

1) Install mlflow and required libraries:

```
Select Anaconda Prompt (anaconda) - pip install numpy pandas matplotlib seaborn scikit-learn mlflow dvc

(base) C:\Users\mohdm>conda create -n mlflow_titanic python=3.8
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.14.0
  latest version: 22.9.0

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\mohdm\anaconda\envs\mlflow_titanic
  added / updated specs:
    - python=3.8

The following packages will be downloaded:

  package | build | size
  -----|-----|-----
  python-3.8.15 | h82bb817_0 | 16.6 MB
  -----|-----|-----
  Total: | | 16.6 MB

The following NEW packages will be INSTALLED:

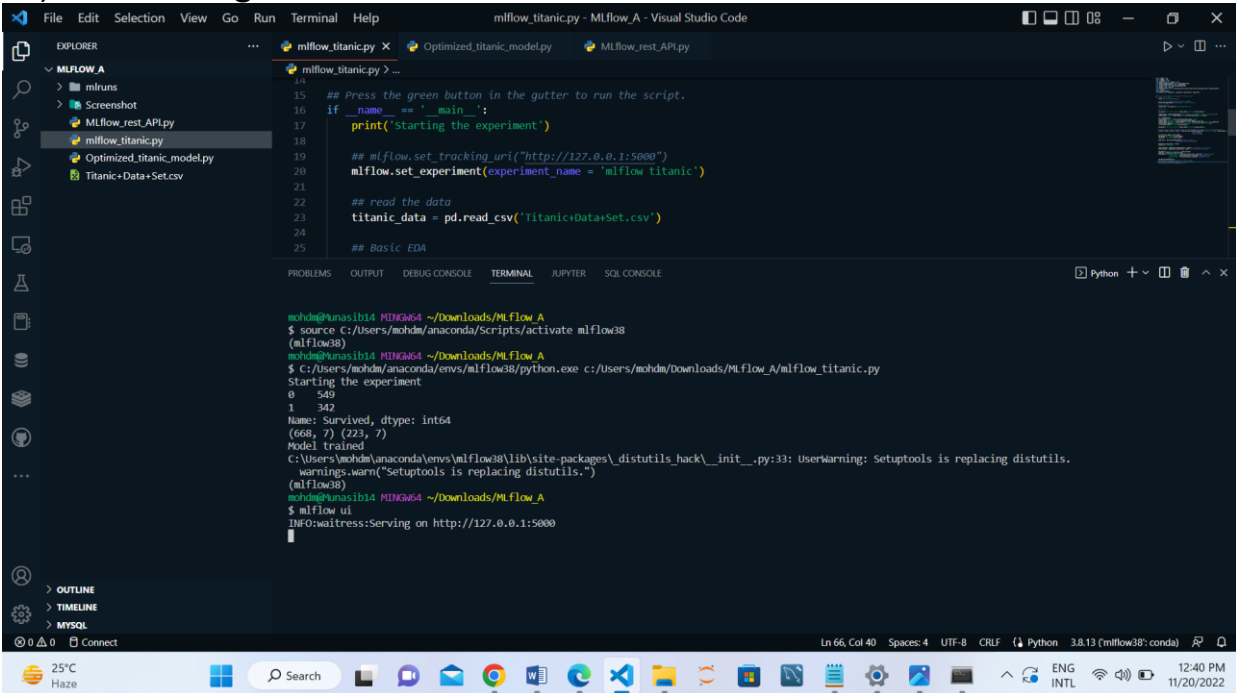
  ca-certificates | pkgs/main/win-64::ca-certificates-2022.10.11-haa95532_0
  certifi | pkgs/main/win-64::certifi-2022.9.24-py38haa95532_0
  openssl | pkgs/main/win-64::openssl-1.1.1s-h2b6fffb_0
  pip | pkgs/main/win-64::pip-22.2.2-py38haa95532_0
  python | pkgs/main/win-64::python-3.8.15-h82bb817_0
  setuptools | pkgs/main/win-64::setuptools-65.5.0-py38haa95532_0
```

```
Select Anaconda Prompt (anaconda) - pip install numpy pandas matplotlib seaborn scikit-learn mlflow dvc
Retrieving notices: ...working... done

(base) C:\Users\mohdm>conda activate mlflow_titanic

(mlflow_titanic) C:\Users\mohdm>pip install numpy pandas matplotlib seaborn scikit-learn mlflow dvc
Collecting numpy
  Downloading numpy-1.23.5-cp38-cp38-win_amd64.whl (14.7 MB)
  -----|----- 14.7/14.7 MB 4.6 MB/s eta 0:00:00
Collecting pandas
  Using cached pandas-1.5.1-cp38-cp38-win_amd64.whl (11.0 MB)
Collecting matplotlib
  Using cached matplotlib-3.6.2-cp38-cp38-win_amd64.whl (7.2 MB)
Collecting seaborn
  Using cached seaborn-0.12.1-py3-none-any.whl (288 kB)
Collecting scikit-learn
  Using cached scikit_learn-1.1.3-cp38-cp38-win_amd64.whl (7.5 MB)
Collecting mlflow
  Using cached mlflow-2.0.1-py3-none-any.whl (16.5 MB)
Collecting dvc
  Downloading dvc-2.34.2-py3-none-any.whl (387 kB)
  -----|----- 387.4/387.4 kB 3.0 MB/s eta 0:00:00
Collecting pytz>=2020.1
  Using cached pytz-2022.6-py2.py3-none-any.whl (498 kB)
Collecting python-dateutil>=2.8.1
  Using cached python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
Collecting cycler>=0.10
  Using cached cycler-0.11.0-py3-none-any.whl (6.4 kB)
Collecting fonttools>=4.22.0
  Using cached fonttools-4.38.0-py3-none-any.whl (965 kB)
Collecting kiwisolver>=1.0.1
  Using cached kiwisolver-1.4.4-cp38-cp38-win_amd64.whl (55 kB)
Collecting packaging>=20.0
  Using cached packaging-21.3-py3-none-any.whl (40 kB)
Collecting pillow>=6.2.0
  Using cached pillow-9.3.0-cp38-cp38-win_amd64.whl (2.5 MB)
Collecting pyparsing>=2.2.1
  Using cached pyparsing-3.0.9-py3-none-any.whl (98 kB)
Collecting contourpy>=1.0.1
  Using cached contourpy-1.0.6-cp38-cp38-win_amd64.whl (163 kB)
Collecting scipy>=1.3.2
```

