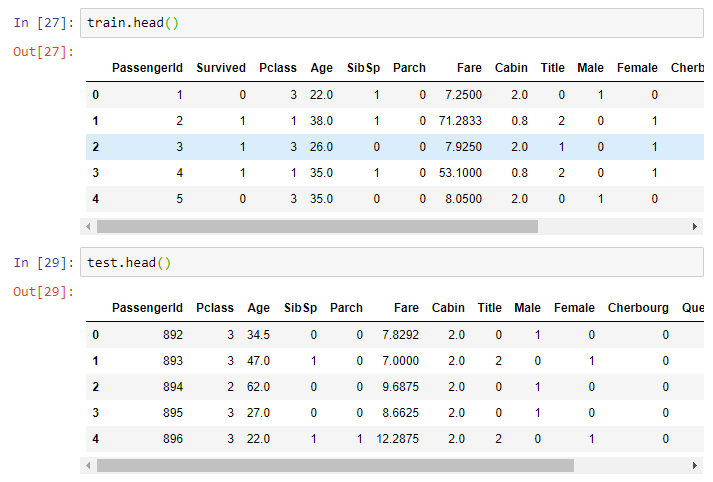
**AI 2 Specifications Document: Aadam Ali – 250911006**

Specification 1:

The Kaggle problem we are tasked with is called the Titanic problem. During the titanic tragedy many people died, however many survived. Since some people were given priority due to either status, age, gender, or even just sheer luck, we must write an algorithm that can accurately determine which passengers survived. There is a significant amount of data given on many passengers. Each passenger has 9 parameters which identify their status and characteristics. Using their parameters, we must compare and train the data in order to figure out which passenger is/was most likely to survive the titanic with the least amount of error. We are to use two models, the perceptron model and the Naïve Bayes model, in order to predict the data.

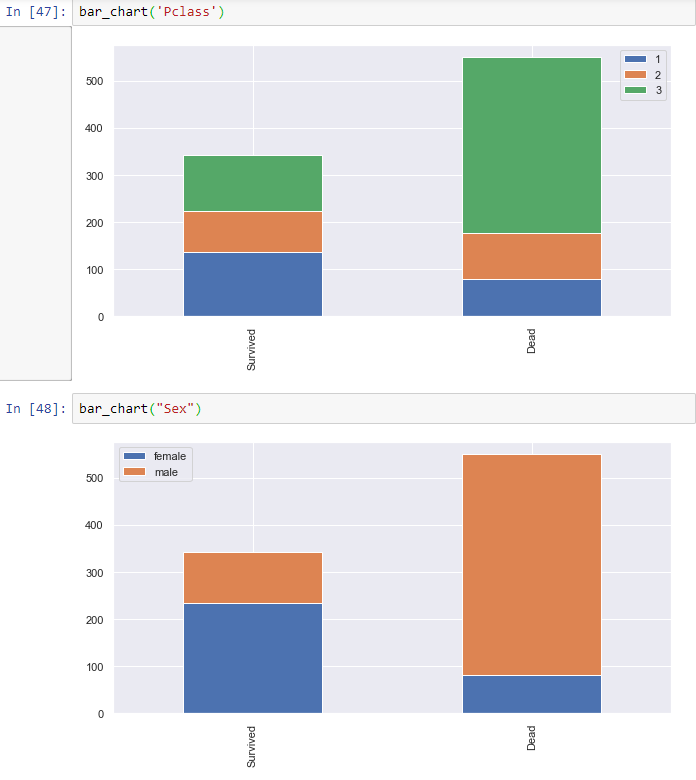
Specification 2:

Using pandas, I created 2 variables which hold the parsed dataset in which I can manipulate anyway I see fit called train and test. The first thing I want to see are the values that the features can have. To do this I use the head() function to show the first 5 rows of the data. After this I use shape to see the amount of entries that the data has which gives me a tuple of data showing how many features I have and number of rows.

