```
Avocado Dataset Analysis and ML Prediction</center>
             Table of Contents
               1. Problem Statement
               2. Data Loading and Description
               3. Data Cleaning
                 3.1 Understanding the Dataset
                 3.2 Profiling
                 3.3 Preprocessing
               4. Data Visualisation
                 4.1 Which type of Avocados are more in demand (Conventional or Organic)?
                 4.2 In which range Average price lies, what is distribution look like?
                 4.3 How Average price is distributed over the months for Conventional and Organic Types?
                 4.4 In which year and for which region was the Average price the highest?
                 4.5 How dataset features are correlated with each other?
               5. Feature Engineering for Model building
               6. Model selection/predictions
                 6.1 Average Price prediction using Linear Regression Regressor
                 6.2 Average Price prediction using Decision Tree Regressor
                 6.3 Average Price prediction using Random Forest Regressor
                 6.4 Lets see final Actual Vs Predicted sample.
               7. Conclusions
             1. Problem Statement
               • The notebooks explores the use fo Pandas and EDA for analysis purpose.
               • In this notebook we will try to predict the Avocado's Average Price based on different features. The features are different
                 (Total Bags, Date, Type, Year, Region etc.).
               • Categorical: 'region', 'type'
               • Date: 'Date'
               • Numerical: 'Total Volume', '4046', '4225', '4770', 'Total Bags', 'Small Bags', 'Large Bags', 'XLarge Bags', 'Year'
               • Target: 'AveragePrice'
             2. Data Loading and Description
             This data provided by INSAID, from the Hass Avocado Board website in May of 2018 & compiled into a single CSV. The
             dataset comprises of 18249 observations of 14 columns.
             Below is a table showing names of all the columns and their description.
                 Features
                                                                Description
               Unamed: 0 Its just a useless index feature that will be removed later
              Total Volume
                                                Total sales volume of avocados
                    4046
                               Total sales volume of Small/Medium Hass Avocado
                    4225
                                      Total sales volume of Large Hass Avocado
                                 Total sales volume of Extra Large Hass Avocado
                Total Bags
                                                    Total number of Bags sold
               Small Bags
                                               Total number of Small Bags sold
               Large Bags
                                               Total number of Large Bags sold
              XLarge Bags
                                              Total number of XLarge Bags sold
  In [2]: import pandas as pd
             import matplotlib
             matplotlib.use("Agg", warn=False)
             import matplotlib.pyplot as plt
             import numpy as np
             import seaborn as sns
             import pandas profiling
             %matplotlib inline
             import plotly.offline as py
             import plotly.graph objs as go
             from plotly.offline import init notebook mode
             init notebook mode(connected=True)
             from plotly import tools
             import warnings
             warnings.filterwarnings("ignore")
             warnings.filterwarnings("ignore", category=DeprecationWarning)
 In [23]: df = pd.read_csv("data/avocado.csv")
 In [24]: df.head()
 Out[24]:
                 Unnamed:
                                                      Total
                                                                                           Total
                                                                                                           Large XLarge
                                                                                                   Small
                             Date AveragePrice
                                                                          4225
                                                                                  4770
                                                                                                                                 type
                                                    Volume
                                                                                           Bags
                                                                                                   Bags
                                                                                                           Bags
                                                                                                                    Bags
                           2015-
                         0
                                                  64236.62 1036.74
                                                                      54454.85
                                                                                 48.16 8696.87 8603.62
                                                                                                                      0.0 conventional 2
                            2015-
                                                  54876.98
                                                             674.28
                                                                      44638.81
                                                                                 58.33 9505.56 9408.07
                                                                                                                      0.0 conventional 2
                            12-20
                                            0.93 118220.22
                                                             794.70 109149.67 130.50 8145.35 8042.21 103.14
                                                                                                                      0.0 conventional 2
                            2015-
              3
                                                                                                                      0.0 conventional 2
                                                  78992.15 1132.00
                                                                      71976.41
                                                                                 72.58 5811.16 5677.40 133.76
                            12-06
                        4 2015-
                                                  51039.60 941.48 43838.39 75.78 6183.95 5986.26 197.69
                                                                                                                      0.0 conventional 2
             3. Data Cleaning
               · Let's analyze and clean the data.
             3.1 Understanding the Dataset
  In [7]: df.shape
  Out[7]: (18249, 14)
  In [8]: df.columns # This will print the names of all columns.
  Out[8]: Index(['Unnamed: 0', 'Date', 'AveragePrice', 'Total Volume', '4046', '4225',
                      '4770', 'Total Bags', 'Small Bags', 'Large Bags', 'XLarge Bags', 'type',
                      'year', 'region'],
                     dtype='object')
