Predicting survival probabilities in a shipwreck: Titanic

Brandon Rodríguez

April 28th, 2021

1. Introduction

1.1 Background

The Titanic shipwreck was one of the most infamous in history, being remembered by today due to its big impact in pop culture and traditional media. This was taken in account by Data Science and Machine Learning platform Kaggle, which made a contest for the first online competition made in 2012.

1.2 Problem

The competition aims to utilize machine learning and diverse data science techniques to predict whether a patient would survive or not the shipwreck.

1.3 Interest

Data scientists, recruiters or people who want to start investigating modern-age safety measures in boats can use this type of approach to improve the safety and probabilities of survival in specific areas of the boat, as well as helping boat staff to prioritize the groups that are more vulnerable in the case of an accident

2. Data acquisition and cleaning

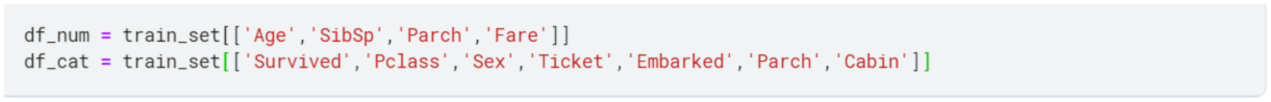
2.1 Data sources

The data is provided from Kaggle, separating in three CSVs. A train set, test set and gender_submission, to analyze and correctly predict the target mentioned above. The data is classified in the following columns:

- 'PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked', 'train_test'