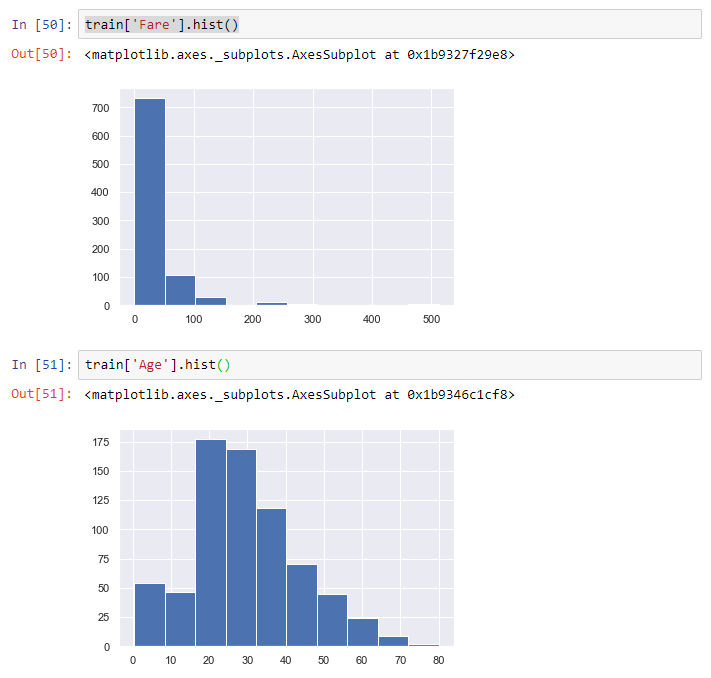


Next, I use info to see that all data has values. In this case some parameters has less data than others. This leads me to believe that some can be cleaned off or even eliminated. In this case the parameters, “Age”, and “Cabin” have null entries. Now we will pick a feature/ parameter of the data and look at it a little more closely. In this I would want to pick data that has no null values. In order to check this I use the .info() to see which of data contains null objects.

After this I visualize the data using a bar chart and I chart the features against survived and dead because that is what we want to predict. I pick 3 and the ones I think are import to see. I chose, ‘Embarked’, ‘Pclass’, and ‘Sex’. I think that all of these are very important when predicting whether a passenger survived or not. These charts allow me to see the feature correlation between survival and the features that I choose.



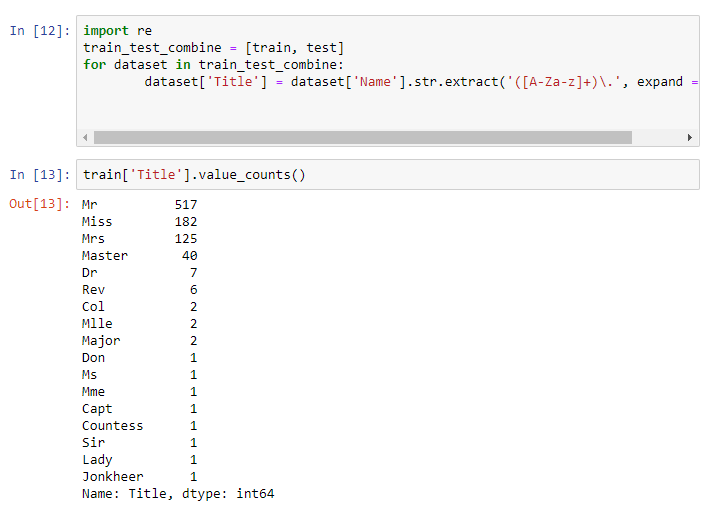
After this I want to see what kind of values I am dealing with so I use a histogram to show the fares and the ages that I am going to be looking at.



This clearly shows that most people paid around $700 for a ticket and most people are between the ages of 20 to 30 years of age.

**Specifications 3**

Next, I am going to use feature engineering to utilize the most important aspects of my data in order to get the best accuracy. My first goal is to make all the data sensible numerical values because the models I am using works best with numerical value. The first thing that I want to change is the name so I use the the titles of everyone in order to extract information from unique names. For example someone who has the title of “Mrs” most likely has an SibSp greater than 0 and families have a higher chance of survival. So, I extract the titles using a regular expression and get the results:

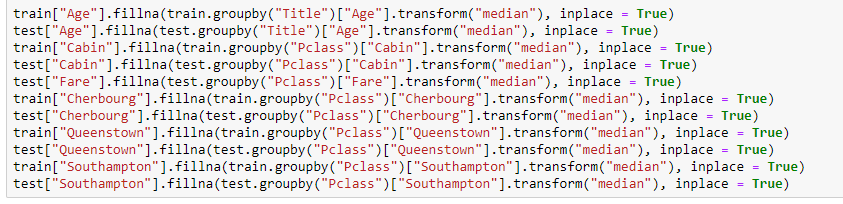


After this I need to map the data and turn each of these into numerical values. I used values 0 to 3. Next I split, “Sex”, into 2 features and, “Embarked”, into 3 features. All the mapping of these are done with 0 and 1.



I also scaled the Cabin feature in order to get rid of cabins that have letters in it. Each Cabin is labelled with a certain letter so I turned that letter into a decimal number that is scaled properly to work with models.

