Sources of data

The data set is mainly found on 'Electric Vehicle Database', which contains information regarding acceleration, battery useable, energy consumption, range, and top speed of each model of electric cars across various brand(X).

In order to make an accurate prediction about whether consumers will purchase a certain brand of electric vehicles in Hong Kong or not(Y), our group chose certain famous electric vehicle brands in Hong Kong, such as Tesla, BMW, and Mercedes Benz. According to corresponding determine factors, comparing their importance and significance of consumers consumption, and make a prediction in consumer purchasing behaviours.

Data Information

We first get some insights of our dataset, including getting its information, statistical description, like maximum and minimum values, quantile, and standard deviation, the correletion of two variables. The higher the values, the higher the correlation between any two variables. For example, battery useable and range have a correlation value of 0.851684, that means there is a strong relationship between them. We have a figure to visualize the correlation information. We then replace the'Purchase' attribute of 'YES', 'NO' with '1','0' to meet our target variable y.

Method 1: Logistic Regression Model

Having several determine variables(X): acceleration, battery useable, energy consumption, range, and top speed, we predict the occurrence of classification variable(Y), the probability of consumers purchase a certain brand of electric vehicle.

Logistic($\theta 0+\theta x$) = 1 / (1 + exp(-$\theta 0$-θx))

When we build the model, we split the dataset into X and Y variables, then we split into training dataset and testing dataset, finally we predict the likelihood of consumer purchasing using the logistic regression body and calculate its accuracy.

Acceleration	Battery Useable	Efficiency	Range	Top Speed
10	7.9	39.0	166	235	144
73	6.0	66.5	190	350	160
48	5.6	64.7	182	355	180
57	4.3	107.8	178	605	210
40	7.3	66.0	191	345	180
66	3.5	90.6	191	480	240
60	6.4	89.0	173	515	210
82	12.1	90.0	295	305	160
65	3.5	90.6	189	485	240
46	4.6	105.2	208	505	200
77	6.2	66.5	199	335	160
67	6.5	90.6	199	455	210
58	6.4	90.6	173	525	210
36	4.5	106.0	226	470	210
76	8.0	66.5	199	335	160

```
27          7.3          74.0        190    390        185
35          5.6         106.0        214    495        200
31          3.3          85.0        210    405        250
52          6.0         108.4        224    485        210
39          4.7          79.0        198    400        180
2           3.3          75.0        163    460        261
33          4.5         106.0        238    445        210
34          5.6          89.0        214    415        200
47          3.8         105.2        217    485        250
15         11.2          46.3        226    205        135
[0 1 1 1 0 1 1 0 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 0 0]
Accuracy:  0.64
```

Method 2: Decision Tree Classifier

In a decision tree, there is a root node, internal nodes, and leaf nodes. When we need to split according to each attribute, we should select the attritube that have the highest score. To understand the model performance, dataset are split into training dataset and tesrting dataset, and model is evaluated using the accuracy score.

```
features_columns = ['Acceleration','Battery Useable','Efficiency','Range','Top Speed']
X = data[features_columns]
y = data.Purchase
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
clf = DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(y_pred)
print("Accuracy: ", metrics.accuracy_score(y_test, y_pred))
```

Output:

```
[1 1 1 1 0 1 0 0 1 1 0 0 0 1 1 1 1 0 1 1 1 0 1 1 0]
Accuracy:  0.6
```

From the above result, we get the accuracy score, which is the classification rate of 0.6. That means it is good accuracy of prediction of purchasing electric vehicles in Hong Kong.

Decision tree can be visualized. Using the features as the classifier, and [‘0’,’1’] be the class names, we can plot the tree using (tree.plot_tree). In the visualized decision tree, the internal nodes are for splitting the data according to different decision rules, such as the range from 200 to 400 is classified as one group, from 400 to 600 is classified as another group. Gini value can also be obtained from the graph of each node.

As the model is easy for understand, it is used widely for classification problems. It can be extended to utilize usage in feature engineering, like selecting the feature variables that can best represent the prediction value, or filling in the missing information in a database. The following display the decision tree for our prediction value.

Method 3: LDA & QDA
To increase the accuracy of the prediction, we can use the method of LDA and QDA.

First, we combined all the five features in predicting purchasing behaviour of Hong Kong consumers. Then, the relavent output is the 'Purchasing' variable.

```
X_data = data.iloc[:, 1:6]
y_label = data.iloc[:, -1]
```

By fitting the model into Linear Discriminant Analysis(LDA) and Quadratic Discriminant Analysis(QDA), we could get the accuracy of approximately 0.6988 and 0.7289.

