VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127

LAB-1

COURSE ID: ITA6016

FACULTY: Dr. Kiruthika S

SUBMITTED BY:

AVANTIKA SHARMA(21MCA1001)

ADITI VERMA(21MCA1086)

ARYAN MITTAL(21MCA1097)

Abstract:

For the medical cost dataset, a linear regression model will be constructed. The dataset includes independent and dependent features such as age, sex, BMI (body mass index), children, smokers, and geography. We'll forecast how much each patient's medical bills from insurance will be.

Definition & Working principle

Linear Regression Model:

Linear regression is a supervised learning algorithm used when target / dependent variable continues real number. It establishes relationship between dependent variable y and one or more independent variable x using best fit line. It work on the principle of ordinary least square (OLS) / Mean square error (MSE) . In statistics ols is a method to estimate unknown parameters of linear regression function, its goal is to minimize the sum of square difference between observed dependent variable in the given data set and those predicted by linear regression function.

Hypothesis representation:

We will use x_i to denote the independent variable and $\mathbf{yi}$ to denote dependent variable. A pair of (x_i,y_i) is called a training example. The subscripe $\mathbf{i}$ in the notation is simply index into the training set. We have $\mathbf{m}$ training example then $\mathbf{i}$=1,2,3,...$\mathbf{m}$.

The goal of supervised learning is to learn a hypothesis function **h**, for a given training set that can be used to estimate **y** based on **x**. So hypothesis function represented as

$$\mathbf{h}_{\theta}(x_i) = \theta_0 + \theta_1 x_i$$

θ0,θ1 are parameter of hypothesis.This is equation for Simple / Univariate Linear regression.

For Multiple Linear regression more than one independent variable exit then we will use **xij** to denote independent variable and **yi** to denote dependent variable. We have n independent variable then **j**=1,2,3.....**n** . The hypothesis function represented as

$$\mathbf{h}_{\theta}(x_i) = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} \mathbf{+.....} \theta_j x_{ij} \mathbf{.....} \theta_n x_{mn}$$

θ0,θ1,....θ**j**....θ**n** are parameter of hypothesis, **m** Number of training examples, **n** Number of independent variable, x_{ij} is i^{th} training example of j^{th} feature.

DATASET DESCRIPTION:

Columns:

- age: age of primary beneficiary
- sex: insurance contractor gender, female, male
- bmi: Body mass index, providing an understanding of 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 24.9
- children: Number of children covered by health insurance / Number of dependents
- smoker: Smoking
- region: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- charges: Individual medical costs billed by health insurance.

Number of rows and columns in the data set: (1338, 7)

LIBRARIES USED:

NUMPY:

A general-purpose array processing package is called NumPy. It offers a high-performance multidimensional array object as well as utilities for interacting with these arrays. It is the core Python package for scientific computing. It is open-source software. It has a number of characteristics, including these crucial ones.An effective N-dimensional array object sophisticated (broadcasting) operations.Tools for combining C/C++ and Fortran code useful linear algebra, fourier transform, and random number capabilities.

NumPy can be used as a productive multi-dimensional container of generic data in addition to its apparent scientific applications. Numpy's ability to establish any data-types enables NumPy to quickly and easily interact with a wide range of databases.

PANDAS:

Pandas is an open-source library designed primarily for working quickly and logically with relational or labeled data. It offers a range of data structures and procedures for working with time series and numerical data. The NumPy library serves as the foundation for this library. Pandas is quick and offers its users exceptional performance & productivity.

MATPLOTLIB:

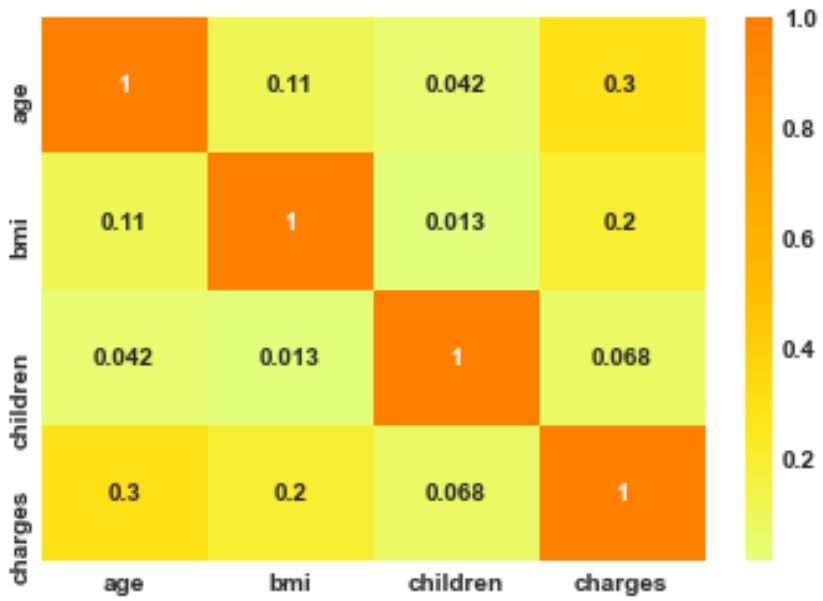
Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

