Brandon Walker

WGU | November 1, 2020

Brandon Walker

C964 Capstone

Contents

[Letter of Transmittal – Section A1 3](#_Toc57918723)

[Proposal – Section A2 4](#_Toc57918724)

[Problem Statement 4](#_Toc57918725)

[Customer Summary 4](#_Toc57918726)

[Existing System Analysis 4](#_Toc57918727)

[Data 5](#_Toc57918728)

[Project Methodology 5](#_Toc57918729)

[Project Outcomes 6](#_Toc57918730)

[Implementation Plan 6](#_Toc57918731)

[Strategy for Implementation 6](#_Toc57918732)

[Phases of rollout 6](#_Toc57918733)

[Details for levels of testing and final distribution 6](#_Toc57918734)

[Dependencies and milestones 7](#_Toc57918735)

[Deliverables 7](#_Toc57918736)

[User testing 7](#_Toc57918737)

[Evaluation Plan 7](#_Toc57918738)

[Resources and Costs 7](#_Toc57918739)

[Programming Environment 8](#_Toc57918740)

[Environment Costs 8](#_Toc57918741)

[Human Resource Requirements 8](#_Toc57918742)

[Timeline and Milestones 9](#_Toc57918743)

[Project Recommendation - Section B 10](#_Toc57918744)

[Problem Summary 10](#_Toc57918745)

[Application Benefits 10](#_Toc57918746)

[Application Description 10](#_Toc57918747)

[Data Description 10](#_Toc57918748)

[Objective and Hypotheses 11](#_Toc57918749)

[Methodology 11](#_Toc57918750)

[Funding Requirements 11](#_Toc57918751)

[Stakeholders Impact 11](#_Toc57918752)

[Data Precautions 11](#_Toc57918753)

[Developer’s Expertise 12](#_Toc57918754)

[Post-Implementation Report - Section D 13](#_Toc57918755)

[Project Purpose 13](#_Toc57918756)

[Datasets 13](#_Toc57918757)

[Data Product Code 15](#_Toc57918758)

[Hypothesis verification 16](#_Toc57918759)

[Effective Visualizations and Reporting 16](#_Toc57918760)

[Accuracy analysis 18](#_Toc57918761)

[Application Testing 19](#_Toc57918762)

[Unit Testing 19](#_Toc57918763)

[Acceptance Testing 19](#_Toc57918764)

[Usability Testing 20](#_Toc57918765)

[Application Files 20](#_Toc57918766)

[User’s Guide 21](#_Toc57918767)

[For using the application 21](#_Toc57918768)

[For viewing the source code and notebook 21](#_Toc57918769)

[Summation of Learning Experience 21](#_Toc57918770)

# Letter of Transmittal – Section A1

November 29, 2020

Norma Buck, Project Manager

WineCo

650 N South Street

Dellberg, WI 45623

Dear Norma Buck:

Every company asks itself, how will my product be received? Companies will go to the ends of the earth to make sure their investment brings a healthy and fruitful return. Fortunately, through data science, it is much easier to come to these conclusions. It also is faster and will work on all your products. Spiralytics specializes in analyzing the quality of wine, beer and spirits through its chemical composition. Recently new research has provided data for quality and how it relates to the chemical composition of wine.

We propose an application that would show the correlations of the chemical makeup to the quality of wine and provide a way to predict the quality of the wines you are developing. This will enable you to test various wines before putting the wine into mass production if it has a chance to meet your target market. We believe this will give you an edge among the competition as there will not be many surprises on how the wine will be received.

As mentioned, we specialize in analytics for beer, wine and spirits. For this application we would have a team of three overseeing development. One of the members is a senior developer, while the other two are intern developers. The development of this project would likely take around 6 months and would be broken into 5 phases. During that time our development team would work with a team of stakeholders to ensure the application meets the expectations of WineCo. The project would cost $30,000 in total, with most of it going toward developer labor. The application will be delivered by hosting it on a webserver, but we will also deliver the source code in case WineCo ever wants to incorporate it into their existing systems. We hope this is as exciting of an idea for you as it is for us and look forward to discussing this with you more.

Sincerely,

Brandon Walker

# Proposal – Section A2

## Problem Statement

The wine industry has been trying to establish objective ways to define the quality of wine for hundreds of years. While these attempts have progressed, they are still built on the pillars of subjectivity. This can add a lot of risk when trying to produce a new wine. If a wine is introduced to the market and it is considered “poor” in quality, it can have a very negative impact on that company. First, sales will be affected, because the product will be hard to distribute to retail. Second, it can damage the reputation of the vineyard, which will make it more difficult to distribute future wine releases. Lastly, it will have wasted time and resources that went into the production of that wine, which could have been used to produce a wine of higher quality. While it isn’t the goal of every vineyard to make the best wine, they are trying to achieve a quality that meets their markets demand.

