# MEDICAL INSURANCE COST PREDICTION

A COURSE PROJECT REPORT

By

**KOLIPAKULA DWARAKADEESH (RA2011026010229)**

**CSE (AI-ML) R1Section**



**FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chengalpattu District**

NOVEMBER 2022

**Table of Contents**

1. TITLE
2. ABSTARCT
3. DATASET DESCRIPTION
   1. INFORMATION OF DATASET
   2. CONTENT OF DATASET
4. MODULES DESCRIPTION
   1. ARCHITECTURE DIAGRAM
   2. ALGORITHM USED
5. RESULTS AND DISCUSSION
   1. CONFUSION MATRIX
   2. MEASURES OF DATASET
   3. SPLITTING THE FEATURES AND TARGET
   4. MODEL ANALYSIS
   5. MODEL EVALUTION
   6. BUILNDING A PREDICTIVE SYSTEM
6. CONCLUSION
7. APPENDIX

**2. Abstract: -**

Different persons have different body and different living culture. So, the diseases effection on them or the treatment expense for them also vary with respect to these situations. For example medical insurance premium of a smoker person may require more amount than a normal person without any bad habits, since the probability of occurrence of chronic disease are more for smokers. The main aim of this application is to help users to get overall idea about the appropriate medical insurance premium suitable for their living culture, gender, and some other factors. By using liner

**3. Dataset Description: -**

The data file is a separated value in the file format with rows and columns. It contains 7 columns and 1338 rows

* 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

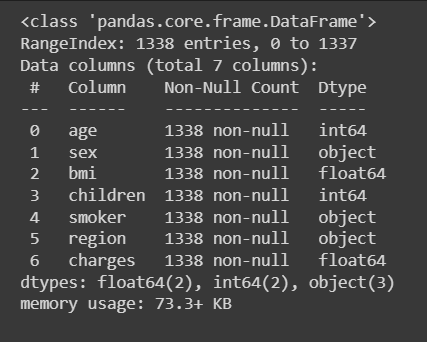
****

Fig3.1 This image talks about the basic information of the dataset

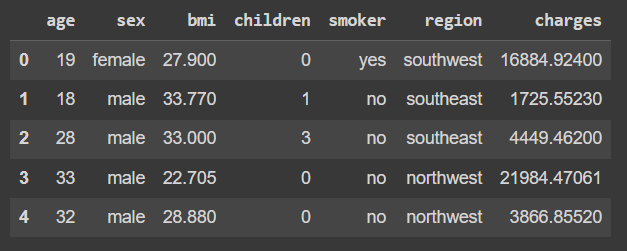
****

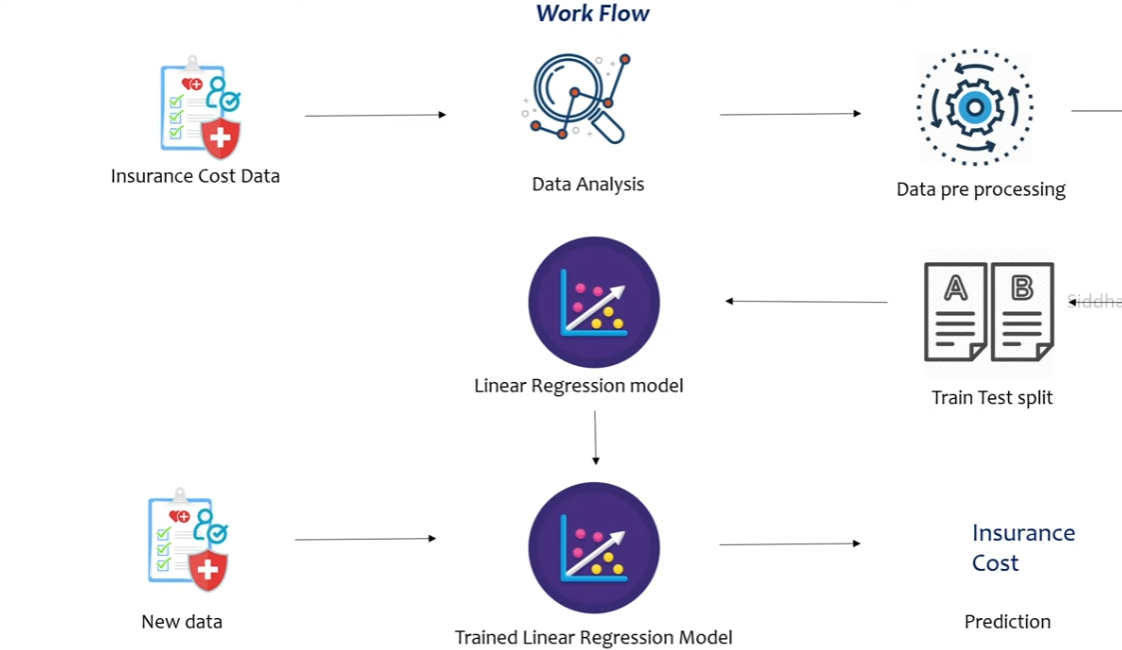
Fig3.2 This image talks about the content of the dataset

