AVACADO PROJECT DOCUMENTATION

Introduction:

Avocados are the darling of the produce section. They’re the go-to ingredient for guacamole dips at parties. And they're also turning up in everything from salads and wraps to smoothies and even brownies.

- Data Pre-processing Done

The purpose of preprocessing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, And sampling.

- Model/s Development and Evaluation

  I have used the below models for regression:

  1. Linear Regression
  2. Decision Tree Regression
  3. KNeighbors Regression
  4. Random Forest Regression

- Identification of possible problem-solving approaches (methods)

- Read the data (from csv)
- Identify the dependent and independent variables.
- Check if the data has missing values or the data is categorical or not.
- If yes, apply basic data preprocessing operations to bring the data in a go to go format.
- Now split the data into the groups of training and testing for the respective purpose.
- After splitting data, fit it to a most suitable model. (How to find a suitable model is answered below)
- Validate the model. If satisfactory, then go with it, else tune the parameters and keep testing. In a few cases, you can also try different algorithms for the same problem to understand the difference between the accuracies.
- From step 7 one can also learn about accuracy paradox.
- Visualize the data.

- Testing of Identified Approaches (Algorithms)

1. KNN
2. Cross validation
3. Linear Regression
4. Decision Tree Regression
5. r2_score
6. mean_squared_error, mean_absolute_error

- Run and Evaluate selected models

I have used the below models for regression:

- Linear Regression

```
2]: lr=LinearRegression()
    lr.fit(x_train,y_train)
    pred=lr.predict(x_test)
    r2_sc=r2_score(y_test,pred)
    print("R2 score :",r2_sc*100)

    R2 score : 23.220027687315238
```

```
3]: from sklearn import metrics
    print('MAE:', metrics.mean_absolute_error(y_test, pred))
    print('MSE:', metrics.mean_squared_error(y_test, pred))
    print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))

    MAE: 0.1233211014559252
    MSE: 0.026926876530485814
    RMSE: 0.16409410876227645
```

- KNeighbors Regression

```
from sklearn.neighbors import KNeighborsRegressor
knn=KNeighborsRegressor()
knn.fit(x_train,y_train)
pred=knn.predict(x_test)
r2_sc=r2_score(y_test,pred)
print("R2 score :",r2_sc*100)
```

```
R2 score : 73.9975166042074
```

```
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
MAE: 0.06700657894736842
MSE: 0.009119118421052633
RMSE: 0.09549407531911408
```

- Random Forest Regressor

```
from sklearn.ensemble import RandomForestRegressor
rdr = RandomForestRegressor()
rdr.fit(x_train,y_train)
pred=rdr.predict(x_test)
```

```
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
MAE: 0.05771282894736837
MSE: 0.006154198782894728
RMSE: 0.0784487016010764
```

- Decision Tree Regression

```
from sklearn.tree import DecisionTreeRegressor
dtr=DecisionTreeRegressor()
dtr.fit(x_train,y_train)
pred=dtr.predict(x_test)
r2_sc=r2_score(y_test,pred)
print("R2 score :",r2_sc*100)
```

```
R2 score : 65.31391408822952
```

```
print('MAE:', metrics.mean_absolute_error(y_test, pred))
print('MSE:', metrics.mean_squared_error(y_test, pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pred)))
```

```
MAE: 0.0712171052631579
MSE: 0.01192730263157895
RMSE: 0.10921219085605302
```

- Conclusion:

- This dataset is Regression type of model.
- Random forest regressor has better r2_score than linear regression model ,Decision Tree model, knn model for this dataset, r2_score is 83.6% it may also denote it is over fitting as it even classifies the outliers perfectly.
- Random Forest Regressors model predicts the **average price** more accurately than linear regression model.
- Well as we can see the RMSE is lower than the three previous models, so the Random Forest Regressors is the best model in this case.