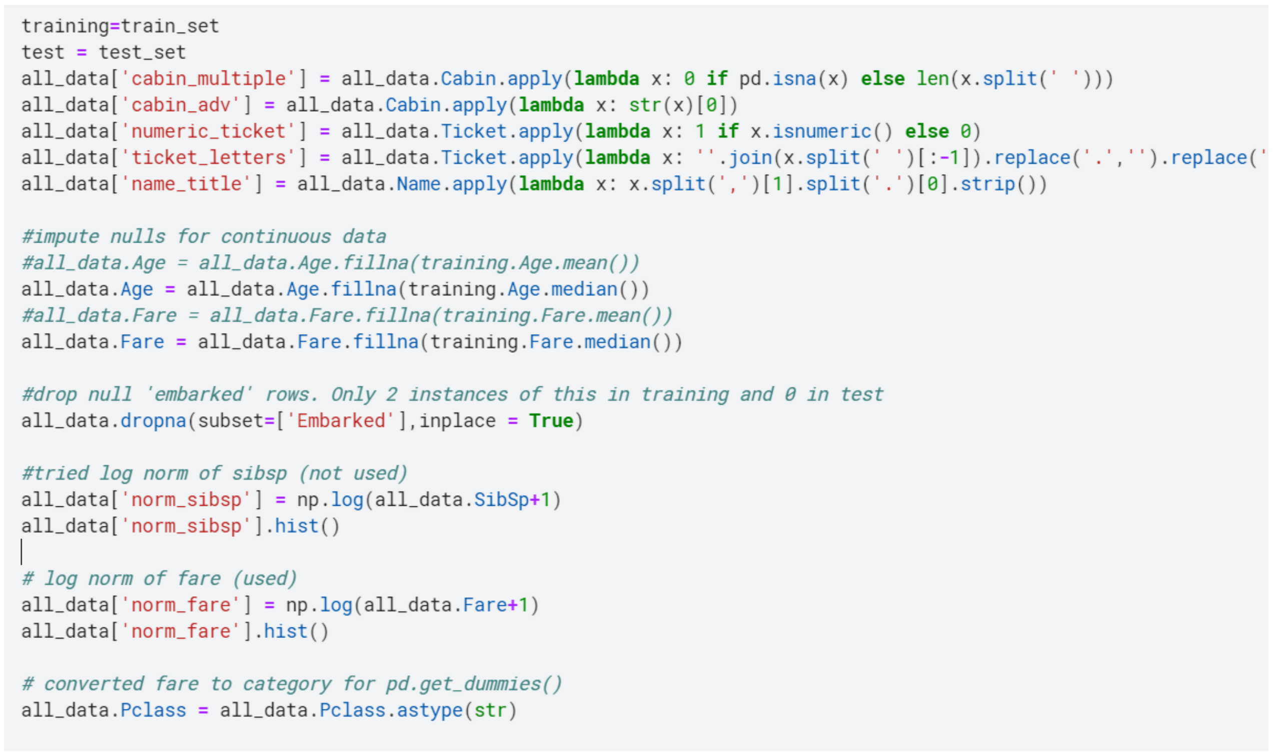
2.2 Data cleaning

The data provided in the data had several obstacles when utilized with Machine Learning. The first one was empty values, which had to be replaced with zeros for the sake of consistency. Then, the categorical features where described by qualitative values, so we had to convert them into 0 and 1s to describe whether the passenger possessed the trait or not. Finally, we had to ensure that the data was properly scaled to fit the Machine-Learning algorithms.



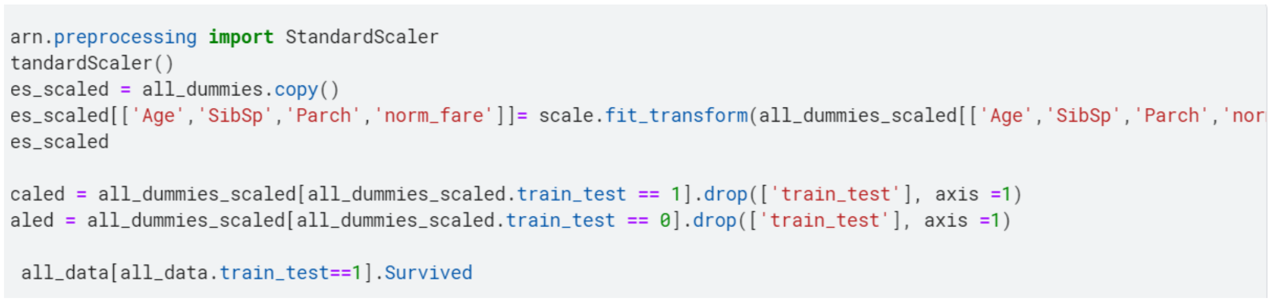
```
df_num = train_set[['Age','SibSp','Parch','Fare']]
df_cat = train_set[['Survived','Pclass','Sex','Ticket','Embarked','Parch','Cabin']]
```

Figure 1: Defining numerical and categorical variables



```
training=train_set
test = test_set
all_data['cabin_multiple'] = all_data.Cabin.apply(lambda x: 0 if pd.isna(x) else len(x.split(' ')))
all_data['cabin_adv'] = all_data.Cabin.apply(lambda x: str(x)[0])
all_data['numeric_ticket'] = all_data.Ticket.apply(lambda x: 1 if x.isnumeric() else 0)
all_data['ticket_letters'] = all_data.Ticket.apply(lambda x: ''.join(x.split(' ')[:-1]).replace('.','').replace('
all_data['name_title'] = all_data.Name.apply(lambda x: x.split(',')[1].split('.')[0].strip())

#impute nulls for continuous data
#all_data.Age = all_data.Age.fillna(training.Age.mean())
all_data.Age = all_data.Age.fillna(training.Age.median())
#all_data.Fare = all_data.Fare.fillna(training.Fare.mean())
all_data.Fare = all_data.Fare.fillna(training.Fare.median())

#drop null 'embarked' rows. Only 2 instances of this in training and 0 in test
all_data.dropna(subset=['Embarked'],inplace = True)

#tried log norm of sibsp (not used)
all_data['norm_sibsp'] = np.log(all_data.SibSp+1)
all_data['norm_sibsp'].hist()

# log norm of fare (used)
all_data['norm_fare'] = np.log(all_data.Fare+1)
all_data['norm_fare'].hist()

# converted fare to category for pd.get_dummies()
all_data.Pclass = all_data.Pclass.astype(str)
```

Figure 2: Normalizing data



```
arn.preprocessing import StandardScaler
tandardScaler()
es_scaled = all_dummies.copy()
es_scaled[['Age','SibSp','Parch','norm_fare']]= scale.fit_transform(all_dummies_scaled[['Age','SibSp','Parch','nor
es_scaled

caled = all_dummies_scaled[all_dummies_scaled.train_test == 1].drop(['train_test'], axis =1)
aled = all_dummies_scaled[all_dummies_scaled.train_test == 0].drop(['train_test'], axis =1)

 all_data[all_data.train_test==1].Survived
```

Figure 3: Scaling for Machine Learning

2.3 Feature selection

After data cleaning, we were kept with 889 entries, and the original columns described above. This means that we discarded only two entries, due to them being incomplete and not having enough information to perform predicting algorithms to them.

