Predicting Healthcare Insurance Expenses: A Machine Learning Approach with Random Forest Regression

V.P. HARRISH
ADITYA M
harrish.vp2022@vitstudent.ac.in
aditya.mukherjee2022@vitstudent.ac.in

Abstract - The increasing volume of insurance costs poses a significant challenge for insurance companies in efficiently processing and assessing costs. This project focuses on developing a predictive modelling system using Python to enhance the accuracy and speed of insurance cost predictions. Leveraging machine learning algorithms and statistical analysis, the system aims to predict the estimated cost of healthcare insurance, providing valuable insights to insurance companies for proactive decision-making.

Keywords: Regression, Correlations, Decision trees, Random forest approach.

I. INTRODUCTION

Healthcare insurance is one of the most sought-out preventive measures in the field of Insurance. It not only provides a safety net for families in case of an unexpected emergencies like car crashes, medical complications etc, but also provide a degree of security for a family's finance in case of such an unexpected complication. 'Healthcare Insurance' is an umbrella term encompassing a wide variety of sub-categories like Medical Insurance, Preventive Care and Wellness Services Coverage, Maternity and Childbirth Service Coverage, etc. The aim of this project is to predict the cost that an individual has to pay yearly, to avail the insurance. This prediction model is based on factors such as Body Mass Index (BMI) of a person, whether a person smokes or not, number of children, and their region.

Insurance cost prediction involves the application of advanced analytical techniques to evaluate historical cost data, identify patterns, and make informed predictions about the likelihood of a cost being valid or fraudulent. The objective is to empower insurance companies with a predictive model that not only expedites cost processing but also adds a layer of intelligence for proactive decision-making.

.

II. LITERATURE REVIEW

Health insurance is a critical component of healthcare financing systems worldwide, providing individuals and families with financial protection against the high costs of medical care. Extensive research has been conducted to understand the factors influencing health insurance costs, improve the affordability and accessibility of insurance coverage, and optimize healthcare financing mechanisms.
Some of the factors influencing the cost are: -

1. Age and Health Status: Healthcare insurance is heavily dependent on the general wellbeing of a person's health. To give an idea, individuals who smoke and have higher Body Mass Index tend to have higher insurance cost payments while relatively young people with no prior history of smoking, alcohol or substance abuse tend to have lesser costs. This shows a highly positive correlation between charges and smoking, BMI, and age.

2. Demographic Characteristics: Gender can influence healthcare utilization patterns and medical expenses, potentially impacting insurance costs. For example, women may have higher healthcare costs during reproductive years due to pregnancy and maternity care. Marital status and household size may affect insurance costs, with family coverage typically being more expensive than individual coverage.

3. Geographic Location: Regional variations in healthcare costs, provider reimbursement rates, and regulatory environments can influence insurance costs. Areas with higher medical inflation rates or provider shortages may experience higher premiums. Urban versus rural disparities in healthcare access and healthcare delivery infrastructure can impact insurance costs and healthcare utilization patterns.

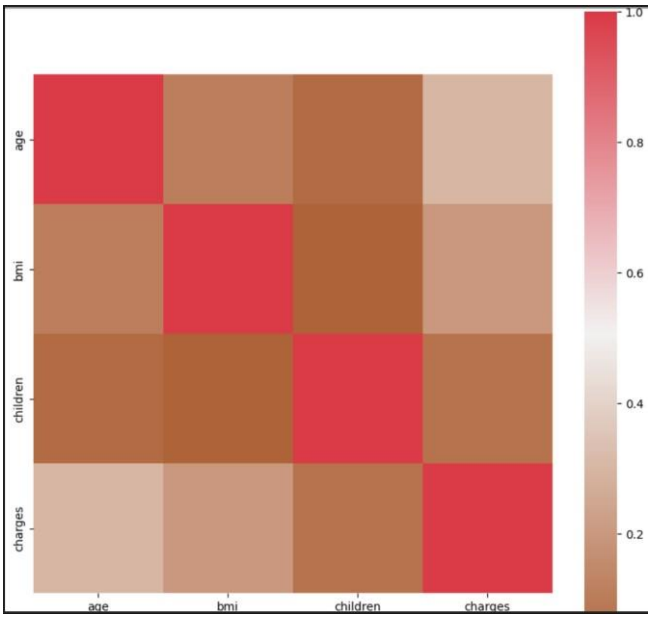
III. OBJECTIVE

The purpose of this project is to provide an accurate estimation of how much it costs for a person to avail healthcare insurance, using various factors such as age, gender, BMI, smoking status, region, and number of children. The predictive machine learning model designed in python (Jupyter Notebook) makes use of various python libraries such as PyTorch, NumPy, Sckitlearn, matplotlib, seaborn sns, and Pandas for analyzing the dataset of previous insurance costs, and visualizing the relationship between each factor in various statistical representations such as HeatMaps, violon charts, regression plots etc, and predicting a person's insurance costs based on the aforementioned factors using machine learning

