1. Business Understanding
   1. Objective

Predict the survival on the Titanic

* 1. Description

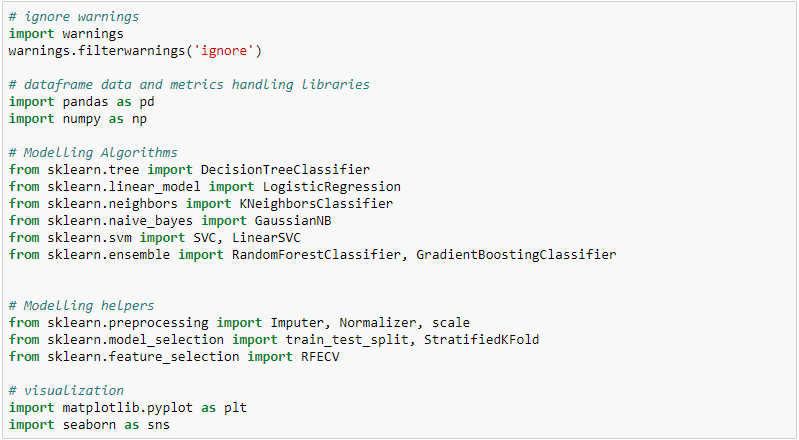
The sinking of the Titanic is one of the most infamous shipwrecks in history. On Apr 15,1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.

One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this project, I will complete the analysis of what sorts of people were more likely to survive. In particular, I will apply machine learning to predict which passengers survived the tragedy.

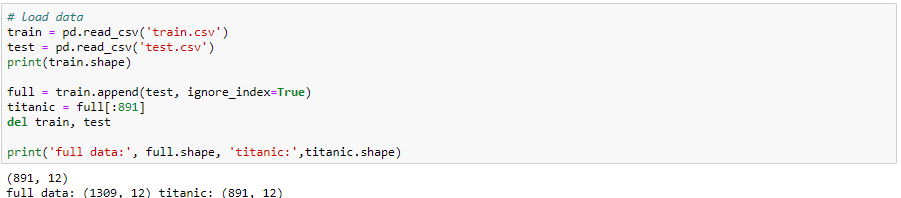
1. Data Understanding
   1. Import library

First of all, I need import python libraries containing the necessary functionality will need.



* 1. Load data

Next, load train.csv and test.csv dataset, append train and test dataset as full dataset. Have a peek at the dataset size.



* 1. Statistical summaries and visualization

To understand the data, I am going to consider some key facts about various variables including their relationship with the target variables, i.e. survived.

Here is the variable descriptions:

Age: passenger’s age

Cabin: cabin

Embarked: port of embarkation

Fare: fare

Name: passenger’s name

Parch: number of parents/children aboard

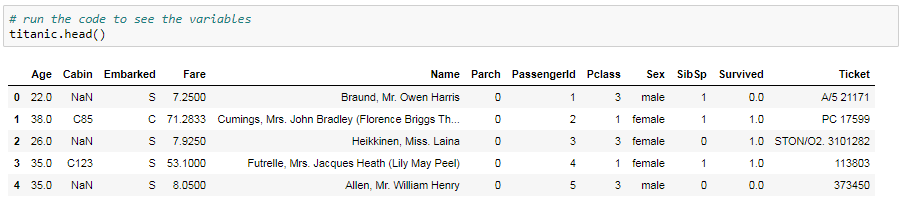
Pclass: passenger’s class

Sex: passenger’s sex

SibSp: number of sibling/spouses aboard

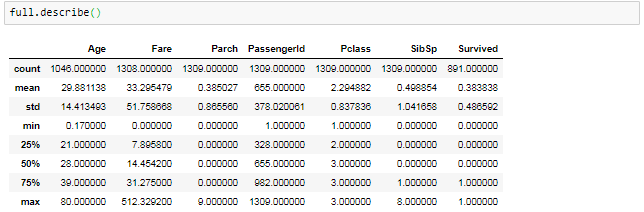
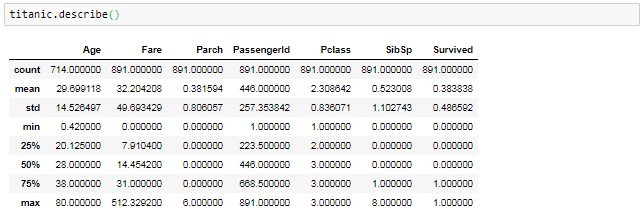
Survived: survived (0) or died (1)

