Documentation for Steps of MLflow

WEEK: 2 - DATA OPS, MLOPS, AIOPS ASSIGNMENT: (TITANIC DATASET)

Gaurav Pal
GREAT LEARNING

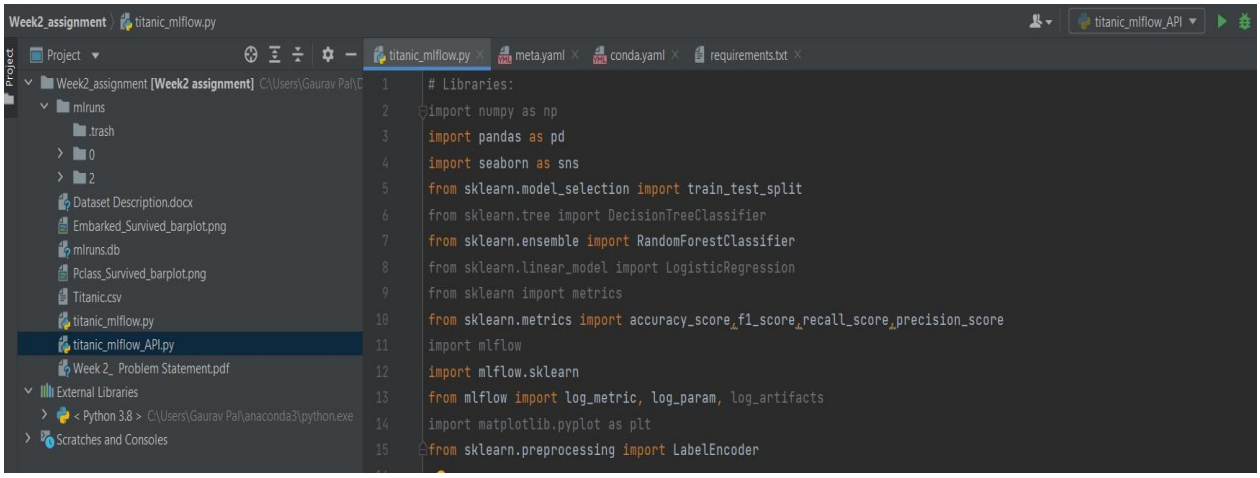
Great Learning
POWER AHEAD

Objective:

To use mlflow for logging and querying machine learning experiments. The task is to build, log, and track multiple versions of a classification model that aims to predict the passengers who survived the titanic shipwreck.

Approaches and Documentation of Steps (Using screen shot):

Installing and importing relevant libraries to solve the problem.



Loaded the given data (Titanic.csv).

Exploratory Data Analysis:

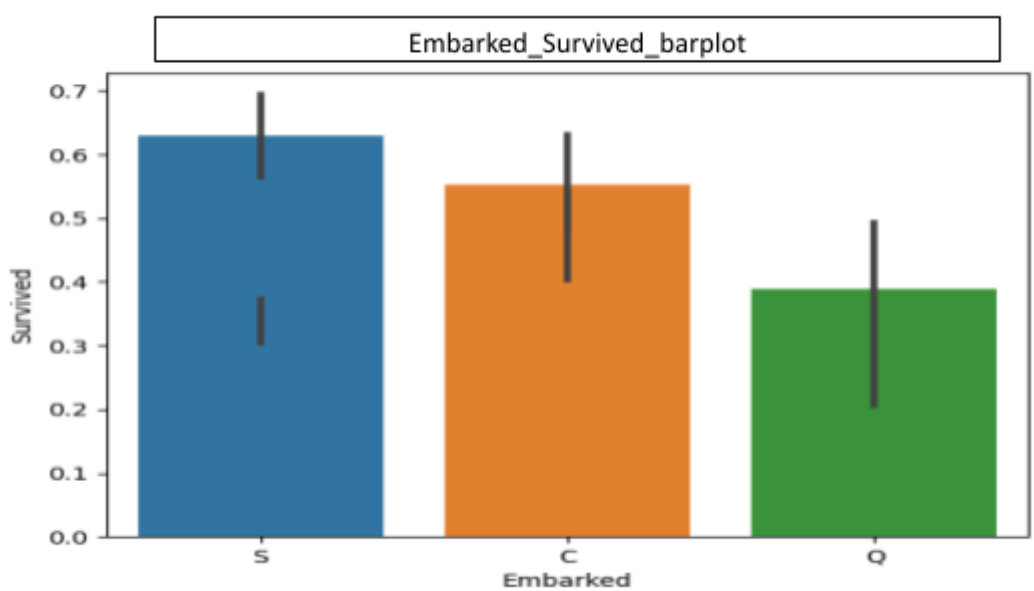
Checking whether any null values present in the dataset or not. As the titanic_df['Age'], titanic_df['cabin'] & titanic_df['Embarked'] has null values and replacing Null values of titanic_df['Age'] & titanic_df['Embarked'] with median and mode values.