2)Load the given data



```
File Edit Selection View Go Run Terminal Help
mlflow_titanic.py - MLflow_A - Visual Studio Code

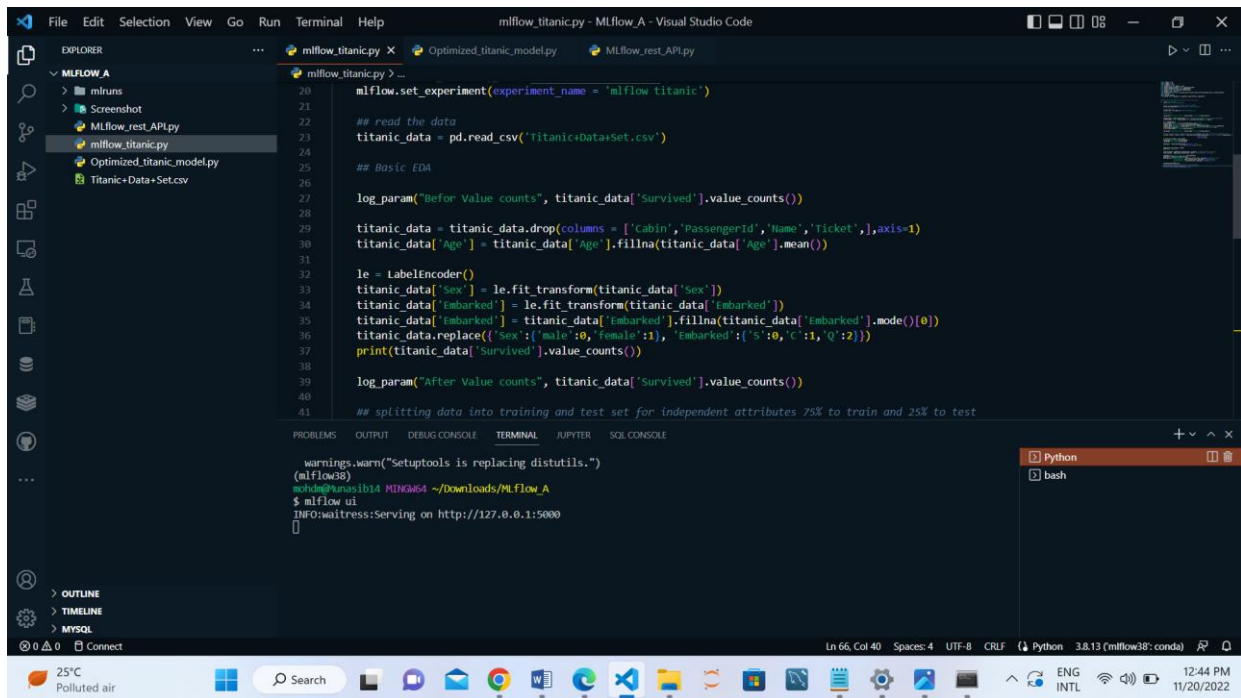
EXPLORER
  MLFLOW_A
    miruns
    Screenshot
    MLflow_rest_API.py
    mlflow_titanic.py
    Optimized_titanic_model.py
    Titanic+Data+Set.csv

mlflow_titanic.py
15  ## Press the green button in the gutter to run the script.
16  if __name__ == '__main__':
17      print('Starting the experiment')
18
19      ## mlflow.set_tracking_uri("http://127.0.0.1:5000")
20      mlflow.set_experiment(experiment_name = 'mlflow titanic')
21
22      ## read the data
23      titanic_data = pd.read_csv('Titanic+Data+Set.csv')
24
25      ## Basic EDA

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER SQL CONSOLE
Python + -

mohdm@Munasi14 MINGW64 ~/Downloads/MLflow_A
$ source C:/Users/mohdm/anaconda/scripts/activate mlflow38
(mlflow38)
mohdm@Munasi14 MINGW64 ~/Downloads/MLflow_A
$ C:/Users/mohdm/anaconda/envs/mlflow38/python.exe c:/Users/mohdm/Downloads/MLflow_A/mlflow_titanic.py
Starting the experiment
0      549
1      342
Name: Survived, dtype: int64
(668, 7) (223, 7)
Model trained
C:/Users/mohdm/anaconda/envs/mlflow38/lib/site-packages/distutils_hack/_init_.py:33: UserWarning: Setuptools is replacing distutils.
warnings.warn("Setuptools is replacing distutils.")
(mlflow38)
mohdm@Munasi14 MINGW64 ~/Downloads/MLflow_A
$ mlflow ui
INFO:waitress:erving on http://127.0.0.1:5000
```