The proposed application offers a more objective approach to predicting the quality of wine, by examining wines chemical composition. This will mitigate all the risks mentioned above by giving an accurate account of future wine releases. Spiralytics uses Python and its data science libraries (Pandas, NumPy, Matplotlib and Scikit Learn) to develop cutting edge web-based dashboards that predict the quality of wine. The dashboard provides data visualizations of the correlation of various chemical components and the quality of the wine, those ratings are “poor”, “normal”, and “excellent”. The application will also include a way to predict both white and red wines simply by entering the chemical components and submitting the information.

## Customer Summary

The application is designed for the vigneron of the vineyard. The vigneron will use the dashboard as a tool to ensure the all wine in development meets its targeted quality. Again, not every wine is meant to be of “excellent” quality, but rather meet the demand of the target market. While testing different variants of a new wine, the vigneron will be able to take the chemical components of each variant and cross reference it to see it predicted quality. If it meets the target it can then move to production. The dashboard is designed to user friendly, while it should not require any special skills, a user manual will be provided for training purposes. With the application being web-based, the facility where the vigneron will be testing, will need to be equipped with an internet connection.

## Existing System Analysis

The proposed application will stand alone, it will not need access or integration of any existing systems. However, if WineCo’s internal IT department wants to integrate the system, Spiralytics will have the source code available through GitHub.

## Data

The data collected for this application comes from a dataset “Wine Quality”, which was created by Paulo Cortez of the University of Minho in Portugal (<https://archive.ics.uci.edu/ml/datasets/wine+quality>). The data is broken up into two comma separated value files (.csv). The files each have the same 13 columns, type, fixed acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality.

There are 1599 records of red wine and 4898 records of white wine. The two different types will be treated independently, to ensure an accurate prediction based on type of wine. All incomplete records will be discarded, this will only be 6 records for red wine data and 28 records for white wine data. All columns will be incorporated into the model as the all have a correlation to the quality of the wine.

## Project Methodology

For the development of this application Spiralytics will us the Agile methodology. Spiralytics believes in producing quality applications thorough testing and receiving feedback on features. Below is an outline of the project:

1. Phase one:
   1. Gather all requirements for project, this include the look and feel of the dashboard as well as how the dashboard will interact with the data.
   2. Meet with key stakeholders who will be responsible for working with development team on feature releases.

*The next three phases will be cyclical as they require each feature to go through each phase.*

1. Phase two
   1. The development team will begin work through 3 sprints; those include:
      1. Clean the data and format it in a way that will make it easy to describe correlations and create the predictive model.
      2. Develop the graphs to give a better understanding of how the chemical components correlate to quality wine and the prediction model that will show what the quality of wine is based on its chemical composition.
      3. Develop the website that will feature a dashboard that will visualize graphs of the descriptive method and a from for the predictive model.
2. Phase three
   1. Test each feature for anomalies and ensure quality results of all visualizations along with the prediction model.
3. Phase four
   1. Deliver the features to the stakeholders assigned from phase one and receive feedback on the feature. If the feature meets standards of all stakeholders, move on to the next sprint, otherwise continue development of current feature until standards are met.
4. Phase five
   1. Usability testing of application.
   2. Deploy application to hosting and release final product to WineCo.

## Project Outcomes

Each phase of the project may include deliverables, below are what to expect from those phases:

* Phase one:
  + During meeting with stakeholders, a wireframe of the application, showing navigation and features of the application will be provided. A schedule will also be provided to the stakeholders of each sprint.
* Phase four
  + During the review of the feature, the feature will available to be reviewed by the stakeholders. A copy of the tests and results of those test will also be available for review.
* Phase five:
  + Final product of application will be available for use as well as the user manual and training documents.
  + Source code will also be available if WineCo would like to have internal team incorporate the application into existing systems.

## Implementation Plan

Below is the implementation plan for the applications. With Spiralytics using Agile, development and the rollout will be intertwined. The plan includes strategy for implementation, phases of the rollout, details of testing and final distribution, dependencies and milestones, deliverables and user testing.

### Strategy for Implementation

With the application being web-based and a standalone product, once the application is completed it will be made available to WineCo. The source code will also be available through GitHub if WineCo decides to integrate it into their existing systems. The user manual and training documents will also be made available for WineCo, so they are able to train all current and future personnel.

### Phases of rollout

The rollout will happen iteratively one feature at a time. As mentioned in above we will have 3 sprints that will release a feature. At the end of each sprint the feature will be reviewed and tested with the stakeholders. Upon signoff from the stakeholders we will move onto the next sprint. However, if the stakeholders do not approve and change need to be made the development team will cycle back through that features development phase and set a new review with the stakeholders. After each feature is completed the final application will be released.

### Details for levels of testing and final distribution

As mentioned above each feature will go though its own testing. Testing will throughout the sprints will include unit testing. Upon completion of each feature and the application, usability and acceptance testing will be performed using sample input data.

