Gurjus Singh

October 11th, 2020

MSDS 422 – Practical Machine Learning

Assignment #4 Random Forests and Gradient Boosting PART B

**Data Preparation, Exploration and Visualisation**

The dataset that we did our data analysis on this week was on the Titanic. Before diving into the dataset, it is important to understand the historical context of the Titanic. The Titanic was a ship that departed from South Hampton, England in the year 1912 [1]. It sank hours before reaching North America and Newfoundland by hitting an Iceberg [1]. With this in mind, it is time to prepare our dataset for our algorithms to learn on. Our goal was to see if our algorithms could classify who survived based on several features in our data.

As before in other assignments, I had to load our dataset which will allow us to manipulate the data using Python. The data was already split into train and training sets which was convenient for this assignment. After loading, I wanted to look at the initial data to get a feel for the data types and variables. I first noticed that this data set had missing values specifically in “Cabin” which I saw in 1-1 and 1-2. I also noticed there was a “Passenger ID” which was used to distinguish passengers. I also noticed a “Name” Column, and a “Sex” Column which shows gender of each passenger. I also noticed “Age” Colum, age of each passenger, “Fare” which is how much each passenger, “Cabin” which is where they resided on the ship and Embarked which is where they got on the ship. I looked up what “SibSp” and “Parch” meant in on Kaggle which was the source of my dataset and it mentioned that “SibSp” represents count of spouses and siblings of passenger while “Parch” was basically a count of the number of Parents and children per passenger [2].

After looking at the initial data set, I then decided to make sure that all the data column types matched from what I inferred from looking at the head of each data set.

Graphical user interface, application

Description automatically generatedA picture containing table

Description automatically generated

*Training Set Table Head 1-1 Test Set Table Head 1-2*

Table

Description automatically generatedTable

Description automatically generated

*Data Types 1-3*

A picture containing text

Description automatically generatedI noticed that most of the types was exactly what I inferred, but the columns that contained strings were actually of object types as seen in 1-3. I then look at the shape of each data set as seen in 1-4 and one column was missing which was “Survived” column from our test set. This is the column which will be the column we want to predict using our classification models. Specifically, this is our response variable.

A picture containing text

Description automatically generated

*Training and Test Data Shapes 1-4*

Graphical user interface, table

Description automatically generated

*Basic Stats for Each Data Set 1-5*

I then wanted to find out initial statistics on the data. I know the stats for “Pclass”, “Survived”, “Passenger Id” were not useful in our case. The “Fare”, “SibSp”, “Parch”, and “Age” column were useful as we could do statistical computations on them. The amazing thing that I found was that most of the people were around age 30 on the Titanic. I thought most of the people would be older around 50-80. This was some good insight.

I then put together some initial EDAs of the training set to see if there was anything useful. I did a barplot to see if there was big difference between survived class and not survived class. What I found is more people died than survived in my results as shown in 1-6. Next I wanted to make a scatter plot to see if there was any connection between variables such as “sex”, “pclass”, “age”, “fare” and “Survival”. What I found was there was a trend between “fare” and survival which shows that the majority of the people who were paid higher fares survived while majority of lower fares did not survive in scatter plot 1-7. In 1-8 I found out that more females than males survived which was an interesting find through the data. I then wanted to see one last EDA to see if there was any finding between “pclass” and survival. What I noticed out of all those people that survived Pclass1 and Pclass3 had the most passengers that survived as shown in 1-9.

*Chart, scatter chart

Description automatically generated*Chart, bar chart

Description automatically generated

*Barplot 1-6 Scatterplot 1-7*

*Chart, pie chart

Description automatically generated*Chart, histogram

Description automatically generated

*Histogram 1-8*

*Piechart of Classes and Survival Rate 1-9*

After doing initial EDA, I had to make sure everything was ready for classification modeling. What I noticed is that some columns in the training set and test set had NA values as shown in 1-10. What I noticed was “Age” was missing a huge chunk of data, as well as “Cabin”. One way to deal with to deal with age is to use K Nearest Neighbors. This involves finding a grouping of variables that are correlated with “Age”. In this case I found using the heat map 1-11 that there was a somewhat large negative correlation between “Age” and “Sibsp”, “Pclass variables. With this in mind I found median Age of each subgroup of “Age”, “Sibsp” and “Pclass” and I imputed the age based on that. This took care of Age imputation. Next to impute Cabin, I made up a new class which I called “O”. I then transformed the Cabin column by only storing the first character which is the letter. I then saw that there missing values in “Embarked” in the train data and “Fare” in the test data. I took care of Embarked by finding the mode of the column which was “S” and imputed using mode, while for Fare I took the mean imputation method.