3) Perform basic EDA



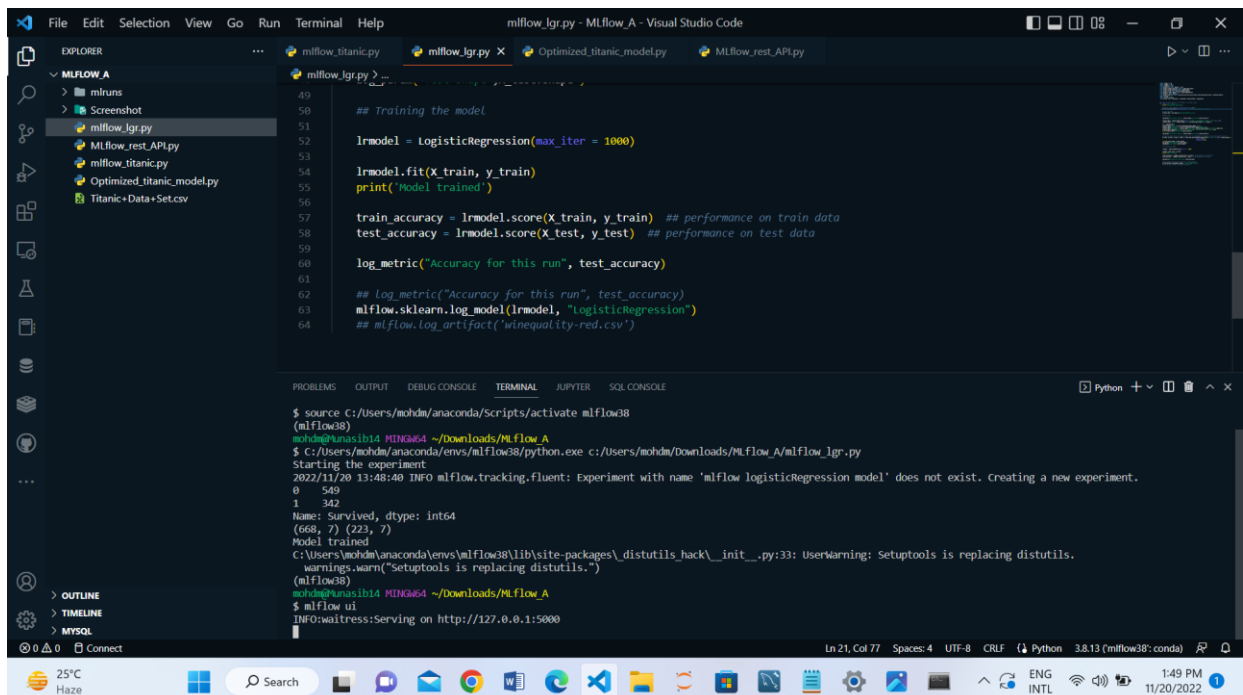
```
File Edit Selection View Go Run Terminal Help
mlflow_titanic.py - MLflow_A - Visual Studio Code

EXPLORER
  MLFLOW_A
    mlruns
    Screenshot
    MLflow_rest_API.py
    mlflow_titanic.py
    Optimized_titanic_model.py
    Titanic+Data+Set.csv

mlflow_titanic.py
20 mlflow.set_experiment(experiment_name = 'mlflow titanic')
21
22 ## read the data
23 titanic_data = pd.read_csv('Titanic+Data+Set.csv')
24
25 ## Basic EDA
26
27 log_param("Before Value counts", titanic_data['Survived'].value_counts())
28
29 titanic_data = titanic_data.drop(columns = ['Cabin','PassengerId','Name','Ticket'],axis=1)
30 titanic_data['Age'] = titanic_data['Age'].fillna(titanic_data['Age'].mean())
31
32 le = LabelEncoder()
33 titanic_data['Sex'] = le.fit_transform(titanic_data['Sex'])
34 titanic_data['Embarked'] = le.fit_transform(titanic_data['Embarked'])
35 titanic_data['Embarked'] = titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0])
36 titanic_data.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}})
37 print(titanic_data['Survived'].value_counts())
38
39 log_param("After Value counts", titanic_data['Survived'].value_counts())
40
41 ## splitting data into training and test set for independent attributes 75% to train and 25% to test

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER SQL CONSOLE
warnings.warn("Setuptools is replacing distutils.")
(mlflow38)
mohd@anasib14 MINGW64 ~/Downloads/MLflow_A
$ mlflow ui
INFO: waitress: Serving on http://127.0.0.1:5000
^
```