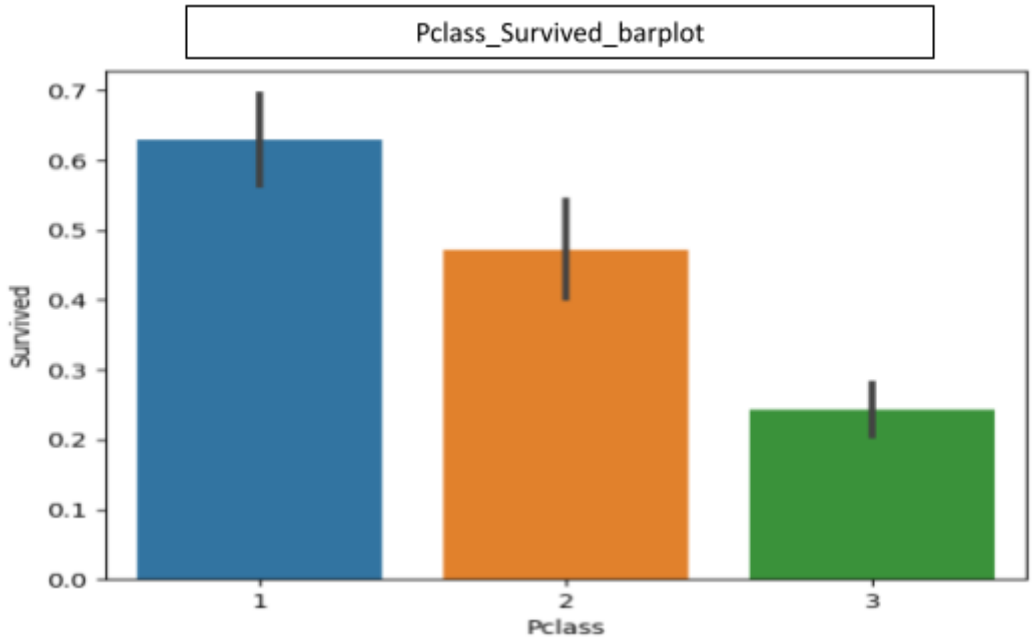
"Pclass_Survived_barplot.png" bar plot graph shows the analysis between titanic_df['Pclass'] and titanic_df['Survived'] column. "Embarked_Survived_barplot.png" bar plot shows the analysis between titanic_df['Embarked'] and the titanic_df['Survived']. Logged these local files as an artifact using mlflow.log_artifact() tracking command.