3. Exploratory Data Analysis

3.1 Calculation of target variable

The target variable in this case was pretty obvious, being the survival of the passenger. This meant that utilizing basic information provided above, we should be able to predict or not if the passenger lived after the tragic shipwreck.

3.2 Age Distribution

Utilizing the information provided, we start exploring the data to see if there is any predominant age group in the Titanic. According to the graph, there were predominantly people between the age of 20-30 years old, which we can later use to start comparing between the survival rate.

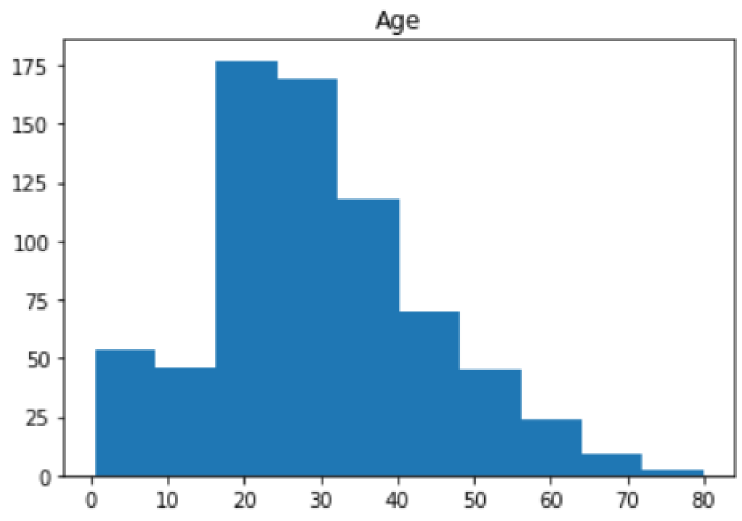


Figure 4: Age Distribution

Figure 2. Box plot of improvement of players of different ages.

3.3 Distribution between Age, Siblings on Board, Parents on Board and Parent Fare

The hypothesis here is that people of different ages have different relationships among their families aboard the Titanic. In other words, people of lower age had to have a parent on board, and if people had any siblings on board their probability of survival would be affected.

```
             Age     SibSp     Parch      Fare
Age     1.000000 -0.308247 -0.189119  0.096067
SibSp  -0.308247  1.000000  0.414838  0.159651
Parch  -0.189119  0.414838  1.000000  0.216225
Fare    0.096067  0.159651  0.216225  1.000000
[8]: <AxesSubplot:>
```

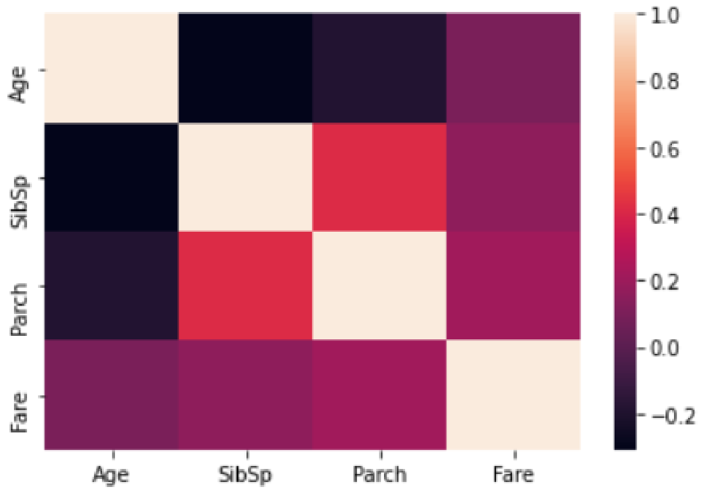


Figure 5: HeatMap

3.4 Relationship between Ticket class and Survival Rate

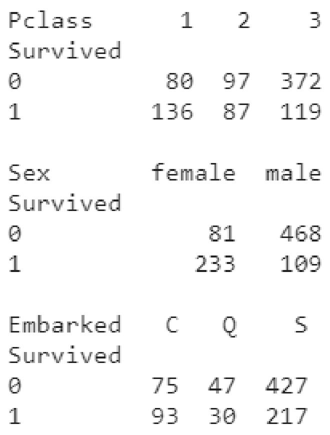
The hypothesis here is that the Ticket-Class would affect the probabilities of survival, because like we saw in James Cameron Titanic, people of higher class would get prioritized in boarding the rescue boats over people of lower class. In this case, we see that this hypothesis was correct, due to the fact that people from the 3rd class only 24% survived, while in first class more than 60% survived, and in middle class about 50% survived.