4) Training the Logistic Regression Model



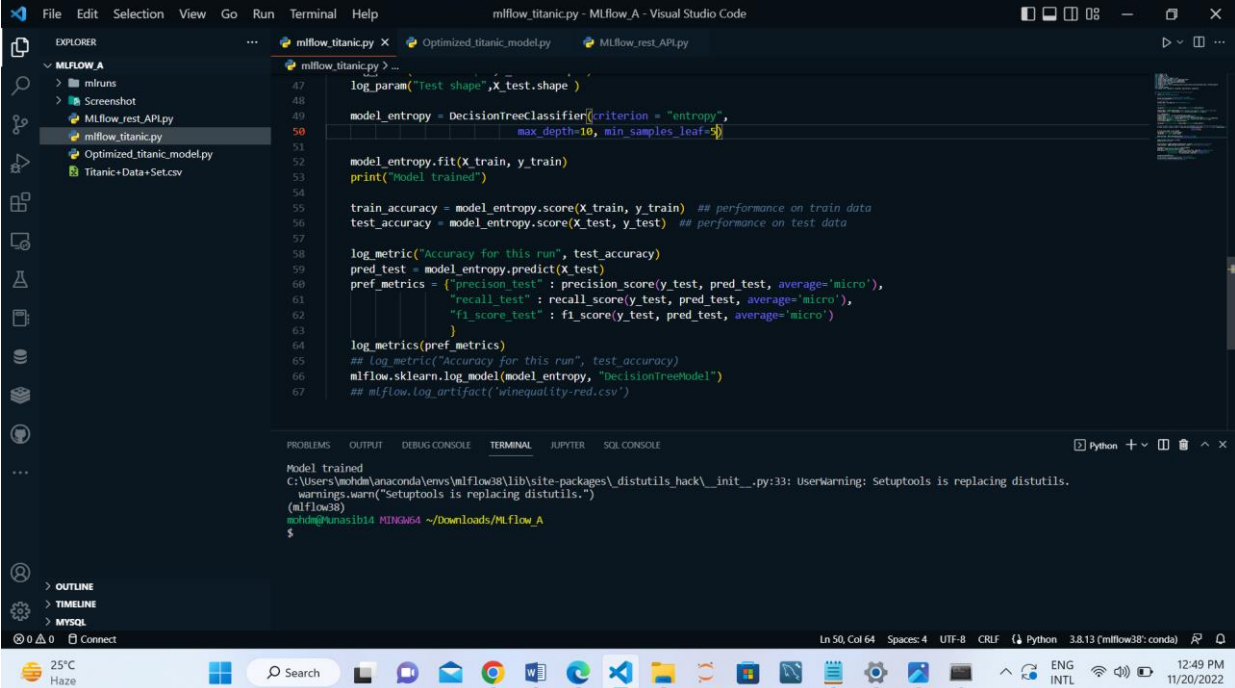
```
File Edit Selection View Go Run Terminal Help
mlflow_lgr.py - MLflow_A - Visual Studio Code

EXPLORER
  MLFLOW_A
    mlruns
    Screenshot
    MLflow_rest_API.py
    mlflow_titanic.py
    Optimized_titanic_model.py
    Titanic+Data+Set.csv

mlflow_lgr.py
49
50 ## training the model
51 lrmodel = LogisticRegression(max_iter = 1000)
52
53 lrmodel.fit(X_train, y_train)
54 print('Model trained')
55
56
57 train_accuracy = lrmodel.score(X_train, y_train) ## performance on train data
58 test_accuracy = lrmodel.score(X_test, y_test) ## performance on test data
59
60 log_metric("Accuracy for this run", test_accuracy)
61
62 ## Log metric("Accuracy for this run", test_accuracy)
63 mlflow.sklearn.log_model(lrmodel, "LogisticRegression")
64 ## mlflow.log_artifact('winequality-red.csv')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER SQL CONSOLE
$ source C:/Users/mohdm/anaconda/Scripts/activate mlflow38
(mlflow38)
mohd@anasib14 MINGW64 ~/Downloads/MLflow_A
$ c:/Users/mohdm/anaconda/envs/mlflow38/python.exe c:/Users/mohdm/Downloads/MLflow_A/mlflow_lgr.py
Starting the experiment
2022/11/20 13:48:40 INFO mlflow.tracking.fluent: Experiment with name 'mlflow logisticRegression model' does not exist. Creating a new experiment.
0 549
1 342
Name: Survived, dtype: int64
(668, 7) (223, 7)
Model trained
c:/Users/mohdm/anaconda/envs/mlflow38/lib/site-packages/distutils_hack/_init_.py:33: UserWarning: Setuptools is replacing distutils.
warnings.warn("Setuptools is replacing distutils.")
(mlflow38)
mohd@anasib14 MINGW64 ~/Downloads/MLflow_A
$ mlflow ui
INFO: waitress: Serving on http://127.0.0.1:5000
^
```

