Exploratory Data Analysis and Classifier

Narayan Kothari(EE15BTECH11020)
Indian Institute of Technology Hyderabad

I. Objective

The Objective of this project is to give an idea about how the work flow in any predictive modeling problem. How do we check features, how do we add new features and some Machine Learning Algorithms. I have tried to keep the ipython notebook, you can access that by clicking here.

II. Exploratory Data Analysis(EDA)

In this section we will try to see data, Analyse the features in the data and finding any relations or trends with other features. The dataset I am using is Titanic dataset(freely available in Kaggle). The data is such that given the details of passenger we need to analyse and predict what kind of people were likely to survive the tragedy.
Figure 1 shows some entries of the dataset and also we can see what kind of features are there in the dataset.

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Fig. 1: Dataset

As we can see there are 11 features and we need to predict who survived from these information. There are some NAN values in the dataset, we will fix them as we proceed. Lets explore some of the features. First let us understand the different types of features:

1) **Categorical Features**: A categorical variable is one that has two or more categories and each value in that feature can be categorised by them.For example, gender is a categorical variable having two categories (male and female).
2) **Ordinal Features**:An ordinal variable is similar to categorical values, but the difference between them is that we can have relative ordering or sorting between the values. For eg: If we have a feature like Height with values Tall, Medium, Short, then Height is a ordinal variable.
3) **Continuous Feature**: A feature is said to be continuous if it can take values between any two points or between the minimum or maximum values in the features column.

A. How many Survived

Figure 2 is the pie chart and count plot of how many people survived the tragedy. From the plots we can see that not many

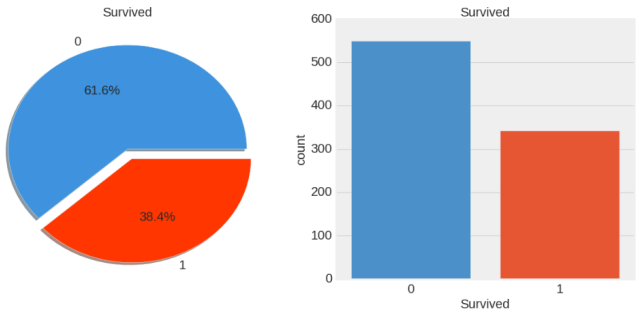

Fig. 2: Survival

people survived the accident(1 means survived). Out of 891 passengers in training set, only around 350 survived i.e Only 38.4% of the total training set survived the crash. We need to dig down more to get better insights from the data and see which categories of the passengers did survive and who didn't.

B. Analysing The Features

*1) **Sex**:* We know sex is a categorical feature. Figure 3 is the graph of (Sex:Survived versus Dead).

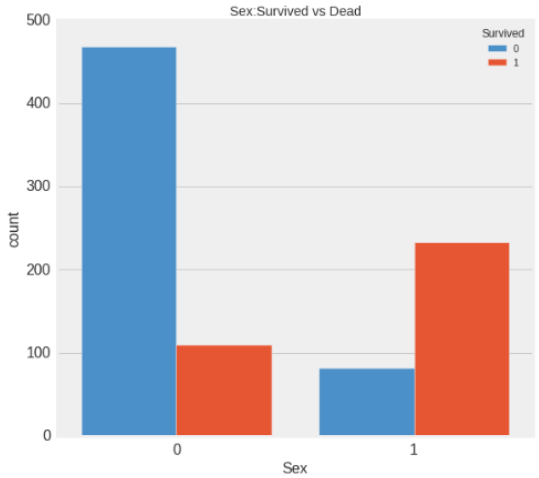

Fig. 3: Sex vs Survival

From the graph we can see that the number of men on the ship is lot more than the number of women but Still the number of women survived is almost twice the number of males survived. The survival rates for a women on the ship is

around 75% while that for men is around 18-19%. So we can see that this feature plays an important role in classification.