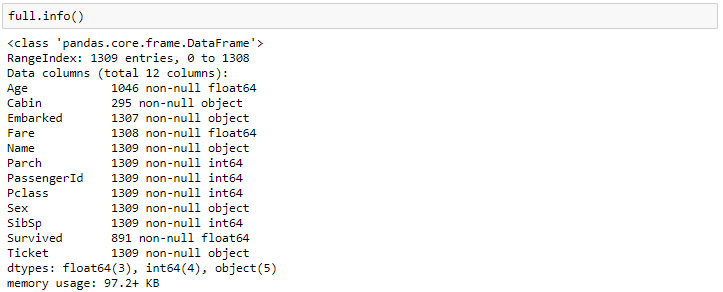
Ticket: ticket number



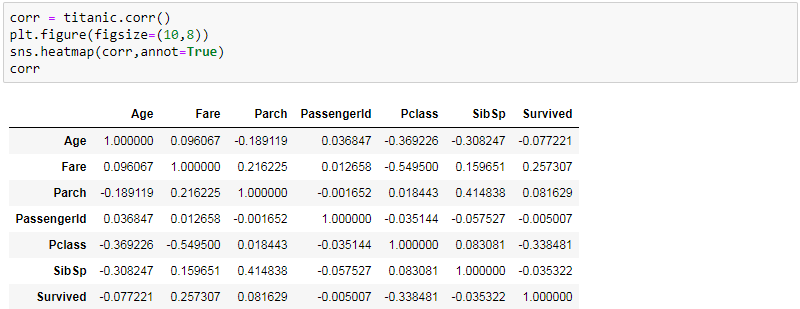
2.3.1 Investigating numerical variables

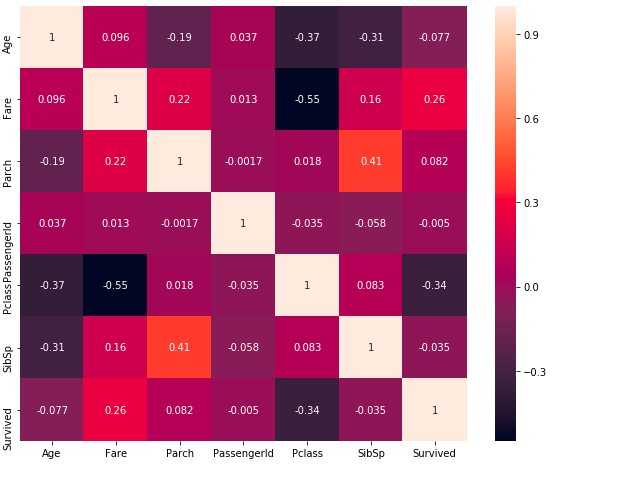
There are 2 kinds of variables, one is numerical variable and the other is categorical variable. Numerical variable is one with values of integers or real numbers, while a categorical variable is a variable that can take on one of a limited , and usually fixed number of possible values.