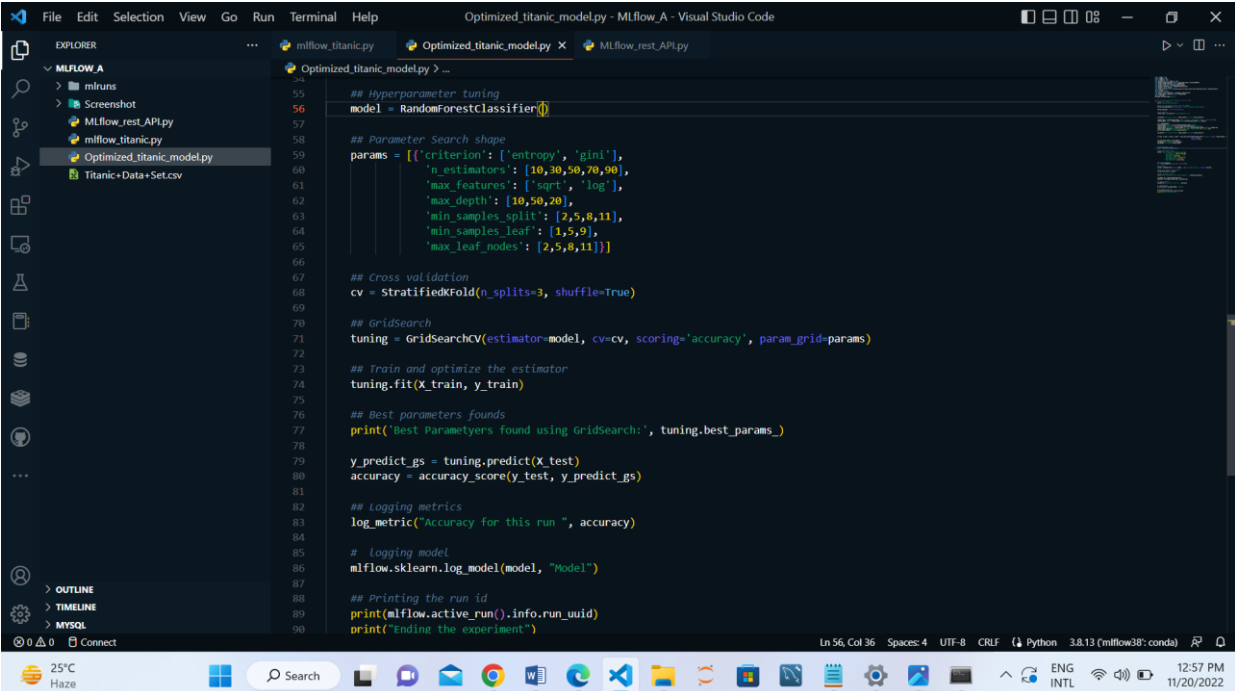
5) Training the Decision Tree Classifier Model



```
47 log_param("Test shape", X_test.shape)
48
49 model_entropy = DecisionTreeClassifier(criterion = "entropy",
50                                     max_depth=10, min_samples_leaf=5)
51
52 model_entropy.fit(X_train, y_train)
53 print("Model trained")
54
55 train_accuracy = model_entropy.score(X_train, y_train) # performance on train data
56 test_accuracy = model_entropy.score(X_test, y_test) # performance on test data
57
58 log_metric("Accuracy for this run", test_accuracy)
59 pred_test = model_entropy.predict(X_test)
60 pref_metrics = {"precision_test": precision_score(y_test, pred_test, average='micro'),
61               "recall_test": recall_score(y_test, pred_test, average='micro'),
62               "f1_score_test": f1_score(y_test, pred_test, average='micro')}
63
64 log_metrics(pref_metrics)
65 # log_metric("Accuracy for this run", test_accuracy)
66 mlflow.sklearn.log_model(model_entropy, "DecisionTreeModel")
67 # mlflow.log_artifact("winequality-red.csv")
```

