INSURANCE CHARGES PREDICTION USING MACHINE LEARNING

INTRODUCTION:

It has become a necessity for almost every person nowadays to sign up for the health insurance facility. These perks are being granted by many of the government as well as private insurance companies all across the globe. Concerning the value of insurance in the lives of individuals, it becomes important for the insurance companies to remain sufficiently precise while quantifying the amount covered by their policy of depicting the insurance charges over the customers.

Our project goal is to provide an idea about the insurance charges that a person require in order to fulfill the terms and conditions of an insurance company. According to our project, the criteria of insurance charges does not rely on any specific insurance company as we have taken out general estimations depending on the health status of an individual. It is purely a machine learning problem justifying multiple linear regression problem. The insurance dataset that we are using in this project is taken from kaggle while it contains seven features i.e. age, gender, bmi, children, smoker, region, and charges. In addition to this, the Machine Learning algorithm that we are up to use for our dataset is LinearSVR (Linear Support Vector Regressor).

PROBLEM STATEMENT:

There are various factors that influence the cost of insurance. These considerations contribute in the origination of some insurance policies. Each of the factors carry an important role when the amounts are being calculated. Ignorance of any factor might cause the policy to change from top to bottom. Hereby, it becomes critical to perform this task with high accuracy. Machine Learning plays an essential part to solve this issue. The technique of supervised learning pulls through the goal to automate the insurance price prediction. The linear regression model learns the insurance data from the past and can give out accurate insurance charges for the new set of data. This on the one hand reduces human effort to assume manual calculations, and on the other hand, can improve insurance company’s expediency through automation.

TOOLS AND LIBRARIES:

In our project, we have demonstrated the insurance charges prediction through the most robust programming language, Python. It is considered as one of the best and efficient tool for Artificial Intelligence, especially to compute and deal the Machine Learning problems. There are various significant packages and libraries with predefined methods which help in solving any of the Machine Learning query in a faster as well as systematic way. All we need is to know about these packages and libraries in order to deal with our data perfectly. Here in our project, we are using the following python libraries for data analysis, preprocessing and building our Machine learning model:

1) Pandas
2) NumPy
3) Matplotlib
4) Seaborn
5) Sci-kit Learn

Let's have a brief look on these libraries and what utilities they provide for the Python developers and A.I enthusiasts.

1) Pandas:

Pandas is an important data preprocessing library written for the Python language. It is an effectual tool to create and manipulate Data Frame objects. In addition, we can also load the in-memory data files with different formats into the Python development environment. For the preprocessing of data, Pandas is sufficient to carry out data cleaning, data alignment, visualization and even handling of the missing data. It is surely the best data representation tool with the benefit to write less and gain more.

2) NumPy:

NumPy, (Abbreviated as Numerical Python), is another fast and robust Python library used to deal with multi-dimensional arrays and numeric data. It possesses large-scale mathematical functions that include in different domains like Linear Algebra, Matrices, Fourier Transform, Fourier Series, Statistics, etc. Furthermore, this Python library also contains such special functions that aid in creating charts and graphs with visualization libraries in Python. Hence, we can consider it the second to none package for array manipulations.

3) Matplotlib:

Matplotlib is a data visualization library integrated with the Python language. It contains numerous packages from which the mostly used is the pyplot package. We can produce

many graphs according to the nature of our data. These different charts and graphs include line plots, scatter plots, bar graphs, pie charts, etc. It deals with both types of plots, 2D, as well as 3D plots.

4) **Seaborn:**

   Another data visualization library is the Seaborn library which is basically the advance version of Matplotlib. If you are working with the statistical methods through Python and you need visuals and graphs, the Seaborn library is the best for that. From line plots to kde-plots and heatmaps, the Seaborn library is the all in one package for extensive and detailed data visualization.

