Titanic: Machine Learning from Disaster

Team size: 1

Name: Priyanka kumari

Introduction

One of the most shocking tragedy happened in the world on 15th of April, 1912. It was sinking of a British Passenger liner RMS *Titanic*, in the middle of the North Atlantic Ocean, during her maiden voyage. It was the largest ship of that time.

The reason for sinking was the hole in the ship made after the collision of the fast moving ship with an iceberg. A combination of icy cold waters, insufficient life boats for all passengers and crew members onboard, took the life of about 1502 out of 2224 passengers and crew members. It was really a heart-breaking news for everyone. The titanic dataset is available in open source, kaggle competition. In this project I have used various machine learning algorithms to build the model using training data based on the factors like age, sex, class. I have used Python for my code. I have tried to tuned the parameter with the help of GridSearchCV and found the best prediction model for the survival of passengers with greatest accuracy for the dataset.

Related work

Most of the Previous publication has compared the accuracy of the various classification model on the dataset. They have first done feature engineering followed by model fitting to find the best accuracy. In one of the previous paper published by Ekin, Neytullah Sevinc (2018), I found that they have used fourteen different machine learning techniques to *titanic* dataset to analyze the survival of the passengers on the boat. They found Gradient Boosting algorithm was giving the best F measure score. Aakriti, Shipra and Neetu (2017) have published the comparison of the algorithms based on accuracy obtained on test dataset. Nadine Farag and Ghada Hassan (2018) worked on only two machine learning algorithms. They found that decision tree gives 90.01% accuracy and Gaussian naïve Bayes gives 92.52% accuracy.

Dataset Description

The data has been given in split of train and test data. The survival attribute in the data set is the target attribute. The survival attribute is absent in test data set, the same will be predicted for the test data set. The training set will be used to build various machine learning models and test dataset will be evaluated on best model.

Data set consists of 11 attributes mixed of categorical and numeric, excluding 'Survival' attribute. PassengerId, Survived, Pclass, Sex, Ticket, Cabin & Embarked comes under qualitative Attributes and SlibSp, Parch & Fare comes under quantitative Attributes. The training set has the information of 891 passengers whether they survived or not, and the test set of 418 passengers with the goal to predict their survival. The dataset has no missing values for the attributes other than Age, Cabin and Embarked.

Data Visualization

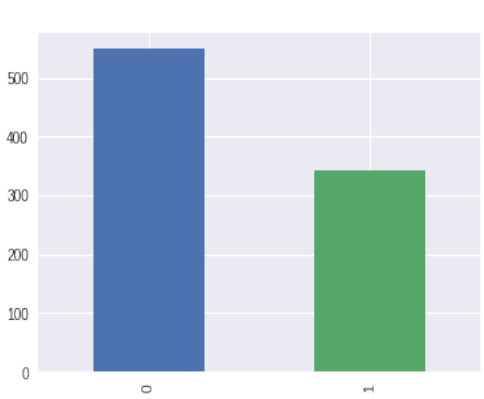


Fig.a Survival

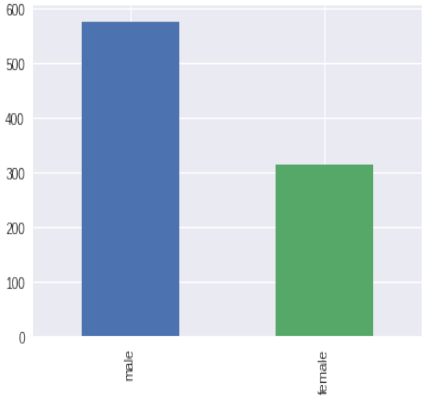


Fig. b Sex distribution

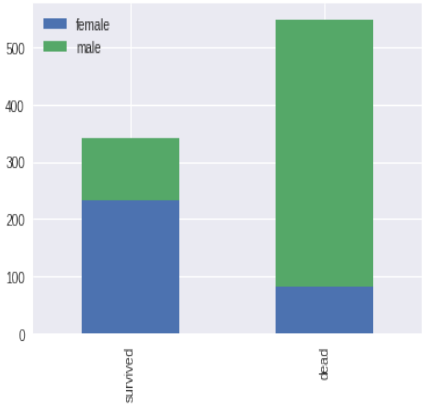


Fig.c (Sex + Survival)

Fig.a shows in training data 549 are survived and 342 are dead.

