can be used to build highly-scalable, data-intensive and real-time backend services that power client applications. It also as the largest amount of open source libraries available.

Flask

- Flask is a Python web framework which means that flask gives you the needed tools, technologies and libraries that allow you to build a web application. It is a micro-framework so it has little to none external libraries.
- Flask is light, there are little dependencies to update and watch out for security bugs. Although because of this you have to do most of the work yourself and install plug-ins and not depend on external libraries.

Django

- Django is a high-level web framework developed with Python. It is extremely fast, secure and exceedingly scalable. It encourages rapid development and a clean design.
- It comes with dozens of external libraries that you can use to help you develop your web framework. It can take care of things such as user authentication, content administration and sitemaps. It has higher security than Flask and a lot of other web frameworks. It is also incredibly versatile and can be used to build numerous types of websites.

Digital Ocean

- Digital Ocean is an Infrastructure as a Service provider, it is a simple cloud hosting environment built for developers. Digital Ocean gives you servers that you can create droplets or containers on. You can run your website/host your database on a droplet or you can use it for test purposes. Controlling your droplets is very easy due to its built in GUI and it is on a similar level to Amazon Web Services.

Amazon Web Services (AWS)

- Amazon Web Services is a secure cloud services platform where you can host your applications on a cloud server, you can then use your databases and store on their services and use them on your application. You can also use their computing power. AWS gives you the resources you need to build sophisticated, scalable applications of any size.

React Native

- React Native allows you to build native mobile acts using JavaScript and React. This gives you a full native app for android and iOS. It is built using Facebook's JavaScript library for building user interfaces. React Native makes it easier to develop cross platform.

Xamarin

- Xamarin was recently purchased by Microsoft and has become open source. It allows you to build cross platform native applications using C#. It gives native user interfaces, native APIs and native user performance. It allows you to build applications and also has an added bonus of the 'Xamarin University'. A free resource to help you get started with Xamarin.

Dash

- Dash is great for creating reactive, web-based applications, it uses an open source python library. It is still very new as it is only three years old. It is used to create analytic web applications. It can help to show stakeholders the data analysis, visualization and exploration of their data in the form of a web application. It uses mostly plot.ly, which is a python library for graphs and displaying data, along with Flask on the backend. It uses pure python to get the same results as you would by using some JavaScript charting libraries.

OTHER RELAVENT RESEARCH

Digital Ocean vs Amazon Web Services

Digital Ocean and Amazon Web Services are similar so the choice to use either one was mostly personal. First here are some similarities and differences…

- Digital Ocean targets small developers who need to start up small high-performance instances. It has a user-friendly , clean interfaces with few features and one-click deployments.
- It has Hourly Billing, Built-In Control Panel, SSH Key Setup and it has 7 Data Centres on 3 Continents. It has Monitoring Charts as well as REST API functionality.
- Amazon Web Services has very high virtual machine performance. It offers a broad range of IaaS/PaaS products with nearly all cloud services.
- It has Hourly Billing, Built-In Control Panel, SSH Key Setup and it has 10 Data Centres on 4 Continents. As well as, Monitoring Charts and REST API functionality.

They are both similar in a lot of ways with few differences. I am deciding to go with Amazon Web Services for this Project as I find it easier to navigate and use. As well as Amazon Web Services having a very good student credit deal.

MongoDB vs MySQL

MongoDB is a NoSQL database which means it does not use Sequel Querying Language. Whereas, MySQL uses SQL, SQL language is the preferred method of database querying for this project as it has been used on many projects before this one and the author is more familiar with the MySQL database than with MongoDB and NoSQL.

Node.js vs Django

Nodejs allows you to use open source JavaScript on the server side of your project and not just on the front end. It is good if you are using JavaScript on the front end and want to keep a similar coding style. Node.js is incredibly scalable but for this project we won’t need to be scaling massively. Node.js follows event driven programming architecture. It has good performance but it is less complex than Django. It is widely used all across the globe, used by many big companies and is quiet ahead technology wise compared to Django. Node.js has good security but the developers have to be aware of this and make sure that it is secure.

Django is an open source web framework coded in Python. It is not as scalable compared to Node.js and it follows the Model, View, Controller architecture, unlike the event driven architecture that Node.js follows. It has better performance and it a lot more complex. It is relatively new compared to Node.js, leading it to be a bit more behind in terms of usage. Django has excellent security and the developer doesn't have to worry about it too much.

Django has been chosen to be used in this project as the project will involve working with a very large dataset and it will need to have a powerful backend. Therefore, Django is a better fit for this use case.