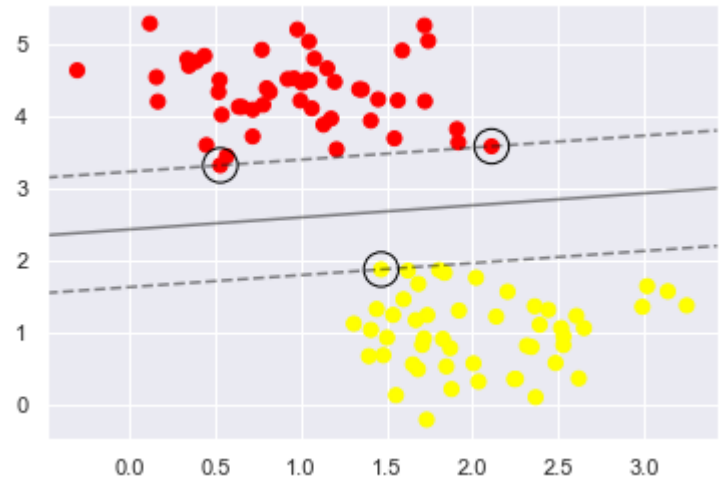
```
lda = LinearDiscriminantAnalysis(store_covariance=True)
lda.fit(X_data,y_label)


qda = QuadraticDiscriminantAnalysis(store_covariance=True)
qda.fit(X_data,y_label)
```

0.6987951807228916
0.7228915662650602

From the result, we could observe that both the classification rate of LDA and QDA are higher than of decision tree, which is 0.6 only. Therefore, we could conclude that usinf LDA/QDA is more predictive than decision tree in analyzing consumer in buying electric vehicles.

Method 4: Support Vector Machines(SVM)
Another approach to increase the accuracy of our prediction model is to define a support vector machine(SVM). We first create a hyperplane in a high dimensional space. For classification, when the hyperplane has the largest distance from the nearest data points, it can be defined as a good seperation. It is because higher classification is achieved by larger margin. (skt-learn for SVM)

In the following figure, we plot a figure in SVM model for our linearly separable problem.



For our figure, there are three samples on the margin of the boundaries, that means there are three support vectors.
Then, we calculate the accuracy of ther model using the following method.

```
from sklearn.svm import SVC
model = SVC(kernel='linear', C=1E10)
model.fit(X, y)
display(model.score(X, y))
```

The result is 1.0. That's mean it is fully classified under using the method of SVM.

Method 5: Bagging and Boosting
First, let's investigate into bagging. In general, bagging is used for reducing variance by preventing the model from overfitting, in order to improve the accuracy score of the model. Overfitting occurs when the dataset has low training error but high testing data, leading to inaccuracy in predicting.

In bagging, there are three processes, that are (i)bootstrapping, (ii)parallel training, and (iii)aggregation.
For (i)bootstrapping, the definition is just creating samples from training dataset. Several subsets can be generated from the original dataset. These subsets have equal tuples and they all can be used for training dataset to train the model.
For (ii)parallel, after several subsets have been generated, they could be fitted using bagging classifier algorithm. The process of training the dataset is called parallel training. The model then produces are called weak learners. Weak learners appear in several data points in each training dataset.
For (iii)aggregation, after several datasets have been created, we combine all weak learners into one single base model, the final model has a higher accuracy score. (Bagging)

For our case, we import 'BaggingClassifier' from sklearn directly. The bagging classifier will undergo all the steps mentioned above to train the dataset and build the best model, fitting all the weak learners into one model, the final result is the accuracy score.

```
from sklearn.ensemble import BaggingClassifier
bagging_model = BaggingClassifier(base_estimator=DecisionTreeClassifier(), n_estimators=100,
max_samples=0.8, bootstrap=True, oob_score=True, random_state=1)
bagging_model.fit(X_train, y_train)
bagging_model.oob_score_
```
Output: 0.6551724137931034

The accuracy score improve from 0.64 in using logistic regression to 0.655 in using bagging. Also, when we calculate the accuracy score from testing dataset, we get an accuracy score of 0.72. Since overfitting occurs when we get a low accuracy using test dataset. Therefore, the score 0.72 we get successfully indicated that bagging truly avoid overfitting and reduce variance, and give high accuracy score.

Then, let's investigate into boosting. In general, boosting is used for reducing bias in underfitting. By combining several weak classifiers into one strong classifier. The principle behind is one classifier is not able to make accurate prediction, but with several weak classifiers, they could be learned from each other and avoid making mistakes, so that one strong classifier could be generated.

In boosting, there are four main steps. First, the weak classifier is made based on the training dataset based on weighted samples. Then, a decision stump is created for each X variables and see if the decision stump classifies the samples well. Finally, if some samples are incorrectly classified, more weights would be assigned to those samples. In the next decision stump, they could be classified correctly. (Boosting)

For example, in our purchasing probability of electric vehicles, when we randomly choose some variables, such as range, accelaration, and battery useable, we know that how many samples are correctly classified as 'Purchase' or 'Not Purchase'. As a result, we can check whether each variables could be classified correctly.

```
from sklearn.ensemble import AdaBoostClassifier
adaboost = AdaBoostClassifier(n_estimators=100, learning_rate=1, random_state=1)
boosting_model = adaboost.fit(X_train, y_train)
y_pred = boosting_model.predict(X_test)
print("Accuracy: ", metrics.accuracy_score(y_test, y_pred))
Output: 0.56
```

Through importing 'AdaBoostClassifier' from sklearn directly, we could get the accuracy of prediction is 0.56. That means our model has a median prediction accuracy.