SEABORN:

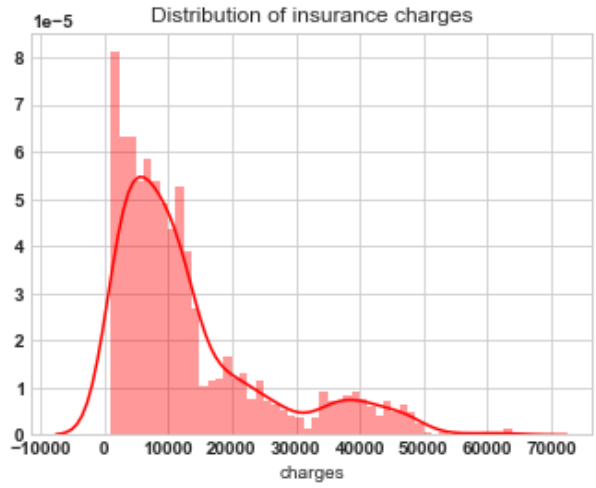
For Python statistical graphics graphing, the Seaborn visualization module is fantastic. To enhance the aesthetics of statistical charts, it offers lovely default styles and color schemes. It is based on the matplotlib package and has a close integration with the data structures from pandas.

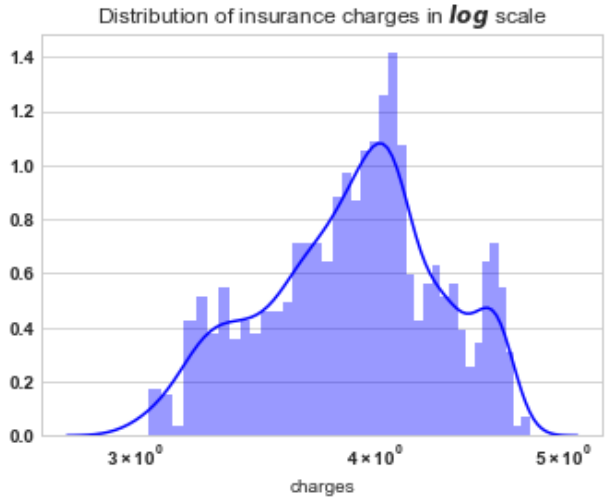
RESULTS AND GRAPHS:

Correlation plot



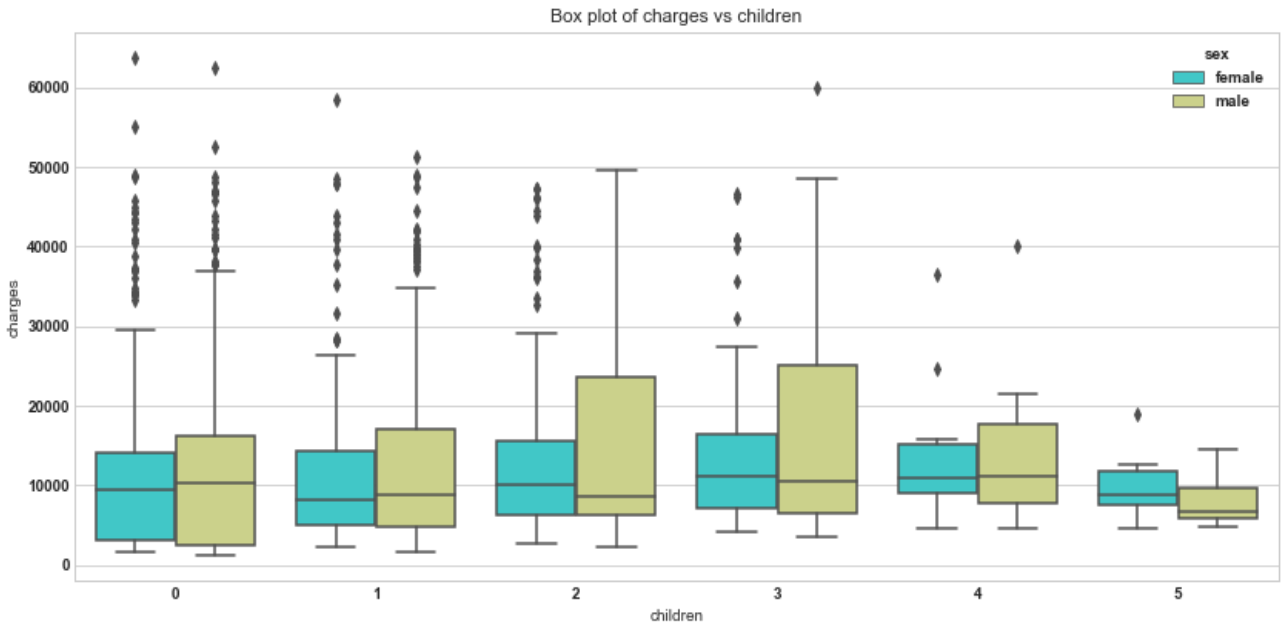
There is no correlation among variables.