React Native vs Xamarin

React Native and Xamarin have major common points. They are both great and do practically the same thing. In this section you will see the comparison of the two and their benefits. React Native is still relatively new, there is not as much support compared to Xamarin. When using Xamarin you used Visual Studio as an IDE, this keeps everything together, you can code your application, connect your API's all through here. React Native has less features like this. When it comes to code compilation, Xamarin wins. Cross platform environment, Xamarin wins here too as everything is done through Visual Studio, React Native usually runs on Expo as an IDE, which doesn't support all of React Natives features. In terms of documentation, React Native is better than Xamarin, everything you need to know is in one place and well documented. Whereas, Xamarin documentation could do with some improvement. Since Xamarin is an older platform, the community is a lot larger than React Native which is relatively new, getting assistance will be a lot easier while using Xamarin. In terms of performance, Xamarin is better as it supports 64-bit Android whereas, React Native does not.

With all these in mind, the application section of this project will be completed using Xamarin. Not only has it better support and more features needed for this project. It has never been used in a Final Year Project before, which will be a great challenge to complete to be the first project to use this platform.

MACHINE LEARNING ALGORITHMS

This section of the report is to discuss the research predictive algorithms and techniques that could be used on this project.

KNearestNeighbors

KNearestNeighbor or KNN for short, is a machine learning algorithm. It can be used for classification or regression, mostly classification. KNN was used in the prototype for this project. KNN is a very simple algorithm. It identifies the K nearest neighbours of C. If we have classes of plus and minus and k = 5 then we have to find the 5 nearest neighbours to C. The algorithm then uses these results to predict the outcome.

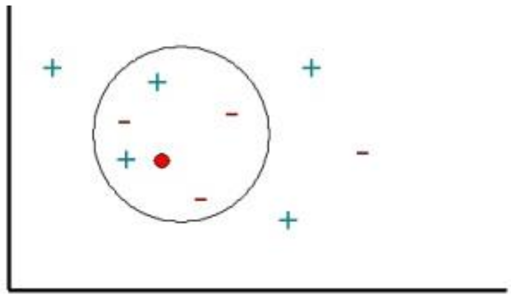
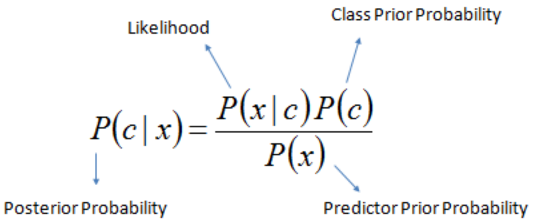


Figure 4: KNN example

Naïve Bayes

This is a machine learning algorithm for classification problems. It is mostly used for text classification like sentimental analysis. It is simple yet extremely effective. It learns the probability of an object with certain features belonging to a particular class. It uses the Bayes theorem which states that probability of event C given X is equal to the probability of the event X given C multiplied by the probability of X upon probability of C.



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

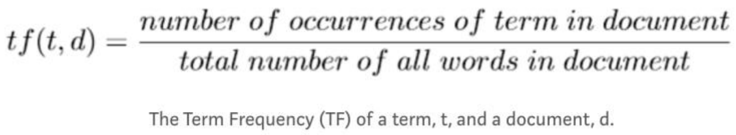
[16]

Figure 5: Bayes Theorem.

Natural Language Processing

There are a lot of different sections to discuss in Natural Language Processing but for this project we will talk about three of them.

- Python has a great library called **Natural Language Tool Kits**. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. [17]
- We will also use **Bag Of Words Counts** – this embeds sentences as a list of 0 or 1, 1 represents containing word. This model extracts features you could use in your predictive model. It is called a “*bag*” of words, because any information about the order or structure of words in the wine descriptions is discarded. The model is only concerned with whether known words occur in the descriptions, not where in the descriptions.
- The next model we will look at is **TF-IDF (Term Frequency, Inverse Document Frequency)** – this model is weighing words by how frequent they are in our dataset, discounting words that are too frequent. It works by finding out how often a word appears over the total number of words.



$$tf(t,d) = \frac{number\ of\ occurrences\ of\ term\ in\ document}{total\ number\ of\ all\ words\ in\ document}$$

The Term Frequency (TF) of a term, t, and a document, d.

Figure 6: Term-Frequency equation

RESULTANT FINDINGS AND REQUIREMENTS

Models

After analysing my research it has been found that KNN was not the best approach for this problem. KNN was not accurate and it would not give the desired results. It was decided that Naïve Bayes will give the needed result using probability. So this is the model that is planned to be used going forward. This along with the use of natural language processing. A massive amount of knowledge on this subject was gained from the domain expert interview. It gave a really good insight into what makes each feature of the dataset different and the factors that determined why each feature is what it is. The data can now be analysed properly and cleaned accordingly as well as the models will be more accurate since we know the background of the features.