2) ***Pclass****:* This is an Ordinal feature as we can sort Pclass according their fare or facility provided. Figure 4 is the graph of (Pclass:Survived versus Dead). We can see that most of

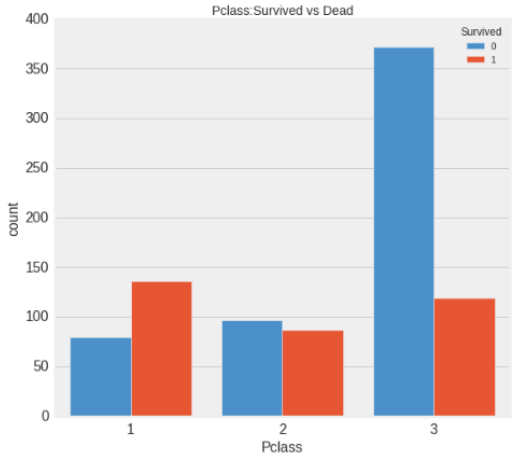

Fig. 4: Pclass:Survived versus Dead

the Passengers were of Pclass3 and number of Passengers in Pclass1 and Pclass3 were almost same. We can also see that Passengers of Pclass1 were given a very high priority while rescue. Even though the the number of Passengers in Pclass3 were a lot higher, still the number of survival from them is very low, somewhere around 25%. For Pclass1 % survived is around 63% while for Pclass2 is around 48%. So money and status matters.

Lets Dive in little bit more and check for other interesting observation. Lets check survival rate with Sex and Pclass Together. Figure 5 is the factor plot of (Sex:Pclass vs Survived).

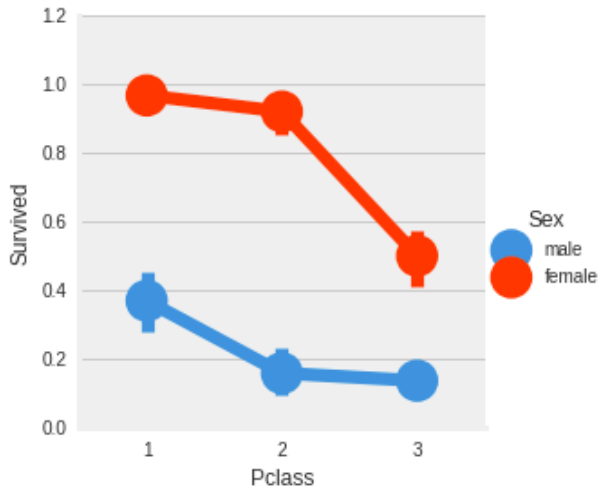

Fig. 5: Sex:Pclass vs Survival

I used factor plot to see the difference clearly. From the factor plot we can clearly see that urvival for Women from Pclass1 is about 95-96%. We can also infer that irrespective of Pclass, Women were given first priority then men while rescuing. Even Men from Pclass1 have a very low survival rate.

3) ***Age****:* We know that Age is a continuous feature and Average value of the age in the dataset is 29.7 years. Figure 6 shows the distribution of age for different Pclass and Sex.

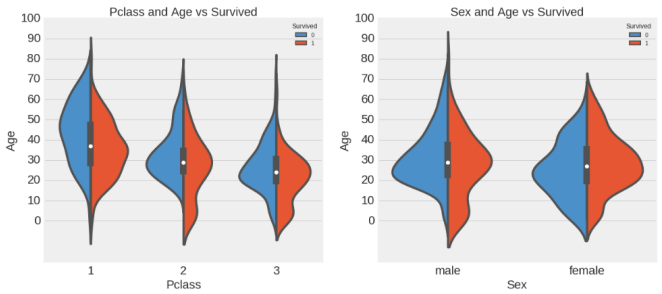

Fig. 6: Age distribution

Following observations can be made by looking on the above figure:

- The number of children increases with Pclass and the survival rate for passenegers below Age 10(i.e children) looks to be good irrespective of the Pclass.
- Survival chances for Passenegers aged 20-50 from Pclass1 is high and is even better for Women.
- For males, the survival chances decreases with an increase in age.
- Maximum number of deaths were in the age group of 30-40.
- The Women and Child first policy thus holds true irrespective of the class.

There are some values in Age column in the dataset which are not available, but we can not just replace them by mean value of age as they can be child also or aged people also. So to fill the empty values we will use the Name feature of the dataset. Looking upon the feature, we can see that the names have a salutation like Mr or Mrs or Master. Thus we can assign the mean values of Mr, Mrs, Master, Miss, etc to the respective groups.