3.5 Relationship between sex and Survival Rate

According to naval policies, women and children are always the first to be evacuated in case of an emergency, so I inferred that we should see a difference between the number of survivors that are female and the number of survivors that are male. In this case, this assumption was correct, and we see that the female survival rate was close to 60%, while the male survival rate was 20.

3.6 Relationship between Port where they embarked and Survival Rate

The hypothesis here is that people from different ports would have different skills or different priorities based on their cultural differences. In this case, we compare if they departed from Cherbourg, Queenstown or Southampton. The analysis shows there wasn’t a clear relationship between these variables, so this will not be utilized for further analysis.

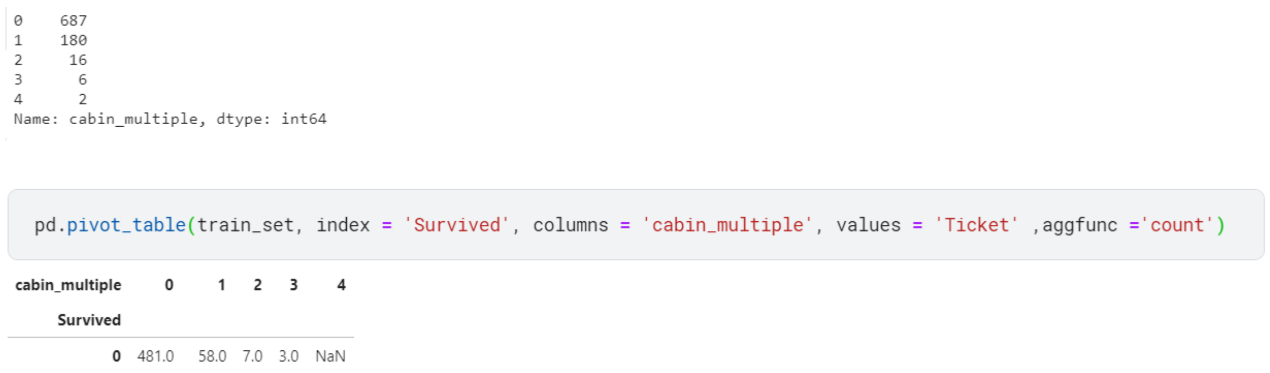


```
Pclass      1   2    3
Survived
0          80  97  372
1         136  87  119

Sex       female  male
Survived
0             81   468
1            233   109

Embarked   C   Q    S
Survived
0         75  47  427
1         93  30  217
```

Figure 6: Independent variables vs Target Variable

3.7 Relationship between being in Multiple Cabins and Survival Rate

The original dataset only contained the information regarding the cabin the ticket was assigned for. However, we can think that a passenger could be moving in different cabins due to other factors, so we want to see in how many cabins this passenger was reported. In this case, we see that there were several passengers that they were in fact moving to other cabins, so we could use this data so see if this affected their survival probabilities or not. In this case we see that the analysis threw an empty value, and there was not a defined relationship between both variables, so this will not be utilized for further analysis.



```
0    687
1    180
2     16
3      6
4      2
Name: cabin_multiple, dtype: int64
```

```
pd.pivot_table(train_set, index = 'Survived', columns = 'cabin_multiple', values = 'Ticket' ,aggfunc ='count')
```