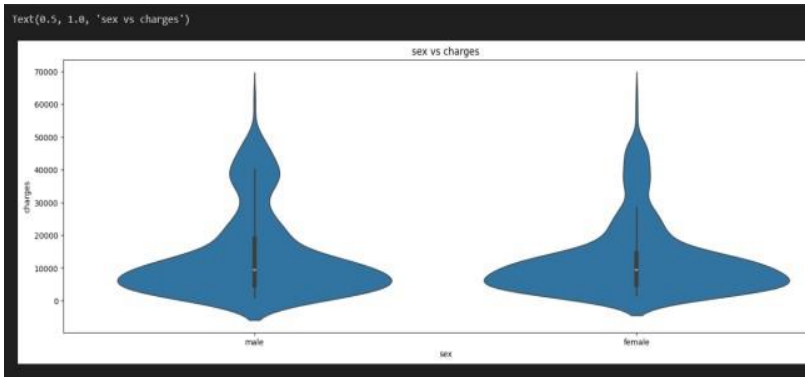
IV. METHODOLOGY

1. Data visualization using Pandas, Seaborn and NumPy: -

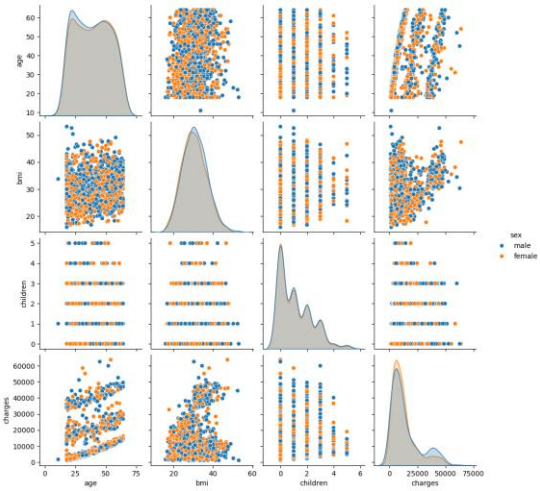
'Pandas' module is used to abstract the csv file containing the dataset of insurance costs based on factors of age, BMI, gender, smoking status, region and number of children. The data is then fed into 'Seaborn' module where functions like sns.lmplot(age, charges) is used to visualize the regressional analysis of age and charge factor. Then, a 4x4 statistical chart is visualized using sns.pairplot(data) to see the relation between every factors. After multiple other visualizations like violin chart between men's insurance charges and women's, a heatmap is plotted to study the correlation between all of these factors. This helps us in effectively interpreting data from the csv file.



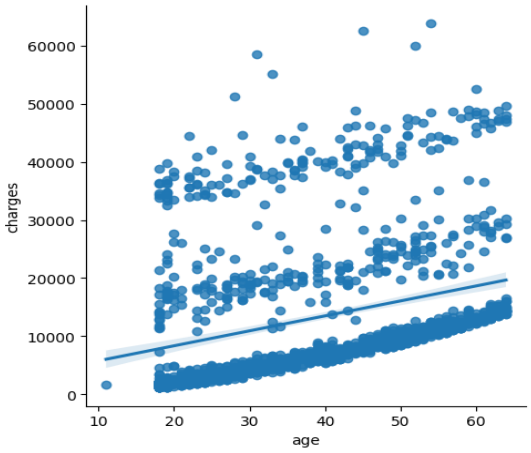
A. A HeatMap showing the correlation between every factor



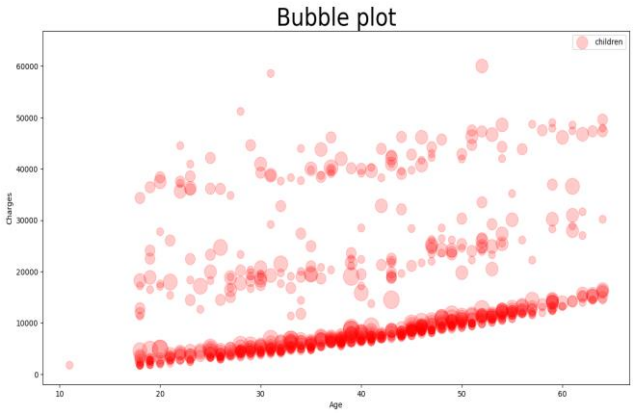
B. A violin chart showing the variations between the insurance costs of men and women. (Note the bump in men's chart around $40,000. This shows that men pay more than women for insurance).