  In [9]: df.head() # Will give you first 5 records
  Out[9]:
                 Unnamed:
                                                      Total
                                                                                           Total
                                                                                                   Small Large XLarge
                             Date AveragePrice
                                                                          4225
                                                                                  4770
                                                                                                                                 type
                                                    Volume
                                                                                           Bags
                                                                                                   Bags
                                                                                                           Bags
                                                                                                                    Bags
                        0 2015-
                                                  64236.62 1036.74
                                                                      54454.85
                                                                                 48.16 8696.87 8603.62
                                                                                                                      0.0 conventional 2
                            12-27
                            2015-
                                                                                 58.33 9505.56 9408.07
                                                  54876.98
                                                                      44638.81
                                                                                                           97.49
                                                             674.28
                                                                                                                      0.0 conventional 2
                            12-20
                            2015-
                                           0.93 118220.22
                                                             794.70 109149.67 130.50 8145.35 8042.21 103.14
                                                                                                                      0.0 conventional 2
                            2015-
              3
                                                                                 72.58 5811.16 5677.40 133.76
                                                  78992.15 1132.00
                                                                      71976.41
                                                                                                                      0.0 conventional 2
                            12-06
                        4 2015-
                                                  51039.60
                                                                      43838.39 75.78 6183.95 5986.26 197.69
                                                                                                                      0.0 conventional 2
               • The data set contians 18249 rows and 14 columns.
 In [11]: df.info() # This will give Index, Datatype and Memory information
             <class 'pandas.core.frame.DataFrame'>
             RangeIndex: 18249 entries, 0 to 18248
             Data columns (total 14 columns):
                                18249 non-null int64
             Unnamed: 0
                                 18249 non-null object
                                18249 non-null float64
             AveragePrice
             Total Volume
                                18249 non-null float64
             4046
                                 18249 non-null float64
             4225
                                 18249 non-null float64
             4770
                                 18249 non-null float64
             Total Bags
                                18249 non-null float64
             Small Bags
                                 18249 non-null float64
                                 18249 non-null float64
             Large Bags
             XLarge Bags
                                 18249 non-null float64
                                 18249 non-null object
             type
                                 18249 non-null int64
             year
                                 18249 non-null object
             dtypes: float64(9), int64(2), object(3)
             memory usage: 1.9+ MB
 In [12]: df.describe()
 Out[12]:
                      Unnamed: 0 AveragePrice Total Volume
                                                                        4046
                                                                                       4225
                                                                                                              Total Bags
                                                                                                                            Small Bags
              count 18249.000000 18249.000000 1.824900e+04 1.824900e+04
                        24.232232
              mean
                                       1.405978 8.506440e+05 2.930084e+05 2.951546e+05 2.283974e+04 2.396392e+05 1.821947e+05
                        15.481045
                                       0.402677 3.453545e+06 1.264989e+06 1.204120e+06 1.074641e+05 9.862424e+05 7.461785e+05
                std
               min
                         0.000000
                                       0.440000 8.456000e+01 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
               25%
                        10.000000
                                       1.100000 1.083858e+04 8.540700e+02 3.008780e+03 0.000000e+00 5.088640e+03 2.849420e+03
               50%
                        24.000000
                                       1.370000 1.073768e+05 8.645300e+03 2.906102e+04 1.849900e+02 3.974383e+04 2.636282e+04
               75%
                        38.000000
                                       1.660000 4.329623e+05 1.110202e+05 1.502069e+05 6.243420e+03 1.107834e+05 8.333767e+04
               max
                        52.000000
                                       3.250000 6.250565e+07 2.274362e+07 2.047057e+07 2.546439e+06 1.937313e+07 1.338459e+07

    We can see all columns having count 18249. Looks like it doesn't contain missing values.

 In [14]: df.isnull().sum() # Will show you null count for each column, but will not count Zeros(0) as null
 Out[14]: Unnamed: 0
             Date
             AveragePrice
             Total Volume
             4046
             4225
             4770
             Total Bags
             Small Bags
             Large Bags
             XLarge Bags
             type
             year
             region
             dtype: int64

    We can obsrve that, there are no missing values exist in dataset, that's great!

             3.2 Profiling
 In [25]: profile = pandas profiling.ProfileReport(df)
             profile.to file(outputfile="avocado before preprocessing.html")
             3.3 Preprocessing

    The Feature "Unnamed:0" is represented as a index, so let's remove it from the dataset!