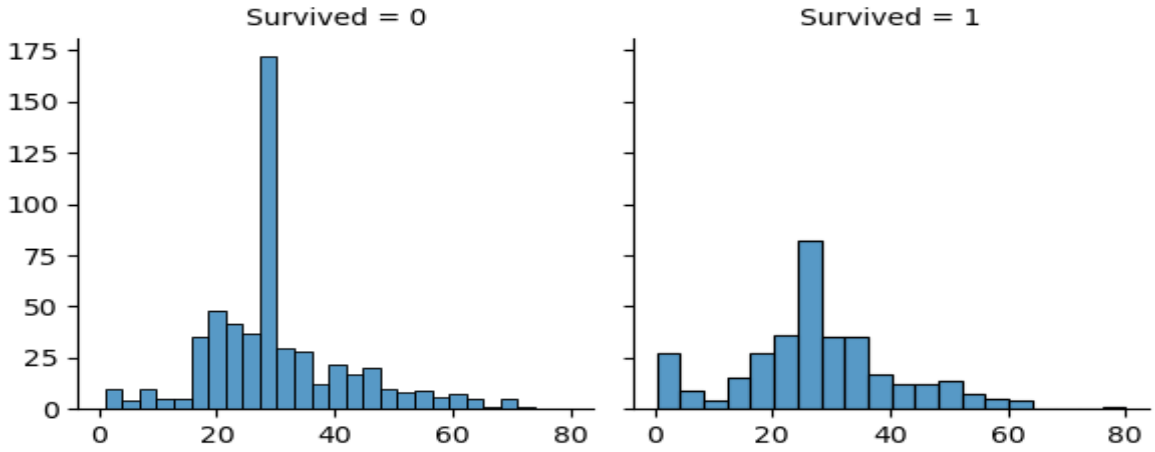
titanic_mlflow.py | Embarked_Survived_barplot.png | requirements.txt

```python

print(titanic_df.isnull().sum()) # Checking whether any null values in the dataset.

# As the titanic_df['Age'], titanic_df['cabin'] & titanic_df['Embarked'] has null values.
# replacing null values of titanic_df['Age'] & titanic_df['Embarked'] with median and mode values
titanic_df['Age'] = titanic_df['Age'].fillna(titanic_df.Age.median())

print(titanic_df['Embarked'].value_counts())

#  imputing the null values with the most frequent values in the Embarked column.

titanic_df['Embarked'] = titanic_df['Embarked'].fillna('S')
print(titanic_df.isnull().sum())

fig = sns.barplot(titanic_df['Pclass'], titanic_df['Survived'], data=titanic_df)
# Saving the Seaborn Figure:
fig.figure.savefig("Pclass_Survived_barplot.png")
mlflow.log_artifact("Pclass_Survived_barplot.png")

fig1 = sns.barplot(titanic_df['Embarked'], titanic_df['Survived'], data=titanic_df)
# Saving the Seaborn Figure:
fig1.figure.savefig("Embarked_Survived_barplot.png")
mlflow.log_artifact("Embarked_Survived_barplot.png")

print(titanic_df['Survived'].value_counts())
log_param("Value counts", titanic_df['Survived'].value_counts())
```

```python
# relationship between Age and Survival
g = sns.FacetGrid(titanic_df, col="Survived")
g.map_dataframe(sns.histplot, x='Age')
g.savefig("Age_Survival.png")
mlflow.log_artifact("Age_Survival.png")
```




Pclass_Survived_barplot
Survived
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1
2
3
Pclass


Survived = 0
Survived = 1
0
25
50
75
100
125
150
175
0
20
40
60
80

As the Sex column & Embarked columns are Categorical columns so we converted the values with the sklearn Label encoder so that it can fit for model building.

```
# Converting Sex column & Embarked column value to Catagorical Value:
le = LabelEncoder()
titanic_df['Embarked'] = le.fit_transform(titanic_df['Embarked'])
print(titanic_df['Embarked'].value_counts())
titanic_df['Sex'] = le.fit_transform(titanic_df['Sex'])
print(titanic_df['Sex'].value_counts())

# As a part of data cleaning we have to drop the column like PassengerId, Name, Ticket, Cabin
# inorder to increase the overall accuracy and efficiency of the data.

titanic_df = titanic_df.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1)
print(titanic_df.head())
X_train, X_test, y_train, y_test = train_test_split(titanic_df.drop('Survived', axis=1), titanic_df['Survived'],
                                                    test_size=.3,
                                                    random_state=20)
print(X_train.shape, X_test.shape)
log_param("Train Shape", X_train.shape)
```

As a part of data cleaning, we have to drop the column like PassengerId, Name, Ticket, Cabin in order to increase the overall accuracy and efficiency of the data and also split the data using sklearn sklearn train test split.