**4. Modules Description: -**

4.1 Architecture Diagram: -

Steps involved:

1. Collection of the data: - Collecting the data for the medical records for the insurance
2. Data analysis: - Analysis the data with help of graphs
3. Data preprocessing
4. Train test split: - To check the performance of the data
5. Machine learning model: - Linear regression model
6. Collection of new data
7. Training of Linear regression model
8. Prediction: - Insurance cost prediction



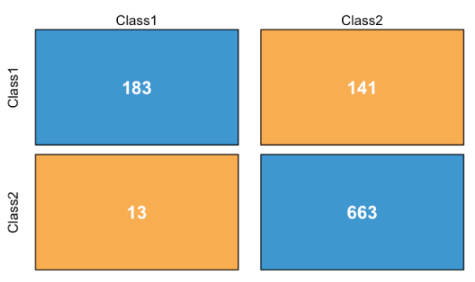
4.2 Algorithm used: -

Linear regression model

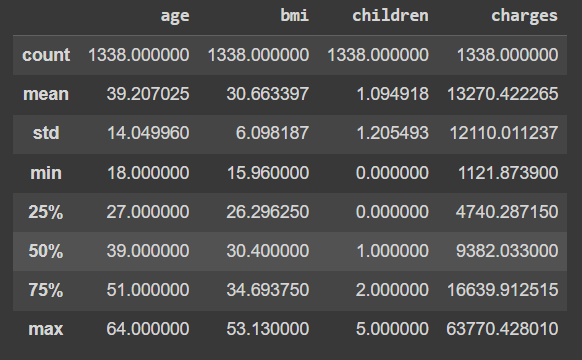
**5. Results and Discussion: -**

5.1 Confusion matrix: -

Confusion matrix is used to measure the accuracy and precision of the machine learning model



5.2 Measures of the dataset: -



5.2.1 Distribution of age value: -

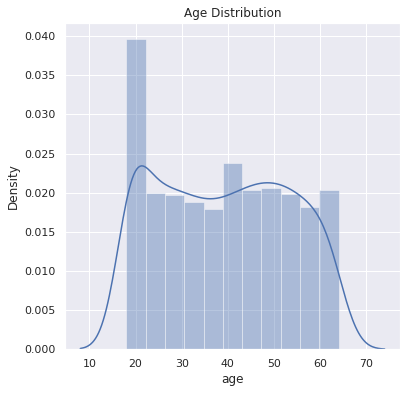
sns.set()

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['age'])

plt.title('Age Distribution')

plt.show()



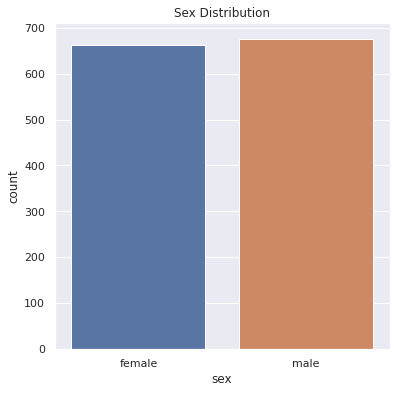
5.2.2 Gender column: -

plt.figure(figsize=(6,6))

sns.countplot(x='sex', data=insurance\_dataset)

plt.title('Sex Distribution')

plt.show()



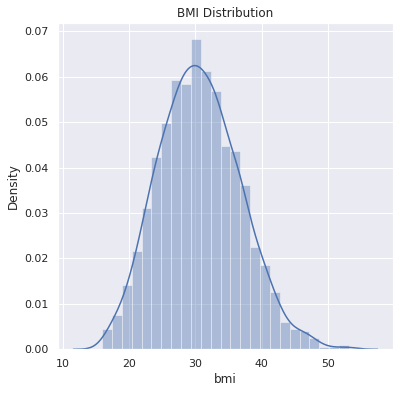
5.2.3 bmi distribution: -

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['bmi'])

plt.title('BMI Distribution')

plt.show()



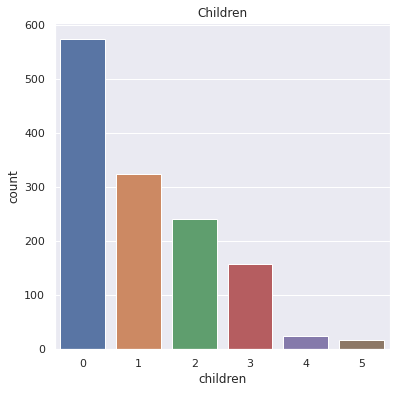
5.2.4 Children column: -

plt.figure(figsize=(6,6))

sns.countplot(x='children', data=insurance\_dataset)

plt.title('Children')

plt.show()