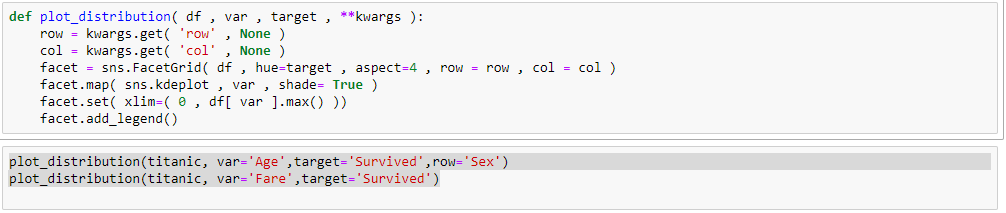
Next, a heat map of correlation may give us an understanding of which variables are important, and the relationship between each variables.

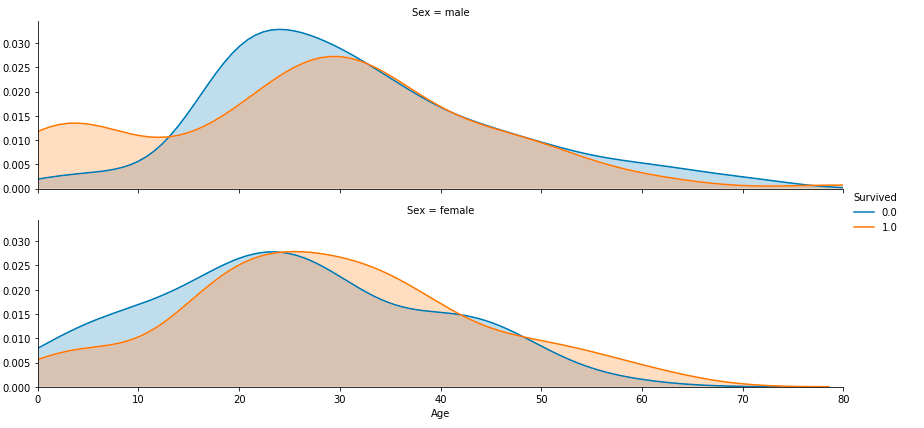


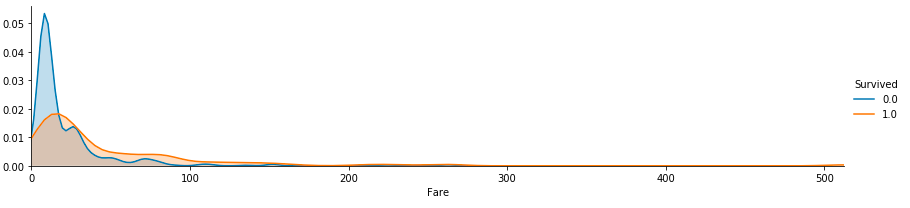


Then, l further explore the relationship between the features and survival of passengers.

Consider the graphs below, differences between survival for different values is what will be used to separate the target variable (survival in this case) in the model. If the two lines had been about the same, then it would not be a good variable for our predictive model.

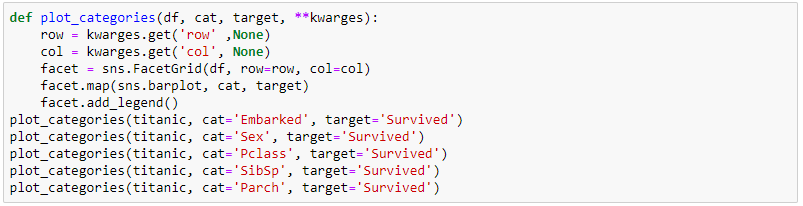


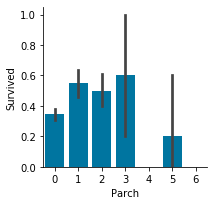
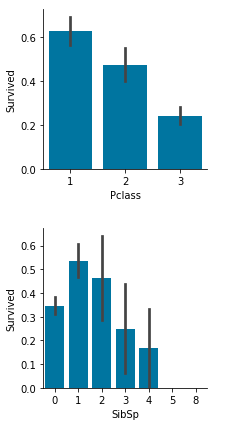
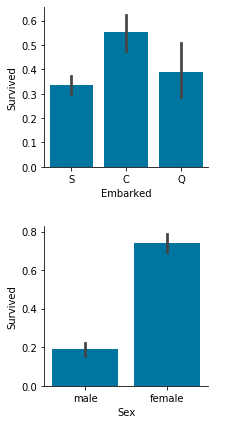




2.3.2 Investigating categorical variables

I also took a look at categorical variables like Embarked, Sex and their relationship with survival.