Model Building: I have used the Random Forest classifier to train the model. Please refer to the below code snippet for that and also evaluate the model performance using scores of accuracies, precision, recall and f1 score, logged the metrices to the mlflow, and also logged the model to mlflow.

```python
rf_model = RandomForestClassifier(n_estimators=100, criterion="gini", max_depth=100,
                                  min_samples_leaf=10, random_state=20, )

rf_model.fit(X_train, y_train)

print("*****Model trained*****")
y_pred = rf_model.predict(X_test)

train_accuracy = rf_model.score(X_train,y_train) #performence on training set
print("train accuracy: ", train_accuracy)
test_accuracy = rf_model.score(X_test, y_test) #performence on test set
print("test accuracy: ", test_accuracy)
accuracy = accuracy_score(y_true=y_test, y_pred=y_pred) #Accuracy Score
p_score = precision_score(y_true=y_test, y_pred=y_pred) #Precision Score
recall = recall_score(y_true=y_test, y_pred=y_pred) #Recall Score
f1 = f1_score(y_true=y_test, y_pred=y_pred) #f1 Score

print('Precision: %.3f' % p_score)
print('Recall: %.3f' % recall)
print('Accuracy: %.3f' % accuracy)
print('F1 Score: %.3f' % f1)
# log_metric("Accuracy for this run", test_accuracy)

# Log in mlflow (metrics)
log_metric("Precision Score", p_score)
log_metric("Recall Score", recall)
log_metric("Accuracy Score", accuracy)
log_metric("F1 Score", f1)


mlflow.sklearn.log_model(rf_model, "RF Model1")
print(mlflow.active_run().info.run_uuid)
if __name__ == '__main__'
```

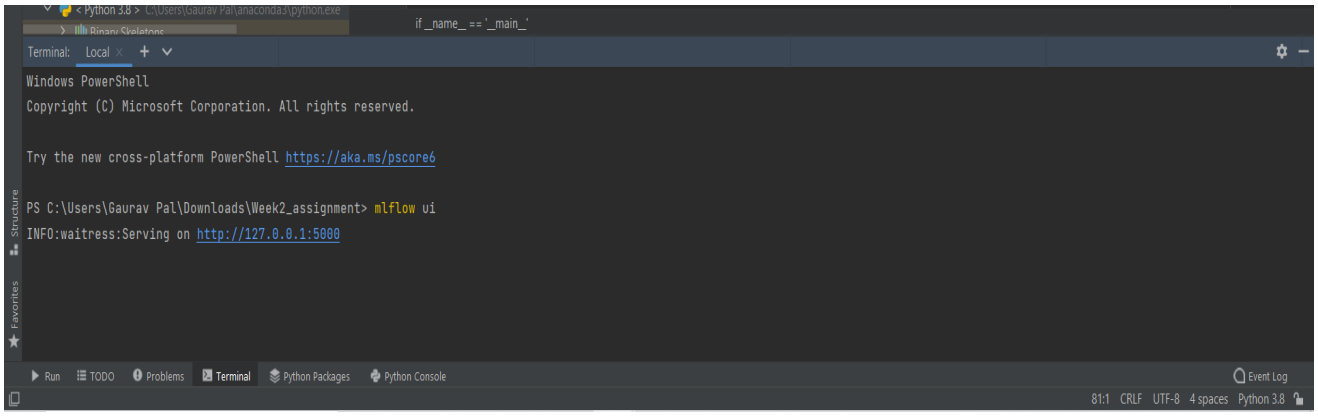
OUTPUT:

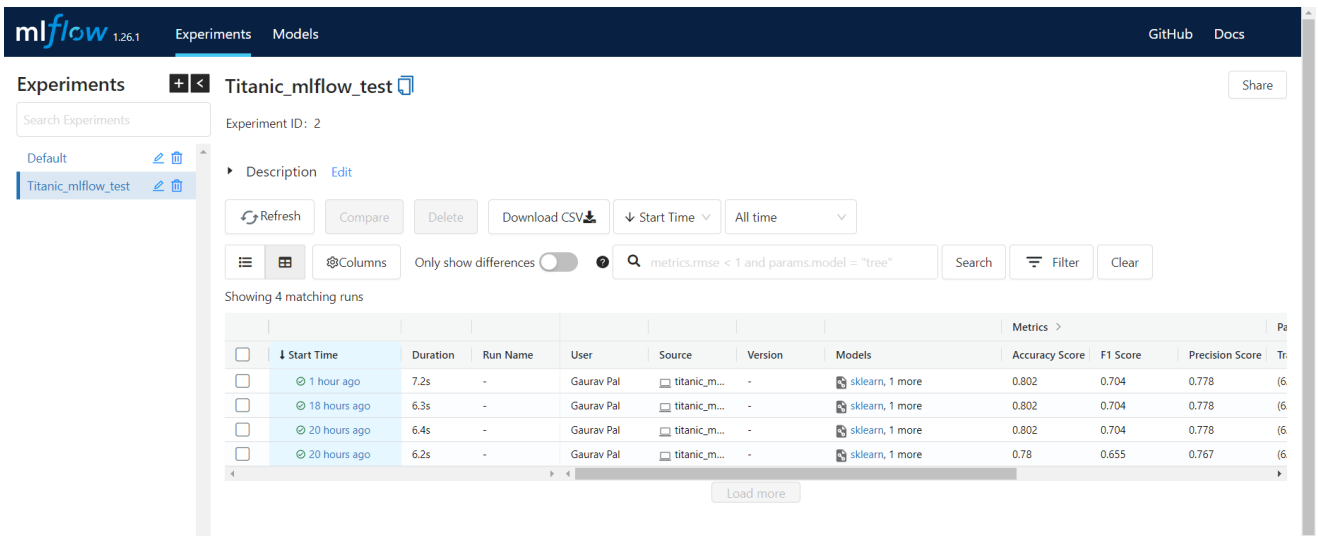
```
"C:\Users\Gaurav Pal\anaconda3\python.exe" "C:/Users/Gaurav Pal/Downloads/Week2_assignment/titanic_mlflow.py"
***Initializing the Experiment***
2022/06/11 16:32:39 INFO mlflow.tracking.fluent: Autologging successfully enabled for sklearn.
**Loading the data**
2022/06/11 16:32:40 INFO mlflow.tracking.fluent: Autologging successfully enabled for statsmodels.
C:\Users\Gaurav Pal\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid posit
  warnings.warn(
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
S    644
C    168
Q     77
Name: Embarked, dtype: int64
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         0
dtype: int64
C:\Users\Gaurav Pal\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid posi
  warnings.warn(
0    549
1    342
Name: Survived, dtype: int64
2    646
0    168
1     77
Name: Embarked, dtype: int64
1    577
0    314
Name: Sex, dtype: int64
   Survived  Pclass  Sex   Age  SibSp  Parch     Fare  Embarked
0         0       3    1  22.0      1      0   7.2500         2
1         1       1    0  38.0      1      0  71.2833         0
2         1       3    0  26.0      0      0   7.9250         2
3         1       1    0  35.0      1      0  53.1000         2
4         0       3    1  35.0      0      0   8.0500         2
(623, 7) (268, 7)
```

```
2022/06/11 16:32:42 WARNING mlflow.utils.autologging_utils: MLflow autologging encountered a warning: "C:\Users\Gaurav Pal\anaconda3\lib\site-packages\mlflow\models\signature.py:12
*****Model trained*****
train accuracy:  0.841091492776886
test accuracy:  0.8022388059701493
Precision: 0.778
Recall: 0.643
Accuracy: 0.802
F1 Score: 0.704
216944ba0b3c41b4904c6509288b8d63

Process finished with exit code 0
```

