EV sale comparison App

03/06/2023

Manikandan C

Devansh Gupta

Aryan

Devika Pratap

Step 1: Prototype Selection

Abstract

Because of the depletion of Fossil Fuels and increasing climate change and environmental damage there is an increase in demand for EV in India.

So in this report we are going to predict the future EV sale price which will be used in predicting and comparing EV sale prices across India.

Goals

1. Predicting the price of EV vehicles for a certain time period.
2. Build a model through which we can check and compare the price ranges.

Problem Statement

To predict the sale price of EV vehicles which can be used to compare the prices from now and the future and also compare the prices of different vehicles this also can be used to predict offer prices and check and also can help automobile companies to set up offers or prices ahead of the competition.

Market/Customer/Business Need Assessment

There has been an increase in the number of EV sales which also increases the demand for EV vehicles, in another 2-3 years India will start producing its own Lithium as it has some of the biggest Lithium reserves in the world and in the next 20-30 years the world will gradually shift to Fossil fuels to an alternate source of energy and EV vehicles will eventually capture the fossil fuel powered automobile market and this will lead to a boom in the EV markets, we already see that companies like Tesla, Tata, Mahindra, Hyundai, BMW are capitalizing into the future prospects of EV Vehicles. In this project we will be predicting the future prices of these companies and also classify them based on the prices.

Target Specifications and Characterization

- Marketing for EV : Using this application or model we can provide Auto Companies with a platform to market various different brands of EV vehicles which might not have a good presence on the internet or in media but can be used to promote various brands of EV vehicles.
- Improve Customer Base : We can use this platform to boost EV sales and improve the customer base for EV vehicles.
- Predict EV sales and Compare the various sale prices : This application can be used to predict the future price and do price comparison which can be used to improve the competition for EV vehicles and this can be used by both the customers and companies to set their target price.

External Search(Information and Data Analysis)

These are some of the sources I visited for more information and need for EV sale pattern analysis of customers.

- https://www.kaggle.com/datasets/aakashshakya/iea-ev-dataset-2023?resource=download
- https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/
- https://www.geeksforgeeks.org/ml-linear-regression/
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- https://www.researchgate.net/publication/336254757_Predict_the_sales_of_New-energy_Vehicle_using_linear_regression_analysis

First import all the datasets and the required Libraries

```
[ ]  1  #importing libraries
```

```
1  import numpy as np
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import seaborn as sns
5  import warnings
6
7  from sklearn.decomposition import PCA
8  from sklearn.preprocessing import LabelEncoder
9  from sklearn import preprocessing
10 from sklearn.model_selection import train_test_split
11 from sklearn.metrics import accuracy_score
12 from sklearn.linear_model import LinearRegression
13 from sklearn.ensemble import RandomForestRegressor
14 from sklearn.decomposition import PCA
```

Data Preprocessing

```
dtype= object )

[63]  1  df.isnull().sum()
      2
```

```
name                0
year                0
selling_price       0
km_driven           0
fuel                0
seller_type         0
transmission        0
owner               0
mileage           221
engine            221
max_power         215
torque            222
seats             221
dtype: int64
```

```
[50]  1  model = LinearRegression()
      2

      1  df1 = df.drop(['name','mileage', 'engine', 'max_power', 'torque'],axis=1)
      2

[52]  1  df1= pd.get_dummies(df1)
      2  df1
```

	year	selling_price	km_driven	seats	fuel_CNG	fuel_Diesel	fuel_LPG	fuel_Petrol	seller_type_Dealer	seller_type_Individual	seller_type_Trustmark Dealer	trar
0	2014	450000	145500	5.0	0	1	0	0	0	1	0	
1	2014	370000	120000	5.0	0	1	0	0	0	1	0	
2	2006	158000	140000	5.0	0	0	0	1	0	1	0	
3	2010	225000	127000	5.0	0	1	0	0	0	1	0	
4	2007	130000	120000	5.0	0	0	0	1	0	1	0	

```
 1   df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8128 entries, 0 to 8127
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           8128 non-null   object
 1   year           8128 non-null   int64
 2   selling_price  8128 non-null   int64
 3   km_driven      8128 non-null   int64
 4   fuel           8128 non-null   object
 5   seller_type    8128 non-null   object
 6   transmission   8128 non-null   object
 7   owner          8128 non-null   object
 8   mileage        7907 non-null   object
 9   engine         7907 non-null   object
 10  max_power      7913 non-null   object
 11  torque         7906 non-null   object
 12  seats          7907 non-null   float64
dtypes: float64(1), int64(3), object(9)
memory usage: 825.6+ KB
```

Business Idea:

We already have pre existing websites and applications which use ML to compare, predict and select a particular type of Vehicle, but using the EV sales data we can predict and then select the specified type of EV vehicle.