                 In [26]: df.drop('Unnamed: 0',axis=1,inplace=True)
 In [27]: df.head()
 Out[27]:
                                          Total
                                                                               Total
                                                                                        Small
                                                                                                Large XLarge
                  Date AveragePrice
                                                               4225
                                                                                                                      type year region
                                         Volume
                                                                                        Bags
                                                                                                Bags
                                                                                                        Bags
                                                                               Bags
                2015-
                                       64236.62
                                                 1036.74
                                                                      48.16 8696.87
                                                                                      8603.62
                                                                                                93.25
                                                                                                          0.0 conventional 2015 Albani
                                                           54454.85
                 12-27
                                                                      58.33 9505.56 9408.07
                                                           44638.81
                                                                                                          0.0 conventional 2015 Alban
                                       54876.98
                                                  674.28
                                                                                                97.49
                 12-20
                2015-
                                0.93 118220.22
                                                  794.70 109149.67 130.50 8145.35 8042.21 103.14
                                                                                                           0.0 conventional 2015 Albani
                 12-13
                                                                      72.58 5811.16 5677.40 133.76
                                                                                                           0.0 conventional 2015 Alban
                                1.08
                                       78992.15 1132.00
                                                           71976.41
                 12-06
                2015-
                                       51039.60
                                                  941.48 43838.39 75.78 6183.95 5986.26 197.69
                                                                                                           0.0 conventional 2015 Alban
                 11-29
 In [28]:
             df['Date']=pd.to datetime(df['Date'])
             df['Month'] = df['Date'].apply(lambda x:x.month)
             df['Day'] = df['Date'].apply(lambda x:x.day)
 In [29]: df.head()
 Out[29]:
                                          Total
                                                                               Total
                                                                                                Large XLarge
                  Date AveragePrice
                                                               4225
                                                                                                                      type year region
                                         Volume
                                                                               Bags
                                                                                        Bags
                                                                                                Bags
                 2015-
                                       64236.62 1036.74
                                                                      48.16 8696.87 8603.62
                                                                                                           0.0 conventional 2015 Albany
                 12-27
                2015-
                                       54876.98
                                                           44638.81
                                                                      58.33 9505.56 9408.07
                                                                                                          0.0 conventional 2015 Alban
                                                  674.28
                 12-20
                                                                                                          0.0 conventional 2015 Albani
                                0.93 118220.22
                                                  794.70 109149.67 130.50 8145.35 8042.21 103.14
                 12-13
                                       78992.15 1132.00
                                                           71976.41
                                                                      72.58 5811.16 5677.40 133.76
                                                                                                           0.0 conventional 2015 Alban
                 12-06
                2015-
                                       51039.60
                                                  941.48 43838.39 75.78 6183.95 5986.26 197.69
                                                                                                          0.0 conventional 2015 Albany
                 11-29
             4. Data Visualisation
             Organic vs Conventional: The main difference between organic and conventional food products are the chemicals involved
             during production and processing. The interest in organic food products has been rising steadily over the recent years with
             new health super fruits emerging. Let's see if this is also the case with our dataset
             4.1 Which type of Avocados are more in demand (Conventional or Organic)?
 In [38]: type=df.groupby('type')['Total Volume'].agg('sum')
             values=[type['conventional'], type['organic']]
             labels=['conventional','organic']
             trace=go.Pie(labels=labels, values=values)
             py.iplot([trace])
                                                                                  97.2%
               • Just over 2% of our dataset is organic. So looks like Conventional is in more demand. Now, let's look at the average price
                 distribution
             4.2 In which range Average price lies, what is distribution look like?
 In [55]: sns.set(font scale=1.5)
             from scipy.stats import norm
             fig, ax = plt.subplots(figsize=(10, 5))
             sns.distplot(a=df.AveragePrice, kde=False, fit=norm)
 Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0xce72cb3828>
              1.0
              8.0
              0.6
              0.4
              0.2
              0.0
                               0.5
