Machine Learning Model for Predicting Insurance Claim Charge.

Dataset

This dataset is taken from Kaggle. Insurance companies are extremely interested in the prediction of the future. Accurate prediction gives a chance to reduce financial loss for the company. A major cause of increased costs are payment errors made by the insurance companies while processing claims. Furthermore, because of the payment errors, processing the claims again accounts for a significant portion of administrative costs.

Tools & Libraries: -

Python • Jupyter Notebook • Pandas • NumPy • Seaborn • Matplotlib • Plotly

• PyCharm • Flask

Data Description

This dataset contains 7 features as shown below:

1.age: age of the policyholder
2.sex: gender of policyholder (female=0, male=1)
3.BMI: Body mass index, providing an understanding of the body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 25
4.children: number of children/dependents of the policyholder

5.smoker: smoking state of policyholder (non-smoke=0;smoker=1)
6.region: the residential area of the policyholder in the US (northeast=0001, northwest=0010, southeast=1000, southwest=0000)
7.charges: insurance costs billed by health insurance.

EDA

We performed two types of EDA — Univariate and Bivariate.

Univariate EDA: For continuous variables like BMI, Age Children, Charges for univariate EDA we use box plot.

- We observed that there are outliers in charges & BMI.

A few Observations about categorical variables like sex, smoker, region, (we use count plot.)

- It was found that the people who applied for insurance are mostly non-Smoker & from south east region.

Bivariate EDA: In Bivariate EDA, we check for the influence of two factors/variables on the distribution of the data. From the bivariate analysis, we draw the following conclusion, Age and BMI has a significant impact on the charges.

Model Creation/Evaluation

I applied the following regression models:

1.Linear Regression

2.Ridge Regression

3.Lasso Regression

4.Random Forest Regression

5.Gradient Boosting regression

From this I chose Gradient Boosting regression as my final model because I got the score as given below:

- R Squared (Train)=0.89
- R Squared (Test)=0.888
- CV score mean (Train)=0.84

Finally, I created an app using PyCharm(flask) then I deployed it in Heroku platform.