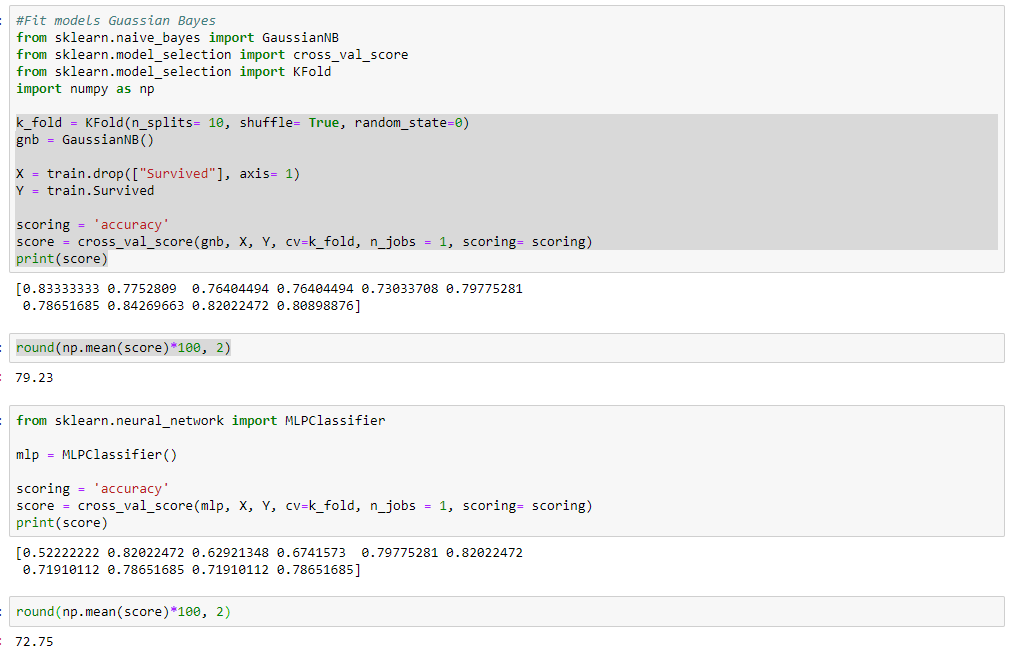
Next, I use a data cleaning method in order to get rid of the null values in the data. So I did this:



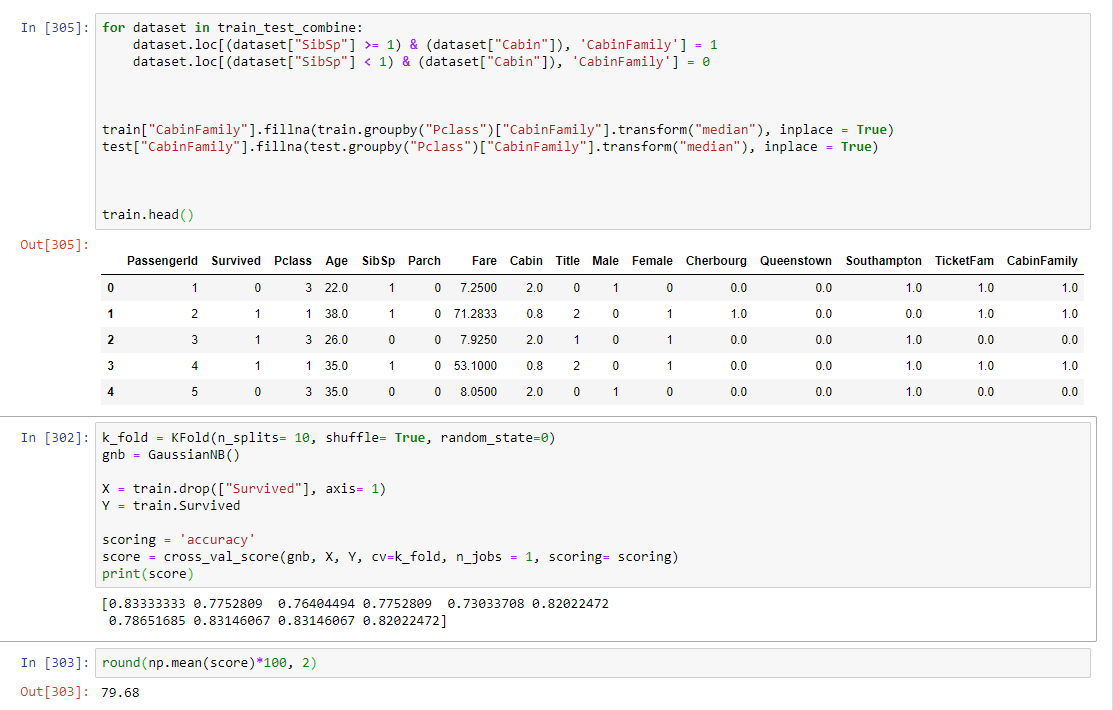
After this I dropped the ones I did not need or want anymore such as, “Sex”, “Name”, “Cabin” and “Ticket”. Ticket was dropped because I related tickets to family and created a new feature with people who bought tickets and had family aboard. This increased my accuracy I saw.

**Specification 4**

Next I used two models the Naïve Bayes and Perceptron in order to cross validate.



As we can see from the screenshot the Gaussian Naïve Bayes was more accurate therefor this will be used to submit to Kaggle. I strongly believed I could fine tune more than I already have so I added a new feature which relates the number of family members and the cabins. If you have a lot of family then your cabin has a higher chance to survive as I found out. Have other graphs showing information that I did not use however is there because I was experimenting with different features. I ran out of time to add more features.



As we can see by fine tuning the features the accuracy raised from what is above.

**Specification 5**

Now I have submitted to Kaggle by generating a csv with the code and received a leaderboard ranking. Below shows how I did this.