                                                                               2.0
                                                1.0
                                                                1.5
                                                                                                2.5
                                                                                                               3.0
                                                                                                                               3.5
                                                                  AveragePrice
               • Average Price distribution shows that for most cases price of avocado is between 1.1, 1.4.
             4.3 How Average price is distributed over the months for Conventional and Organic Types?
 In [52]: plt.figure(figsize=(18,10))
             sns.barplot(x="Month", y="AveragePrice", hue='type', data=df)
             plt.show()
                             type
                            conventional
                1.75
                       organic
                1.50
                1.25
             AveragePrice
.u
00
                0.75
                0.50
                0.25
                0.00
                                                                                                                      11
                                                                                                                               12
                                                                          Month
               • Looks like there was a hike between months 8 - 10 for both Conventional and Organic type of Avocados prices
             4.4 In which year and for which region was the Average price the highest?
 In [64]: g = sns.factorplot('AveragePrice', 'region', data=df,
                                     hue='year',
                                     size=20,
                                     aspect=0.7,
                                     palette='Greens',
                                     join=False,
                            Albany
                            Atlanta
                BaltimoreWashington
                             Boise
                            Boston
                    BuffaloRochester
                          California
                          Charlotte
                           Chicago
                    CincinnatiDayton
                         Columbus
                      DallasFtWorth
                            Denver
                            Detroit
                       GrandRapids
                        GreatLakes
                  HarrisburgScranton
                  HartfordSpringfield
                           Houston
                        Indianapolis
                        Jacksonville
                         LasVegas
                        LosAngeles
                          Louisville
                  MiamiFtLauderdale
                          Midsouth
                                                                                                                              year
                          Nashville
                                                                                                                                 2015
                  NewOrleansMobile
                                                                                                                                 2016
                          NewYork
                                                                                                                                 2017
                          Northeast
                                                                                                                                 2018
                NorthernNewEngland
                           Orlando
                        Philadelphia
                     PhoenixTucson
                         Pittsburgh
                             Plains
                           Portland
                 RaleighGreensboro
                    RichmondNorfolk
                          Roanoke
                        Sacramento
                          SanDiego
                      SanFrancisco
                            Seattle
                      SouthCarolina
                       SouthCentral
                         Southeast
                          Spokane
                            StLouis
                          Syracuse
                            Tampa
                           TotalUS
                             West
                 WestTexNewMexico
                                            1.0
                                                         1.2
                                                                                     1.6
                                                                                                   1.8
                                                                                                                 2.0
                                                                         AveragePrice

    Looks like there was a huge increase in Avocado prices as the demand was little high in Year 2017 in SanFranciso

                 region.
             4.5 How dataset features are correlated with each other?
 In [70]: plt.figure(figsize=(15,6))
             sns.heatmap(df.corr(),cmap='coolwarm',annot=True)
 Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0xce74b3f5c0>
                                                                                                                                  1.00
                                                                   -0.18 -0.17 -0.17
                                                                                          -0.12 0.093 0.16 0.027
              AveragePrice
                                      -0.19 -0.21 -0.17 -0.18
                                                                                                0.017 -0.025-0.0097
                               -0.19
                                                                    0.96
                                                                           0.97
                                                                                   0.88
              Total Volume
                                             0.98
                                                     0.97
                                                            0.87
                                                                           0.93
                                                                                                 0.0034 -0.026 -0.01
                               -0.21
                                      0.98
                                                     0.93
                                                            0.83
                                                                    0.92
                                                                                   0.84
                      4046
                                                                                                                                - 0.75
                                             0.93
                                                                            0.92
                                                                                                 -0.0096-0.022 -0.012
                                                             0.89