Model trained
C:\Users\mohdm\anaconda\envs\mlflow38\lib\site-packages\distutils\hack__init__.py:33: UserWarning: Setuptools is replacing distutils.
warnings.warn("Setuptools is replacing distutils.")
(mlflow38)
mohdm@Kunasib14 MINGW64 ~/Downloads/MLflow_A
\$

6) Hyperparameter tuning of Random forest Model and evaluation metrics like accuracy, precision, recall, f1 score.



```
55 # Hyperparameter tuning
56 model = RandomForestClassifier()
57
58 # Parameter Search shape
59 params = [{"criterion": ['entropy', 'gini'],
60           'n_estimators': [10, 30, 50, 70, 90],
61           'max_features': ['sqrt', 'log'],
62           'max_depth': [10, 50, 20],
63           'min_samples_split': [2, 5, 8, 11],
64           'min_samples_leaf': [1, 5, 9],
65           'max_leaf_nodes': [2, 5, 8, 11]}]
66
67 # Cross validation
68 cv = StratifiedKFold(n_splits=3, shuffle=True)
69
70 # GridSearch
71 tuning = GridSearchCV(estimator=model, cv=cv, scoring='accuracy', param_grid=params)
72
73 # Train and optimize the estimator
74 tuning.fit(X_train, y_train)
75
76 # Best parameters found
77 print('Best Parameters found using GridSearch:', tuning.best_params_)
78
79 y_predict_gs = tuning.predict(X_test)
80 accuracy = accuracy_score(y_test, y_predict_gs)
81
82 # Logging metrics
83 log_metric("Accuracy for this run ", accuracy)
84
85 # Logging model
86 mlflow.sklearn.log_model(model, "Model")
87
88 # Printing the run id
89 print(mlflow.active_run().info.run_id)
90 print("ending the experiment")
```

7) Tracking the logged parameters in mlflow ui

The screenshot shows the mlflow 1.30.0 Experiments page. On the left, a sidebar lists experiments: 'Default', 'mlflow titanic', 'mlflow Optimized titanic mo...', and 'mlflow logisticRegression mo...'. The main area is titled 'Displaying Runs from 3 Experiments'. It includes a search bar with the query 'metrics.rmse < 1 and params.model = "tree"', a 'Columns' dropdown, and a 'Only show differences' toggle. Below, a table shows 34 matching runs. The table has columns: Created, Experiment Name, Duration, Run Name, Accuracy for this, best_cv_score, f1_score_test, and Param. The runs are sorted by 'Created' time, showing various runs from the three experiments.