Frontend

For the prototype, Dash was used. This is an incredible front end choice and it is a possibility that it could be used in the project depending on how time requirements are. But, the plan is to create a cross platform native application using Xamarin. This application will visualize the wine reviews data for the user. I chose these technologies as I love what dash can do in regards to visualizing data, but I also chose Xamarin as I see great potential in this application if Xamarin is used.

Backend

I have chosen MySQL as the backend for this project as I have more knowledge and practise with MySQL compare to MongoDB so it was decided that this is the best option. It also runs nicely with Django and Amazon Web Services.

Server Side

Amazon Web Services with Django.
I have chosen this combination as I feel like Amazon Web Services has a lot of support in regards of documentation and help. It also has a lot of features that could be useful throughout the production of this project. I have researched that it works well with Django and Django was chosen as it is a python backend and this project involves a great amount of python coding. For the server side on the prototype, it is Amazon Web Services with Flask since is a built in feature of Dash.

PROPOSED SOLUTION

Using the above findings I plan to create an Amazon Web Services Server which will host all of the code. Django will contain the backend APIs which will call the Xamarin application and also connect to the MySQL database using Restful API calls. The data cleaning and predictive analysis will be completed through Jupyter Notebooks, An interactive Python/R IDE.

SYSTEM REQUIREMENTS AND PRIORITIES

Below is the system requirements for this project as well as their level of priority. As you can see the application priority is quite low. That set to low because, depending on how complex the machine learning and natural language processing becomes, the final product may have to be completed as a web application. For now, we are going to include it.

Requirement No.	Name	Description	Priority
1.	Database	Setup a connection to a database on a webserver.	High
2.	Clean Data	Data must be properly cleaned and manipulated.	High
3.	Natural Language Processing	Natural Language Processing will have to be implemented on the data to predict a variety from the description.	High
4.	Machine Learning	Machine Learning Algorithm will have to be implemented on the data.	High
5.	Cloud Service	Server and database will be available through Amazon Web Services.	High
6.	System on Django Server	The project is running on a server and is accessible not on a local host. Must be running on Django	High
7.	Other Data visualization features	The application will contain multiple ways to visualize and view the data and predictive models.	High
8.	WebApp	Data and data visualisation will be available on a web app, if not a cross platform native application.	High
9.	Analyse User Wine Reviews.	Users will review wines, wine reviews will be submitted and a wine variety predicted.	Medium
10.	User Login /Logout	The user needs to be able to log in and log out.	Medium
11.	Cross Platform Application	The system will run on a cross platform native application	Medium
12.	Search for wine	The user must be able to search for wines.	Medium
13.	Wine review	Users will review wines	Low
14.	Data Heatmaps/Data Maps	Display data to user on heat maps and data maps using geolocation	Low
15.	Account setup page	Users will have to be able to create an account	Low

Figure 7: Priorities of System

BIBLIOGRAPHY

[1] J. D. Kelleher, B. M. Namee, and A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. MIT Press, 2015.

[2] J. Robinson and J. Harding, Eds., *The Oxford Companion to Wine*. Oxford University Press, 2015.

[3] A. Kassam and N. Davis, "Evidence of world's earliest winemaking uncovered by archaeologists," *The Guardian*, 13-Nov-2017.

[4] Y. Lun, K. Gu, and S. Yang, "Predicting Wine Points using sentiment analysis," p. 7.

[5] B. Chen, V. Velchev, J. Palmor, and T. Atkinson, "Wineinformatics: A Quantitative Analysis of Wine Reviewers."

[6] M. Bowles, *Machine Learning in Python: Essential Techniques for Predictive Analysis*. John Wiley & Sons, 2015.

[7] "Wine Enthusiast Magazine | Wine Ratings, Wine News, Recipe Pairings," *Wine Enthusiast Magazine*. [Online]. Available: https://www.winemag.com/. [Accessed: 30-Nov-2018].

[8] "Kaggle: Your Home for Data Science." [Online]. Available: https://www.kaggle.com/. [Accessed: 30-Nov-2018].

[9] "Bitter Grapes," *Bitter Grapes*. [Online]. Available: http://www.bittergrapes.net/. [Accessed: 04-Dec-2018].

[10] "Vivino.com - Find and buy wine in seconds." [Online]. Available: https://www.vivino.com/. [Accessed: 04-Dec-2018].

[11] M. LTD, "Mintec Global." [Online]. Available: https://www.mintecglobal.com. [Accessed: 01-Dec-2018].

[12] "R: What is R?" [Online]. Available: https://www.r-project.org/about.html. [Accessed: 27-Nov-2018].

[13] "Java Programming Language." [Online]. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html. [Accessed: 27-Nov-2018].

[14] "What Is MongoDB? | MongoDB." [Online]. Available: https://www.mongodb.com/what-is-mongodb. [Accessed: 27-Nov-2018].

[15] "MySQL :: MySQL 5.7 Reference Manual :: 1.3.1 What is MySQL?" [Online]. Available: https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html. [Accessed: 28-Nov-2018].

[16] "Naive Bayesian." [Online]. Available: https://www.saedsayad.com/naive_bayesian.htm. [Accessed: 02-Dec-2018].

[17] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.

[18] "White Box Testing," *Software Testing Fundamentals*, 19-Dec-2010. .

[19] "Black Box Testing," *Software Testing Fundamentals*, 19-Dec-2017. .

APPROACH AND METDHODOLOGY

For this project there will be two approaches and methodologies used. The project will take on the Agile Methodology as well as CRIP-DM as it is data and machine learning heavy. In this section there will be two methodologies discussed.

Agile Methodology

Agile Methodology is a term that covers several project management approaches that allow teams to respond to changing requirements through incremental and iterative work. Agile came about in the 1990's when the software industry began to rise, this is when rigid, sequential software development methods couldn't keep up with the rapidly changing requirements and priorities in projects. How this works is Business Users will relay their product requirements to developers through user stories. The team takes the user stories and figure out how to make the product in stages and make sure that the product gets delivered on time to the business users. User stories are sorted into order of what needs to get done first according to the priority of the user story. These are then completed in sprints, which are usually one or two week blocks of where certain issues and user stories are assigned to developers to be completed within that time frame. This is all over looked by a Scrum Master. Developers add to the product as they go and improve 'on the go' then adjust accordingly depending on feedback from other developers. This feedback stage is called a retrospective. The sprints continue until the product is delivered and the cycle starts over again.

This model suits this project as a lot of variables will change throughout the project and this is a more flexible model compared to the waterfall model.

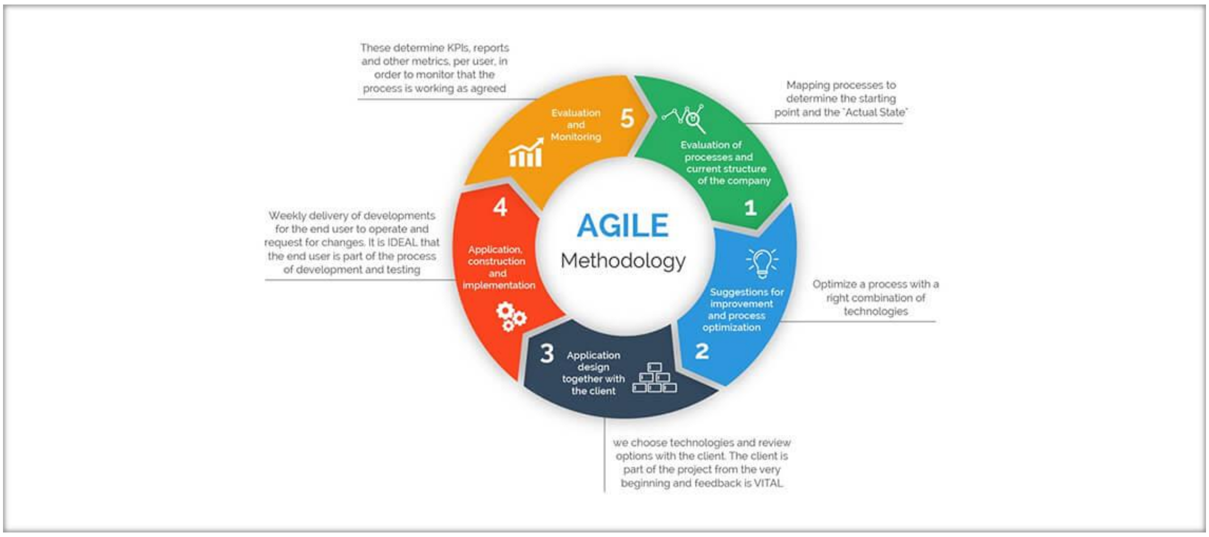

These determine KPIs, reports and other metrics, per user, in order to monitor that the process is working as agreed
Mapping processes to determine the starting point and the 'Actual State'
Evaluation and Monitoring
5
Evaluation of processes and current structure of the company
1
4
AGILE
Methodology
Weekly delivery of developments for the end user to operate and request for changes. It is IDEAL that the end user is part of the process of development and testing
Application, construction and implementation
Suggestions for improvement and process optimization
Optimize a process with a right combination of technologies
3
Application design together with the client
2
we choose technologies and review options with the client. The client is part of the project from the very beginning and feedback is VITAL

Figure 8: Agile Methodology

Crisp-DM

When working with data it is a good idea to follow a Crisp-DM model. Crisp-DM stands for 'Cross Industry Process for Data Mining. It is a very popular methodology which uses a structured approach to planning a data mining project. It has six phases which are Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment.