5) **Sci-kit Learn:**

   Sci-kit learn (shortly referred as sklearn) is a Python library that supports predictive data analysis tasks. It is a complete package for machine learning algorithms and data mining techniques. All of these methods, techniques and algorithms are based on statistical methods and modeling. Some of the significant techniques that the Sci-kit Learn library affords are Regression, Classification, Clustering, Association Rules, etc. It also features methods and functions that are relevant for finding out model accuracies, losses, deviations, and many other evaluation metrics. Moreover, there is also an essence of useful data preprocessing methods.

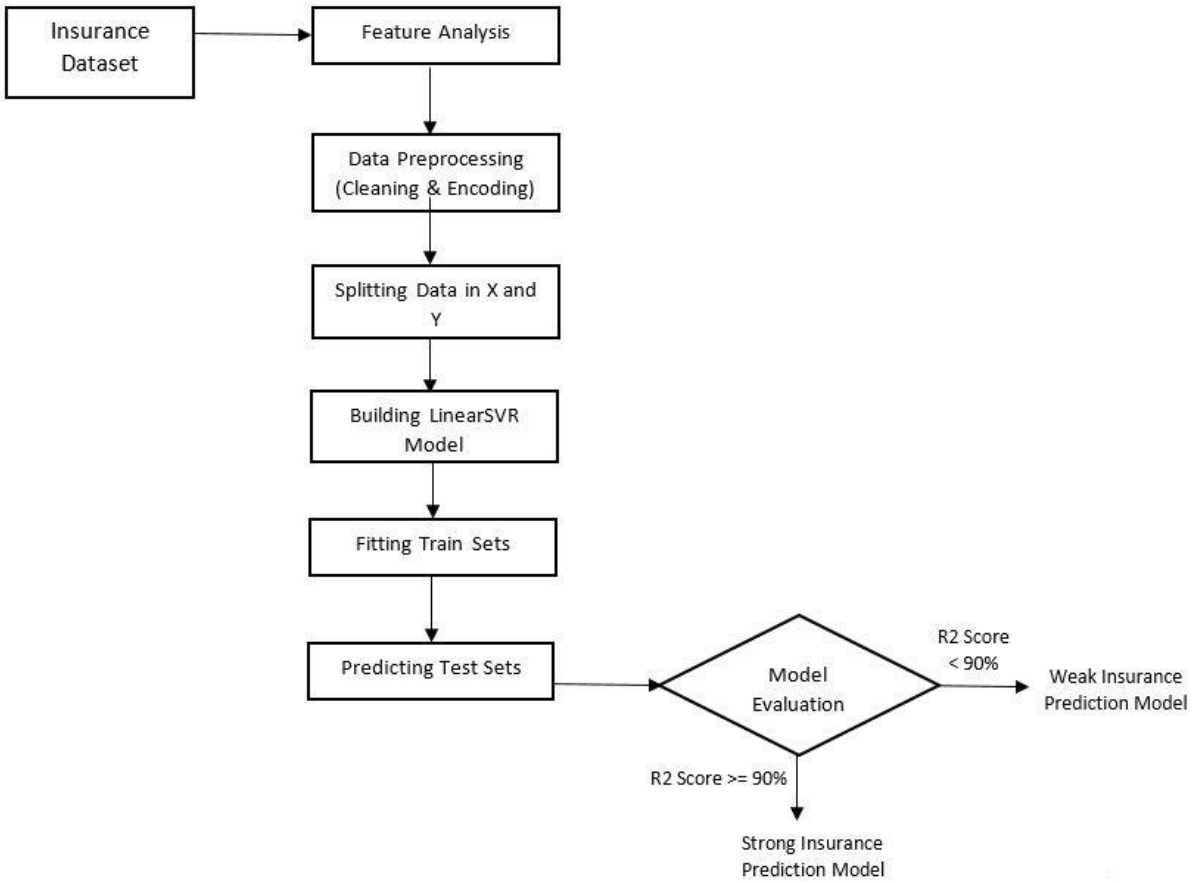
THE LINEAR SUPPORT VECTOR REGRESSOR:

Support Vector Regressor is the predecessor of the SVC (Support Vector Classifier) associated with the SVM machine learning technique. The SVR works same as the SVC i.e. both of them aim to attain maximal margin. The main focus is on minimizing the regression error of the hyperplane through taking different traces over the dataset.