As Age is a continuous feature it will create problem when we implement Machine Learning algorithms. So, we will convert this continuous feature into categorical by making group or bins i.e group a range of ages into a single bin or assign them a single value.

4) ***SibSip****:* This feature represents whether a person is alone or with his family members, So This is a categorical feature. Figure 7 shows the barplot of Sibsip versus Survived.

The barplot shows that if a passenger is alone onboard with no siblings, he have 34.5% survival rate. The graph roughly decreases if the number of siblings increase. This makes sense. That is, if I have a family on board, I will try to save them instead of saving myself first. Surprisingly the survival for

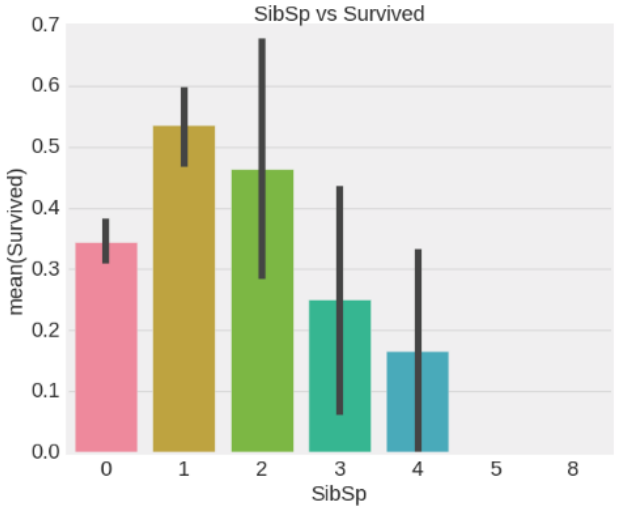

Fig. 7: SibSip vs Survived

families with 5-8 members is 0%. The reason may be Pclass as most of the Passengers were from Plass3.
We took consider of most of the features, similarly we can also plot the graphs of remaining features.
So I am mentioning some more change which I did:

- There is a Fare feature which is continuous, so I made categories of Fare feature(Farecat) as I did for Age feature.
- There are two features(Sibsip feature and Parch feature(number of guardians of the Passenger)) which are more or less same, so I made another feature Familysize(Sibsip+Parch) comprising of these two
- I made another feature initials(consist of Mr., Mrs, etc) which uses the Name feature, So this can be categorized.

*5) **Dropping of Unwanted Features**:* I am going to drop some of the features which are not necessary for the model:

- **Name**: We don't need name feature as it cannot be converted into any categorical value.
- **Age**: We have the Ageband feature, so no need of this.
- **Ticket**: It is any random string that cannot be categorized.
- **Fare**: We have the Farecat feature, so unneeded
- **Cabin**: A lot of values under this column are empty also many passengers have multiple cabins. So this seems to be a useless feature.
- **PassengerId**: Cannot be categorized.

*6) **Correlation Plot**:* Different types of correlation are:

- **Positive**: If an increase in feature A leads to increase in feature B, then they are positively correlated. A value 1 means perfect positive correlation.
- **Negative**: If an increase in feature A leads to decrease in feature B, then they are negatively correlated. A value -1 means perfect negative correlation.

If two features are highly or perfectly correlated, so the increase in one leads to increase in the other. This means that both the features are containing highly similar information and there is very little or no variance in information. This is known as Multicollinearity as both of them contains almost the same information. So While making or training models, we should try to eliminate redundant features as it reduces training time and many such advantages. Figure 8 is the Correlation Plot of the features we have.