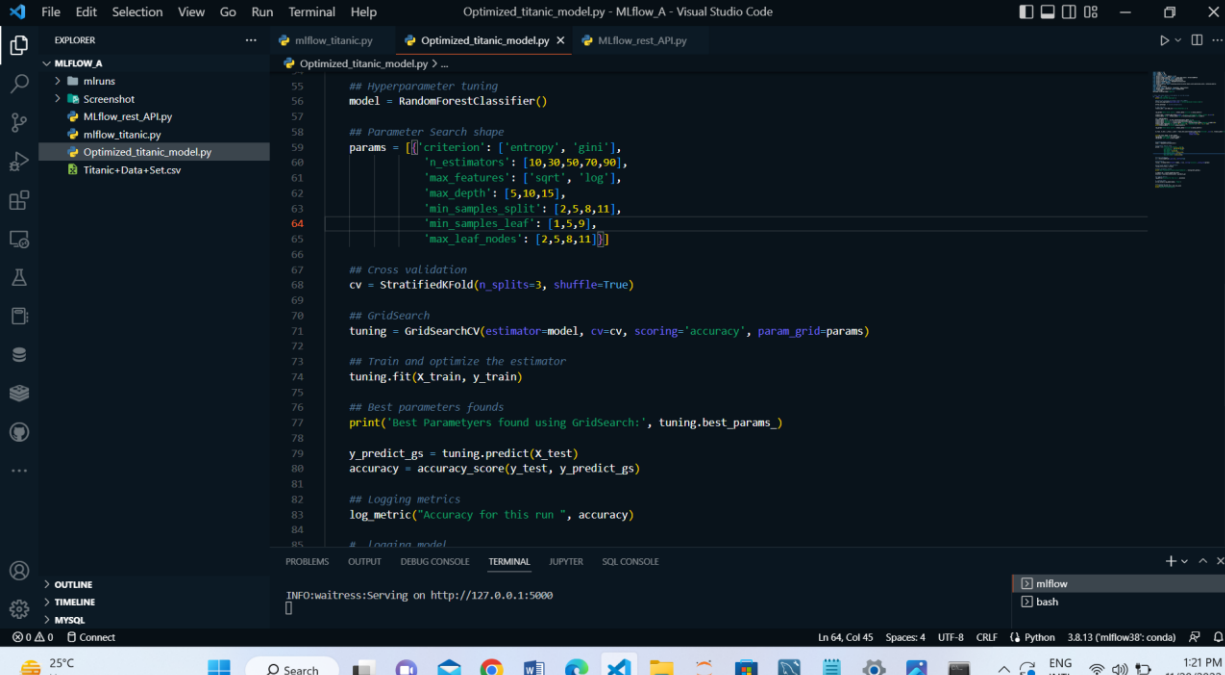
Created	Experiment Name	Duration	Run Name	Accuracy for this	best_cv_score	f1_score_test	Param
2 minutes ago	mlflow logisticRegre...	3.8s	zealous-mar...	0.785	-	-	0.549
4 minutes ago	mlflow titanic	4.3s	valuable-be...	0.785	-	-	0.549
43 minutes ago	mlflow Optimized ti...	9.1min	merciful-car...	0.812	0.844	-	0.549
1 hour ago	mlflow titanic	7.0s	mercurial-sh...	0.812	-	0.812	0.549
1 hour ago	mlflow titanic	6.3s	polite-chim...	0.794	-	0.794	0.549
1 hour ago	mlflow titanic	9.8s	clean-ant-991	0.78	-	0.78	0.549
20 hours ago	mlflow titanic	4.4s	caring-donk...	0.794	-	0.794	0.549
21 hours ago	mlflow Optimized ti...	7.0min	thundering-...	0.785	0.841	-	0.549
21 hours ago	mlflow titanic	3.8s	awesome-b...	0.812	-	0.812	0.549

9) Making some changes in Hyperparameter of the model

a) Run Name: merciful-carp-654 model parameters:

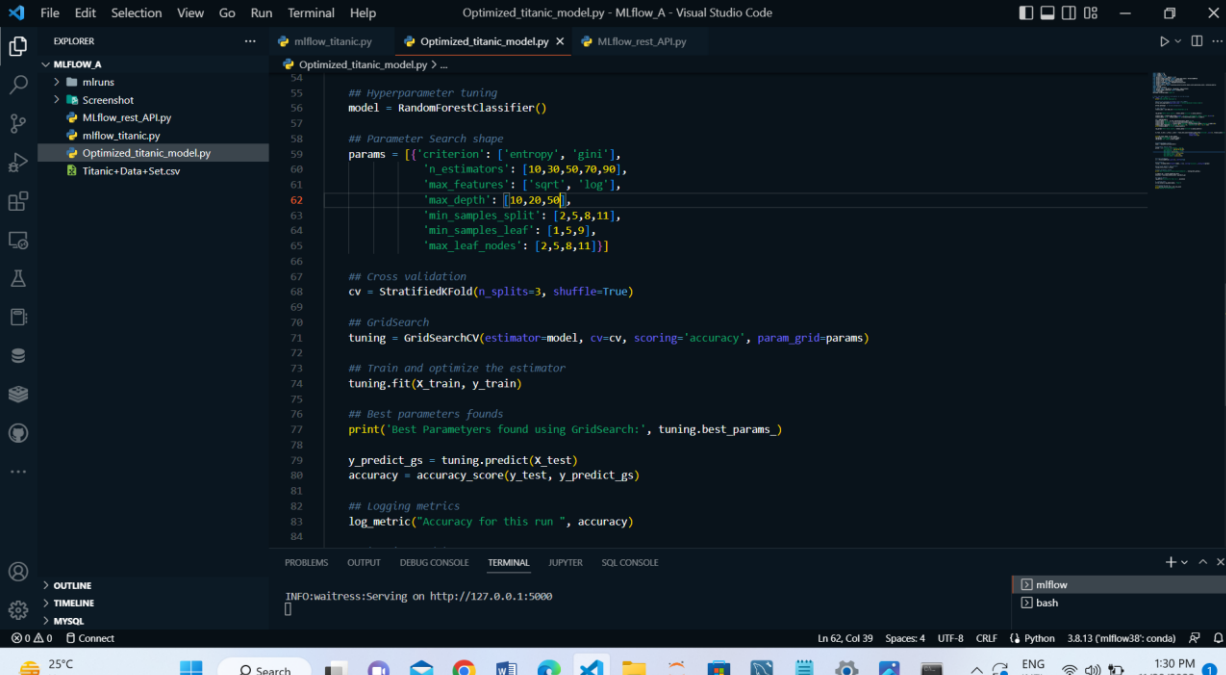
```
55 # Hyperparameter tuning
56 model = RandomForestClassifier()
57
58 # Parameter Search shape
59 params = [{"criterion": ['entropy', 'gini'],
60            'n_estimators': [10, 30, 50, 70, 90],
61            'max_features': ['sqrt', 'log'],
62            'max_depth': [5, 10, 15],
63            'min_samples_split': [2, 5, 8, 11],
64            'min_samples_leaf': [1, 5, 9],
65            'bootstrap': ['bootstrap'],
66            'max_leaf_nodes': [2, 5, 8, 11]}]
67
68 # Cross validation
69 cv = StratifiedKFold(n_splits=3, shuffle=True)
70
71 # GridSearch
72 tuning = GridSearchCV(estimator=model, cv=cv, scoring='accuracy', param_grid=params)
73
74 # Train and optimize the estimator
75 tuning.fit(X_train, y_train)
76
77 # Best parameters found
78 print('Best Parameters found using GridSearch:', tuning.best_params_)
79
80 y_predict_gs = tuning.predict(X_test)
81 accuracy = accuracy_score(y_test, y_predict_gs)
82
83 # Logging metrics
84 log_metric("Accuracy for this run ", accuracy)
85
86 # Logging model
87 mlflow.sklearn.log_model(model, "Model")
88
89 # Printing the run id
90 print(mlflow.active_run().info.run_uuid)
```