### Dependencies and milestones

Each Sprint will serve as a milestone with the final release marking the completion of the project.

### Deliverables

As mentioned in the project outcomes, the deliverables are as followed (listed in chronological order):

* Wireframe of application
* Schedule of the development Lifecyle (subject to change based on feature results)
* Test results for each feature
* User Manual
* Training Documents
* Final Application

### User testing

During the final phase of development WineCo will be asked to have the stakeholders perform an acceptance test on the application. Spiralytics will perform its own usability testing as well. This will ensure ultimate quality and the desire outcome of both parties.

## Evaluation Plan

Agile methodology builds testing into the development lifecycle. Throughout all phases of the project we will be testing to ensure quality and accuracy. It is important that when gathering the requirements all expectations are voiced, this will ensure a smooth development life cycle. During development the developers will use unit testing to test the code and data to ensure that each feature is performing as intended. The review of each feature will act as acceptance testing, as we will review each feature with the assigned stakeholders. After development, Spiralytics will perform usability testing simultaneously while WineCo is also performing acceptance testing.

With the application being a relatively small project, Spiralytics feels the aforementioned testing is more than sufficient to meet the standards and expectations of WineCo. The data being used is public and will not require any regulatory procedures or policies.

Spiralytics will use the targeted range of 75% to 95% accuracy of the predictive model, client satisfaction of the interface, and visualizations of all chemical components to correlation of wine quality as a measure of success. All three measures must be attained in order for the application to be deemed successful.

## Resources and Costs

Below are all the resources and cost the project will require. These include the programming environment, environment costs and human resource requirements.

Spiralytics estimates the Total Cost will be $30,000 for the application.

### Programming Environment

The programming environment will include Python, Pandas, Matplotlib, NumPy, Scikit Learn, Jupyter Notebooks. Heroku will be used to host the website and dashboard which will have a cost associated. With the application being web-based there will be no concern whether WineCo is using MacOS or Windows based computers. If each computer has an up to date browser and an internet connection, the application will run.

Total Cost for Programming Environment: $2,750

### Environment Costs

Other than a computer, internet connection and up to date browser, there is no additional hardware requirements. It has been established that WineCo already has all three of these components.

Total Cost for Environment: $0

### Human Resource Requirements

The cost of development will also rely on the amount of developer hours. Spiralytics estimates that the total amount of hours needed to develop the application will be 250 hours. Our developer’s hourly rate is $110 per hour. This is subject to change based on change requests of requirements or change to features.

Total Cost for Human Resources: 27,500

## Timeline and Milestones

Below is a table featuring the timeline of the project and each milestone associated with the development of the application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event** | **Start Date** | **End Date** | **Duration (Days)** | **Dependencies** | **Resources** |
| **Phase One** | **4-Jan-21** | **8-Jan-21** | **5** |  |  |
| Requirement Gathering | 4-Jan-21 | 6-Jan-21 | 3 | N/A | Owners and Management |
| Stakeholder meeting | 7-Jan-21 | 8-Jan-21 | 2 | N/A | Stakeholders and developers |
| **Phase two and three** | 11-Jan-21 | 17-Feb-21 | 24 | Phase 1 |  |
| Sprint 1 (test and development) | 11-Jan-21 | 20-Jan-21 | 8 | Phase 1 | Developers |
| Sprint 2 (test and development) | 25-Jan-21 | 3-Feb-21 | 8 | Feature review 1 | Developers |
| Sprint 3 (test and development) | 8-Feb-21 | 17-Feb-21 | 8 | Feature review 2 | Developers |
| **Phase 4** | **21-Jan-21** | **19-Feb-21** | **6** | **Phase 2 & 3** |  |
| Feature review 1 | 21-Jan-20 | 22-Jan-21 | 2 | Sprint 1 | Stakeholders and developers |
| Feature review 2 | 4-Feb-21 | 5-Feb-21 | 2 | Sprint 2 | Stakeholders and developers |
| Feature review 3 | 18-Feb-21 | 19-Feb-21 | 2 | Sprint 3 | Stakeholders and developers |
| **Phase 5** | **22-Feb-21** | **1-Mar-21** | **6** | **Phase 4** |  |
| Usability Testing | 22-Feb-21 | 26-Feb-21 | 5 | Feature review 3 | Developers |
| Acceptance Testing | 22-Feb-21 | 26-Feb-21 | 5 | Feature review 3 | Stakeholders |
| Deployment and Release | 1-Mar-21 | 1-Mar-21 | 1 | Usability and Acceptance Testing |  |

# Project Recommendation - Section B

## Problem Summary

There are several types of wine and there are only a few components needed to make wine. Those being your fruit, usually grapes, water and yeast. Unfortunately, combining these three ingredients will not necessarily produce quality wine. While there are only three ingredients, there are several chemical components, and those components can tell us a lot about the quality of the wine. There thousands of records of wine that show the overall quality of that wine and what the chemical composition is. This information can be analyzed to predict what the quality of a new wine would be based on these specific components. An application could be developed to process the information and train a model that would give incredible insight into future wines the company is interested in producing. This would enable production to be confident in an outcome of a newly created wine prior to mass production.

