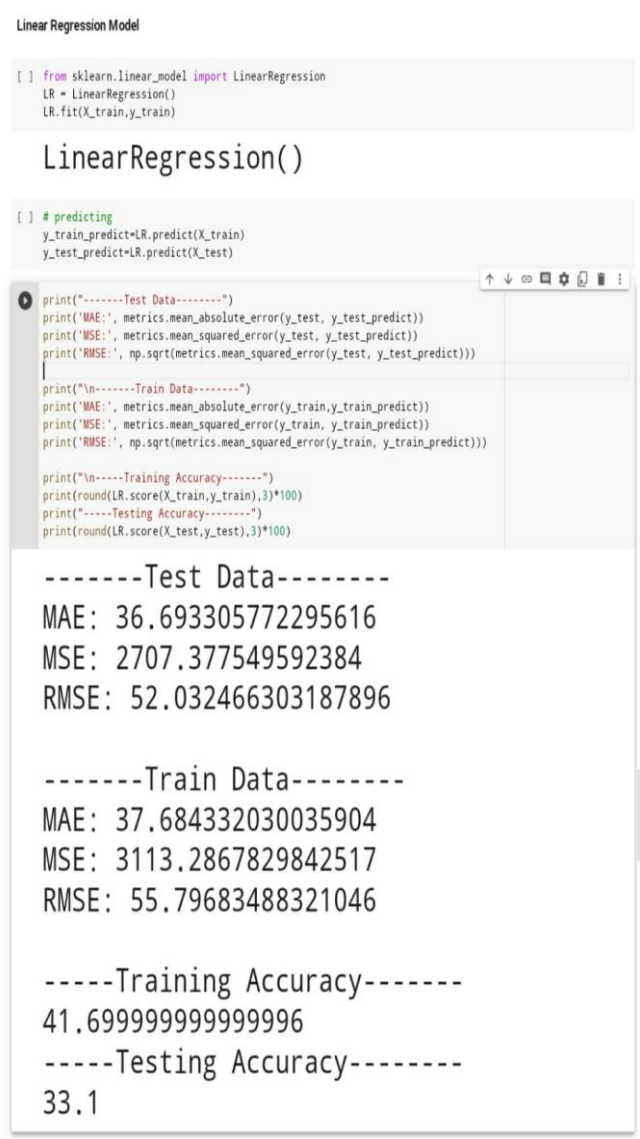
Project Development Phase
Model Performance Test

Date	19 November 2022
Team ID	PNT2022TMID49531
Project Name	Project -Exploratory Analysis of Rainfall Data in India for Agriculture.
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	**1.Linear Regression** ----------Test Data---------- MAE : 36.693305772295616 MSE : 2707.377549592384 RMSE : 52.032466303187896 ----------Train Data---------- MAE : 37.684332030035904 MSE : 3113.2867829842517 RMSE : 55..79683488321046	

		2.Lasso model ----------Test Data---------- MAE : 41.774633175550605 MSE : 3011.482049035098 RMSE : 54. 8769719375540 ----------Train Data---------- MAE : 46.667686894462854 MSE : 3948.760899348929 RSME : 62.839166921188	Lasso Model ``` from sklearn import metrics print("-------Test Data--------") print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict)) print('MSE:', metrics.mean_squared_error(y_test, y_test_predict)) print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict))) print("\n-------Train Data--------") print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict)) print('MSE:', metrics.mean_squared_error(y_train, y_train_predict)) print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict))) print("\n-----Training Accuracy-------") print(round(lasso.score(X_train,y_train),3)*100) print("-----Testing Accuracy--------") print(round(lasso.score(X_test,y_test),3)*100) ``` ``` -------Test Data-------- MAE: 41.774633175550605 MSE: 3011.4820490350985 RMSE: 54.87697193755408 -------Train Data-------- MAE: 46.667686894462854 MSE: 3948.760899348929 RMSE: 62.839166921188 -----Training Accuracy------- 26.1 -----Testing Accuracy-------- 25.6 ```