Building ML model

```
[74]  1  X_train, X_test, y_train, y_test = train_test_split(X[:3000], y[:3000], test_size=0.2)
      2

[75]  1  model = RandomForestRegressor(n_estimators=300, random_state = 42, n_jobs = -1)
      2

[76]  1  model.fit(X_train, y_train)
      2

                    RandomForestRegressor
RandomForestRegressor(n_estimators=300, n_jobs=-1, random_state=42)

[77]  1  y_pred = model.predict(X_test)
      2

[78]  1  model_accuracy = round(model.score(X_train, y_train)*100,2)
      2  print(round(model_accuracy, 2), '%')
```

Predicting the sale price using Linear Regression

```
[80]  1  y_predl = lr.predict(X_test)
      2

      1  print(y_predl)

 434871.26935194  470976.76810253  403341.63217142  347129.48591976
 -34260.86690958 1597311.7770526    64599.07599784  949448.28073296
 751916.66152564  269460.54343982  696740.0417922   546772.70477258
 470612.78812799 1707818.38390075  419858.92157567 2440248.35369821
 269460.54343982 1257895.87084766  468848.09788196  -56495.95031495
 329523.75938171  -23942.67315374  485968.33816108  398182.5352935
 741598.46776979 1073935.29092859 2018608.06266782  383792.97000824
 463754.03247343 1712977.48077868  496315.48902954  776607.34999166
1642246.88672082  549132.99433303 2530692.98071416 2215107.86249954
 401480.88996384   29517.21722798 1057912.02654769  558195.694452
 604518.29463685  735334.57496828 1963168.81718075  603718.45835605
 541613.60789464  189551.1541042   875763.9437084   894215.75502662
 473765.12300532  781766.4468696   547636.23209997 2049717.4168417
 442847.41925293  956764.77062778  726758.8274245  1066885.75965714
 362365.50795734  516943.69714634   84920.18089069  -78878.02429099
 802969.71089938  414226.70244537  -98947.53528458  782333.32338768
 408500.72904935 1070246.94434176 1640183.24796966  450340.38059083
-337073.22378236  559815.1680634   840644.56944978 1670106.0098616
2417548.32743534   39835.41098382  503426.86332618  474189.87044616
 698654.02030472  -96361.660634   1606598.15143286  601654.81960489
 158029.69631857  776607.34999166  520054.86582936   -3306.28564204
-286162.30348587  448883.59292996 2454693.82495639  473243.30339727
```

Then we can Build a Random Forest model and use this to Classify them into different Categories:

```
84]  1  print(y_pred)

294379.05730159  525737.77777778  494574.19444444  384112.21722222
161781.56701852 1057203.33333333  186881.80595382  627383.33333333
824482.3015873   550000.          454182.60582011  274552.45698413
287640.4152381   623195.5952381   292743.32666667 3084500.
550000.          450000.          176985.2047619   233249.37777778
435180.42063492  183796.83575517  465043.48544974  380803.39856926
795710.27777778  479900.02783333 1273133.33        224483.31634921
400377.60777778 1568111.11111111  359011.434       803808.16666667
578616.66666667  426844.99       3325266.66666667 1870349.81
336214.68127452  229082.72359524  594019.67782903  643966.66666667
208055.55555556  711446.17460317  495583.33333333  502480.
295654.31539683  309420.69095238  852701.19047619  420967.46517989
384420.09        712958.33333333  547093.28333333 3105257.77777778
512101.48770193  668669.75190476  889290.25444444  701333.31666667
404367.13666667  587460.          181461.2268254   110683.83333333
783811.32022607  429340.06179101  126498.88222222 1016856.80555556
410290.50986051  432880.09744444  542860.          441319.31319439
 63370.2         569280.65225428  219337.81746032  830882.22222222
3200000.         241729.11920635  442001.98412698  546603.29
535385.87037037  155532.22222222 3526550.          465136.86298886
239331.38222222  803808.16666667  431033.18        186130.48678174
 95616.66666667  254649.08008658 2625000.          374157.52469841
```

Applicable Regulations(Government and Environmental)

- Data collection and Privacy of Regulations of Customers.
- https://morth.nic.in/Motor-Vehicle-Legislation

Concept Generation

This model is built using Linear Regression, this is used to predict the sale prices and this uses the previous Data and predicts the future selling price, this helps in increase in predictability and provides companies with an automated option to setup the seling_price, after this the other ML model used is Random forests to classify them and it can be used to compare and classify the different models and this can be used to help in selecting the model.

Hyperparameters

43

- X_train, X_test, y_train, y_test = train_test_split(X[:3000], y[:3000], test_size=0.2)
- model = RandomForestRegressor(n_estimators=300, random_state = 42, n_jobs = -1)

The above hyperparameters will be used to build the Random forest classifier model which will be used in Classification and will be used to select the specific model for the price variant.