## Application Benefits

The application will aim to describe the correlation between components as well as predict the quality of the wine based on those components. There will be multiple graphs depicting the components relationship to the quality of the wine. The prediction portion will use the data to predict if a wine will be of “poor”, “normal”, or “excellent” quality. There will a snapshot of each wine type and give the ability to analyze how components correlate to the quality of the wine and given the components what the quality should be.

## Application Description

Python will be used as the primary programming language to develop the application. Python has all the tools needed to analyze, describe and predict using the data we have available. Some of the tools include pandas, NumPy, Scikit learn. Miniconda will be used to create our environment and it will incorporate a Jupyter Notebook. The application will be in the form of a dashboard hosted through a website. All the data will be shown using various graphs as mentioned above.

## Data Description

The data is accessed using a comma separated file, aka a .csv file, in it contains 6,500 different examples of wine and its 13 characteristics. Of those characteristics there are 11 different chemical components. The remaining two other descriptions which are the type of wine, in this case red or white, and the quality, which is graded on a scale between zero and nine (the values only range between three and nine). The chemical components are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates and alcohol.

The dataset comes from the Portuguese “Vinho Verde” wine. The data includes both white and red wine varieties and does not give additional details on the type of wine to protect privacy. There are more wines that fit the “normal” category than “excellent” or “poor.

## Objective and Hypotheses

The primary objective of the application is to predict the quality of a wine based on its chemical makeup. The hypothesis is that if the chemical composition of x wine has y variables it should have a quality of z. This will give incite on how the wine will be received by the market prior to it going into mass production. Of course, there is always subjectivity to how things taste, this will be a fail-safe on a wine being of “poor” quality. The application will attempt to do this by employing two methods, one descriptive and the other predictive. Both will offer insights on how the chemical composition will affect the quality.

## Methodology

The development methodology for this project will use Agile development. As each feature is created, testing will be performed and the feature will be released to gather feedback. This will ensure that WineCo is receiving the product according to their expectations and that it is of the highest quality.

The project will be broken down into three sprints. The first sprint will be to clean the data and format it in a way that will make it easy to describe correlations and create the predictive model. The second sprint will be the development of graphs to give a better understanding of how the chemical components correlate to quality wine and the prediction model that will show what the quality of wine is based on its chemical composition. The last sprint will be to develop the website that will feature the dashboard and visualize the graphs and predictive model.

Throughout the development cycle feedback from WineCo will be critical in assuring that the model and correlations are meeting what is expectations. This will ensure the success of predicting the quality of wine and investment WineCo puts into developing their future wine releases.

## Funding Requirements

The project costs primarily come from the hosting of the application and the development of the application. The software tools used to build the application are all open source and will not add to the cost of the project. As discussed in the letter of transmittal the total cost of the application will be $30,000. This includes 250 hours of development at $110 per hour and $2,750 for hosting the application.

## Stakeholders Impact

It is important that not only WineCo’s stakeholders, but Spiralytics is stakeholders are fully vested in the project and its success. The stakeholders of both parties are in agreement that for the project to be successful the accuracy of the prediction model has to fall with 75-95%. This ensures that all wines produced by WineCo will fall within their targeted quality and have a successful launch. Other considerations include that project is completed on time and that the model ‘s usability is in line with what is expected. This will allow WineCo to continue with production.

## Data Precautions

With the data having all data removed that could compromise the privacy of the creators of the wine, there are no precautions that need to take place to protect the data. There information regarding person(s) and because of that there is no healthcare, education or payment information. This will protect WineCo and Spiralytics from any HIPPA, FERPA and PCI DSS violation concerns. The data is from a public study and will not need to be protected from anyone inside or outside WineCo and Spiralytics, however the analytics of the data will be kept private as the predictive model will be designed for WineCo only.

## Developer’s Expertise

The team that will be dedicated to this project consists of two interns that are attending Western Governor’s University, one of which is majoring in Computer Science and the other in Software Engineering. Both have experience in Python and some data science experience. They will be working with a senior developer who has worked on similar projects and will help guide them through any unknowns in the development process. While this will be developed mostly by interns, both have shown the ability to move through projects faster than projected. The senior developer will also be testing the code and features as they move through each of the three assigned sprints.