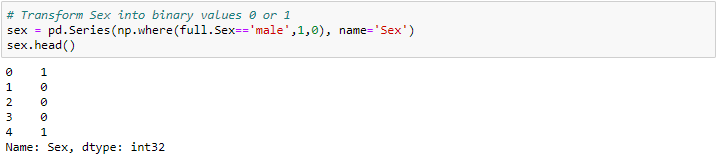


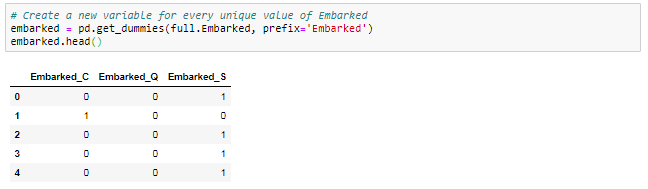


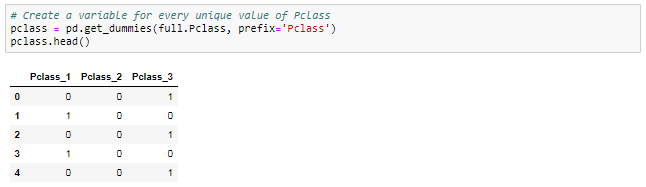
1. Data Preparation
   1. Categorical variables need to be transformed to numerical variables

The variables Embarked, Sex, Pclass are treated as categorical variables. Some of our model algorithms can only handle numeric values, so we need to create a new variable (dummy variable) for every unique value of the categorical variables.

For variable Embarked and Pclass, it will have a value 1 if the row has a particular value and a value 0 if not. Sex is a dichotomy, so it will be encoded as one binary variable (0 or 1).



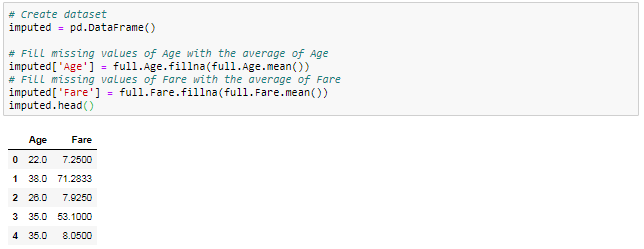




* 1. Fill missing values in variables

Most machine learning algorithms require all variables to have values in order to use it for training the model.

From 2.3.1 I found that for numerical variables, only Age and Fare variable has missing value. The simplest method is to fill missing values with the average of the variable across all observations in the training set.



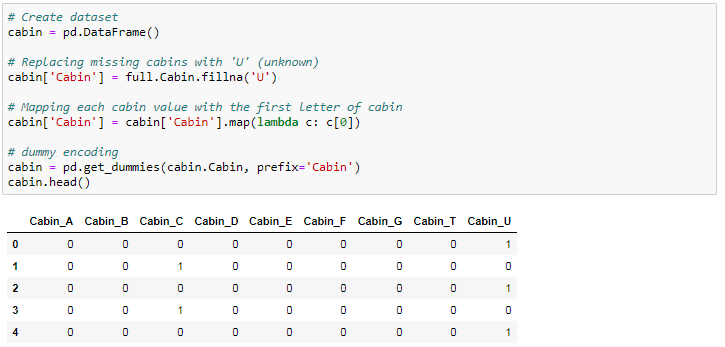
* 1. Feature engineering – creating new variables
     1. Extract titles from passenger names

If take a look at Name column, I can see there is a title in it. In that case, we might introduce an additional information about the social status by simply parsing the name and extracting the title and converting to a binary variable. Titles may have an influence on survival probability.