Fig.b show out of 891 Passengers in training data 577 are male and 314 are female.

Fig.c show distribution of sex according to survival.

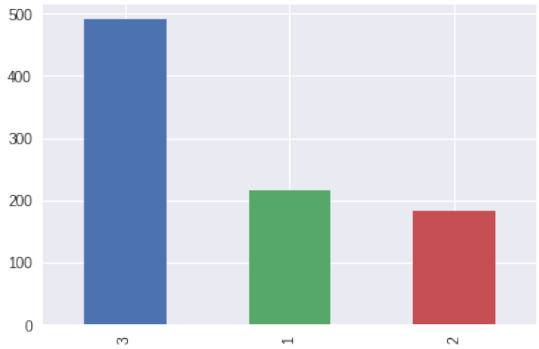


Fig.c Pclass

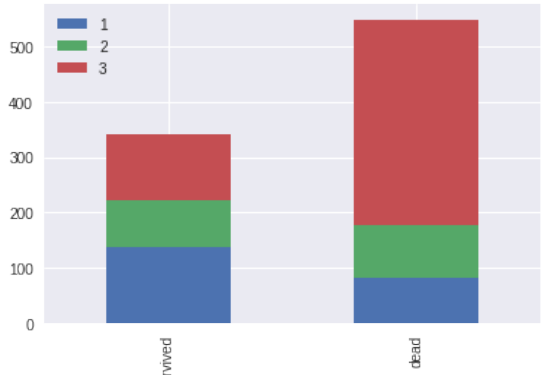


Fig.d Pclass+Survival

Fig.c and Fig.d represents there are three classes having their respective number of passengers. Pclass 1 has 261, Pclass2 has 180 and Pclass3 has 491 passengers.

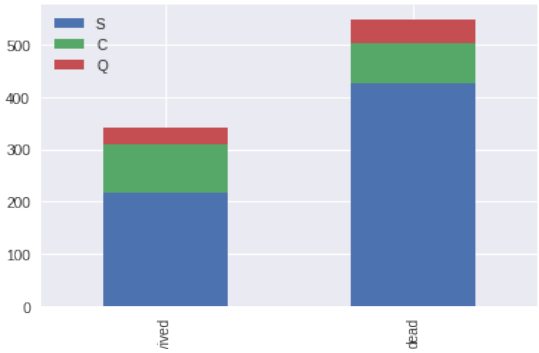


Fig.e Embarked+ Survival

Fig.e represents passengers were embarked for three ports. They were symbolled in data as C for Cherbourg, Q for Queenstown and S for Southampton.

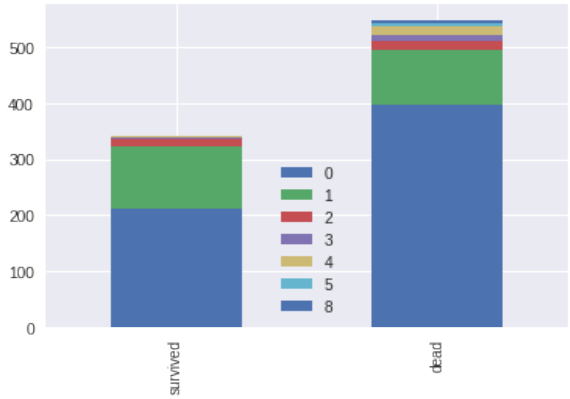


Fig.f SibSp + Survival

Fig.f represents SibSp attribute is for siblings and spouse like brother, sister, stepbrother, spouse like husband or wife of passenger aboard on *Titanic*.

Pre-processing techniques

Both train and test data was combined under the name Titanic and preprocessed for some attributes like Name, Fare, Age, Embarked.