Table

Description automatically generated

*A picture containing application, calendar

Description automatically generated*

*Missing Values 1-10 Heat Map 1-11*

After imputation, it was time to convert the object columns to numerical. This was done by creating separate functions to deal with encoding. For each column except for “Sex” such as “Embarked”, “Cabin” I made N-1 dummy columns with 1 and 0s. After doing encoding, I did feature engineering creating a relevant variable that could be better use. This column was created by taking “Sibsp” and “Parch” and summing them to find the total family members on board for a given passenger. For Forests, I did not need to make N-1 columns instead I needed to convert “Cabin” and “Embarked” to int category type. Once I did all this it was time to start model preparation phase.

**Review research design and modeling methods**

The three models we will be using for prediction were Logistic Regression, Random Forest and Extra Randomized Trees. Logistic Regression is used to estimate the probability that a given instance belongs to a class [3]. The way this works is a sigmoid function is used to give a probability between 0 and 1 based on the features from the training set. From there the data scientist uses a cutoff to classify each instance [3]. Random Forest is another special algorithm which involves decision trees [3]. Decision trees are trees that use feature cutoffs to decide on the final value of the prediction [3]. With Random Forests it involves an ensembled method which involves different trees which all aggregate together to form a predicted result [3]. Each tree uses a method to collect training data which is known as “Bagging”. “Bagging” involves sampling which collects n data points from the training set to train on. Thirdly another Tree learning algorithm used is Extra Randomized Trees [3]. This is similar to Random Forests in a sense because they both pick a random number of features to train on, but this also involves random thresholds to use as a cutoff for each feature chosen [3]. I think these algorithms are important because Linear Regression as mentioned before is mostly used to predict numerical values. Trees are kind of different in that one can see how the algorithm works and it is not a “black-box” like Linear Regression is. Trees do not assume linearity like Regression does and does not need transformation and scaling [3]. I think it is important to define several models/algorithms, so a data scientist knows how they work. It is also important to use several models to get an idea of one being better over the other. I also think that several of these methods can be combined to get one score known as Ensembled methods.

Before running each model it is important to drop features that are unimportant in prediction or cannot be used and in this case “Passenger ID”, “Name”, “Parch”, “SibSp” which were used in feature transformation, “Cabin” and “Embarked” which we converted to numerical. Once we have dropped these features, we can then split the train dataset to 80% train and 20% test. I cannot use the original test data in my model since it did not come with the response column. For the tree algorithms I only needed to drop “Passenger ID”, “Name”, “Parch”, and “SibSp” as the other columns were just converted to numeric.

**Review results, evaluate models**

After training the models, I noticed from the results of AUC, and accuracy score, that the Random Forests model was the best model. The scores were both pretty high were Accuracy and AUC for the Test Data in 1-13 while score while the Accuracy was lower for Logistic Regression and was too high for AUC score at 0.89. I also checked the OOB score for Random Forests and what I saw was that the OOB score increased for the test set from the training set from 0.80 to 0.866.

I also wanted to how the linear regression model was doing by looking at the coefficients and which features the extra trees classifier was marking as important. The formulas for logistic regression from 1-14 and 1-15 are:

1.71 – 1.53\*Pclass -2.4\*Sex -0.37\*Age + 0.29\*Fare + 0\*CabA + 0.74\*CabB +0.63\*CabC + 1.12\*CabC – 0.0079\*CabD + 0.62\*CabE – 0 .394\*CabF + 0\*CabT – 0.049\*EmbS + 0.733\*EmbC – 0.205\*Fammemb

*Train Data Formula*

3.15 – 1.67\*Pclass -2.51\*Sex -2.13\*Age + 0.56\*Fare + 0.26\*CabA + 0.20\*CabB -0.26\*CabC + 0.51\*CabC + 1.17\*CabD + 0.81\*CabE + 0.017\*CabF – 0.245\*CabT – 0.362\*EmbS + 0.312\*EmbC – 1.287\*Fammemb

*Test Data Formula*

From looking at the train and test data formula, Pclass and Sex are the most heavily weighted which suggests that they contribute the most to the survival rate. What Pclass of -1.53 says for the train data that for every increase in Pclass the odds of survival decrease by 21.7%. With Sex for every increase, the odds of survival decrease by 9%. In the test data, what Pclass of -1.67 says for the train data that for every increase in Pclass the odds of survival decrease by 18.8%. With Sex for every increase, the odds of survival decrease by 8.1%. I also notice the weight of Age has gone up which says for every increase in age the odds of survival decrease by 11.89%. I also went ahead to look at what the Extra Trees said about importance features for both train data and test data in 1-16 and 1-17. What it said is that the importance of “Sex” was at 28%, the importance of “age” was at 24% and the importance of “Fare” was at 22.4% for the train data. I was a little surprised at how it marked these important features. For the test data, it said that 10% feature importance to “Pclass”, 27% importance was to “Sex”, 15.3% importance to “age”, 19.34% to “Fare” and 12% feature importance was to “Cabin”. I thought this was surprising as well considering the EDA findings.