# Post-Implementation Report - Section D

## Project Purpose

Since the existence of wine, the industry has been trying to establish objective ways of defining its quality. In many ways progress has been, but most of the methods of rating a wines quality hinges on subjectivity. This of course adds significant risk for the vineyards producing the wine. If a new wine is introduced and its target quality does meet the market’s expectations, it can have a very negative effect on the company or vineyard. Those impacts can include a damaged relationship with their distributors, a damaged reputation of the company and the wasted time and resources that went into the development of the wine. While not every wine is meant to be of the highest quality, it is important the company meets the expectations of the markets demand.

The purpose of this application was to add objectivity into the mix of rating wine. While it wasn’t meant to be an “end-all” solution. The target was to make a more minimal investment when developing new wine. The vineyard can now see if the wines composition meets the quality target before moving to next phase of production.

The application was developed using Python and its data science libraries (Pandas, NumPy, Matplotlib and Scikit Learn). The front-end was developed using HTML/CSS and Flask (a python framework). This allowed the vigneron to login using a computer with an internet connection. In addition to the web application, the source code was also made available in case there was any need to integrate the application into existing systems.

## Datasets

The data collected for this project was from a data set “Wine Quality”. This dataset was created by Paulo Cortez of the University of Minho in Portugal (<https://archive.ics.uci.edu/ml/datasets/wine+quality>). The comma separated value file (.csv) was split into two sets representing both types of wine. The files have the same 12 columns, fixed acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality.

The red wine csv file has 1599 records and the white wine file has 4898 records. To ensure accurate prediction the two sets have their own models. All incomplete data was discarded from both sets. All columns were kept except for the “quality” column which gave values 3-9, was changed to the “target” column and given a scale of 1-3. This gave the ability for the sets to use multi-nominal classification.

The following figures give examples of the two data sets and how they were cleaned up.

Fig 1: Datasets prior to transformation and cleanup

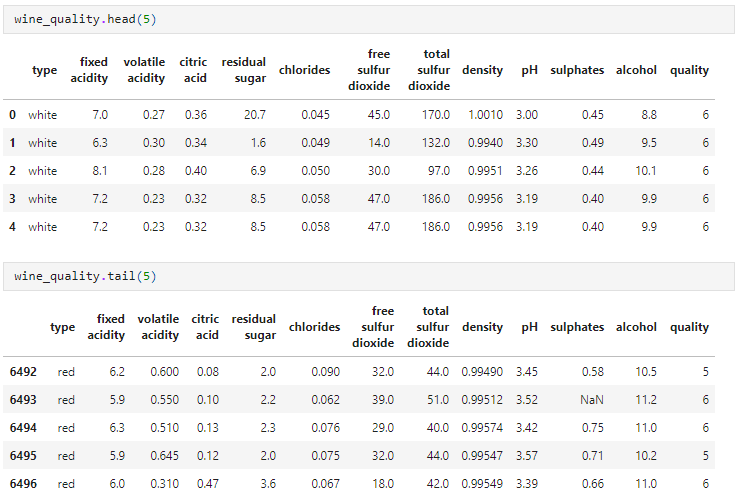
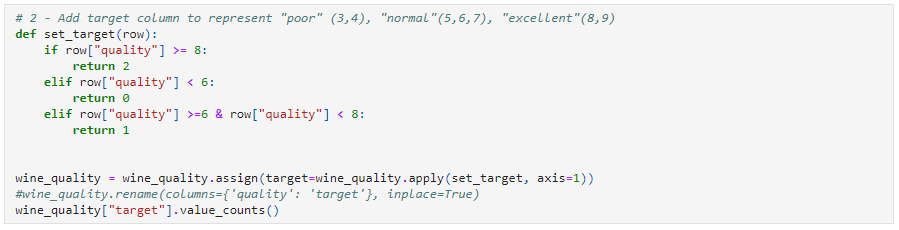


Fig 2: Example of code used to transform and cleanup the two datasets



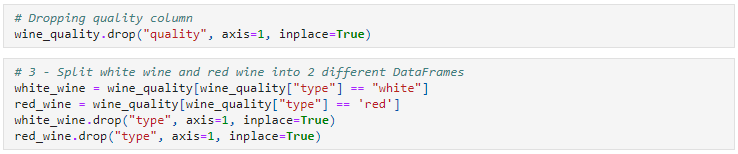
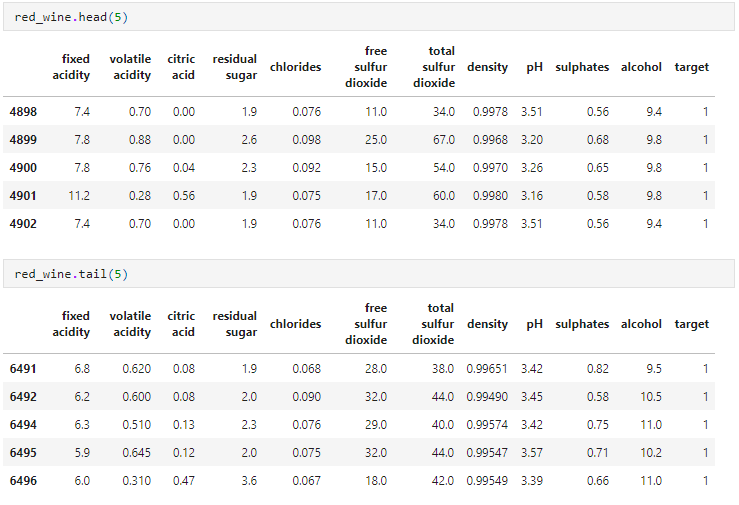