                                                                    0.91
                                                                                   0.81
                                                                                           0.69
                               -0.17
                                      0.97
                      4225
                                             0.83
                                                     0.89
                                                                    0.79
                                                                            0.8
                                                                                                  -0.037 -0.033 -0.009
                               -0.18
                                      0.87
                                                                                    0.7
                                                                                           0.68
                      4770
                                                                                                                               - 0.50
                                                            0.79
                                                                                   0.94
                                                                                                  0.072 -0.023 -0.005
                Total Bags
                               -0.18
                                      0.96
                                             0.92
                                                     0.91
                                                                            0.99
                                                                                           0.8
                                                                                                  0.064 -0.023-0.0039
                                             0.93
                                                     0.92
                                                                    0.99
                Small Bags
                               -0.17
                                      0.97
                                                             0.8
                                                                                    0.9
                                                                                           0.81
                                                                            0.9
                                      0.88
                                             0.84
                                                     0.81
                                                             0.7
                                                                    0.94
                                                                                                 0.088 -0.02 -0.0084
                               -0.17
                                                                                           0.71
                Large Bags
                                                                                                                               - 0.25
                                                                            0.81
                                                                                                 0.081 -0.0130.0003
                               -0.12
                                              0.7
                                                     0.69
                                                            0.68
                                                                     8.0
                                                                                   0.71
              XLarge Bags
                              0.093 0.017 0.0034-0.0096-0.037 0.072 0.064 0.088 0.081
                                                                                                        -0.18 0.0045
                       year
                                                                                                                               - 0.00
                                     -0.025 -0.026 -0.022 -0.033 -0.023 -0.023 -0.02 -0.013 -0.18
                     Month
                              0.027 -0.0097 -0.01 -0.012 -0.009 -0.005-0.00390.00840.000320.0045 0.011
                       Day
                                AveragePrice
                                                                             Small Bags
                                                                                    Large Bags
                                                                                                                   Day
                                       Total Volume
                                                                     Total Bags
                                                                                            XLarge Bags
                                                                                                           Month
               • As we can see from the heatmap above, all the features are not correleted with the Average Price column, instead most
                 of them are correlated with each other, so we can say that it is not a good model.
             5. Feature Engineering for Model building
 In [71]: df['region'].nunique()
 Out[71]: 54
             df['type'].nunique()
 Out[72]: 2
               • Type has 2 unique values, so it's going to be easy to to transform the type feature into dummies
               • Region has 54 unique values, so I decided to drop the entire column.
               • I will drop the Date Feature as well because I already have 3 other columns for the Year, Month and Day.
 In [73]: df final=pd.get dummies(df.drop(['region', 'Date'], axis=1), drop first=True)
 In [74]: df_final.head()
 Out[74]:
                                   Total
                                                                        Total
                                                                                 Small Large XLarge
                 AveragePrice
                                             4046
                                                        4225
                                                              4770
                                                                                                        year Month Day type_organic
                                  Volume
                                                                        Bags
                                                                                 Bags
                                                                                         Bags
                                                                                                  Bags
                                                    54454.85
                                                                                                    0.0 2015
                                64236.62 1036.74
                                                               48.16 8696.87 8603.62
                                                                                                                  12
              1
                               54876.98
                                                                                                                                      0
                         1.35