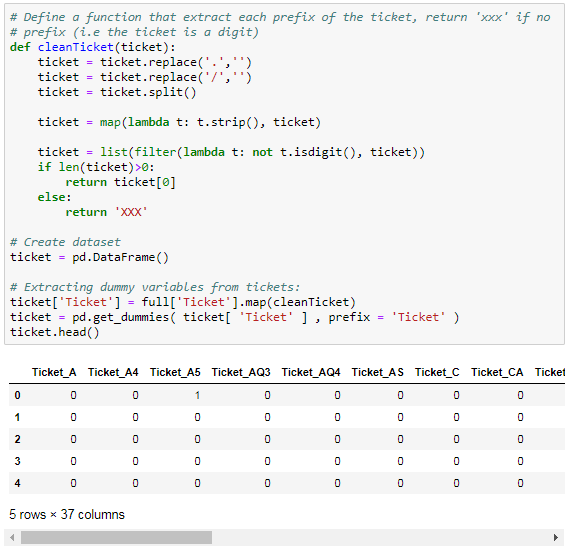
* + 1. Extract cabin category information from Cabin number

From 2.3.1, I can see out of 1309 cells, only 295 cells have cabin information, that means 77.4% data is missing for cabin variable. Due to high volume of missing data, one method is get rid of this column, another method is replacing the missing value as ‘U’ (means unknown) and extract cabin with cabin letter.



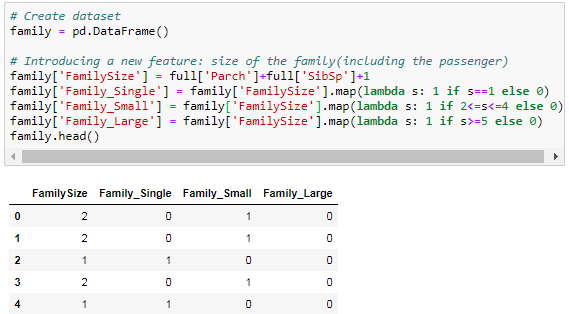
* + 1. Extract ticket class from ticket number

The ticket variable seems unsystematic, so let me extract the prefix of the ticket. If only digit in the ticket number, then replace it with ‘XXX’; otherwise extracting the prefix code.



* + 1. Create family size and category for family size

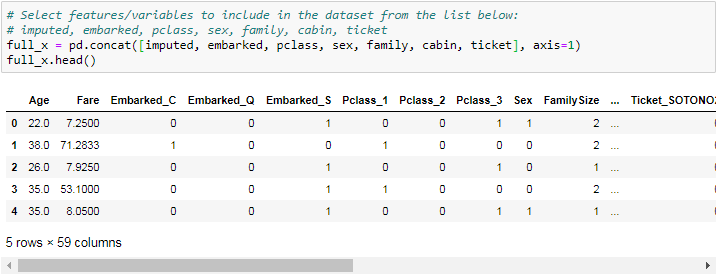
Here I want to use two variables Parch and SibSp to create the family size variable.



* 1. Assemble final datasets for modeling
     1. Variable selection

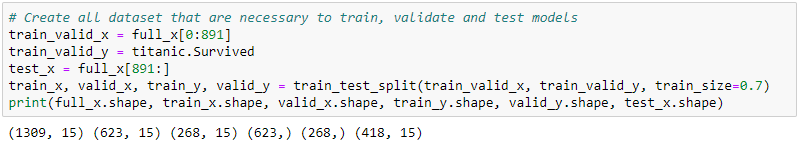
Select below features in the datasets:

* Imputed
* Embarked
* Pclass
* Sex
* Family
* Cabin
* Ticket



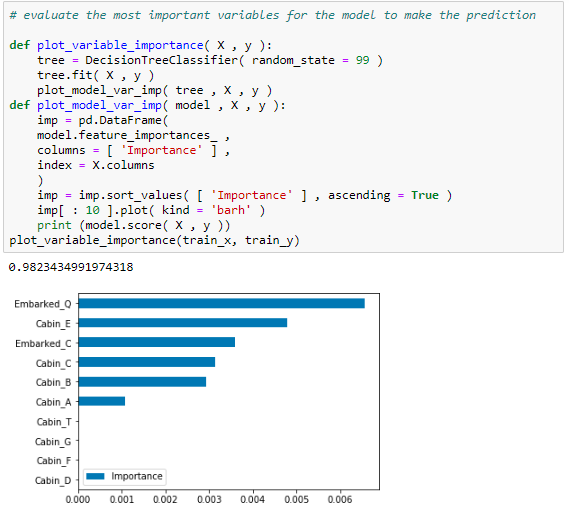
* + 1. Create datasets

Below I will separate the data into training and test datasets



* + 1. Feature importance

Selecting the optimal features in the model is important. Now, I am trying to evaluate what the most important variables are for the model to make the prediction.



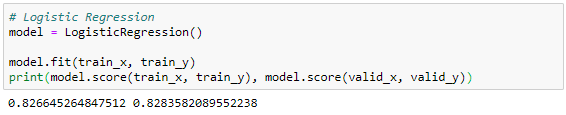
1. Modeling

Now, I need select a model which I would like to try then use the training dataset to train this model and thereby check the performance of the model using the test set

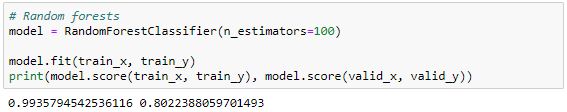
* 1. Model selection

There are several options to choose from when it comes to models. So a good starting point is logistic regression.

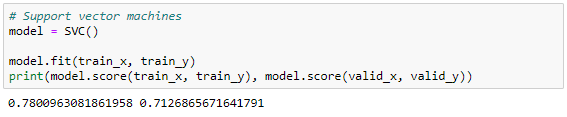
* + 1. Logistic Regression



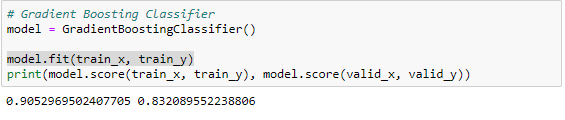
* + 1. Random forests



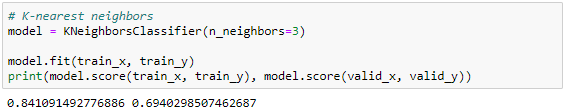
* + 1. Support vector machines



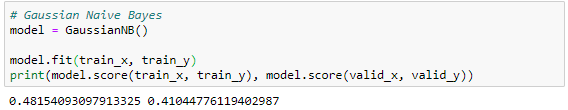
* + 1. Gradient boosting classifier



* + 1. K-nearest neighbors



* + 1. Gaussian Naïve Bayes



1. Evaluation

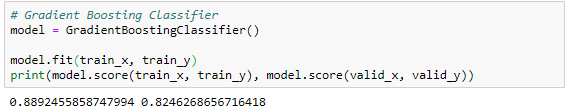
Now I am going to evaluate model performance and feature importance

* 1. Model performance

We can evaluate the accuracy of the model by using the validation set where we know the actual outcome. This dataset have not been used for training the model, so it’s completely new to the model.

We then compare this accuracy score with the accuracy when using the model on the training data. If the difference between these are significant this is an indication of over-fitting. We try to avoid this because it means the model will not generalize well to new data and is expected to perform poorly.

From 4.1, comparing the score for these models, I though gradient boosting classifier is the best model. It has the highest score for validation data, also the difference between training data and validation data is not big, that means the model does not have over-fitting problems.



* 1. Feature importance – selecting the optimal features in the model

有待补充

1. Deployment

Publishing the resulting prediction from the model to the Kaggle leaderboard.