Name: attribute various title of passenger's name has been mapped to common title like Mr., Mrs., Miss and Others followed by converting and mapping it to categorical variable like 0,1,2 and 3. Name attribute has been replaced by **Title** attribute in dataset. The bar graph below is showing title and survival relationship. Fig.g represents Title and survival relation.

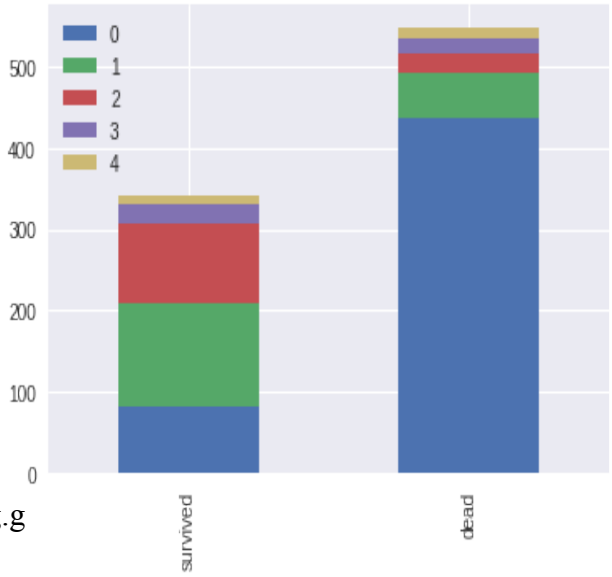


Fig.g Title+ Survival

Age: attribute has been grouped in range and converted from numerical to categorical variable. variable has missing value that has been filled by median age of title value accordingly.

Sex: variable has been converted to categorical. Female and male has been mapped to 1 and 0 respectively.

Family_size : A new attribute has been created by merging **SibSp** and **Parch** attribute which is for parent and child.

Embarked: It has two missing values which has been replaced by mode of the variable and made categorical.

Fare: It has been made categorical like 0,1,2,3 according to range of ticket fare.

Finally less significant attributes like PassengerId, Name, SibSp, Parch, Ticket, Cabin has been removed from dataset.

Proposed Solution and Methods

Data Normalization: I have normalized the attributes so that each variable will have mean =0 and standard deviation=1

Data splitting: I have split the training data into 80:20 ratio of train and test for model fitting and validation.

The function train_test_split will make 80% data as training data which will fit the model on X_train and y_train and 20% data as test data to predict the model as the accuracy of the fitted model.

GridSearch: I have used the GridSearchCV function which do exhaustive search over specified parameter values for an estimator. GridSearch find all the combinations of parameter and give the best combination of the best parameter.

AUC: I have used the area under curve to measure the effectiveness of the model. It's value lies between 0 and 1.

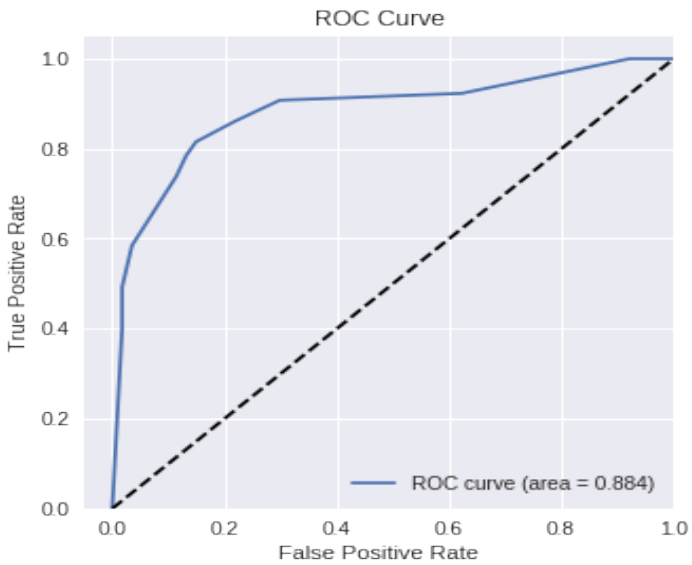
ROC: I have plotted the receiver operating characteristic curve between true positive rate (TPR)and False positive rate(FPR) for different classification models.