- In the Business understanding phase you have to figure out what is that has to be done. The business objectives have to be determined, determine the project goals and produce the project plan.
- The Data understanding stage is where the data has to be found and has to be analysed to see if the data will suit to the needs of the project as well the data's strengths and limitations. The data has to be collected, explored and the quality has to be verified.
- In the data preparation stage the data has to be prepared. This stage is the most time consuming. The data has to be selected, cleaned, constructed and integrated with the product.
- In the Modelling stage the model you wish to use has to be applied to the data. This stage is completed side by side with the data preparation stage. Models are built as you find your data and changed depending on the results of the data preparation stage.
- In the Evaluation stage the models have to be evaluated for suitability with the project and end product. Does the model help to satisfy the business goals? If so then the project moves to the deployment stage if not then the data gets brought back to the business.
- The Deployment stage is when the prototype has to be developed to a working product and it then gets deployed.

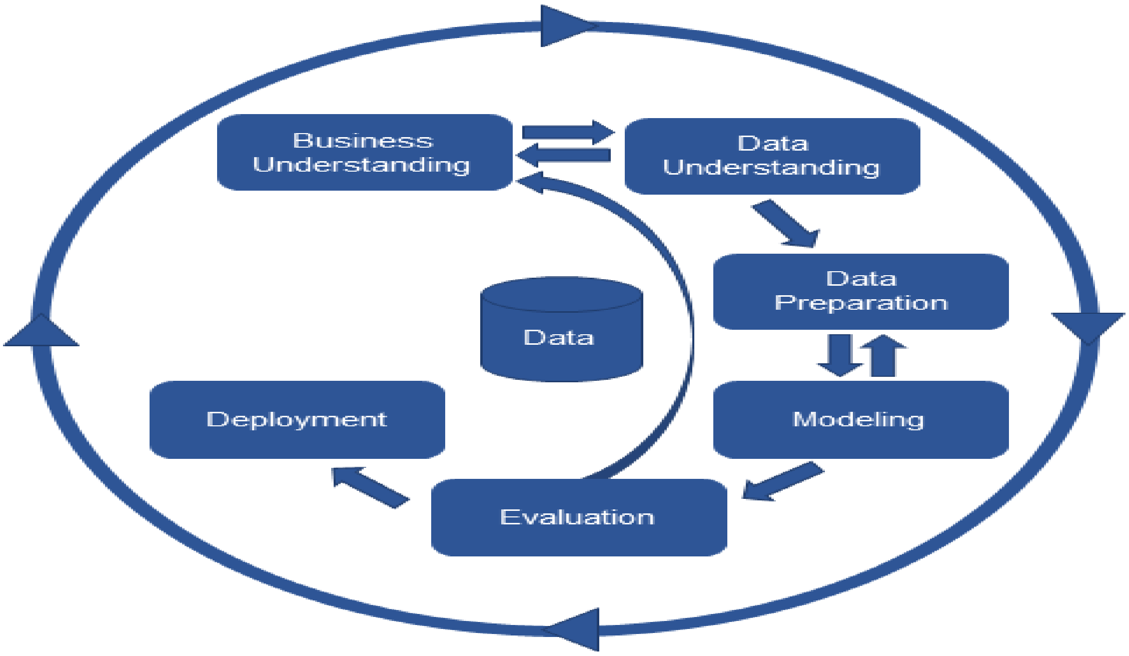


Figure 9: Crisp-DM Methodology

DESIGN

TECHNICAL ARCHITECTURE DIAGRAM

Below is a diagram of this projects technical architecture. This is how everything flows and connects together. Theses connections used Restful API calls and data will be sent through these connections using the JSON format.

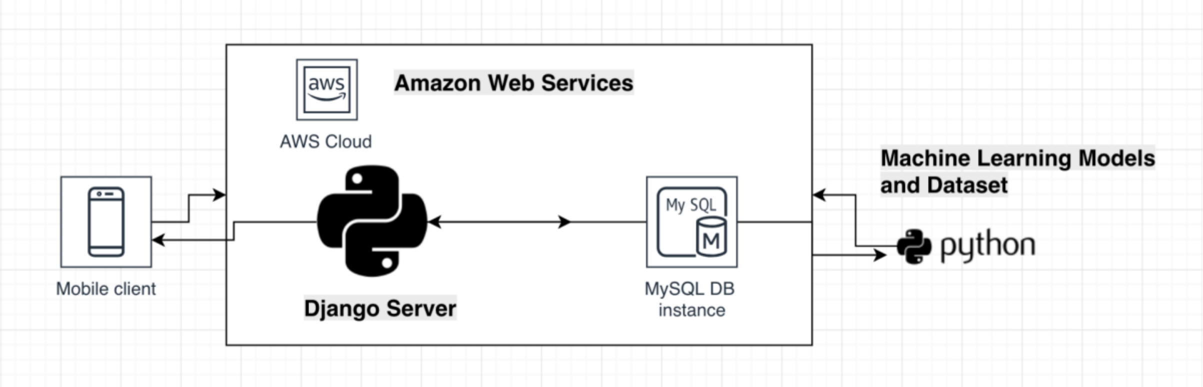


Figure 10: Technical Architecture

USE CASE

Below is a small simplified use case for this project. The Author will feed data to the predictive pnalysis process and it will display through an application for a user to use.