Now let us take a look on the one of the versions of SVR algorithm that we have used here in this project i.e. the Linear Support Vector Regressor. It is the type of Support Vector Regressor which holds the linear kernel (to approximate target value through an estimating single line). There is a tolerance function of epsilon which aims to minimize the regression error while C (the regularization parameter) adjusts the hyperplane while controlling the fitting parameters of LinearSVR.

The LinearSVR is one of the popular algorithms for regression problems especially when we have a dataset with more than 1000 values. This is one of the reasons that we are using this exceptional regressor for our Insurance Price Prediction in this project as we initially have 1338 values in our dataset.

METHODOLOGY:



The above diagram shows the methodological model and the workflow of our Insurance Price Prediction project. There are a few steps that initiate from the loading of our dataset in the Jupyter IDE to the predictions that we have carried out. Obviously, we have utilized all the necessary Python libraries discussed above for the step by step and clear workflow.

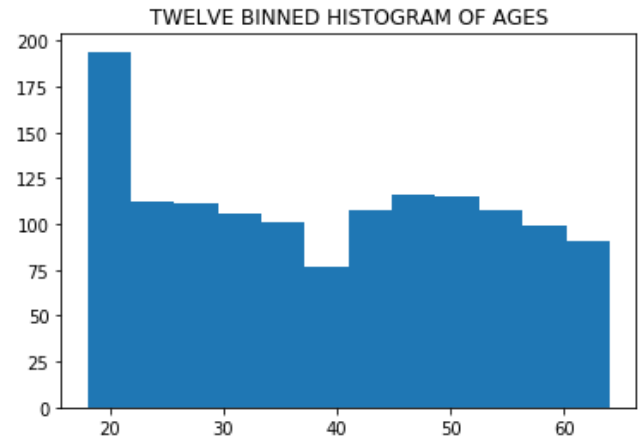
The details about each of the steps is given below:

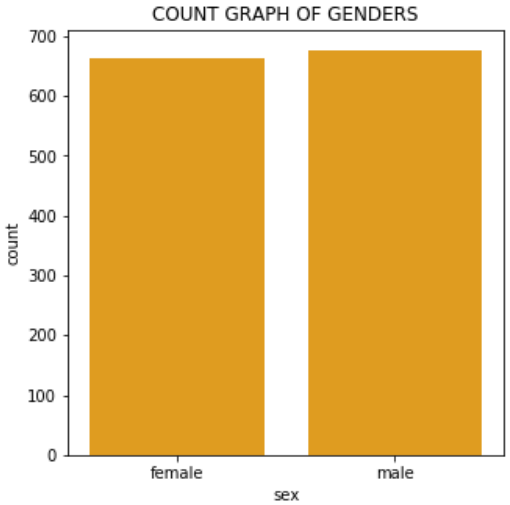
Loading The Insurance Dataset:

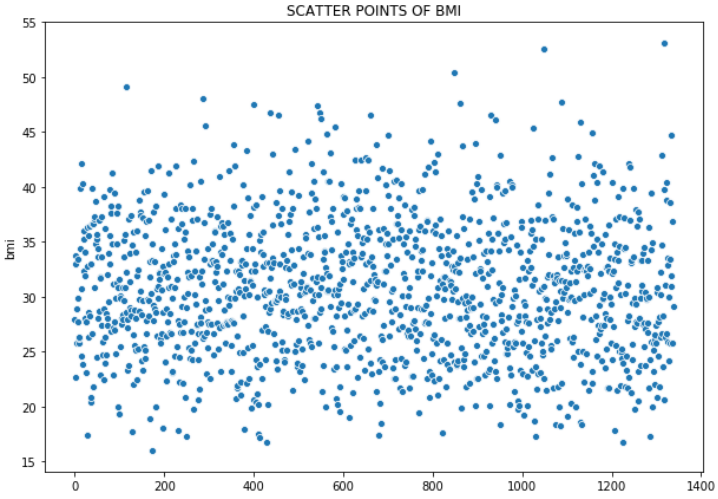
The first and the most essential step of our project is to load the dataset on the Jupyter Notebook environment. We have done it through the Pandas function i.e. read_csv. Here we have given the path where our data file exists. We have just given the name of file as the path as our insurance.csv file was previously uploaded in the Jupyter IDE.

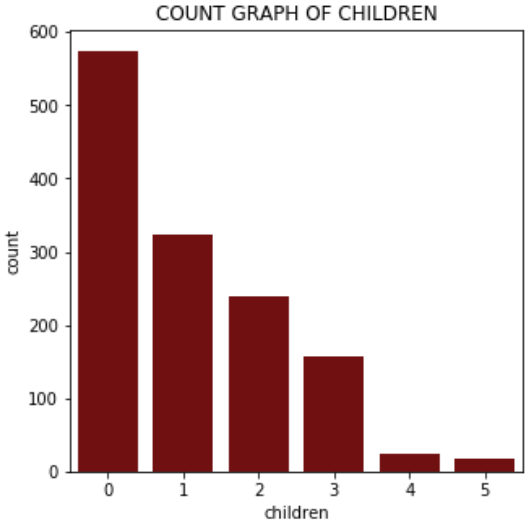
Feature Analysis:

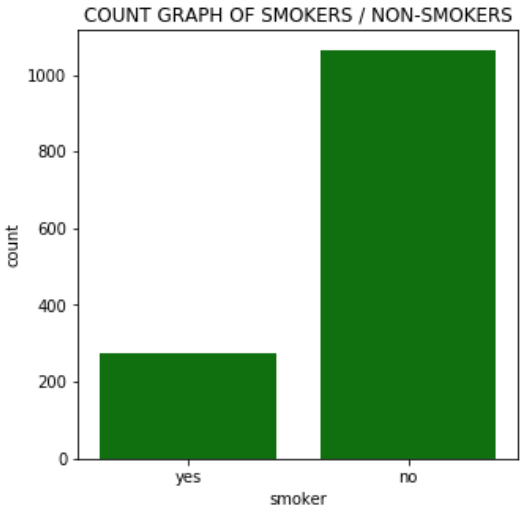
Analyzing our features is our very next step of execution. In this analysis, we have described the features according to their value counts (for categorical variables), and their frequencies (for numerical variables). Here are the results that we have taken out from our analysis.

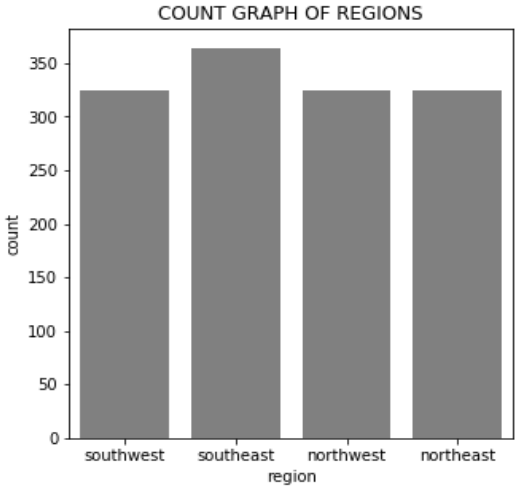












Data Preprocessing:

The next and the most critical part of this project is to maintain the data through preprocessing. Here, we have done some cleaning in our data which seems relevant for making a proper regression model. In addition, we have encoded the categorical features in the form of numbers so that it could become easy for the algorithm to recognize those instances. For the cleaning of our data, we first have deleted the feature namely "region" in order to make the data general as these regions were specific for the U.S only. After this we performed a careful analysis of outliers in our data and came to know that majority of our target feature i.e. "charges" lie in the range less than 15000. So we removed the data having charges values more than 15000 through the Panda's data.query() function. There were 980 values that remained after the cleaning process which seemed to be perfect for our regression model. For the encoding of our categorical features (sex, smoker), we used the sklearn's preprocessing function, that is LabelEncoder(). This function encoded the sex values as: female – 0, male – 1, whereas, the smoker values as: no – 0, yes – 1.

Splitting Data in X and Y Sets:

In regression, there is a concept of dependent and independent variable. Firstly, we divide our dataset into two parts. The first one contains independent features (the training features). We denote them as X. The second part possesses the dependent variable (target feature) which we have to predict after model training. We denote this part as Y.

Now, for every supervised learning workflow, it is necessary to split the data into two sets, the one for training and the other for testing. The training set fits to generate a machine learning model whereas, the test set fulfills the prediction segment. For our insurance data, we have used the sklearn library's model_selection package in which we have a function (train_test_split) to split our data into four different parts i.e. X_train (contains training features), y_train (contains training targets/ labels), X_test (contains testing features), y_test (contains test targets that are to be compared with our predictions).

For the parameter setting of our train_test_split function, we chose the 75^{th} order of the shuffle in the random state, and preferred 90% of our data for the training purpose of the Linear SVR model.

Building LinearSVR Model:

After splitting the insurance data into X and Y sets, here comes the step where we have built the Linear SVR model. We have used the instance of the sklearn.svm package and carried out fine parameter tuning in order to attain the best model accuracy. We have set the regularization parameter C with the value 7.5 to have best hyperplane fitting. The pseudo

random state for the model is set on 1, and the model iterations are 12 which seemed enough to get high precision.

Fitting Train Sets:

Building the LinearSVR model isn't enough as we have to fit the X_train and Y_train values through the fit() method of the model. This applies the backend calculation of the LinearSVR model and sets the hyperplane on the basis of the data that we have provided, thus, completing the training stage of the model.

Predicting Test Sets:

Now, we predicted the Y_pred sets through applying the model.predict() function, giving the X_test parameter in it. This predicts the Y values for our X_test set followed by our Linear SVR model built on the training insurance data (X_train, Y_train)

MODEL EVALUATION AND RESULTS:

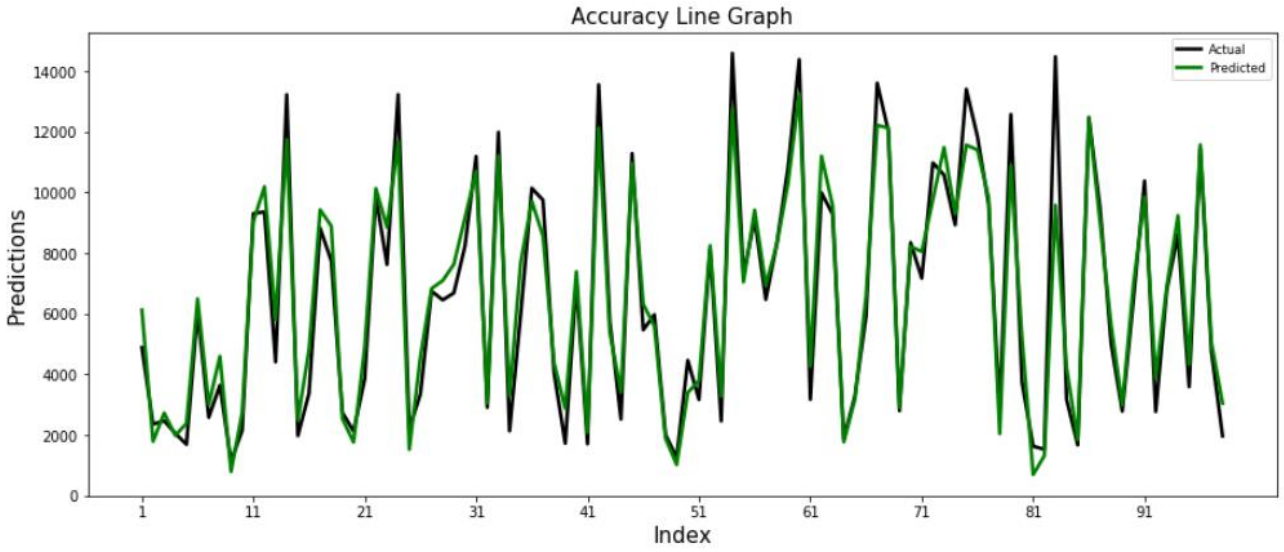
Evaluating the prediction values ultimately gives the precision of our Machine Learning model for our dataset. For this purpose, we have used two popular evaluation metrics for regression problems available in the sklearn.metrics package as mean_absolute_error, and r2_score.

The R2_Score (Accuracy Score):

The R2 score in regression depends on the variances of dependent and independent variable. It refers that, "the proportion of the variance in the dependent variable that is predictable from the independent variable(s)."

The value of R2 Score can vary from 0 to 1 in float. The higher the value of R2 Score, the best our model is. However, the value 1 which depicts 100% accuracy causes the model to overfit and conclusions become invalid.

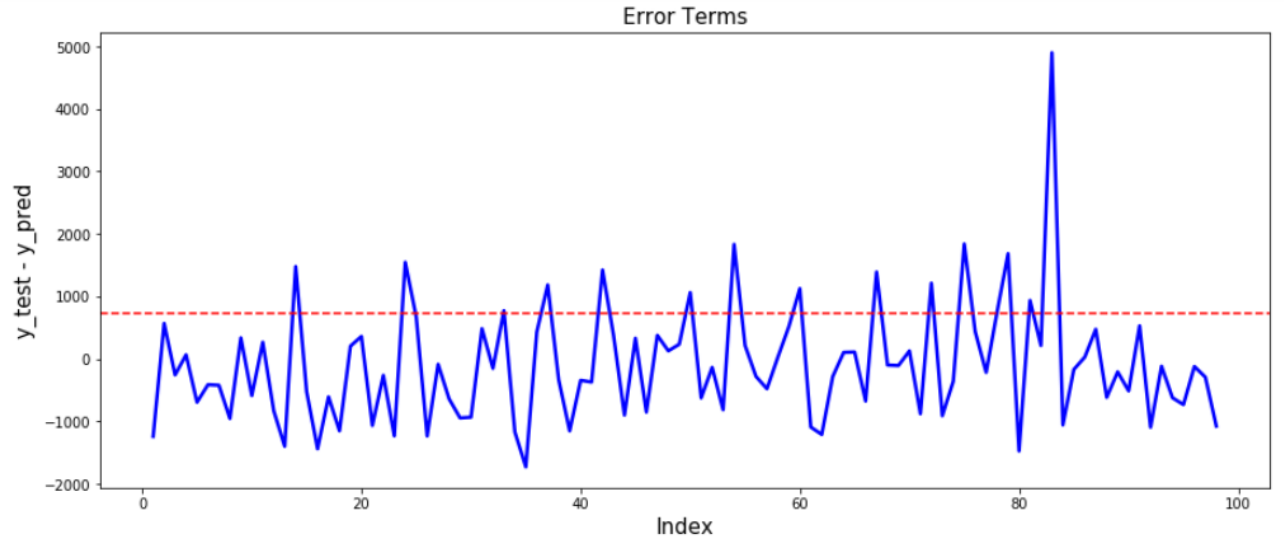
As we predicted the insurance charges values and evaluated our Linear SVR model, we got R2 Score as 0.9402 or we can say 94.02 %. Here is the graph of the Y_test and Y_pred values comparison.



Mean Absolute Error (Loss Score):

The Mean Absolute Error defines the average of deviation in the original Y values and predicted Y values. It takes the differences as absolute values and gives mean of those absolute differences, thus, putting the loss score of our model in front.

In our insurance prediction model, we have got the MAE loss score as 723.73. Here is the graph which shows the error terms of differences in the values of Y_test and Y_pred:



The red dotted line shows the mean absolute value of errors i.e. 723.73

CONCLUSION:

As a result of using Linear SVR regression algorithm to predict the insurance charges, performing fine hyperparameter tuning of the model, and analyzing result metrics, we can conclude that our Machine Learning model is successful and strong enough to predict insurance charges for the unseen data. However, increase in the number of features in our insurance dataset can help us generalize the problem and take out results more related to the real life.