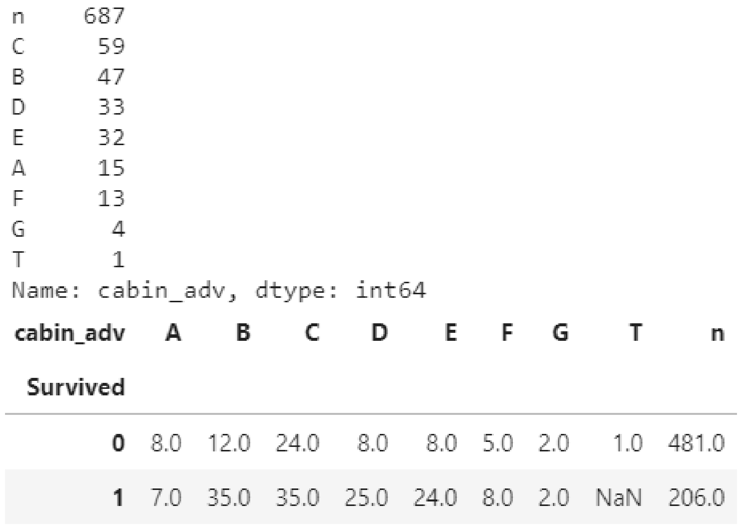
cabin_multiple	0	1	2	3	4
Survived					
0	481.0	58.0	7.0	3.0	NaN

Figure 7

3.8 Relationship between Ticket Type and Survival Rate

Finally, we can see that tickets had different letters, which could mean they were separated by something geographical or class-based. Separating the initial letters, we can see that we had 9 different types of tickets, so I proceeded to make an analysis to see if there is any relationship between ticket type and survival rate.

However, the report showed an empty value and there wasn't any direct relationship between the ticket type and survival rate, which most probably means that for this contest the tickets were written like that for the purpose of historic accuracy.



```
n    687
C     59
B     47
D     33
E     32
A     15
F     13
G      4
T      1
Name: cabin_adv, dtype: int64
```

cabin_adv	A	B	C	D	E	F	G	T	n
Survived									
0	8.0	12.0	24.0	8.0	8.0	5.0	2.0	1.0	481.0
1	7.0	35.0	35.0	25.0	24.0	8.0	2.0	NaN	206.0

Figure 8

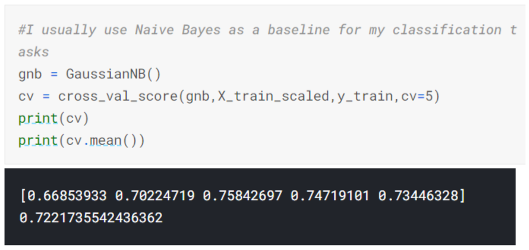
4. Predictive Modeling

To continue analyzing the data, we utilized different models to see how they perform with the default parameters. To avoid having under-fitted or over-fitted models, we utilized a 5-fold cross validation to get the baseline.

4.1 Model Application

4.1.1 Applying Naïve Bayes

This is the most basic and easiest model to implement. This was done with the purpose to have a baseline and be able to see the progress made in accuracy utilizing the models learned in the course. We had an accuracy of 72.6%, which meant the Machine Learning algorithms will be very efficient with the data sourced.



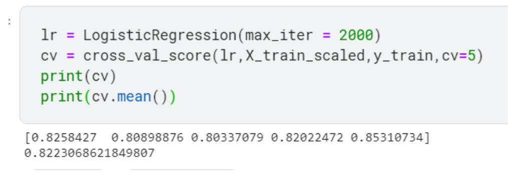
```
#I usually use Naive Bayes as a baseline for my classification t
asks
gnb = GaussianNB()
cv = cross_val_score(gnb,X_train_scaled,y_train,cv=5)
print(cv)
print(cv.mean())
```

```
[0.66853933 0.70224719 0.75842697 0.74719101 0.73446328]
0.7221735542436362
```

Figure 9: Naive Bayes

4.1.2 Applying Logistic Regression

As a second step, I followed the initial approach of trying several Machine Learning Models to see which one got me the best accuracy. Logistic Regression performed, as expected, better than the Naïve Bayes, giving us 82.22% when compared to the test set.



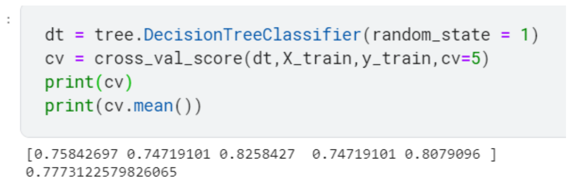
```
lr = LogisticRegression(max_iter = 2000)
cv = cross_val_score(lr,X_train_scaled,y_train,cv=5)
print(cv)
print(cv.mean())
```

```
[0.8258427  0.80898876 0.80337079 0.82022472 0.85310734]
0.8223068621849807
```

Figure 10: Logistic Regression

4.1.3 Applying Decision Tree Classifier

Third, we have the Decision Tree Approach. In this case, utilizing the random state as 1, we see that the accuracy was 77.77% when comparing to the Test Set.