Fig 3: Red Wine dataset after the transformation and cleanup was completed

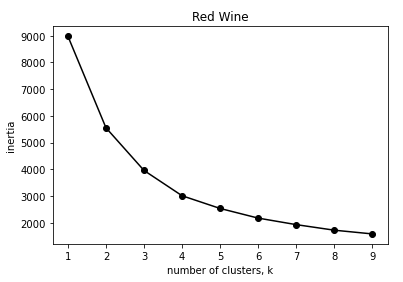


## Data Product Code

As mentioned above, in order to analyze the data and give accurate predictions the data set had to be split into two datasets, one for red wine and one for white wine. Prior to separating the two datasets they were cleaned and transformed. Once the two sets were separated, Principle Component Analysis was run to reduce the dimensionality of the data. This took the two data set from having 11 features to have just 2 components.

After PCA was completed, next was to show the correlation of these components to the quality of the wine. A descriptive method, K-Means Clustering, which clusters the data based on an inputted number of clusters. Intuitively because there are three classes this was the first thought. However, the Elbow Method was used to discover how many clusters would be used. After discovering the three clusters were indeed the amount needed from the Elbow Method, the algorithm was run on both sets.

Figure 4: Elbow Method used on Red Wine dataset



A second method, Decision Trees, were used to help predict the quality of wine. With the predictive method cross validation was used to ensure the model used the best parameters, ensuring an accurate prediction. All source code available at <https://mybinder.org/v2/gh/branlewalk/C964/HEAD?filepath=wine-quality-spiralytics.ipynb>

## Hypothesis verification

The hypothesis for this project was, if the chemical composition of x wine is y, the quality would be z. With the accuracy being over 75% for both the white and red wine models, the hypothesis appears to be true. The primary objective was to predict the quality of wine based on its chemical composition and the application performs this for both sets. It also gives incite on the influence the different chemical components have on the quality.

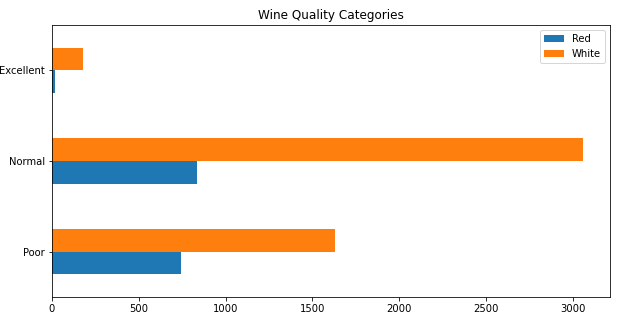
With the data having an overwhelming amount of its examples be of normal quality, this does make lean toward predicting “normal” quality and because most of the data is of that targeted quality it very likely to predict “normal” quality. The data may be this way because of the data samples provided, or it could be an accurate depiction of reality and that being that most wine is of “normal” quality.

## Effective Visualizations and Reporting

There are three visualizations, first, the amount of each quality node (“poor”, “normal”, “excellent”) for both datasets, followed by Principle Component Analysis on each set, and finally, K-Mean Clustering on both sets.

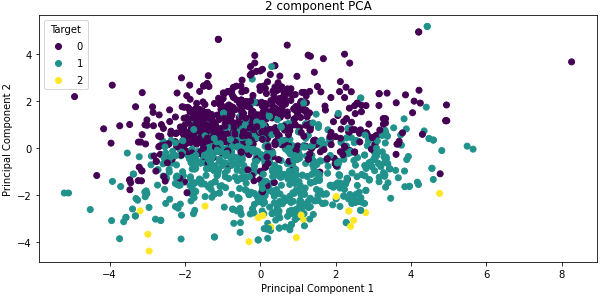
The first graph (Figure 5) give the idea of how many of each type of quality are going to be in the following graphs. Note that the number of “excellent” samples is limited and may make our prediction model slightly skewed when predicting “excellent” quality.