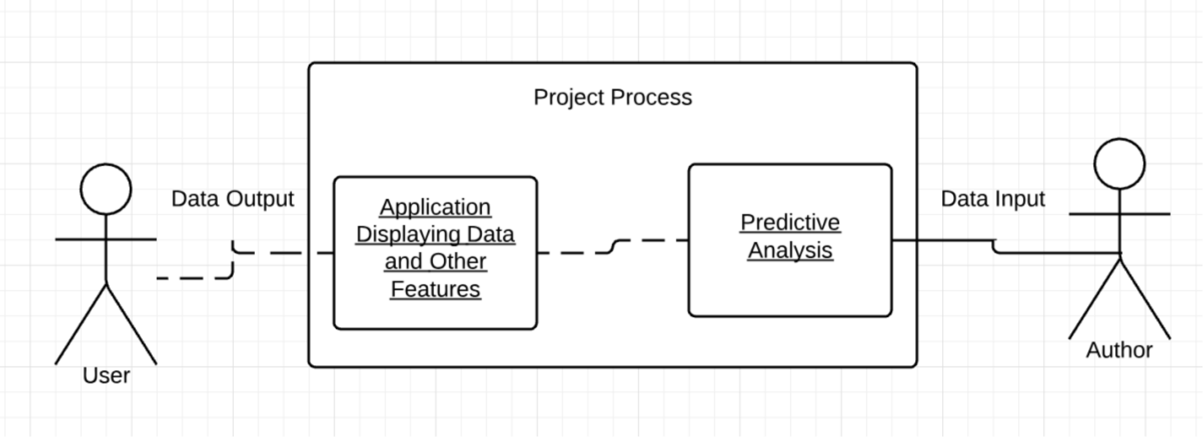
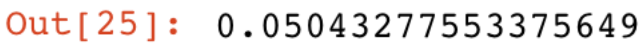


Figure 11: Project UseCase

PROTOTYPING AND DEVELOPMENT

The prototype is an important part of a project as it gives an idea of the direction of a project. It will show whether the proposed approach will succeed or fail. Prototyping reduces the overall time spent developing the project as you realise what works and what doesn't. It is easier to identify potential problems and what changes need to be made before beginning the product development.

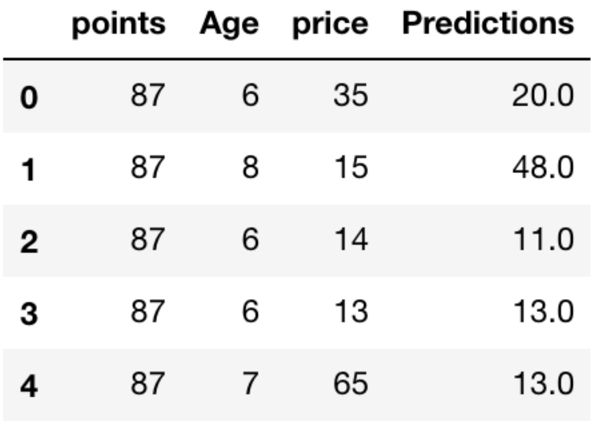
In this project a prototype was created using the KNearestNeighbor Classifier as well as Dash and Flask. When reviewing the results of the prototype classifier it was determined that KNearestNeighbor was not an ideal approach. The accuracy was returning at only 0.05%.



```
Out[25]: 0.05043277553375649
```

Figure 12: KNN Prototype Accuracy

This is not enough for the final product. The model needs to have good accuracy to return good, realistic results. KNearestNeighbor only used three features of the dataset, The Age and Points/Rating and the Price. These were used to try and predict the price. While the model was giving a prediction, the outcome cannot be trusted as the accuracy was so low. The model needs more features to get an accurate return. When compared to the dataset the predictions are extremely off. For example:
In Figure 13, you can see the first 5 predictions that were made under the 'Predictions' column. We can compare these to the 'price' column which is the actual price of the wine given its rating /points and age. You can see that the predictive algorithm is extremely off from the actual result. Only predicting correctly once in this example, with all other predictions being +10 away from the actual price except for one row.