Final Solution Package and Detailed Results

Analysis of features in purchasing electric vehicles

By using feature engineering, we can choose several variables that have the strongest relationship with the target variable Y. We use the chi-squared test to select the top three features that can best predict the occurence probability of purchasing electric car in Hong Kong. The chi-square value is use to quantify our prediction result.

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2

X = data.iloc[:, 1:6]
Y = data.iloc[:, -1]

top_features = SelectKBest(score_func = chi2, k = 3)
fit = top_features.fit(X,Y)
data_scores = pd.DataFrame(fit.scores_)
data_columns = pd.DataFrame(X.columns)

features_scores = pd.concat([data_columns, data_scores], axis = 1)
features_scores.columns = ['Features','Scores']
features_scores.sort_values(by = 'Scores')
```

Output:

	Feature	Scores
2	Efficiency	1.801554
0	Acceleration	4.023778
4	Top Speed	10.801959
1	Battery Useable	31.014170
3	Range	240.54801

From the result of the influencing features and their corresponding scores, we can see that top speed, battery useable, and range to be the top three features in predicting Y, having scores of 10.8, 31.0, and 240.5 respectively. These three factors are the factors that affecting the consumer's willingness of purchasing a electric vehicle. In general, when the electric car of a certain brand have higher top speed, longer battery use time, and larger range, people will tends to buy an electric car, so these factors affect the sales volume and revenue of different brands of electric vehicles in Hong Kong. (Electric Vehicle Marketing Planning)

Model Evaluation on consumer willingness of purchasing electric vehicles

As the last step, our group evaluate the performance of our predictive model, that is the factors of influencing consumer willingness of purchasing electric vehicles in Hong Kong.

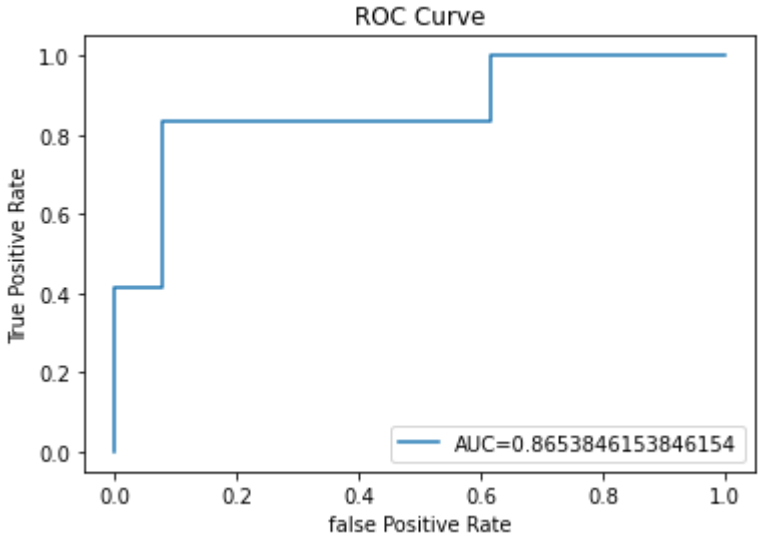
Under accuracy, the formula of accuracy is the ratio of the number of samples to the total number of samples, which can be represented by Accuracy = (TP + TN) / (TP + FN + FP + TN), where TP is true positive, TN is true negative, FP is false positive, FN is false negative. For our case, we simply create a classification report. A classification report is a report that evaluate the performance of our model by different aspects, including accuracy, precision, recall, F-score, and support.

Report:	precision	recall	f1-score	support
0	0.75	0.46	0.57	13
1	0.59	0.83	0.69	12
accuracy			0.64	25
macro avg	0.67	0.65	0.63	25
weighted avg	0.67	0.64	0.63	25

Precision: 0.5882352941176471
Recall: 0.8333333333333334
Accuracy: 0.64

The accuracy of our model is 0.64(64%), which is a medium accuracy. Also, the recall is 0.83(83%), and the precision is 0.59(59%). According to the performance values, we could conclude that our model has a relatively good prediction in the likelihood of consumer purchasing electric vehicles according to different determine factors.

Besides, the Receiver Operating Characteristic (ROC) curve is to show the effectiveness of our predictive model by calculating True Positive Rate (TPR) and False Positive Rate (FPR). The Area Under Curve (AUC) is from 0 to 1, the closer the value to 1, the better the effectiveness of our prediction. (Electric Vehicle Marketing Planning)



For our case, the AUC is approximately 0.8654, which represent that our predictive model performs well in predicting consumer willingness in purchasing electric vehicles.
In conclusion, as our area under curve is close to 1, the prediction of our model is good overall.