Figure 5: Quality Nodes for both sets



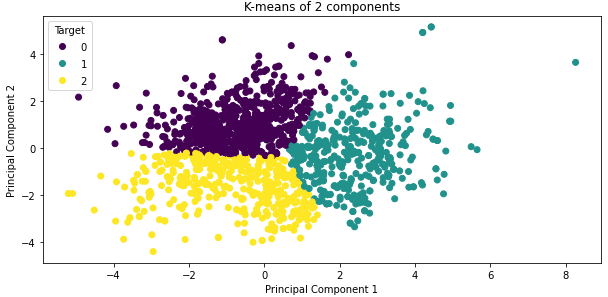
The second graph (figure 6) shows the Principle Component Analysis of Red Wine. It is very clear that “excellent” quality lies at the bottom, “Normal” in the middle and “poor” quality is near the top. These results show that there is clear distinction based on the components, specifically PC2, of what makes a particular wine “poor”, “normal” or “excellent quality. Another graph showing white wine gives a very similar result and is available on the application.

Figure 6: Red Wine Principle Component Analysis



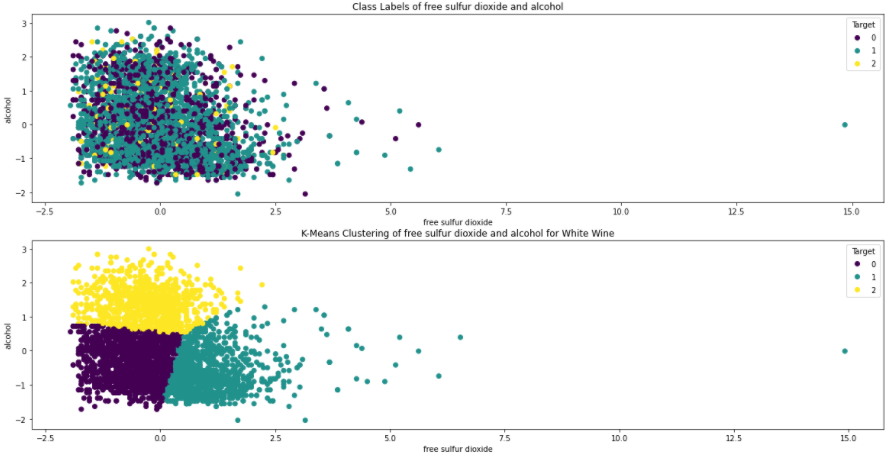
The last graph (figure 7) shows the K-Means clusters. The reason we wanted to use K-Means is to give a better picture of the quality based on the components. However, after many iterations of adjusting the model, K-Means is not able to give us an additional understanding of how the components effect the quality of wine. This is caused by the data clusters not having any distinct shapes.

Figure 7: Red Wine K-Means Clustering



It should be noted that prior to incorporating PCA, another forms a feature selection was used. It showed the top, most influential features for the two data sets. Among red wine the top four features were alcohol, sulphates, total sulfur dioxides, and volatile acidity. For white wine the top 4 features were alcohol, volatile acidity, density and free sulfur dioxide. Three graphs were used in order to compare the labels (quality) to the K-Means clustering prediction. The same results were found here as they were in the assessment that followed the dimensional reduction using PCA. The failure to find good cluster analysis using these features is what lead to the use of PCA and it was able to give a clear understanding that the components do in fact show a strong connection between the wine’s chemical composition and the quality.

Figure 9: Example of failed K-Means analysis on Red wine using 2 features

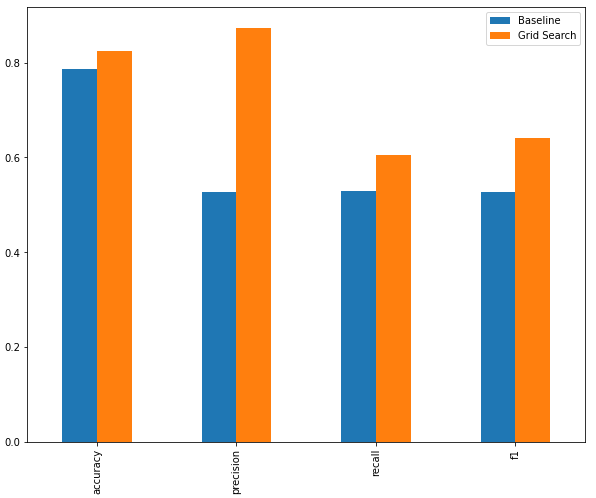


## Accuracy analysis

Two methods were used to score the prediction model, one was using the “score” method of our model and the second was to use cross validation to discover if there are better parameters for the model. The results showed that the cross validation results gave the best score in every metric.

For Red Wine the accuracy from the cross validation parameters was 78.68% (figure 8) and for White Wine the results were 81.82%.

Figure 8: Red Wine Score (Baseline vs Cross Validation)



## Application Testing

For testing Spiralytics primarily employed three methods, unit testing, acceptance testing, and usability testing. Unit testing was used throughout the development of the backend (the Jupyter Notebook). While acceptance testing and usability testing was used after the development of each feature and a more in-depth version was performed once the application was completed

### Unit Testing

Developing the descriptive method and prediction model was done in a Jupyter Notebook. Unit testing was used throughout the process of cleaning the data, transforming the data, creating the visualization, and creating models to ensure the data and models were accurate.