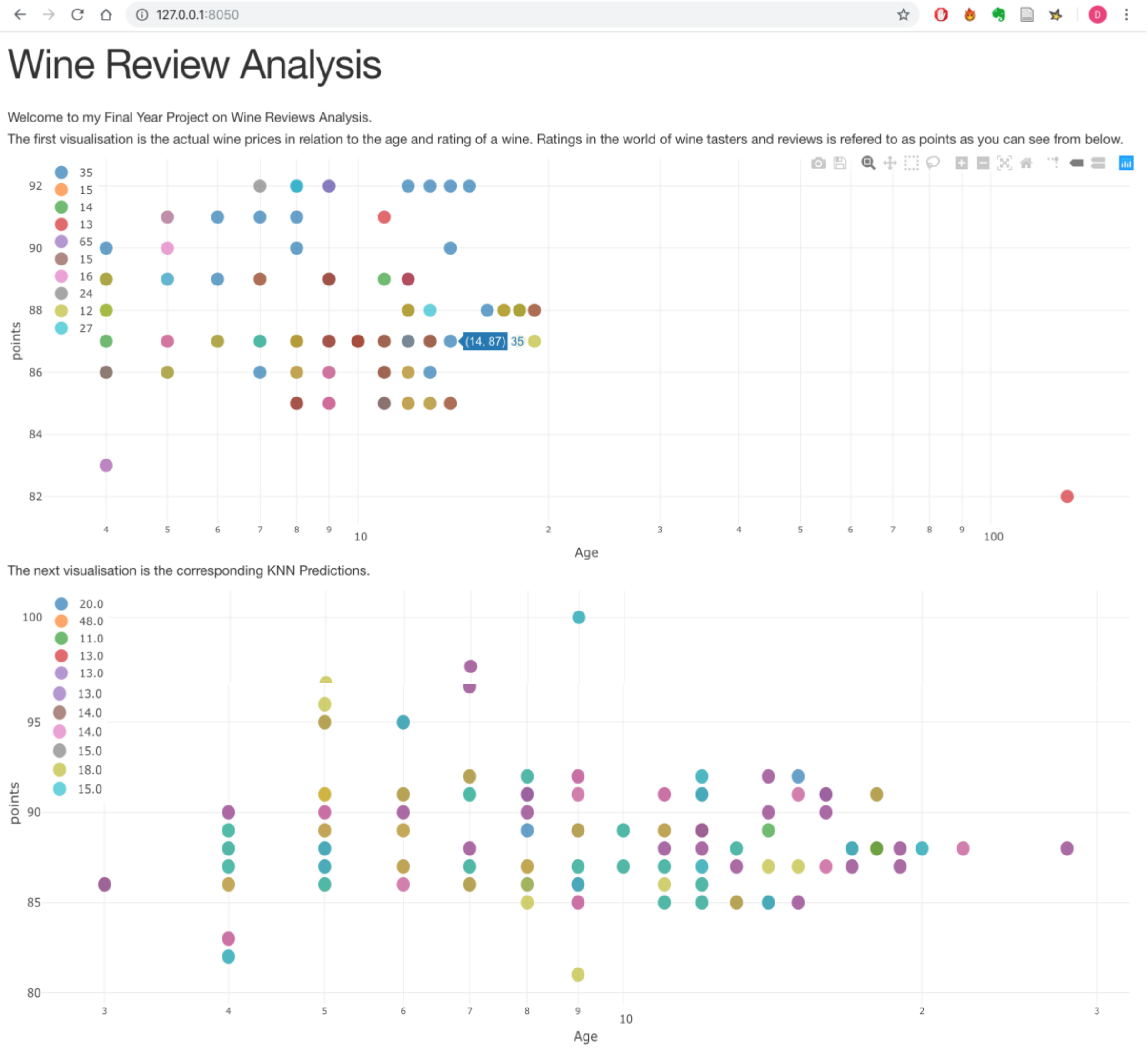


	points	Age	price	Predictions
0	87	6	35	20.0
1	87	8	15	48.0
2	87	6	14	11.0
3	87	6	13	13.0
4	87	7	65	13.0

Figure 13: KNN Comparison

The Server Side was created using a combination of Dash and Flask. Dash is visualisation web framework developed with Python. It runs on a flask server and it is excellent to use as a dashboard for your data. Although, it came to the realisation that this wasn't going to be a viable solution or approach for this project. The project needs to be more customer/consumer focused, letting the user have control and make the product more than just the visualisation of data models and their results.

This is the Flask Sever running with Dash on a localhost:



Wine Review Analysis

Welcome to my Final Year Project on Wine Reviews Analysis.

The first visualisation is the actual wine prices in relation to the age and rating of a wine. Ratings in the world of wine tasters and reviews is refered to as points as you can see from below.