3.Ridge model

----------Test data---------

MAE : 36.694264997117806

MSE : 2700.404122847211

RMSE : 51.96541275547815

----------Train data--------

MAE : 37.71478463865123

MSE : 3113.4499194422324

RSME : 55.79826743200254

Ridge Model

```
[ ] from sklearn.linear_model import Ridge
    from sklearn.model_selection import GridSearchCV

    ridge=Ridge()
    parameters={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,1,5,10,20,30,35,40,45,50,55,100]}
    ridge_regressor=GridSearchCV(ridge,parameters,scoring='neg_mean_squared_error',cv=5)
    ridge_regressor.fit(X_train,y_train)

    print(ridge_regressor.best_params_)
    print(ridge_regressor.best_score_)
```

```
{'alpha': 1e-15}
-3139.0798658992653
```

```
[ ] print("Best Parameter for Ridge:",ridge_regressor.best_estimator_)
```

```
Best Parameter for Ridge: Ridge(alpha=1e-
```

```
[ ] ridge=Ridge(alpha=100.0)

    # fit into the object
    ridge.fit(X_train,y_train)
```

```
Ridge(alpha=100.0)
```

```
[ ] # predicting the train and test values
    y_train_predict=ridge.predict(X_train)
    y_test_predict=ridge.predict(X_test)
```

```
[ ] from sklearn import metrics
    print("-------Test Data--------")
    print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict))
    print('MSE:', metrics.mean_squared_error(y_test, y_test_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict)))

    print("\n-------Train Data--------")
    print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict))
    print('MSE:', metrics.mean_squared_error(y_train, y_train_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict)))

    print("\n-----Training Accuracy-------")
    print(round(ridge.score(X_train,y_train),3)*100)
    print("-----Testing Accuracy--------")
    print(round(ridge.score(X_test,y_test),3)*100)
```

```
-------Test Data--------
MAE: 36.694264997117806
MSE: 2700.404122847211
RMSE: 51.96541275547815

-------Train Data--------
MAE: 37.71478463865123
MSE: 3113.4499194422324
RMSE: 55.798296743200254

-----Training Accuracy-------
41.699999999999996
-----Testing Accuracy--------
33.300000000000004
```

4.SVM model

--------Test data--------

MAE : 76.73671497584542

MSE : 9936.306763285023

RMSE : 99.6102509146375

----------Train data--------
MAE : 78.82815734989649

MSE : 11555.623188405798

RSME : 107.4970845577023

Svm Model

```
[ ] from sklearn import preprocessing
    from sklearn import svm

    svm_regr = svm.SVC(kernel='rbf')
    svm_regr.fit(X_train, y_train)
```

SVC()

```
[ ] y_test_predict = svm_regr.predict(X_test)
    y_train_predict = svm_regr.predict(X_train)
```

SVM Model

```
[ ] from sklearn import metrics
    print("-------Test Data--------")
    print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict))
    print('MSE:', metrics.mean_squared_error(y_test, y_test_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict)))

    print("\n-------Train Data--------")
    print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict))
    print('MSE:', metrics.mean_squared_error(y_train, y_train_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict)))


    print("\n-----Training Accuracy-------")
    print(round(svm_regr.score(X_train,y_train),3)*100)
    print("-----Testing Accuracy--------")
    print(round(svm_regr.score(X_test,y_test),3)*100)
```

```
-------Test Data--------
MAE: 76.73671497584542
MSE: 9936.306763285023
RMSE: 99.68102509146374

-------Train Data--------
MAE: 78.82815734989649
MSE: 11555.623188405798
RMSE: 107.4970845577023

-----Training Accuracy-------
3.5000000000000004
-----Testing Accuracy--------
1.7000000000000002
```