### Acceptance Testing

Stakeholders were asked to review the features as they were released. They also were given the final application to review against the requirements. After finding that all requirements had all been met, they also tested the application as an end user. They inputted data from samples of WineCo’s wine and reported any bugs in the application.

### Usability Testing

The developers tested out the application, inputting sample data provided from the test sets, for ease of access the samples were put on the about page of the application. They were also asked to put in obscure data results to see if there were any errors that occurred.

## Application Files

The source code for project is located on GitHub (<https://github.com/branlewalk/c964>). The application is available at <https://spiralytics-wine-app.herokuapp.com>. Below is a breakdown of the file hierarchy:

|  |  |  |  |
| --- | --- | --- | --- |
| **Main folder/files** | **Subfolders** | **Files** | **Notes** |
| static |  |  |  |
|  | images |  | Pictures for the dashboard |
|  |  | amount.png |  |
|  |  | Red-wine-pca.png |  |
|  |  | sample\_red copy.png |  |
|  |  | sample\_white copy.png |  |
|  |  | White-wine-pca.png |  |
|  | js |  |  |
|  |  | …js and bootstrap files needed for project |  |
|  | styles |  |  |
|  |  | …bootstrap files needed for project |  |
|  |  | style.css | Styling for the application |
|  | templates |  |  |
|  |  | about.html | Page for “About Spiralytics” and how to use the application |
|  |  | dashboard.html | The “dashboard” page and where the dashboard is located |
|  |  | index.html | The landing page for the application |
|  |  | login.html | Login page for the application |
|  |  | predict.html | The page where the predict form is located |
| red\_wine\_model.joblib |  |  | The model for red wine predictions |
| white\_wine\_model.joblib |  |  | The model for white wine predictions |
| wine\_quality\_spiralytics.ipynb |  |  | The notebook for the data cleaning and models |
| wine\_quality.csv |  |  | The raw data set |
| wine\_app.py |  |  | Application |
| Procfile |  |  | file needed for hosting site |
| requirements.txt |  |  | Requirements for application |
| runtime.txt |  |  | Python version used |

The Jupyter Notebook is also available at: <https://mybinder.org/v2/gh/branlewalk/C964/HEAD?filepath=wine-quality-spiralytics.ipynb>

## User’s Guide

With Spiralytics releasing the source code as well as providing the application, see the user guide below on how to access either the source code or the application.

### For using the application

1. Visit the website at <https://spiralytics-wine-app.herokuapp.com>
2. Log-in using the following credentials:
   * Username: WineCo
   * Password: Red&White
3. This will navigate to the dashboard
   * There are 2 versions of the same two graphs, notes are present to breaks down the results. There is also a graph showing the amount of quality nodes that are present in each graph.
4. Click on the “Predict” Tab to navigate to the prediction form
   * In the predict form, first select the wine type and then enter the chemical components metrics. Once the metrics have been entered, press “submit”. The box labeled “results” will show which quality classification the wine is.
5. Click on the “About” Tab to read about the application
   * The about page has some sample data to test the model. Directions on how to use the site are also on the about page.

### For viewing the source code and notebook

For downloading the source code using Git, type the following commands:

1. “git clone https://github.com/branlewalk/c964”

To view the source code of the Jupyter Notebook:

1. Visit <https://mybinder.org/v2/gh/branlewalk/C964/HEAD?filepath=wine-quality-spiralytics.ipynb>

## Summation of Learning Experience

With this project, the application portion came fairly easy. A combination of WGU’s courses that used Python for the development of the projects and my own personal projects made developing this application very exciting. There were still many resources I needed to use in order to complete this, but over the course of my time with WGU I have developed a certain comfortability with not knowing how to do something, but trusting my ability to figure it out. Having a baseline knowledge of programming fundamentals and experience from past projects in the language all assisted. This made the project very enjoyable. The data science portion was the most challenging task. While I have some analytics experience from my current career, I had never used any of these methods, or understood the concepts. I enrolled in a Udemy course to help me better understand these concepts. After completing that course and finishing this project I feel more comfortable with the subject matter. As for the writing, Technical Communication setup this very nicely. For me, coming into WGU, writing was my biggest obstacle. Over the various project related writing courses, I no longer feel that way.

Before my college experience I worked on several projects, whether it was home improvement or hiring a new team for a store opening at work. I always approached projects as a 1 step process, trying to accomplish everything all at once and fast. This was something that worked really well for me to a point. Through my time at WGU I have learned that an iterative approach is not only more effective, but more enjoyable. I have always enjoyed learning and seeking knowledge about new subjects, I have to many hobbies to count. Now having a better strategy on how to learn, where I am enjoying the process and delivering on better outcomes, make the lifelong journey less overwhelming and more enjoyable.