Learning outcome from the project

# Task-3

# Group-6

In this project we are going to make a machine learning model using python language. We are going to take an in-built dataset and predict some values.

Here, we will work on Boston Housing Pricing dataset that we can find through scikit learn.

# Understanding the dataset

The first step is to understand the data on which we will be working . This is really important, by this we will understand the nature of data. In this dataset we have a number of features like CRIM (crime rate by town), ZN( residential land zoned), RM(number of rooms), etc.

# Preparing dataset

There are a number of independent features in the provided data. We will combine them by using dataframe.

Then we will add one more column of target variable named as “Price” in our data which will be considered as dependent variable.

# Checking missing value

It’s really important to check missing values as this could miss lead the outcome and increase the possibility of errors. We can check at by using is.null.sum() fuction. Here we are not having any missing value.

# Performing EDA (Exploratory Data Analysis)

It refers to the process of examining data sets to detect patterns, identify outliers, and etc.

some of the steps are

# Correlation:

It is really important to check the correlation as it indicates the strength and direction of relationships between variables. It basically tells how independent features and output features are correlated. If it shows highly positive or negative correlation then it means our model performance will definitely be high.

The value close to ZERO indicates NO correlation.

### Two type of correlation:

a) Correlation among the independent features

b) Correlation between independent and dependent variables

If there is very much correlation between independent variables you may remove one of the variables

You can use dataset.corr() to find out the correlation and you can also use scatterplot for that.

# Regression scatter plot:

It is a type of scatterplot which consists of a line drawn though the points with the nearest distance. For this we have to import seaborn which is a library on Matplotlib that helps to plot complex scatterplot , regression plot, box plot etc.

Linearity must be there between the independent variable and dependent variable to produce regression model. We can use sns.regplot() fuction to make this.

For eg:

If we check a regression model between column “RM” which is the average number of rooms and column “Price” which is the price of house, it will give positive correlation and the graph indicates that the average number of rooms increases with the increase in price of the house.

# Normalize or standardize the data:

All the independent features in the data or calculated with respect to different units. We have to normalize the data so the the converging of that specific algorithm of the gradient descent will take place faster. Our main aim is to come to the global minima. We can use fit\_transform() and transform() for this.

# Train and test split

We split data in training set and testing set to understand how good the model we have trained is, so that we can decide if we want to deploy the model

Without a dedicated testing set, the risk of overfitting increases when a model adapts too closely to the training data. Here, we will take the test size= 0.3 and random state = 42.

We can import and run train\_test\_split() fuction for this.

# Predicting with test data

Prediction are done on the model by regression.predict() fuction. This helps to predict the outcome. Further we will check whether the predictions we made is good or bad.

Some of the assumptions for predictions are:

a) To plot a scatterplot between the y\_test and reg\_pred

b) Find the residual or error , then plot the residual using sns.displot() fuction which helps to plot a univariate graph( graph for only one variable). Then we will see that there are some outliers in the data.

We can also perform some performance matrix to double sure the model performance. Performance matrix like mean squared error, mean absolute error, R square and adjusted R square.

# Predictions on new data

We are going to take up a new data and predict it through our regression model. Take first few data points and then reshape them into 2D then predict the output. Reshaping is done by using reshape() function. Further transform the reshaped data by using transform() fuction and then predict by using predict() function.

# Pickling the model file

In Picking python objects are converted into a byte stream. This byte stream can then be stored in a file or sent over a network. Pickling allows easy storage and retrieval of complex data structures in python. For this firstly import pickle then use pickle.dump() for converting into pickle file.

Pickle.dump( regression, open (‘regmodel.pkl’, ‘wb’)) If we haven’t given the path where to store the file this command will create a pickle file in the same folder where this file will be running.

We can also predict by using predict() fuction.