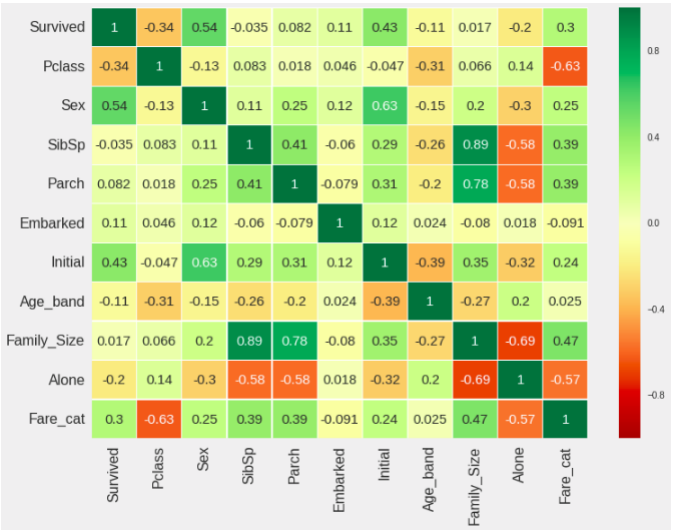

Fig. 8: Correlation Plot

From the Correlation Plot we can see that Familysize feature, Sibsip feature and Parch feature are in good correlation with each other. So we can get rid of one feature as for making training faster but I did not removed them for the training purpose.

*7) **t-SNE PLot**:* (t-SNE) t-Distributed Stochastic Neighbor Embedding is a non-linear dimensionality reduction(representing multi-dimensional data in 2 or 3 dimension) algorithm used for exploring high-dimensional data. It maps multi-dimensional data to two or more dimensions suitable for human observation. To see what t-NSE algorithm is click here. Figure 9 is the t-NSE plot of our features. By looking

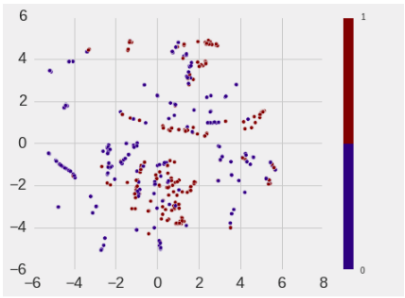

Fig. 9: t-NSE Plot

on the t-NSE plot it doesn't seem to be linearly separable in two dimension.

III. Predictive Modeling Using Machine Learning Algorithms

For the modeling purpose I split the data into 70% for training the model and 30% for testing the model. Some of the Algorithms I used to model are(Sklearn libraries are used to train model) :

A. Support Vector Machines

Support vector machines are a famous and a very strong classification technique which does not uses any sort of probabilistic model like any other classifier but simply generates hyperplanes or simply putting lines, to separate and classify the data in some feature space into different regions. Another important concept in SVM is of **maximal margin classifiers**.What it means is that amongst a set of separating hyperplanes SVM aims at finding the one which maximizes the margin M. This simply means that we want to maximize the gap or the distance between the 2 classes from the decision boundary(separating plane). This concept of separating data linearly into 2 different classes using a linear separator or a straight linear line is called linear separability. For more details on SVM click here. We use linear kernel and Radial kernel function(which will project data into another dimension where it could be linear separable also known as **Kernel trick**). I used sklearn libraries to run SVM(both Linear SVM and Radial SVM) and the accuracy of Radial SVM was 0.836% and of Linear SVM was 0.81% on test-dataset.

B. Logistic Regression

Logistic regression is generally used where the dependent variable is Binary. Even though logistic regression is frequently used for binary variables (2 classes), it can be used for categorical dependent variables with more than 2 classes. In this case its called Multinomial Logistic Regression. The underlying algorithm of Maximum Likelihood Estimation (MLE) determines the regression coefficient for the model that accurately predicts the probability of the binary dependent variable. The algorithm stops when the convergence criterion is met or maximum number of iterations are reached. For more details on logistic regression click here. The accuracy of the Logistic Regression is 0.817% on test-dataset.

C. Decision Tree

A decision tree is a map of the possible outcomes of a series of related choices. It allows an individual or organization to weigh possible actions against one another based on their costs, probabilities, and benefits. They can can be used either to drive informal discussion or to map out an algorithm that predicts the best choice mathematically.

A decision tree typically starts with a single node, which branches into possible outcomes. Each of those outcomes leads to additional nodes, which branch off into other possibilities. This gives it a treelike shape. An example of Decision tree is shown below. For more details on Decision Tree click here. The accuracy of the Decision Tree is 0.802% on test-dataset.

D. K-Nearest Neighbours(KNN)

The k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small).

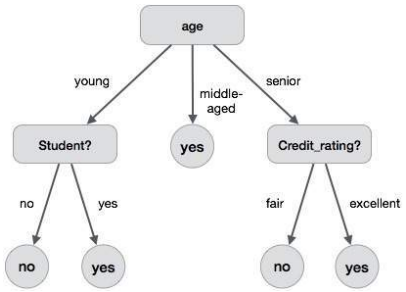

Fig. 10: Decision Tree

If k = 1, then the object is simply assigned to the class of that single nearest neighbor. For more details on K-Nearest Neighbours(KNN) click here. I ran KNN for different values of K and below is the plot of accuracies with different values of K.

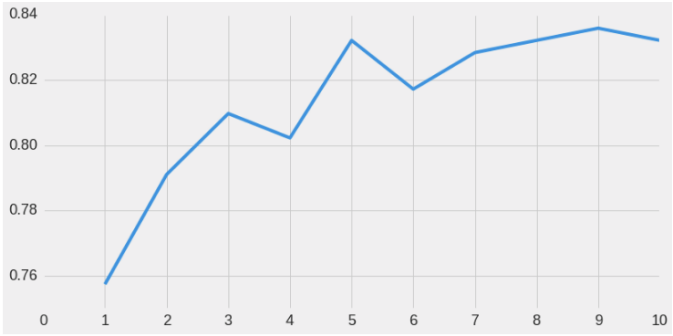

Fig. 11: K vs Accuracy

From the plot we can see that maximum accuracy was for K=5 and k=9, which is close to 0.8358% on test-dataset.

E. Naive Bayes

Naive Bayes is a simple technique for constructing classifiers- models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle- all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features. For more details on Naive Bayes click here. The accuracy of the NaiveBayes classifier is 0.8134% on test-dataset.

F. Random Forest

Random forest is like bootstrapping algorithm with Decision tree model. Say, we have 1000 observation in the complete

population with 10 variables. Random forest tries to build multiple Decision tree model with different sample and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a Decision tree model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction. For more details on Random Forest click here. The accuracy of the Random Forests is 0.821% on test-dataset with 100 number of estimators. We can tune the hyper-parameter(number of estimators for Random Forest) using Sklearn and by doing this I got optimum number of estimators as 900.

G. Artificial Neural Networks

A neural network consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. Generally the networks are defined to be feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another, and it is these weightings which are tuned in the training phase to adapt a neural network to the particular problem at hand. This is the learning phase. For more detailsclick here.
There are many parameters involved in training of ANN like loss function, optimizer, etc. I used categorical-crossentropy as a loss function, Stochastic gradient descent as an optimizer, activation function as RELU and one hidden layer. The accuracy using ANN was 0.821% on test-dataset.

IV. Ensembling

Ensembling is a good way to increase the accuracy or performance of a model. In simple words, it is the combination of various simple models to create a single powerful model. Lets say we want to buy a phone and ask many people about it based on various parameters. So then we can make a strong judgment about a single product after analyzing all different parameters. This is Ensembling, which improves the stability of the model. Ensembling can be done in ways like:

A. Voting Classifier

It is the simplest way of combining predictions from many different simple machine learning models. It gives an average prediction result based on the prediction of all the submodels. The submodels or the base models are all of different types. The accuracy for ensembled model is 0.825% on test data.

B. Bagging

Bagging is a general ensemble method. It works by applying similar classifiers on small partitions of the dataset and then taking the average of all the predictions. Due to the averaging,there is reduction in variance. Unlike Voting Classifier, Bagging makes use of similar classifiers. Bagging works best with models with high variance. An example for this can be Decision Tree or Random Forests. We can use KNN with small value of neighbors.

- **Bagged KNN**: The accuracy for bagged KNN is 0.836%.
- **Bagged DecisionTree**: The accuracy for bagged Decision Tree is 0.825%.

V. Conclusion

We have first explored the features first and then applied different-different classifying algorithms on the dataset. Also, I had tried to remove some features which does not seem important and then applied classifiers with that features but the result is best when we incorporate all the mentioned features which means there were no redundant features.
And then we saw two simple Ensemble learning techniques. We got the best accuracy on non-linear SVM and Bagged KNN which is 0.836%.
But when we split training and testing data into 80 to 20 ratio then some of the algorithms gave lesser accuracy when we split the data into 70 to 30 ratio. The reason was mostly because of over fitting of the dataset i.e it is not trying to generalize. You can see the code for all the above performed things by clicking here.