b)Run Name : thundering-yak-634 model parameters



```
55 ## Hyperparameter tuning
56 model = RandomForestClassifier()
57
58 ## Parameter Search shape
59 params = [
60     {'criterion': ['entropy', 'gini'],
61      'n_estimators': [10, 30, 50, 70, 90],
62      'max_features': ['sqrt', 'log'],
63      'max_depth': [5, 10, 15],
64      'min_samples_split': [2, 5, 8, 11],
65      'min_samples_leaf': [1, 5, 9],
66      'max_leaf_nodes': [2, 5, 8, 11]}]
67
68 ## Cross validation
69 cv = StratifiedKFold(n_splits=3, shuffle=True)
70
71 ## GridSearch
72 tuning = GridSearchCV(estimator=model, cv=cv, scoring='accuracy', param_grid=params)
73
74 ## Train and optimize the estimator
75 tuning.fit(X_train, y_train)
76
77 ## Best parameters founds
78 print('Best Parameters found using GridSearch:', tuning.best_params_)
79
80 y_predict_gs = tuning.predict(X_test)
81 accuracy = accuracy_score(y_test, y_predict_gs)
82
83 ## Logging metrics
84 log_metric("Accuracy for this run ", accuracy)
85
86 ## Learning model
```

c)Run Name: bedecked-bug-891 model parameters



```
54
55 ## Hyperparameter tuning
56 model = RandomForestClassifier()
57
58 ## Parameter Search shape
59 params = [
60     {'criterion': ['entropy', 'gini'],
61      'n_estimators': [10, 30, 50, 70, 90],
62      'max_features': ['sqrt', 'log'],
63      'max_depth': [10, 20, 50],
64      'min_samples_split': [2, 5, 8, 11],
65      'min_samples_leaf': [1, 5, 9],
66      'max_leaf_nodes': [2, 5, 8, 11]}]
67
68 ## Cross validation
69 cv = StratifiedKFold(n_splits=3, shuffle=True)
70
71 ## GridSearch
72 tuning = GridSearchCV(estimator=model, cv=cv, scoring='accuracy', param_grid=params)
73
74 ## Train and optimize the estimator
75 tuning.fit(X_train, y_train)
76
77 ## Best parameters founds
78 print('Best Parameters found using GridSearch:', tuning.best_params_)
79
80 y_predict_gs = tuning.predict(X_test)
81 accuracy = accuracy_score(y_test, y_predict_gs)
82
83 ## Logging metrics
84 log_metric("Accuracy for this run ", accuracy)
85
```


10) Comparing all three models

Run details

Run ID:	07b1c884e6ba4e609aff0e5e2ab56861	4387d05080c64c99ae0cf7f2ddc0c821
Run Name:	aqed-crane-480	enthused-colt-408
Start Time:	2022-11-20 13:07:20	2022-11-20 13:07:20
End Time:	2022-11-20 13:16:21	2022-11-20 13:16:21
Duration:	9.0min	9.0min

Parameters

☐ Show diff only

bootstrap	bootstrap	bootstrap
ccp_alpha	0.0	0.0

Metrics

☐ Show diff only

mean_fit_time	0.084	0.068
mean_score_time	0.011	0.017
mean_test_score	0.841	0.841
rank_test_score	2	5
std_fit_time	7.867e-7	4.700e-4
std_score_time	4.232e-4	4.702e-4

Tags

☐ Show diff only

