Model Development Phase

Date	12 July 2024
Team ID	SWTID1720077079
Project Title	Wild Blueberry Yield Prediction
Maximum Marks	4 Marks

Initial Model Training Code:

1. Linear Regression

```
In [43]: from sklearn.linear_model import LinearRegression

In [44]: lr=LinearRegression()

In [45]: lr.fit(X_train,y_train)

Out[45]: LinearRegression
         LinearRegression()
```

2. Decision Tree

```
In [51]: from sklearn.tree import DecisionTreeRegressor

In [52]: dt=DecisionTreeRegressor(criterion="squared_error", random_state=42, max_depth=5)

In [53]: dt.fit(X_train,y_train)

Out[53]: DecisionTreeRegressor
         DecisionTreeRegressor(max_depth=5, random_state=42)
```

3. Random Forest

```
In [56]: from sklearn.ensemble import RandomForestRegressor

In [57]: rf = RandomForestRegressor(n_estimators=100, random_state=42, max_depth=5)

In [58]: rf.fit(X_train,y_train)

Out[58]: RandomForestRegressor
         RandomForestRegressor(max_depth=5, random_state=42)
```

4. XGBoost

In [64]:
```
import xgboost as xgb
```

In [65]:
```
xg = xgb.XGBRegressor(objective='reg:squarederror', max_depth=5, n_estimators=100, learning_rate=0.1)
```

In [66]:
```
xg.fit(X_train,y_train)
```

Out[66]:
```
XGBRegressor
colsample_bylevel=None, colsample_bynode=None,
colsample_bytree=None, device=None, early_stopping_rounds=None,
enable_categorical=False, eval_metric=None, feature_types=None,
gamma=None, grow_policy=None, importance_type=None,
interaction_constraints=None, learning_rate=0.1, max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=5, max_leaves=None,
min_child_weight=None, missing=nan, monotone_constraints=None,
multi_strategy=None, n_estimators=100, n_jobs=None,
num_parallel_tree=None, random_state=None, ...)
```

5. SVM Regression

In [69]:
```
from sklearn.svm import SVR
```

In [70]:
```
sv = SVR(kernel='linear')
```

In [71]:
```
sv.fit(X_train,y_train)
```

Out[71]:
```
SVR
SVR(kernel='linear')
```

In [70]:
```
from sklearn.model_selection import GridSearchCV
```

In [71]:
```
svr = SVR(kernel='linear')

param_grid = {
    'C': [0.1, 1, 10, 100],
    'epsilon': [0.001, 0.01, 0.1, 1],
    'gamma': ['scale', 'auto']
}

grid_search = GridSearchCV(estimator=svr, param_grid=param_grid, scoring='neg_mean_squared_error', cv=5)

grid_search.fit(X_train, y_train)

print("Best hyperparameters: ", grid_search.best_params_)
print("Best score: ", grid_search.best_score_)

best_model = grid_search.best_estimator_
y_pred = best_model.predict(X)

mse = mean_squared_error(y, y_pred)
print("Mean Squared Error: ", mse)
```

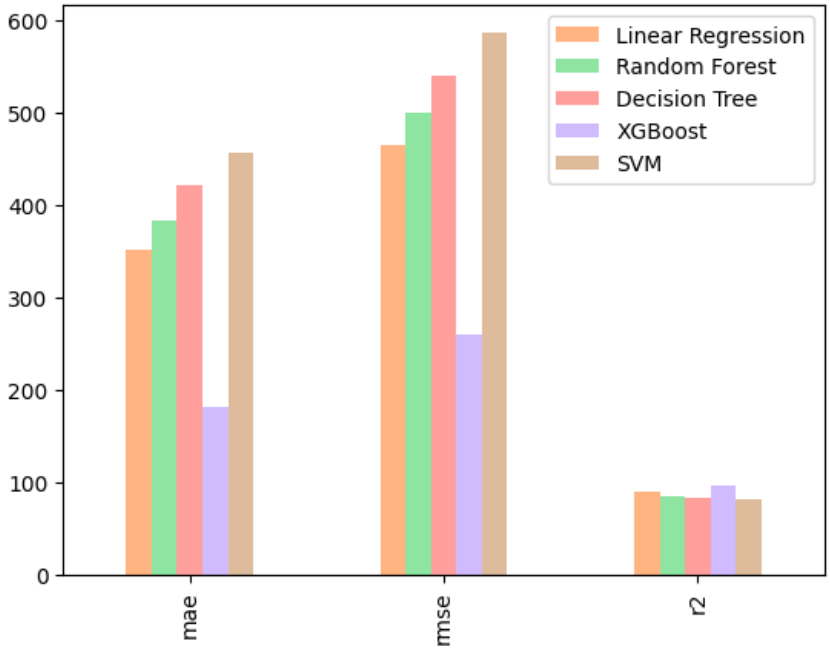
```
Best hyperparameters:  {'C': 100, 'epsilon': 1, 'gamma': 'scale'}
Best score:  -393657.4129916722
Mean Squared Error:  363111.59314407565
```

In [72]:
```
sv2= SVR(C=100, epsilon=0.001, gamma='auto', kernel='linear')

sv2.fit(X_train,y_train)
```

Out[72]:
```
SVR
SVR(C=100, epsilon=0.001, gamma='auto', kernel='linear')
```

Model Validation and Evaluation Report:



Model	Performance Metrics
Linear Regression	Mean Absolute Error: 351.5273933689664 Root Mean Squared Error: 463.7929580320785 R2: 88.81392550043651
Decision Tree	Mean Absolute Error: 421.6096623777866 Root Mean Squared Error: 539.5911930066827 R2: 82.80694144829309
Random Forest	Mean Absolute Error: 382.0077129253888 Root Mean Squared Error: 499.75198453244883 R2: 84.93700199371773
XGBoost	Mean Absolute Error: 180.54001388799836 Root Mean Squared Error: 260.16663946930686 R2: 96.60866093301176
SVM Regression	Mean Absolute Error: 455.29076862926655 Root Mean Squared Error: 586.6050431338977 R2: 81.513327311015