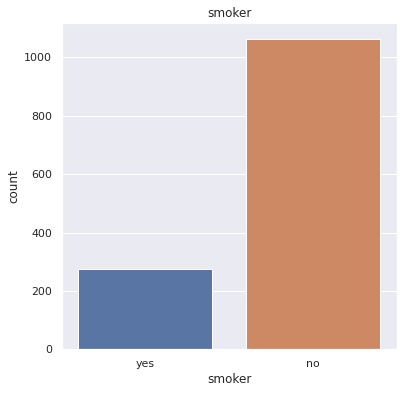
5.2.5 Smoker column: -

plt.figure(figsize=(6,6))

sns.countplot(x='smoker', data=insurance\_dataset)

plt.title('smoker')

plt.show()



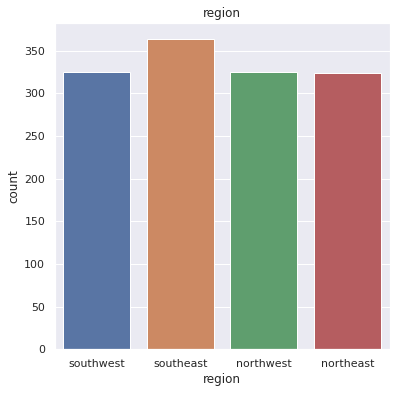
5.2.6 Region column

plt.figure(figsize=(6,6))

sns.countplot(x='region', data=insurance\_dataset)

plt.title('region')

plt.show()



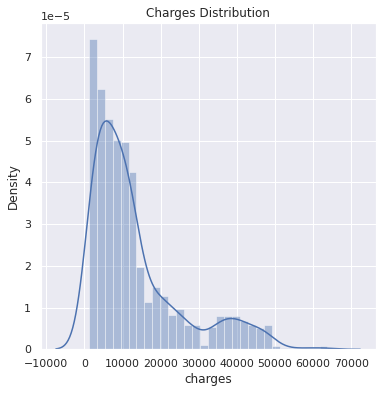
5.2.7 Distribution of charges value

plt.figure(figsize=(6,6))

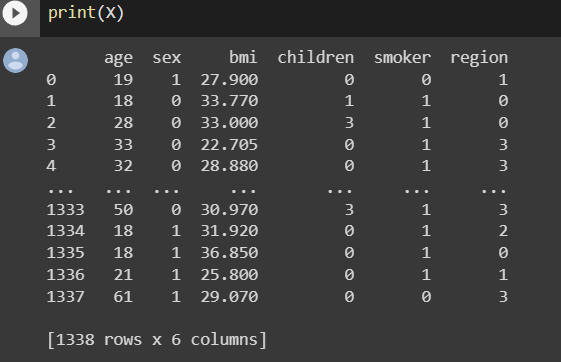
sns.distplot(insurance\_dataset['charges'])

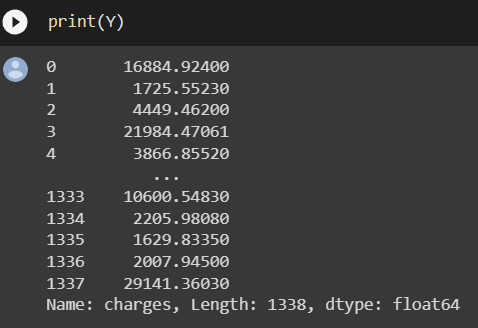
plt.title('Charges Distribution')

plt.show()



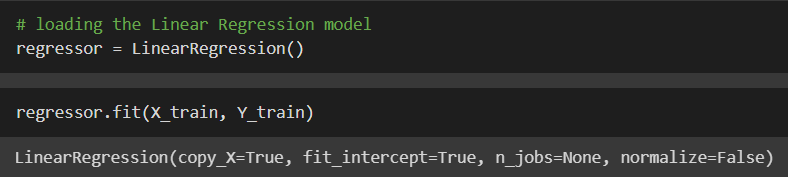
**5.3 Splitting the features and target: -**