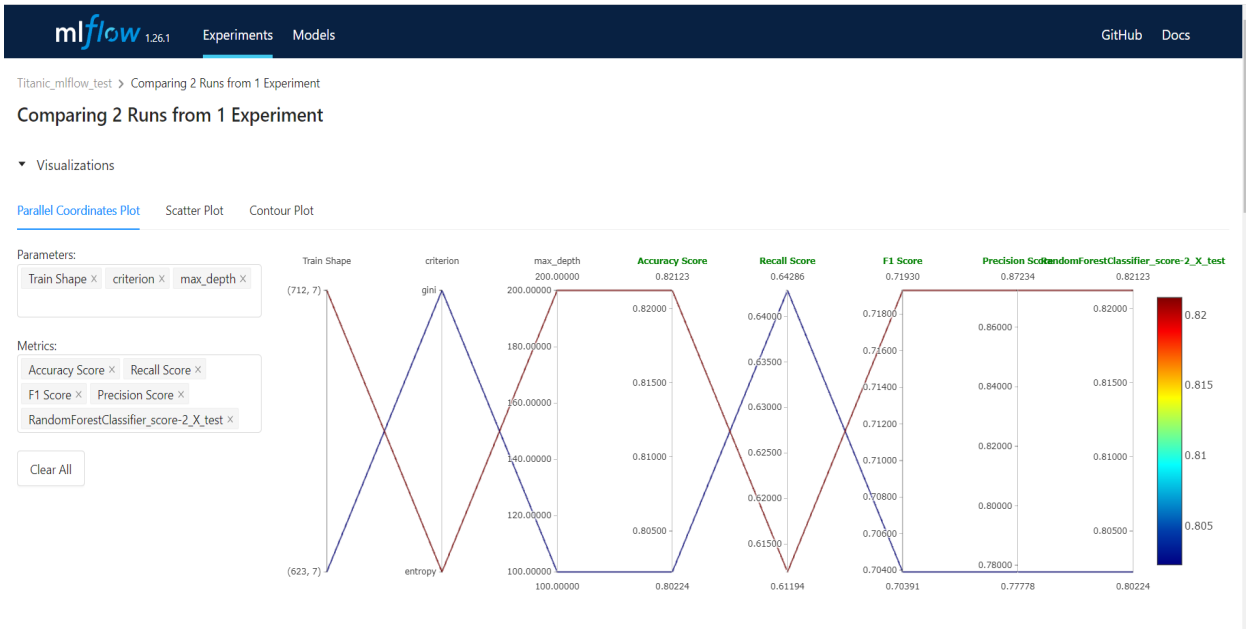
<u>Mlflow tracking in the UI</u>: I have used the "**mlflow ui**" command at the terminal for tracking and analyzing the created model.





Comparing the Two models:

Run ID:	8692c9a694f0476695524c1376dfd442	216944ba0b3c41b4904c6509288b8d63



Parameters

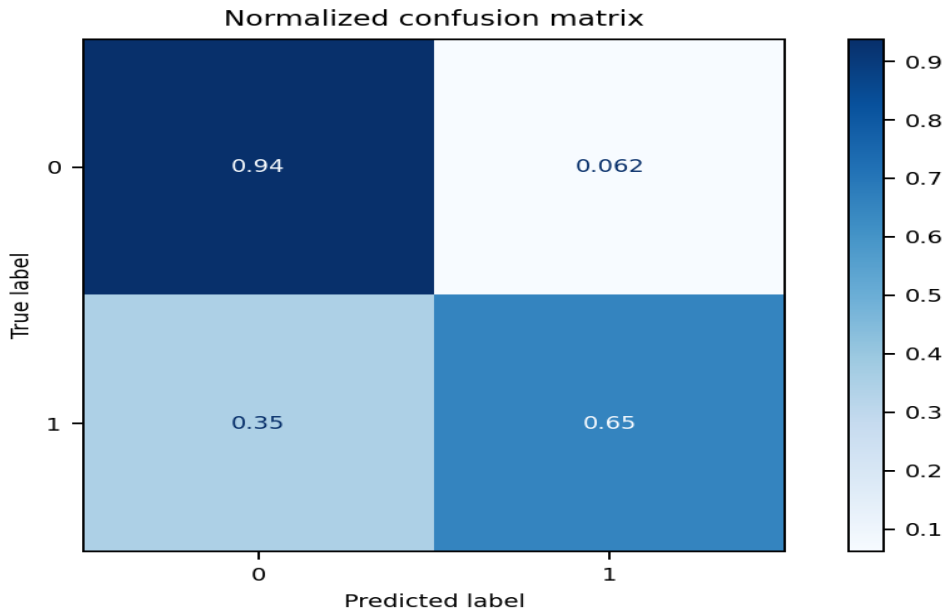
Show diff only

Train Shape	(712, 7)	(623, 7)
criterion	entropy	gini
max_depth	200	100
min_samples_leaf	20	10
n_estimators	200	100

Metrics

Show diff only

Accuracy Score	0.821	0.802
F1 Score	0.719	0.704
Precision Score	0.872	0.778
RandomForestClassifier_score-2_X_test	0.821	0.802
RandomForestClassifier_score_X_train	0.827	0.841
Recall Score	0.612	0.643


Normalized confusion matrix
0.94
0.062
0.35
0.65
True label
Predicted label
0
1
0
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1