I have used the Scikit- learn machine learning library to import the various classifier like DecisionTreeClassifier, KNeighborsClassifier, GaussianNB, LogisticRegression, MLPClassifier, RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier, keras and xgboost classifier.

K-Nearest neighbors:

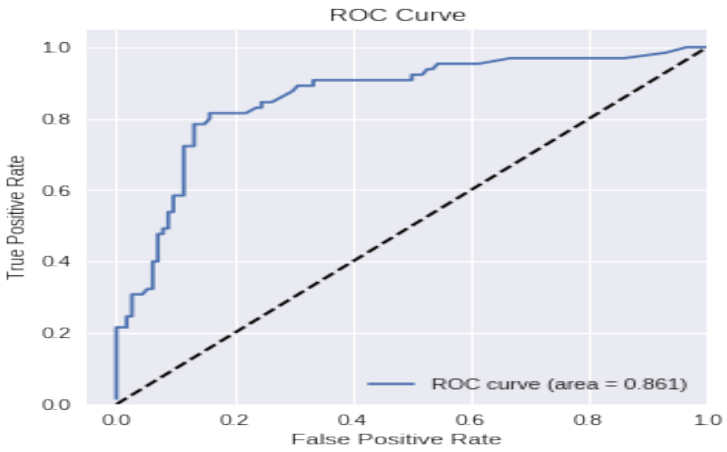
Best parameters set found on development set:

{'algorithm': 'auto', 'metric': 'minkowski',
'n_neighbors': 10, 'p': 1, 'weights': 'uniform'}
confusion matrix:
[[99 15]
[14 51]]
Accuracy Score: 0.8379888268156425
ROC AUC: 0.884



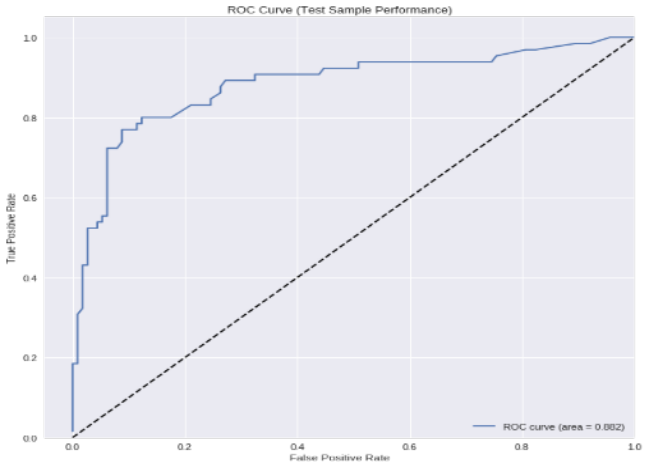
Gaussian naïve Bayes

Best parameters:
{'priors': (0.7, 0.3)}
confusion matrix:
[[96 18]
 [13 52]]
Accuracy Score: 82.7%
ROC AUC: 0.861



Logistic Regression

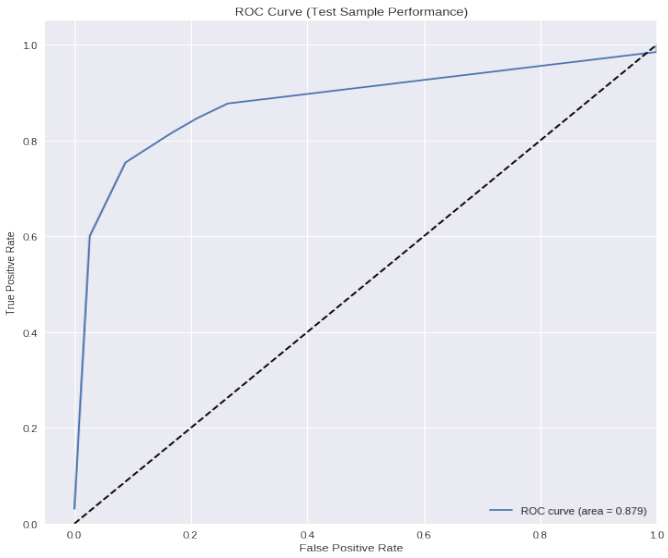
Best parameters set found on development set:
{'C': 10, 'max_iter': 100, 'multi_class': 'ovr', 'random_state': 19, 'tol': 0.0001}
confusion matrix:
[[98 16]
 [13 52]]
Accuracy Score: 83.8%
ROC AUC: 0.882