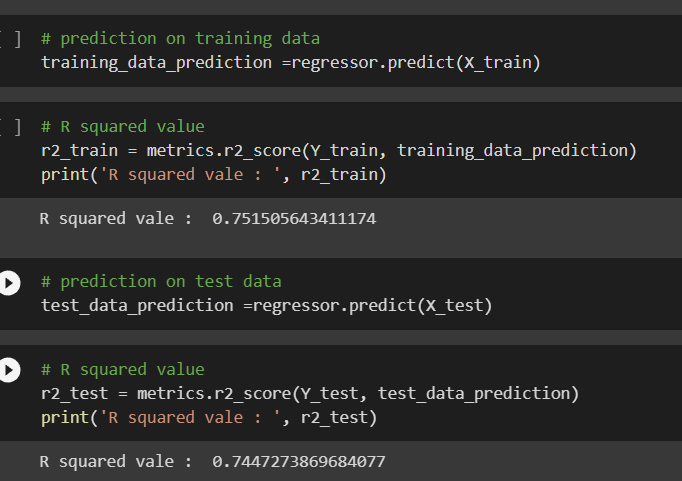


**5.4 Model analysis: -**

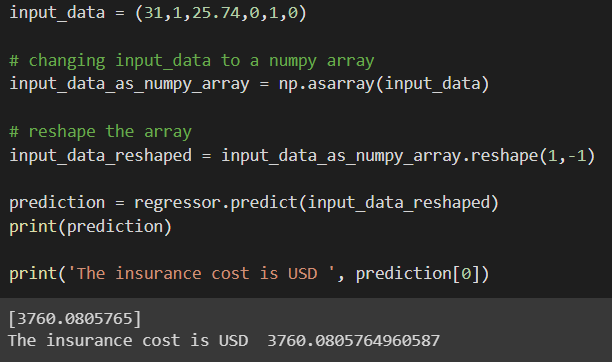
Linear regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.



**5.5 Model evaluation: -**

****

**5.6 Building a predictive system: -**

****

**6. Conclusion: -**

The model is built for the medical insurance prediction using linear regression model. By using this model, we can predict the medical insurance cost of the person.

1. **Appendix: -**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

# loading the data from csv file to a Pandas DataFrame

insurance\_dataset = pd.read\_csv('/content/insurance.csv')

# number of rows and columns

insurance\_dataset.shape

# getting some informations about the dataset

insurance\_dataset.info()

# checking for missing values

insurance\_dataset.isnull().sum()

# statistical Measures of the dataset

insurance\_dataset.describe()

# distribution of age value

sns.set()

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['age'])

plt.title('Age Distribution')

plt.show()

# Gender column

plt.figure(figsize=(6,6))

sns.countplot(x='sex', data=insurance\_dataset)

plt.title('Sex Distribution')

plt.show()

insurance\_dataset['sex'].value\_counts()

# bmi distribution

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['bmi'])

plt.title('BMI Distribution')

plt.show()

# children column

plt.figure(figsize=(6,6))

sns.countplot(x='children', data=insurance\_dataset)

plt.title('Children')

plt.show()

insurance\_dataset['children'].value\_counts()

# smoker column

plt.figure(figsize=(6,6))

sns.countplot(x='smoker', data=insurance\_dataset)

plt.title('smoker')

plt.show()

insurance\_dataset['smoker'].value\_counts()

# region column

plt.figure(figsize=(6,6))

sns.countplot(x='region', data=insurance\_dataset)

plt.title('region')

plt.show()

insurance\_dataset['region'].value\_counts()

# distribution of charges value

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['charges'])

plt.title('Charges Distribution')

plt.show()

# encoding sex column

insurance\_dataset.replace({'sex':{'male':0,'female':1}}, inplace=True)

3 # encoding 'smoker' column

insurance\_dataset.replace({'smoker':{'yes':0,'no':1}}, inplace=True)

# encoding 'region' column

insurance\_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest':3}}, inplace=True)

X = insurance\_dataset.drop(columns='charges', axis=1)

Y = insurance\_dataset['charges']

print(X)

print(Y)

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=2)

print(X.shape, X\_train.shape, X\_test.shape)

# loading the Linear Regression model

regressor = LinearRegression()

regressor.fit(X\_train, Y\_train)

# prediction on training data

training\_data\_prediction =regressor.predict(X\_train)

# R squared value

r2\_train = metrics.r2\_score(Y\_train, training\_data\_prediction)

print('R squared vale : ', r2\_train)

# prediction on test data

test\_data\_prediction =regressor.predict(X\_test)

# R squared value

r2\_test = metrics.r2\_score(Y\_test, test\_data\_prediction)

print('R squared vale : ', r2\_test)

input\_data = (31,1,25.74,0,1,0)

# changing input\_data to a numpy array

input\_data\_as\_numpy\_array = np.asarray(input\_data)

# reshape the array

input\_data\_reshaped = input\_data\_as\_numpy\_array.reshape(1,-1)

prediction = regressor.predict(input\_data\_reshaped)

print(prediction)

print('The insurance cost is USD ', prediction[0])