**HOUSE PRICE PREDICTION MODEL**

***Question:* *Apply Support Vector machine (SVM) & predict the house prices using python and the given dataset (melb\_data.csv).***

**Created By: Student of COEP Technological University, PUNE**

* AKSHAY GIDDE

**Objectives**

* To predict the Housing Prices from the provided Dataset of **melb\_data.csv**  using Support Vector Machine, with the help of Support Vector Regressor in Python
* To Understand the Data and Perform Data Preparation (Exploratory Data Analysis).
* Business – The final output of the analysis would be to Predict the Housing Prices.

**Steps:**

**Loading datasets:**

* Import all the libraries needed i.e. Pandas, NumPy, matplotlib and seaborn.
* Load the csv file provided.

**Understanding Data types:**

* Next we understood the data types of all the columns and description of each column.

**Handling Null Values:**

* There are 4 columns with null values in it.
* For these null value attributes we have checked as to what percentage of null values are there out of the total values.
* We have filled the columns which have less than or around 10% null values and dropped the columns with null values more than 30%.
* We have used Statistical Measure - Mode to fill the Null values in the columns Car and CouncilArea.

**Understanding the Distribution of Target variable:**

* We checked the distribution of the data and found it to be Right Skewed reflecting the outliers in the dataset.

**Handling Outlier:**

* Checked for the outliers in the data and handled them with methods like ‘Z-score method’, ‘IQR method’, ‘Percentile Method’.
* We have handled Dataset Outliers by using the Z score method taking values up to 3 standard deviations from the mean.

**Categorical & Numerical Divination:**

* The dataset has been further divided into Categorical features and Numerical Features as per the data types.
* As the unique values in Attributes Suburb, Address and Date are huge, which can affect the complexity and computationalism of the model, these attributes have been **dropped** from further processing.

**Encoding Categorical Data:**

* The Categorical features data has been transformed into numeric data by using **.get\_dummies** from pandas library of Python, this does encoding similar to one hot encoding which makes a new column for each unique feature and assigns binary to it.

**Splitting Data into Training and Target Variables:**

* The price column has been removed from the list of column names and then that list is provided to form a training data frame. And that data frame is assigned as training variables(x).
* Then Price column has been assigned as Target Variables(y

**Train-Test Split:**

* split data into training and validation data, for both features and target.
* The split is based on a random number generator.
* Supplying a numeric value to the random\_state argument guarantees we get the same split every run on this script.
* split test size will be of test = 25%

**Model Building:**

* Using the Support Vector Machine as per the guidelines.
* Built the SVR model by importing SVR from Sklearn.SVM library
* Fitted the training data into it .

**Accuracy Testing :**

* As it’s a prediction model we can’t use confusion matrix so, we will be using mean absolute error and r^2 error
* To check the accuracy of the model we used the Mean absolute error and r^2 error,
* which returned the results as
  + mean squared error = 405039.415208357
  + R^2 Score: -0.07412121092052248
* Since the model based on SVR has low accuracy, we have also tested the dataset on Other models.
* Other models used: Decision Tree regressor, linear regression and rigid regression, trained them with the training data and checked their accuracy using the testing data.
* Their accuracies are:
  + Decision Tree regressor - 0.6712065409384984
  + Linear regression - 0.6712971061367757
  + Rigid regression - 0.6716602213734507

**Conclusion:**

* All other models showed accuracy close to 70%, therefore it can be concluded that for the given dataset the SVR model is not the best fit and better results can be derived using either of the models from Decision Tree regressor, linear regression and rigid regression for the final prediction models.

**SVR is not fit for the this data due to :**

* SVR, a machine learning algorithm, has some limitations.
* First, it may struggle to capture complex nonlinear relationships in the data, especially when working with significant amounts of information.
* Second, if the hyperparameters are not correctly tuned, the model may overfit, resulting in poor generalization.
* Third, outliers in the data can significantly impact the performance of the model.
* Fourth, the decision boundary created by SVR is not easily interpretable, making it challenging to understand the model's behaviour.
* Finally, it can be time-consuming to train an SVR model, especially when working with large datasets.

Thank you… :)