DecisionTree Classifier

Best parameters:

{'max_depth': 5, 'max_leaf_nodes': 10, 'min_samples_leaf': 2, 'min_samples_split': 2, 'random_state': 11}
confusion matrix:
[[104 10]
 [16 49]]
Accuracy Score: 85.47%
ROC AUC: 0.882



Support Vector Machine

Best parameters set found on development set:
{'C': 1, 'gamma': 0.1, 'kernel': 'rbf', 'max_iter': 1000, 'random_state': 10}
Detailed confusion matrix:
[[107 7]
 [17 48]]
Accuracy Score: 86.6%

Multi-layer Perceptron (MLP)

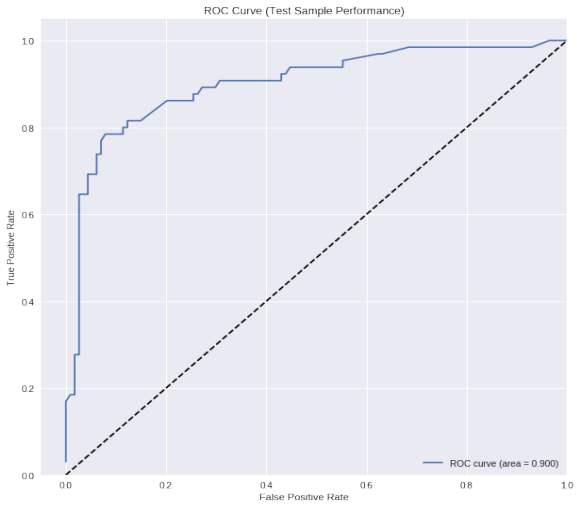
Best parameters set found on development set:

{'activation': 'relu', 'alpha': 1, 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'random_state': 11}
Detailed confusion matrix:
[[104 10]
 [14 51]]
Accuracy Score: 86.6%

RandomForest Classifier

Best parameters:

{'criterion': 'entropy', 'max_depth': 5, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}
Detailed confusion matrix:
[[103 11]
 [14 51]]
Accuracy Score: 86%
ROC AUC: 0.901



Gradient Boosting Classifier

Best parameters set found on development set:

{'learning_rate': 1, 'max_depth': 4, 'max_features': 4, 'n_estimators': 1000}

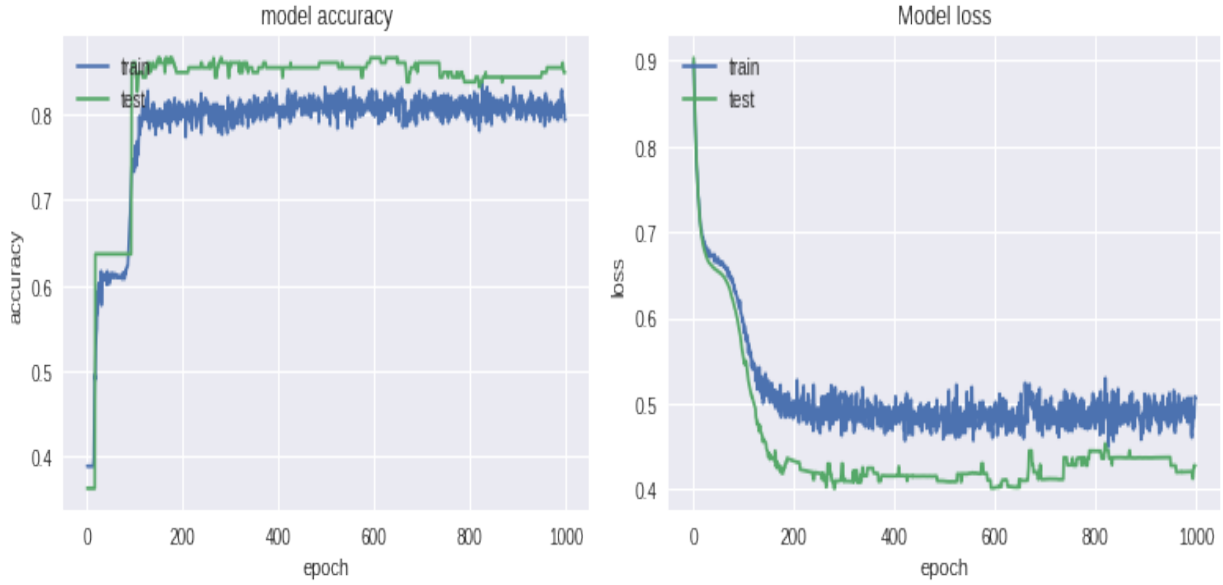
Detailed confusion matrix:

[[99 15]
 [14 51]]
Accuracy Score: 83.8%
ROC AUC: 0.832

Keras Neural Net

Best parameters:
Hidden layer of one size 8
Activation = ‘sigmoid’
Dropout = 20%
Second hidden layer size 4
Third hidden layer of size 2
Outer layer of size 1
Accuracy Score: 85%

The dropout helps us get rid of overfitting. For Binary classification I have used sigmoid activation function.



It is clear from the curves that accuracy for test is better than train set and loss is less for test than that of train set. These characters are good for a predictive model.

XGBoost classifier

Best parameters set found on development set:
{'learning_rate': 0.05, 'max_delta_step': 1, 'n_estimators': 100, 'seed': 1234}
Detailed confusion matrix:
[[103 11]
 [14 51]]
Accuracy Score: 86%

Experimental results and analysis

I have used ten machine learning algorithm for fitting the model for *Titanic* dataset like DecisionTree classifier, Multi-Layer Perceptron, Support vector Machine, Gaussian naïve Bayes, Logistic Regression, K-Nearest neighbors, GradientBoost classifier, XGBoost and RandomForest classifier. I got average 85% accuracy. Strong classifiers like MLP, SVM, XGboost, RandomForest and Keras Neural Net has given average accuracy of 86%.

Tuning of the parameters with GridSearch increased the accuracy of classifiers by 10% from average of 76% obtained by simple train_test split of 80:20 ratio.

Multi-layer perceptron (MLP) is a class of artificial neural network and it is a strong classifier. It used the best parameter for the dataset like Relu, nonlinear activation function and constant learning rate to give maximum accuracy of 86.6%.

Below is the table showing comparison between performance of different classifier.

Comparison of performance of Classifiers

Algorithm	Avg Precision	Avg Recall	Avg F1	Accuracy score %
DecisionTree	0.85	0.85	0.85	85.3
MLP	0.86	0.87	0.86	86.6
Support vector Machine	0.87	0.87	0.86	86.5
Gaussian Naive Bayes	0.83	0.83	0.83	82.7
LogisticRegression	0.84	0.84	0.84	83.7
kNN	0.84	0.84	0.84	83.7
GradientBoost	0.84	0.84	0.84	83.7
XGBoost	0.86	0.86	0.86	86
RandomForest	0.86	0.86	0.86	86.0

Conclusion

In this project survival of the passengers from the Titanic disaster has been predicted with machine learning model. I have tried to tune the model with combination of best parameter to give maximum accuracy of the model on the dataset. Before fitting the model date has been preprocessed to avoid overfitting of the model due to co-relation of the data and unnecessary features. It is clear from the result table that strong and powerful learner has shown better accuracy for the dataset than that of the week learner. However, the accuracy of the model could be increased with more feature engineering.

References

- https://scikit-learn.org/stable/
- https://keras.io/
- Ekinci, Ekin & Omurca, Sevinc & Acun, Neytullah. (2018). A Comparative Study on Machine Learning Techniques Using Titanic Dataset.
- Farang N, Hassan G. (2018) Predicting the survivors of the Titanic Kagglle, machine learning from disaster. ICSIE'18 Proceedings of the 7th International Conference on Software and Information Engineering. 32-37