Predicted Values of both the Models

```
[80]  1  y_predl = lr.predict(X_test)
      2

      1  print(y_predl)

 434871.26935194  470976.76810253  403341.63217142  347129.48591976
 -34260.86690958 1597311.7770526    64599.07599784  949448.28073296
 751916.66152564  269460.54343982  696740.0417922   546772.70477258
 470612.78812799 1707818.38390075  419858.92157567 2440248.35369821
 269460.54343982 1257895.87084766  468848.09788196  -56495.95031495
 329523.75938171  -23942.67315374  485968.33816108  398182.5352935
 741598.46776979 1073935.29092859 2018608.06266782  383792.97000824
 463754.03247343 1712977.48077868  496315.48902954  776607.34999166
1642246.88672082  549132.99433303 2530692.98071416 2215107.86249954
 401480.88996384   29517.21722798 1057912.02654769  558195.694452
 604518.29463685  735334.57496828 1963168.81718075  603718.45835605
 541613.60789464  189551.1541042   875763.9437084   894215.75502662
 473765.12300532  781766.4468696   547636.23209997 2049717.4168417
 442847.41925293  956764.77062778  726758.8274245  1066885.75965714
 362365.50795734  516943.69714634   84920.18089069  -78878.02429099
 802969.71089938  414226.70244537  -98947.53528458  782333.32338768
 408500.72904935 1070246.94434176 1640183.24796966  450340.38059083
-337073.22378236  559815.1680634   840644.56944978 1670106.0098616
2417548.32743534   39835.41098382  503426.86332618  474189.87044616
 698654.02030472  -96361.660634   1606598.15143286  601654.81960489
 158029.69631857  776607.34999166  520054.86582936   -3306.28564204
-286162.30348587  448883.59292996 2454693.82495639  473243.30339727
```

The Above predicted values are the Predicted values of the selling_price.

```
84]  1  print(y_pred)

     294379.05730159  525737.77777778  494574.19444444  384112.21722222
     161781.56701852 1057203.33333333  186881.80595382  627383.33333333
     824482.3015873   550000.          454182.60582011  274552.45698413
     287640.4152381   623195.5952381   292743.32666667 3084500.
     550000.          450000.          176985.2047619   233249.37777778
     435180.42063492  183796.83575517  465043.48544974  380803.39856926
     795710.27777778  479900.02783333 1273133.33        224483.31634921
     400377.60777778 1568111.11111111  359011.434       803808.16666667
     578616.66666667  426844.99       3325266.66666667 1870349.81
     336214.68127452  229082.72359524  594019.67782903  643966.66666667
     208055.55555556  711446.17460317  495583.33333333  502480.
     295654.31539683  309420.69095238  852701.19047619  420967.46517989
     384420.09        712958.33333333  547093.28333333 3105257.77777778
     512101.48770193  668669.75190476  889290.25444444  701333.31666667
     404367.13666667  587460.          181461.2268254   110683.83333333
     783811.32022607  429340.06179101  126498.88222222 1016856.80555556
     410290.50986051  432880.09744444  542860.          441319.31319439
      63370.2         569280.65225428  219337.81746032  830882.22222222
    3200000.          241729.11920635  442001.98412698  546603.29
     535385.87037037  155532.22222222 3526550.          465136.86298886
     239331.38222222  803808.16666667  431033.18        186130.48678174
      95616.66666667  254649.08008658 2625000.          374157.52469841
     358461.24278499  208895.81033333   63856.66666667  712058.33333333
                                                        completed at 8:15PM
```

The Following Values are the predicted values of the classification values.

Final Product Prototype/ Product Details

The Final product will be in the form of an website or a application which will be of two types, one for the consumer and the other for the client or Manufacturer, the Manufacturer will use the ML model the fix the selling price using previous data available in their databases and then the classifier model will compare the prices and then the proper model will be selected, a similar web app or phone application can be developed to select a Ev vehicle using a similar method.

Development

- This project can be developed using Python programming, Machine Learning methods.
- The data can be provided through public platforms, Company Databases, APIs etc..
- The Application can be built using Streamlit, Django and Flask.

Long Term Prospects:

In the Long term this application will be extremely feasible and can be upgraded by adding a lot of features and this App can be used to automate the Marketing sector of the companies and this can be used a Subscription based App and can be integrated with pre existing applications.

Conclusion

In conclusion we can say that EV vehicles will see a massive rise in India and we will need a lot of integration with AI and ML applications to help in the Marketing strategy for the EV markets of India.

Github

- https://colab.research.google.com/gist/Mani512996/cb735d43ef05b1dc9e56988e7a967e30/project3-cars.ipynb

Team Members:

- Manikand C
- Devansh Gupta
- Aryan
- Deepika P.