		5.Random Forest Regressor ----------Test data-------- MAE : 34.11254680110697 MSE : 2343.0259771944416 RSME : 48.40481357462749 ----------Test data-------- MAE : 25.907192325293526 MSE : 1457.4100445477081 RSME : 38.1760401894658	Random Forest Model [ ] from sklearn.ensemble import RandomForestRegressor random_forest_model = RandomForestRegressor(max_depth=100, max_features='sqrt', min_samples_leaf=4, min_samples_split=10, n_estimators=800) random_forest_model.fit(X_train, y_train) RandomForestRegressor(max_depth=100, max_features='sqrt', min_samples_leaf=4, min_samples_split=10, n_estimators=800) [ ] y_train_predict=random_forest_model.predict(X_train) y_test_predict=random_forest_model.predict(X_test) [ ] print("-------Test Data--------") print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict)) print('MSE:', metrics.mean_squared_error(y_test, y_test_predict)) print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict))) print("\n-------Train Data--------") print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict)) print('MSE:', metrics.mean_squared_error(y_train, y_train_predict)) print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict))) -------Test Data-------- MAE: 34.05103074139167 MSE: 2341.6039068475443 RMSE: 48.3901219966177 -------Train Data-------- MAE: 25.907838970434035 MSE: 1456.8105348382642 RMSE: 38.168187471220904 [ ] print("----------Training Accuracy----------") print(round(random_forest_model.score(X_train,y_train),3)*100) print("----------Testing Accuracy----------") print(round(random_forest_model.score(X_test,y_test),3)*100) ----------Training Accuracy---------- 72.7 ----------Testing Accuracy---------- 42.199999999999996

2.	Tune the Model	Validation Method:
		
		1.Linear Regression
		
		Training Accuracy
		
		41.6999999999999996
		
		Testing Accuracy
		
		33.1
			Linear Regression Model
			
			[ ] from sklearn.linear_model import LinearRegression
			LR = LinearRegression()
			LR.fit(X_train,y_train)
			
			LinearRegression()
			
			[ ] # predicting
			y_train_predict=LR.predict(X_train)
			y_test_predict=LR.predict(X_test)
			
			print("-------Test Data--------")
			print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict))
			print('MSE:', metrics.mean_squared_error(y_test, y_test_predict))
			print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict)))
			
			print("\n-------Train Data--------")
			print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict))
			print('MSE:', metrics.mean_squared_error(y_train, y_train_predict))
			print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict)))
			
			print("\n-----Training Accuracy-------")
			print(round(LR.score(X_train,y_train),3)*100)
			print("-----Testing Accuracy--------")
			print(round(LR.score(X_test,y_test),3)*100)
			
			-------Test Data--------
			MAE: 36.693305772295616
			MSE: 2707.377549592384
			RMSE: 52.032466303187896
			
			-------Train Data--------
			MAE: 37.684332030035904
			MSE: 3113.2867829842517
			RMSE: 55.79683488321046
			
			-----Training Accuracy-------
			41.699999999999996
			-----Testing Accuracy--------
			33.1

2.Lasso model

Training Accuracy

26.1

Testing Accuracy

25.6

Lasso Model

```
from sklearn import metrics
print("-------Test Data--------")
print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict))
print('MSE:', metrics.mean_squared_error(y_test, y_test_predict))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict)))

print("\n-------Train Data--------")
print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict))
print('MSE:', metrics.mean_squared_error(y_train, y_train_predict))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict)))

print("\n-----Training Accuracy-------")
print(round(lasso.score(X_train,y_train),3)*100)
print("-----Testing Accuracy--------")
print(round(lasso.score(X_test,y_test),3)*100)
```

```
-------Test Data--------
MAE: 41.774633175550605
MSE: 3011.4820490350985
RMSE: 54.87697193755408

-------Train Data--------
MAE: 46.667686894462854
MSE: 3948.760899348929
RMSE: 62.839166921188

-----Training Accuracy-------
26.1
-----Testing Accuracy--------
25.6
```

3.Ridge model

Training Accuracy

41.699999999996

Testing Accuracy

33.30000000004

Ridge Model

```
[ ] from sklearn.linear_model import Ridge
    from sklearn.model_selection import GridSearchCV

    ridge=Ridge()
    parameters={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,1,5,10,20,30,35,40,45,50,55,100]}
    ridge_regressor=GridSearchCV(ridge,parameters,scoring='neg_mean_squared_error',cv=5)
    ridge_regressor.fit(X_train,y_train)

    print(ridge_regressor.best_params_)
    print(ridge_regressor.best_score_)
```

```
{'alpha': 1e-15}
-3139.0798658992653
```

```
[ ] print("Best Parameter for Ridge:",ridge_regressor.best_estimator_)
```

```
Best Parameter for Ridge: Ridge(alpha=1e-
```

```
[ ] ridge=Ridge(alpha=100.0)

    # fit into the object
    ridge.fit(X_train,y_train)
```

```
Ridge(alpha=100.0)
```

```
[ ] # predicting the train and test values
    y_train_predict=ridge.predict(X_train)
    y_test_predict=ridge.predict(X_test)
```

```
[ ] from sklearn import metrics
    print("-------Test Data--------")
    print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict))
    print('MSE:', metrics.mean_squared_error(y_test, y_test_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict)))

    print("\n-------Train Data--------")
    print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict))
    print('MSE:', metrics.mean_squared_error(y_train, y_train_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict)))

    print("\n-----Training Accuracy-------")
    print(round(ridge.score(X_train,y_train),3)*100)
    print("-----Testing Accuracy--------")
    print(round(ridge.score(X_test,y_test),3)*100)
```

```
-------Test Data--------
MAE: 36.694264997117806
MSE: 2700.404122847211
RMSE: 51.96541275547815

-------Train Data--------
MAE: 37.71478463865123
MSE: 3113.4499194422324
RMSE: 55.798296743200254

-----Training Accuracy-------
41.699999999999996
-----Testing Accuracy--------
33.300000000000004
```

4.SVM model

Training Accuracy

3.50000000000000004

Testing Accuracy

1.700000000000000002

Svm Model

```
[ ] from sklearn import preprocessing
    from sklearn import svm

    svm_regr = svm.SVC(kernel='rbf')
    svm_regr.fit(X_train, y_train)
```

```
SVC()
```

```
[ ] y_test_predict = svm_regr.predict(X_test)
    y_train_predict = svm_regr.predict(X_train)
```

SVM Model

```
[ ] from sklearn import metrics
    print("-------Test Data--------")
    print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict))
    print('MSE:', metrics.mean_squared_error(y_test, y_test_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict)))

    print("\n-------Train Data--------")
    print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict))
    print('MSE:', metrics.mean_squared_error(y_train, y_train_predict))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict)))


    print("\n-----Training Accuracy-------")
    print(round(svm_regr.score(X_train,y_train),3)*100)
    print("-----Testing Accuracy--------")
    print(round(svm_regr.score(X_test,y_test),3)*100)
```

```
-------Test Data--------
MAE: 76.73671497584542
MSE: 9936.306763285023
RMSE: 99.68102509146374

-------Train Data--------
MAE: 78.82815734989649
MSE: 11555.623188405798
RMSE: 107.4970845577023

-----Training Accuracy-------
3.5000000000000004
-----Testing Accuracy--------
1.7000000000000002
```

		5.Random Forest Regressor **Training Accuracy** 72.7 **Testing Accuracy** 42.199999999999996	Random Forest Model [ ] from sklearn.ensemble import RandomForestRegressor random_forest_model = RandomForestRegressor(max_depth=100, max_features='sqrt', min_samples_leaf=4, min_samples_split=10, n_estimators=800) random_forest_model.fit(X_train, y_train) RandomForestRegressor(max_depth=100, max_features='sqrt', min_samples_leaf=4, min_samples_split=10, n_estimators=800) [ ] y_train_predict=random_forest_model.predict(X_train) y_test_predict=random_forest_model.predict(X_test) [ ] print("-------Test Data--------") print('MAE:', metrics.mean_absolute_error(y_test, y_test_predict)) print('MSE:', metrics.mean_squared_error(y_test, y_test_predict)) print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_test_predict))) print("\n-------Train Data--------") print('MAE:', metrics.mean_absolute_error(y_train,y_train_predict)) print('MSE:', metrics.mean_squared_error(y_train, y_train_predict)) print('RMSE:', np.sqrt(metrics.mean_squared_error(y_train, y_train_predict))) -------Test Data-------- MAE: 34.05103074139167 MSE: 2341.6039068475443 RMSE: 48.3901219966177 -------Train Data-------- MAE: 25.907838970434035 MSE: 1456.8105348382642 RMSE: 38.168187471220904 [ ] print("----------Training Accuracy----------") print(round(random_forest_model.score(X_train,y_train),3)*100) print("----------Testing Accuracy----------") print(round(random_forest_model.score(X_test,y_test),3)*100) ----------Training Accuracy---------- 72.7 ----------Testing Accuracy---------- 42.199999999999996