The next visualisation is the corresponding KNN Predictions.

Figure 14: Prototype Web Deployment

All of the Data Analysis and Predictive Modelling was completed in Jupyter Notebooks, which is an open source web application that lets you produce live code and visualisation. It is extremely useful for data analysis and predictive modelling as you can see how you progress bit by bit. The Flask and Dash was developed through PyCharm which is a well-known Python IDE.

TESTING

Testing your product is an essential step of any project. Your product needs to be tested so that it can functionally work without error when it is released to production. There is multiple types of testing that can be carried out.

White Box Testing

White Box Testing or also known as Clear Box Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.[18] What is being looked for while testing is being complete is, internal security holes, broken or poorly structured paths in any code, The flow of inputs and the expected output. It is checking pre-determined inputs and trying to find expected outputs from that. There are two main steps to white box testing which fall under these two headings:

1. Understand the source code.
   The tester needs to understand how the source code works. Why does 'that loop' work with 'that function'.
2. Create test cases and execute.
   When creating the test cases you have to cover every aspect of the code. All paths (if-loops etc.). You want to cover around 90% of your code to test.

Examples of types of white box testing would be, Unit Testing – Usually the first call of action when testing an application. This is where each block of code or unit of code is tested. . It helps to identify almost all of the bugs early on in your testing which will lead to a better product. Another type would be White Box Penetration Testing – During this test the tester wants to try to attack the code from every angle and aspect of the code to test the applications level of security.

Advantages of White Box Testing are that:

- Hidden errors will be found and code can be optimised.
- These types of tests can be easily automated.
- Covers every aspect of your code and every path.

Black Box Testing

Black Box Testing, also known as Behavioural Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This type of testing tests for how the application reacts to user inputs, interface errors, errors in the database and termination errors. The tester is able to test the input and the output without knowing about the internal structure of the application. [19] The steps to black box testing are:

1. Functional Testing
2. Non-Functional Testing
3. Regression Testing

To perform black box testing the tester chooses valid inputs and checks if the application processes them correctly and compares the output to the desired output.

White box and Black box testing will be carried out throughout this project as there will be an internal code structure as well as an external product. People that have relevant knowledge of the code base will perform the white box testing and other people without knowledge of the internal structure will test the outer layers of the product. Testing will be executed continuously throughout the project cycle. It will then be tested after the project has been completed. Test scripts will be written up and testers will use these scripts as a basis for their testing.

ISSUES AND RISKS

Every project will have issues along with a number of risks. The issues and risks related to this project are:

- Cross Platform Application – The author has very little experience with Xamarin so it will be a steep learning curve for this project.
- Amazon Web Services – The same as above. The author is still learning all the needed fundamentals for Amazon Web Services but is determined to get everything working smoothly for the final project.
- Displaying the predictive models on an application – Displaying the predictive models will also be a challenge as there is a steep learning curve involved.

PLAN AND FUTURE WORK

Below is my projects GANTT Chart or Scrum Plan. Tasks are separated into mostly two week sprints, with bigger objectives getting more time than objectives with smaller tasks. When a task is complete it will be tested to make sure that it will work with the project. After all sprints have been completed testing will happen again on the final product. These sprints could change throughout the project, varying in length and objective.

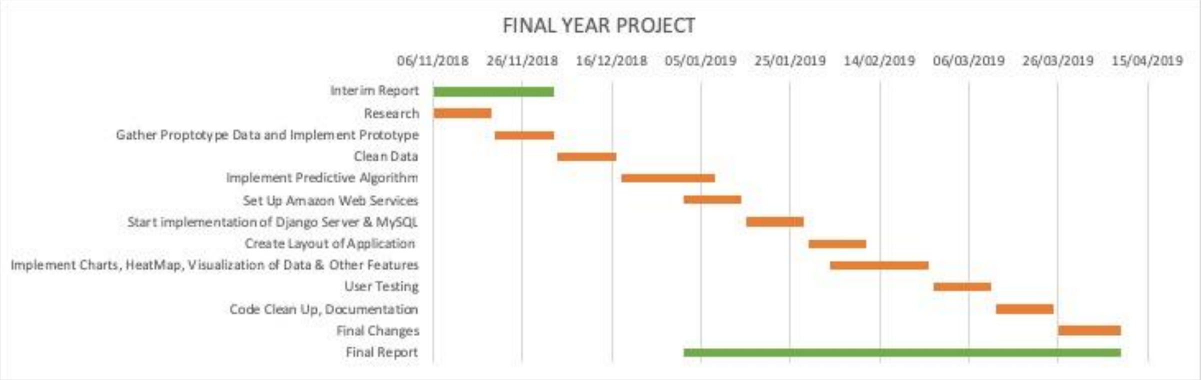


Figure 15: GANTT CHART