Project Development Phase
Model Performance Test

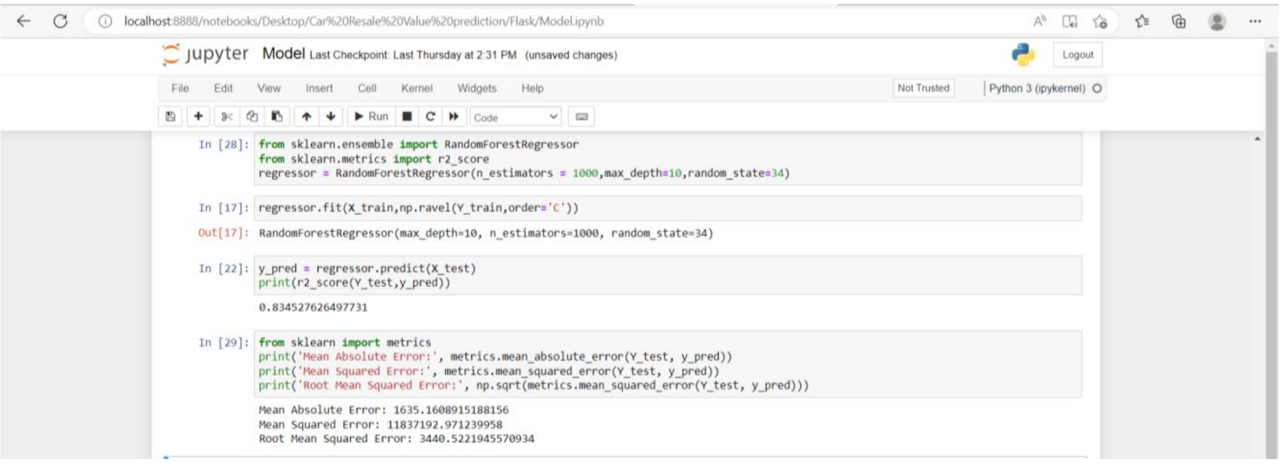
Date	19 November 2022
Team ID	PNT2022TMID08581
Project Name	Project – Car Resale Value Prediction
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

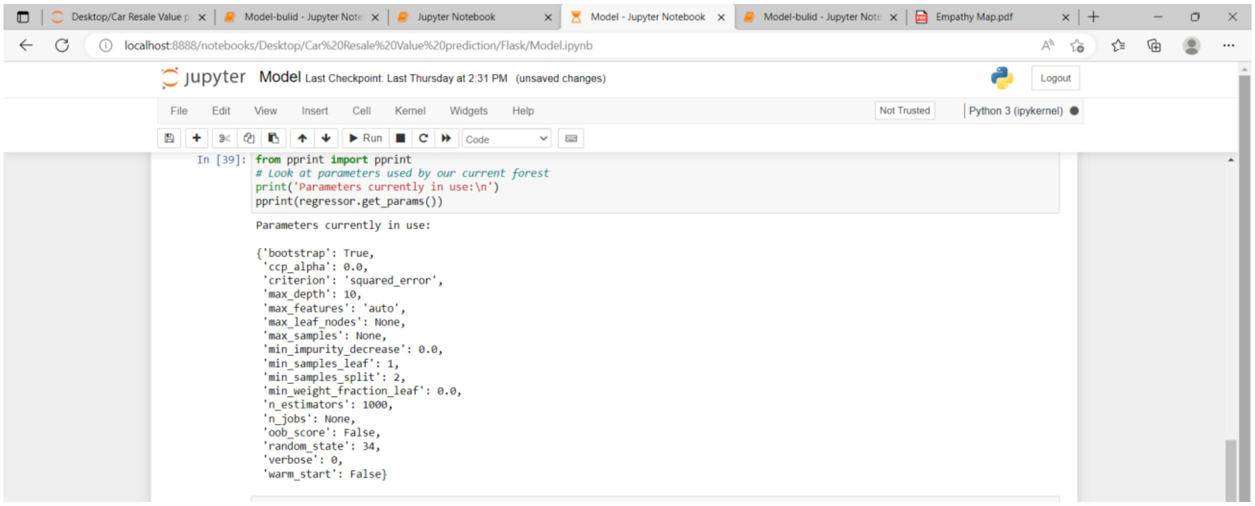
S.No.	Parameter	Values	Screenshot
1.	**Metrics**	**Regression Model:** **Mean Absolute Error:** 1635.1608915188156 **Mean Squared Error:** 11837192.971239958 **Root Mean Squared Error:** 3440.5221945570934 **R2 Score:**0.83427626497731	
2.	**Tune the Model**	**Hyperparameter Tuning :** Bootstrap: [True, False], **Max_depth:** [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None], **Max_features**: ['auto', 'sqrt'], **Min_samples_leaf'**: [1, 2, 4], **Min_samples_split'**: [2, 5, 10], **n_estimators'**: [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000] **Validation Method:** Cross Validation	

1.Metrics:

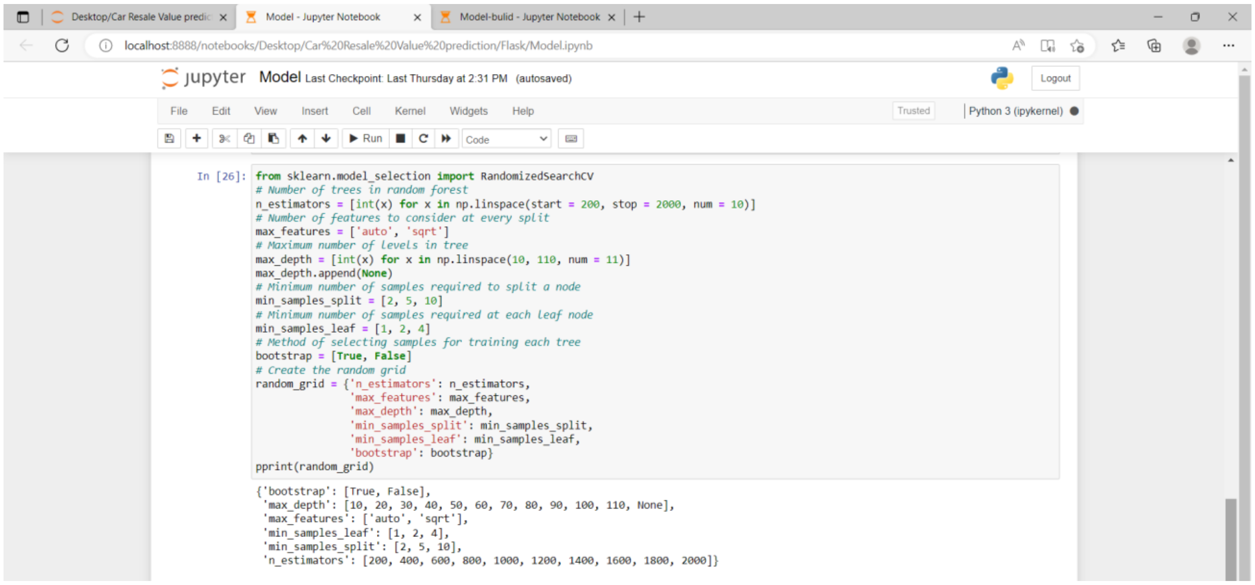

localhost:8888/notebooks/Desktop/Car%20Resale%20Value%20prediction/Flask/Model.ipynb
jupyter Model Last Checkpoint: Last Thursday at 2:31 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [28]: from sklearn.ensemble import RandomForestRegressor
         from sklearn.metrics import r2_score
         regressor = RandomForestRegressor(n_estimators = 1000,max_depth=10,random_state=34)

In [17]: regressor.fit(X_train,np.ravel(Y_train,order='C'))

Out[17]: RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)

In [22]: y_pred = regressor.predict(X_test)
         print(r2_score(Y_test,y_pred))

         0.834527626497731

In [29]: from sklearn import metrics
         print('Mean Absolute Error:', metrics.mean_absolute_error(Y_test, y_pred))
         print('Mean Squared Error:', metrics.mean_squared_error(Y_test, y_pred))
         print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(Y_test, y_pred)))

         Mean Absolute Error: 1635.1608915188156
         Mean Squared Error: 11837192.971239958
         Root Mean Squared Error: 3440.5221945570934
```

2.Tune the Model:


Desktop/Car Resale Value p
Model-bulid - Jupyter Note
Jupyter Notebook
Model - Jupyter Notebook
Model-bulid - Jupyter Note
Empathy Map.pdf
localhost:8888/notebooks/Desktop/Car%20Resale%20Value%20prediction/Flask/Model.ipynb
jupyter Model Last Checkpoint: Last Thursday at 2:31 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [39]: from pprint import pprint
         # Look at parameters used by our current forest
         print('Parameters currently in use:\n')
         pprint(regressor.get_params())

         Parameters currently in use:

         {'bootstrap': True,
          'ccp_alpha': 0.0,
          'criterion': 'squared_error',
          'max_depth': 10,
          'max_features': 'auto',
          'max_leaf_nodes': None,
          'max_samples': None,
          'min_impurity_decrease': 0.0,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'min_weight_fraction_leaf': 0.0,
          'n_estimators': 1000,
          'n_jobs': None,
          'oob_score': False,
          'random_state': 34,
          'verbose': 0,
          'warm_start': False}
```


Desktop/Car Resale Value predic
Model - Jupyter Notebook
Model-bulid - Jupyter Notebook
localhost:8888/notebooks/Desktop/Car%20Resale%20Value%20prediction/Flask/Model.ipynb
jupyter Model Last Checkpoint: Last Thursday at 2:31 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [26]: from sklearn.model_selection import RandomizedSearchCV
         # Number of trees in random forest
         n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
         # Number of features to consider at every split
         max_features = ['auto', 'sqrt']
         # Maximum number of levels in tree
         max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
         max_depth.append(None)
         # Minimum number of samples required to split a node
         min_samples_split = [2, 5, 10]
         # Minimum number of samples required at each leaf node
         min_samples_leaf = [1, 2, 4]
         # Method of selecting samples for training each tree
         bootstrap = [True, False]
         # Create the random grid
         random_grid = {'n_estimators': n_estimators,
                        'max_features': max_features,
                        'max_depth': max_depth,
                        'min_samples_split': min_samples_split,
                        'min_samples_leaf': min_samples_leaf,
                        'bootstrap': bootstrap}
         pprint(random_grid)

         {'bootstrap': [True, False],
          'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None],
          'max_features': ['auto', 'sqrt'],
          'min_samples_leaf': [1, 2, 4],
          'min_samples_split': [2, 5, 10],
          'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}
```