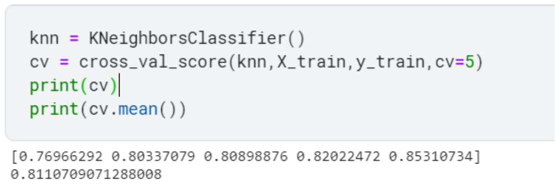
```
dt = tree.DecisionTreeClassifier(random_state = 1)
cv = cross_val_score(dt,X_train,y_train,cv=5)
print(cv)
print(cv.mean())
```

```
[0.75842697 0.74719101 0.8258427  0.74719101 0.8079096 ]
0.7773122579826065
```

Figure 11: Decision Tree

4.1.4 Applying K Neighbors Classifier

Utilizing the model we used the most in the course, the K Neighbors approach seemed like a really good fit to the data, due to the fact that predicting missing values according to results from similar kind of entries can be very effective. In this case, we see that Logistic Regression had a better approach, because KNN had an overall accuracy of 81.11%, compared to 82.22%.



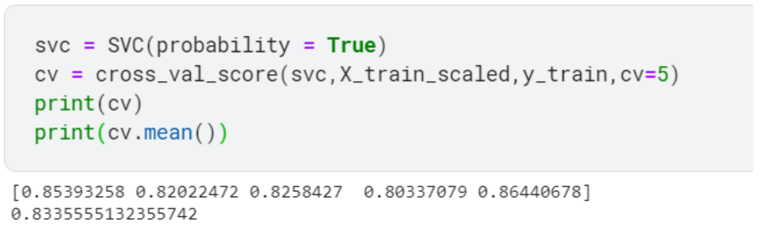
```
knn = KNeighborsClassifier()
cv = cross_val_score(knn,X_train,y_train,cv=5)
print(cv)
print(cv.mean())
```

```
[0.76966292 0.80337079 0.80898876 0.82022472 0.85310734]
0.8110709071288008
```

Figure 12: K- Nearest Neighbors

4.1.4 Applying Support Vector Machine

Finally, the implementation of SVM proved to be the most accurate and effective for this dataset. In this case, with the default values we managed to obtain 83.36% accuracy, surpassing KNN and Logistic Regression.



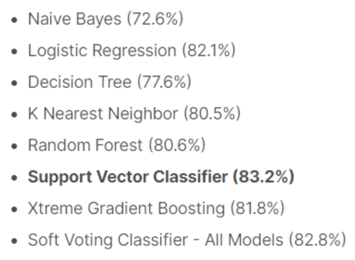
```
svc = SVC(probability = True)
cv = cross_val_score(svc,X_train_scaled,y_train,cv=5)
print(cv)
print(cv.mean())
```

```
[0.85393258 0.82022472 0.8258427  0.80337079 0.86440678]
0.8335555132355742
```

Figure 13: Support Vector Machine

5. Conclusions

In this study, I was successfully able to predict and learn the different factors that could affect the survival probability of a shipwreck. At first glance, factors like sex or age can be thought of as very intuitive, but their degree of impact was proved to be extremely high. Apart from that, other factors that were discussed in the study also influenced the survival probability, proving that data can seriously help us improving current technology and their uses.



- Naive Bayes (72.6%)
- Logistic Regression (82.1%)
- Decision Tree (77.6%)
- K Nearest Neighbor (80.5%)
- Random Forest (80.6%)
- **Support Vector Classifier (83.2%)**
- Xtreme Gradient Boosting (81.8%)
- Soft Voting Classifier - All Models (82.8%)

Figure 14: Machine Learning Results

6. Future directions

I was able to achieve the best accuracy in the Machine Learning models utilizing the default values, which means that by modifying and tuning their parameters, we can improve their accuracy, which could also mean that maybe the best model is not Support Vector Machine.

Apart from that, using external data linked to the information provided of the people aboard the Titanic, to see if we can relate any other variable towards the target variable.