                                           674.28
                                                    44638.81
                                                               58.33 9505.56 9408.07
                                                                                         97.49
                                                                                                    0.0 2015
                                                                                                                  12
                                                                                                                       20
                         0.93 118220.22
                                           794.70 109149.67 130.50 8145.35 8042.21 103.14
                                                                                                    0.0 2015
                                                                                                                  12
                                                                                                                       13
                               78992.15 1132.00
                                                   71976.41
                                                               72.58 5811.16 5677.40 133.76
              3
                                                                                                    0.0 2015
                                                                                                                  12
                         1.08
                                                                                                                                      0
                               51039.60
                                           941.48
                                                   43838.39 75.78 6183.95 5986.26 197.69
                                                                                                    0.0 2015
             6. Model selection/predictions
               • Now our data is ready. Let's apply Linear Regression Algorithm because our Target variable 'AveragePrice' is continuous.
               • Let's now begin training the regression model! We will need to first split the into an X array that contains the features to
                 train on, and a y array with the target variable.
             6.1 Average Price prediction using Linear Regression Regressor
 In [91]: X=df_final.iloc[:,1:14]
             y=df_final['AveragePrice']
             from sklearn.model_selection import train_test_split
             X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2, random_state=42)
 In [92]: from sklearn.linear_model import LinearRegression
             lr=LinearRegression()
             lr.fit(X_train,y_train)
             pred=lr.predict(X_test)
 In [93]: 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: 0.23297133291665678
             MSE: 0.09108802805350158
             RMSE: 0.3018079323899582
               • The RMSE is low so we can say that we do have a good model, but lets check to be more sure.

    Lets plot the y_test vs the predictions.

 In [82]: plt.scatter(x=y_test,y=pred)
 Out[82]: <matplotlib.collections.PathCollection at 0xce76b4a2e8>
               1.8
               1.6
               1.4
              1.2
              1.0
              0.8
              0.6
              0.4
                                                         1.5
                                                                                        2.5
                        0.5
                                         1.0
                                                                         2.0
                                                                                                        3.0
               • As we can see that we don't have a straight line so I am not sure that this is the best model we can apply on our
                 data.
               • Lets try working with the Decision Tree Regression model.
             6.2 Average Price prediction using Decesion Tree Regressor
 In [95]: from sklearn.tree import DecisionTreeRegressor
             dtr=DecisionTreeRegressor()
             dtr.fit(X_train,y_train)
             pred=dtr.predict(X_test)
 In [96]: plt.scatter(x=y_test,y=pred)
             plt.xlabel('Y Test')
             plt.ylabel('Predicted Y')
 Out[96]: Text(0, 0.5, 'Predicted Y')
                   3.0
                   2.5
              Predicted Y
                   2.0
                   1.5
                   1.0
                   0.5
                                                                             2.0
                             0.5
                                             1.0
                                                             1.5
                                                                                             2.5
                                                                                                             3.0
                                                                     Y Test
               • Nice, here we can see that we nearly have a straight line, in other words its better than the Linear regression
                 model, and to be more sure lets check the RMSE.
 In [87]: 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: 0.1373123287671233
             MSE: 0.046453890410958903
             RMSE: 0.2155316459616984
               • Very Nice, RMSE is lower than as comapred to Linear Regression.
               • Now let's try one last model to see if the predictions are improved. For this, we will use Random Forest Regressor.
                 6.3 Average Price prediction using Random Forest Regressor
 In [99]: from sklearn.ensemble import RandomForestRegressor
             rdr = RandomForestRegressor()
             rdr.fit(X train,y train)
             pred=rdr.predict(X_test)
 In [89]: 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: 0.10660109589041096
             MSE: 0.024027293698630138
             RMSE: 0.15500739885124884
               . Well as we can see the RMSE is lower than the two previous models, so the Random Forest Regressor is the
                 best model in this case.
In [100]: plt.scatter(x=y_test,y=pred)
             plt.xlabel('Y Test')
             plt.ylabel('Predicted Y')
Out[100]: Text(0, 0.5, 'Predicted Y')
```

2.5

2.0

1.5

1.0

0.5

data.head(5)

8604

2608

14581

4254

16588

3.0

2.5

2.0

1.5

1.0

0.5

7. Conclusions

predictions.

is highly correlated with.

Y Test Pred

0.82 0.918

0.97 1.026

1.44 1.416

0.97 0.905

1.45 1.444

Out[122]:

0.5

1.0

In [122]: data = pd.DataFrame({'Y Test':y_test , 'Pred':pred},columns=['Y Test','Pred'])

2

Used Pandas and other plotting libraries like matplotlib, plotly and seaborn for EDA.

Used profiler for data cleaning and identifying missing and incorrect values present in the data.

Used various algorithms to know which model will work better with the help of low RMSE scores.
This project helped me to gain lots of insights on how to choose the best model for predcition.

Y Test

3

. The most important inference drawn from all this analysis is, I get to know what are the features on which price

Also get to know how to use Linear Regression, Decision Tree and Random Forest models to fine tune the

1

Used dummification for categorical values like type.

sns.lmplot(x='Y Test',y='Pred',data=data,palette='rainbow')

6.4 Lets see final Actual Vs Predicted sample.

1.5

2.0

Y Test

2.5

3.0

Predicted Y