Gurjus Singh

October 4th, 2020

MSDS 422 – Practical Machine Learning

Assignment #3 Evaluating Classification Models

**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.

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 columns. 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. 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 is the Logistic Regression, Support Vector Machines, and KNeighbors Classifiers. 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]. In Support Vector Machines, we have a margin which can be thought as of the flexibility to errors allowed [3]. This can be thought of as overfitting in linear regression if the margin is small as possible [3]. Support Vector Machines try to find the maximum margin that can correctly classify instances [3]. Support Vector Machines can also be used for nonlinear regression [3]. Lastly KNeighbors Classifier is an unsupervised classifier compared to the other two [4]. The way it works is the programmer inputs a particular parameter to the method which will then find the smallest distance between a certain number of defined instances in certain classes and the new instance [4]. Whichever defined instance is closest in distance/features to the new instance will be the class of the new instance [4].

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.

**Review results, evaluate models**

Table

Description automatically generatedTable

Description automatically generatedAfter implementing the models, it was time to see their results. Using the train data called x\_train and y\_train the AUC score was highest for KNN at 0.872 and was lowest for Support Vector Machines at 0.733. The Accuracy was highest for Logistic Regression and lowest for Support Vector Machines as well as shown in 1-12. I wanted to see if the same model was best for the test data. As seen in 1-13 on the test data, the highest AUC score came from KNNs, while the lowest was from Support Vector Machines. The accuracy score was lowest for Support Vector Machines, while the highest was for Logistic Regression. In my opinion I believe Linear Regression was the best model as it had high Accuracy as well as a high AUC score. I have learned we cannot rely purely on AUC score so that is why Linear Regression performs the best overall.

*Train Data Results 1-12 Test Data Results 1-13*

After going over the overall models, I looked at the parameters for logistic regression which helped fit the model to both the train data and test data. What I found was these two formulas computed for the train data from 1-14 and the test data from 1-15 :

A picture containing graphical user interface

Description automatically generated

*A picture containing text

Description automatically generatedTrain Data Coefficients and Intercept 1-14*

*Test Data Coefficients and Intercept 1-15*

Train Data Formula: 4.26 – 0.76\*Pclass-2.53\*Sex – 0.04\*Age + 0.004\*Fare + 0.57\*CabA + 0.25\*CabB -0.08\*CabC +0.79\*CabD + 1.5\*CabE + 0.89\*CabF – 0.0141\*CabG – 0.12\*CabT – 0.285\*EmbS + 0.0285\*EmbC -0.211\*FamilySize

Test Data Formula: 2.42 – 0.65\*Pclass-2.4\*Sex – 0.02\*Age + 0.009\*Fare + 0\*CabA + 0.8\*CabB + 0.53\*CabC + 1.25\*CabD -0.06\*CabE + 0.71\*CabF – 0.39\*CabG – 0\*CabT – 0.303\*EmbS + 1.02\*EmbC - 0.213\*FamilySize

From looking at the formulas above, the Logistic Regression model on the train data thinks that the most important feature is “Sex”. “Sex” has 1 for every passenger that is a male and 0 for every person that is a female. The coefficient is -2.53 which indicates that females have 8% more of surviving than males since by taking the exponential function of -2.53 one will get 0.08 which is 8 percent. Comparing the odds of survival by gender on the test set, the model says that males’ survival rate is 9 less than females. Secondly “Pclass” also has an influence on the model and it says for every increase in the variable the chance of survival decreases by 46.7 percent. While for the test set the odds of survival decreases by 52.2 percent. Lastly “Family Size” seems to be also contributing to the model. The model says for the training set that for every increase in family size there is an 80 percent decrease in odds of survival and similarly for the test set the odds of survival decreases by 80 percent for increase in family variable. With Fare, the odds of survival actually increase 100% with every increase in Fare.

Lastly, I want to look at results from the confusion matrices. What the matrices told me is that the logistic regression model seemed to perform better and predict less instances as false positives and negatives. I looked at this for both data sets, but usually we want to see performance on the test set.

Table

Description automatically generated

*Confusion matrix for Test set 1-16*

**Implementation and programming**

For implementation I first imported the packages as seen in 1-17 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 1-17 Imports*

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, SVC and KNN. I had to first call the instance of each model and use the **.fit()** method on each. 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.

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

The overall goal of this analysis was to find a way to predict the “Survived” response variable on several key features. When thinking about the features involved, I think the obvious relevant features in the dataset were “Sex” which had the biggest negative coefficient in 1-14, 1-15, then came “Pclass” and lastly “Family Size”. After evaluating each model, I would recommend Logistic Regression for predicting the survivors of the Titanic because when looking at the results from 1-12 and 1-13, Logistic has both AUC and Accuracy scores that are not too low and not too high which shows that the model is not overfitting such as “too good to be true” and not very bad. The other models specifically seem too high and too low specifically AUC socre for KNN is around 0.9 which is what you want, but I think that is too high to my liking as the Accuracy score is around 0.76, while for Support Vector Machine method the Accuracy score is too low at 0.70. For Logistic Regression, I think both are right where you want them not too high and not too low. Overall in looking at ways to improve the model, I think we need to look at other features, maybe there are attributes that contributed to why more Females survived, an underlying correlation?

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.

[4] <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>