A picture containing text

Description automatically generatedTable

Description automatically generated

*Training Set Results 1-12 Test Set Results 1-13*

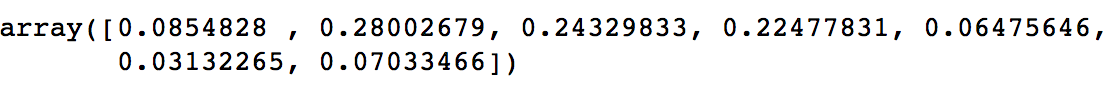
*Graphical user interface, text, application

Description automatically generated*

**Text

Description automatically generated**

*Test Data Coefficients 1-14**Train Data Coefficients 1-15*

****

****

*Train Data Feature Importance 1-16 Test Data Feature Importance 1-17*

**Implementation and programming**

For implementation I first imported the packages as seen in 1-18 in the such as numpy for using the mean function, pandas for dataframe methods such as groupby, sklearn for the classification models, matplotlib which I used to plot several of the EDA plots.

Graphical user interface, text, application

Description automatically generated

*Code Imports 1-18*

I had to load the data using the **Pandas.read\_csv()** function. After loading the packages and data, I used **.head(), describe(),** and **.dtypes** to get general information about the two datasets. One important thing to do in data prep is to also see if we have NA values. Specifically, the code to do this is **.isnull().sum()** to find the missing values in each column. I then used different EDA methods to show the data using matplotlib package. The methods I used were **.scatter()** which was used to create the scatter plot in 1-7. I used **.bar()** method for creating the barplot 1-6 and .**hist()** to create the histogram. The piechart was created was created by creating three ratio for each class and passengers that survived. I then used the **plt.pie** function to create the pie chart as shown below. For imputation techniques to handle NA values I had to use **groupby** in the Pandas package as it was important to group by columns which were highly correlated with age to find a trend. I then added **.median()** on to the group by to find the median of each subgroup for age. I then used the function **.fillna()** to fill in the specific age columns. This method was used in the other imputations. Before imputing Cabin, I did not need the whole sequence only the first character which was a letter, in order to get only the first character it could be done with the **lambda** function using indexing passed into the **map** method

For feature transformation, I used the add **+** expression to add two columns together which would be used to find family members of each passenger on board the Titanic. After Feature transformation, I encoded the columns, using **np.where()** to find columns that matched the specific Boolean expression such as for **cabin = = A, 1, 0.** This expression specifically meant to return 1 in the output encoded column if the value was equal to 1. This was done for columns such as Embarked, Cabin and I had only N-1 column for encoded N values. The Sex column was also encoded but did not need N-1 columns for N values since it was already binary between Male and Female. After imputation, feature transformation and encoding, I had to drop irrelevant columns using .**drop()** column. After dropping non-relevant columns such as nonnumerical columns, and untransformed columns, I then split the train set specifically using **.split().** I also had to save the target variable which was “Survived” in its separate dataframe. I then started the training using Logistic Regression, RandomForests and Extra Trees. I had to first call the instance of each model and use the **.fit()** method on each. The way to call the instance is writing **LogisticRegression(),** **RandomForestClassifier()** and **ExtraTreesClassifier().** After using the **.fit()** method, I then computed the AUC and Accuracy scores for each method which each required a custom-made method. Accuracy is simply defined as predicted correctly/total predicted while AUC is the area under ROC curve which is the curve that is computed by True Positive rate by False Positive rate [3]. I used the confusion matrix custom made function to find the confusion matrix. I also used **ExtraTreesClassifier().feature\_importances\_** to find which features Extra Trees put more weight in percentages on. I also did something similar to get coefficeints an y intercept for logistic regression. See Sklearn package for more details on the attributes to apply.

**Exposition, problem description, and management recommendations**

After reviewing the above formulas, results from each algorithm, **I recommend** the Random Forests classifier for this data set as it seems to have performed the best by looking at the table for the test data 1-13. I chose this algorithm as Logistic Regression had too low of an accuracy score for the test-data at 0.787 while Extra Trees had too low of a AUC score for the test data at 0.774. I think by looking at the formulas and feature importance in 1-14-1-17 it made sense to me why those were not performing well as those algorithms though features such as “Cabin”, “Age” were important or heavily weighted. Another reason why Random Forests was a great algorithm, is because the OOB score was around 0.86 for the test data. This was also a big factor in my interpretation. Therefore, I recommend the Random Forests algorithm to management. In the future I also want to explore how I can make Extra Trees classifier, and Logistic Regression perform better. I think it will take some hyperparameter tuning which may help them both perform better for this type of problem.

References

[1] <https://www.history.com/topics/early-20th-century-us/titanic#:~:text=The%20RMS%20Titanic%2C%20a%20luxury,their%20lives%20in%20the%20disaster>.

[2] <https://www.kaggle.com/c/titanic/data>

[3] Géron, A. *Hands-On Machine Learning with Scikit-Learn & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* 2d Edition. Sebastopol, Calif.: O'Reilly. [ISBN 9781492032649], 2019.