If we look at the left plot the charges vary from 1120 to 63500, the plot is right skewed. In the right plot we will apply a natural log, then plot

approximately tends to normal. for further analysis we will apply log on target variable charges.



We will predict the value for the target variable by using our model parameter for the test data set. Then compare the predicted value with the actual value in the test set. We compute Mean Square Error using formula

$$\mathbf{J}(\theta) = \frac{1}{\mathbf{m}} \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{m}} (\hat{\mathbf{y}}_\mathbf{i} - \mathbf{y}_\mathbf{i})^2$$

R Square is a statistical measure of how close data are to the fitted regression line. R square is always between 0 to 100%. 0% indicated that the model explains none of the variability of the response data around it's

mean. 100% indicated that the model explains all the variability of the response data around the mean.

$$R^2 = 1 - \frac{SSE}{SST}$$

SSE = Sum of Square Error
SST = Sum of Square Total

$$SSE = \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

$$SST = \sum_{i=1}^{m} (y_i - \bar{y}_i)^2$$

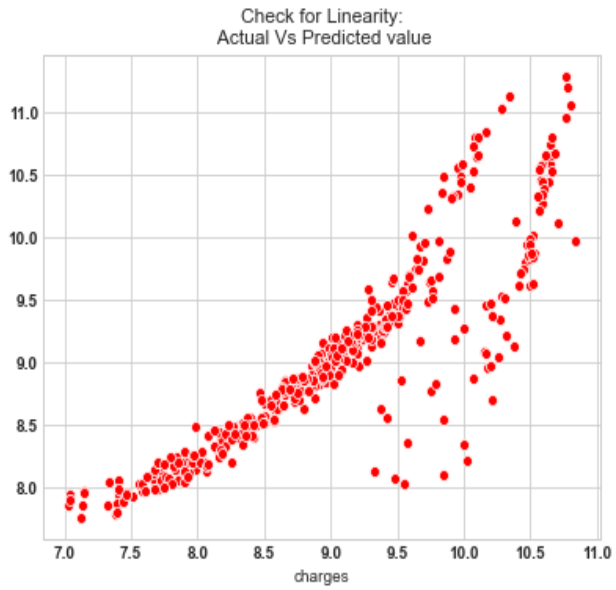
Accuracy returned by the model with the normal equation:

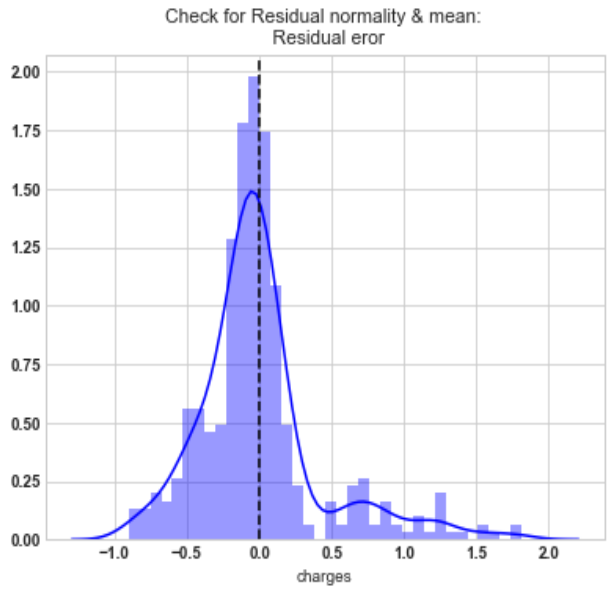
	MSE / J(theta)	**RMSE**
ACCURACY	0.18729622322981962	0.779568754505531

Accuracy returned by the model by using scikit learn library:

	MSE / J(theta)	**RMSE**
ACCURACY	0.1872962232298189	0.7795687545055318

Model Validation:





Variance Inflation Factor: `4.536561945911135`

CONCLUSION:

1. In our model the actual vs predicted plot is curve so linear assumption fails
2. The residual mean is zero and residual error plot right skewed
3. The plot is heteroscedastic, error will increase after a certain point.
4. Variance inflation factor value is less than 5, so no multicollearity.

APPENDIX:

Link to the dataset:

https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction