Report on

Titanic Machine Learning problem

by

Name: Anamika Chatterjee

St. ID: 202004102

Email: x2020dxz@stfx.ca

Problem Definition:

The titanic dataset is available on Kaggle website, we need to predict the survivors based on 10 features available in the dataset by using different machine learning techniques. This problem can be used to compare different machine learning model performances.

Approach:

In this implementation I compared the performance of mainly two classifiers: Random Forest classifier and Gradient Boosting classifier.

Steps taken:

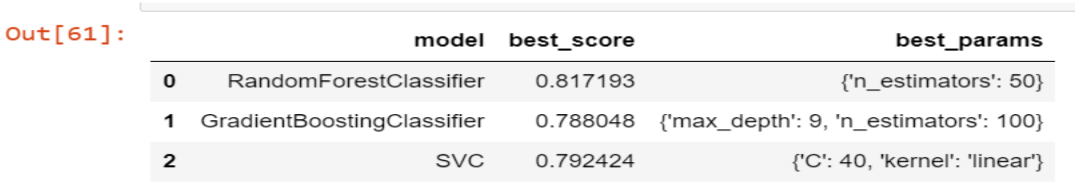
1. Load the train dataset using pd.read_csv.
2. Check for null values in the dataset.
3. Fill the null value of Age with the mean value of Age column.
4. Fill the null value of Embarked column with the mode of the column i.e., ‘S’.
5. Drop inconsequential features like PassengerId, Name, Ticket, Cabin.
6. Convert categorical values of Embarked and sex columns to numerical data. Assign numerical value to S, C and Q of Embarked column and male, female of Sex column.
7. Normalize the value of Age and Fare column using StandardScaler()
8. Import grid search cross validation function. This is used to compare different models with helps in hyper parameter tuning.
9. Import different classifier function.
10. Fit the train data and get the comparison between different models.
11. Load the test data clean and treat the test data same way as the train data.
12. Fit the most well performing model and predict the survivor for test data.

Attempts:

1st Attempt: -

Data cleaning method: At first, I dropped all the features like PassengerID, Embarked, Cabin, SibSp, Parch, Name, Ticket. I also used dummy values to convert the categorical data of Sex column.

Classifiers compared: RandomForest, GradientBoosting, SVC



Out[61]:

	model	best_score	best_params
0	RandomForestClassifier	0.817193	{'n_estimators': 50}
1	GradientBoostingClassifier	0.788048	{'max_depth': 9, 'n_estimators': 100}
2	SVC	0.792424	{'C': 40, 'kernel': 'linear'}

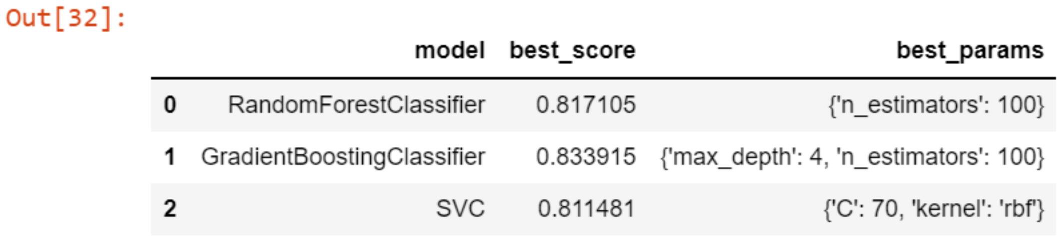
Classifier used with test data: SVC (C=40, kernel= ‘linear’)

Accuracy with test data: 75.5%

2nd Attempt: -

Data cleaning method: I added back the Embarked, SipSb, Parch features.

Classifiers compared: RandomForest, GradientBoosting, and SVC



Out[32]:

	model	best_score	best_params
0	RandomForestClassifier	0.817105	{'n_estimators': 100}
1	GradientBoostingClassifier	0.833915	{'max_depth': 4, 'n_estimators': 100}
2	SVC	0.811481	{'C': 70, 'kernel': 'rbf'}

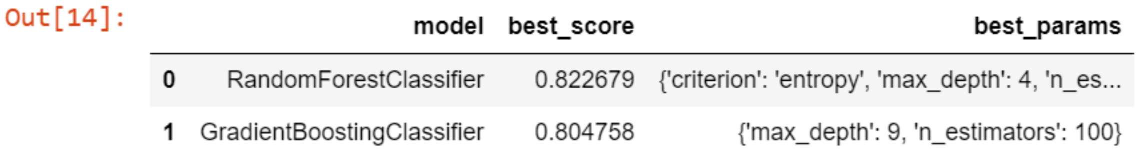
Classifier used with test data: GradientBoostingClassifier(max_depth =4, n_estimators=100)

Accuracy with test data: 77.03%

Final Attempt:

Data cleaning method: Instead of using dummy variable for I replaced the of categorical value with numerical data with .replace() function. For Embarked feature S=1, C=2, Q=3; for Sex feature male=0 and female=1.

Classifiers compared: RandomForest, GradientBoosting



Out[14]:

	model	best_score	best_params
0	RandomForestClassifier	0.822679	{'criterion': 'entropy', 'max_depth': 4, 'n_es...
1	GradientBoostingClassifier	0.804758	{'max_depth': 9, 'n_estimators': 100}

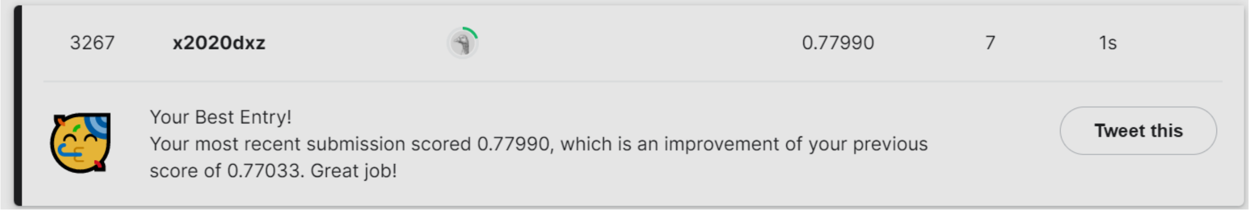
Classifier used with test data: RandomForestClassifier(n_estimators=250, max_depth=4, criterion= ‘entropy’)

Accuracy with test data: 77.99%

Attempts summary table:

S.no.	Classifier	Score
1	SVC(C=40, kernel= ‘linear’)	75.5%
2	GradientBoostingClassifier(max_depth =4, n_estimators=100)	77.03%
3	RandomForestClassifier(n_estimators=250, max_depth=4, criterion= ‘entropy’)	77.99%

Public Leaderboard Rank :



3267 x2020dxz 0.77990 7 1s

Your Best Entry!
Your most recent submission scored 0.77990, which is an improvement of your previous score of 0.77033. Great job!