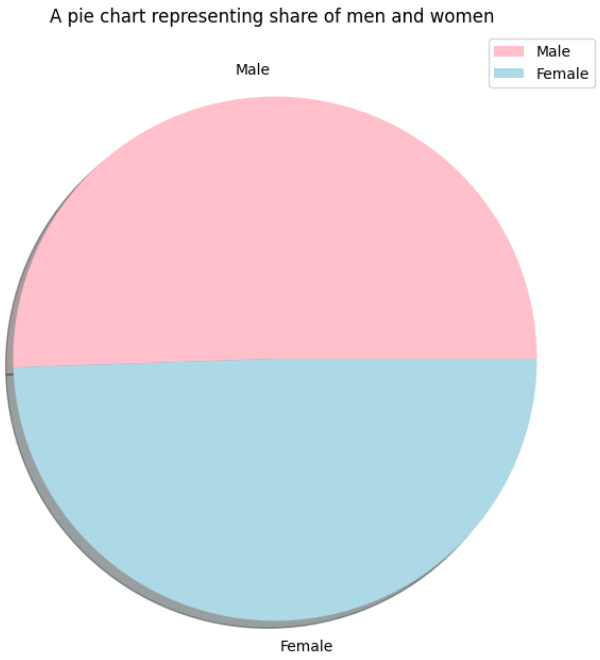
C. A 16x16 plot visualising the various data parameters, from the classification point of view of male and female.



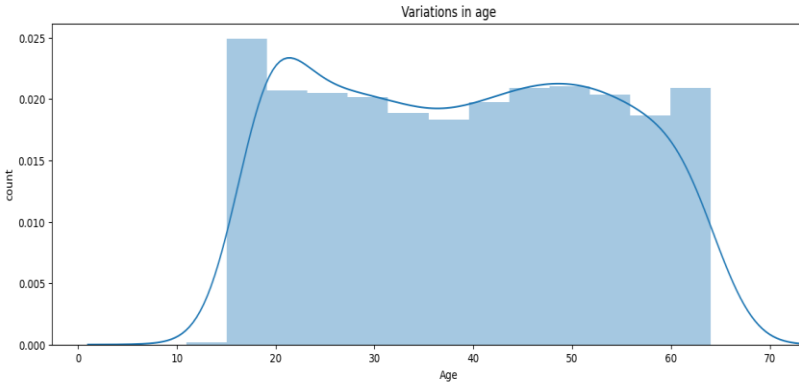
D. Regressional relation between age vs charges.



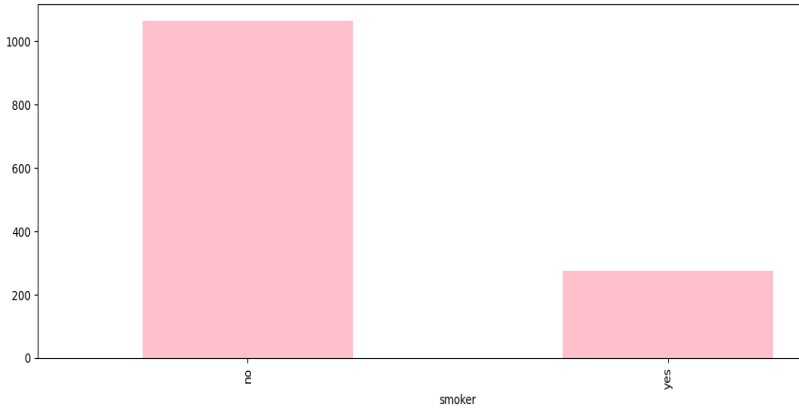
E. Bubble plot showcasing the relation between age, charges and children (bubbles – children)



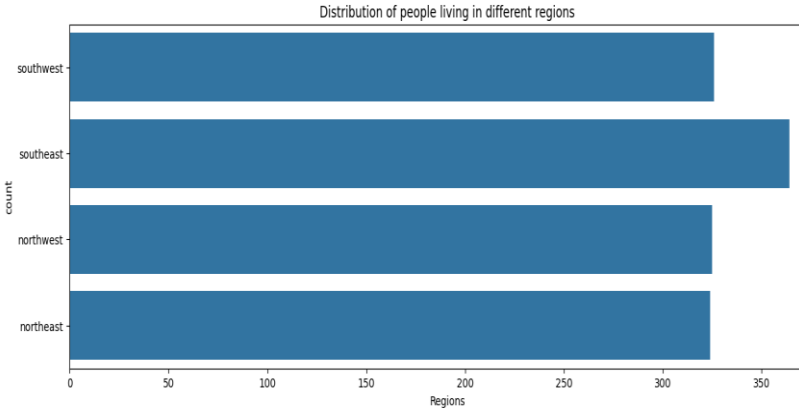
F. Pie chart representing the ratio of Male vs Female in the dataset.



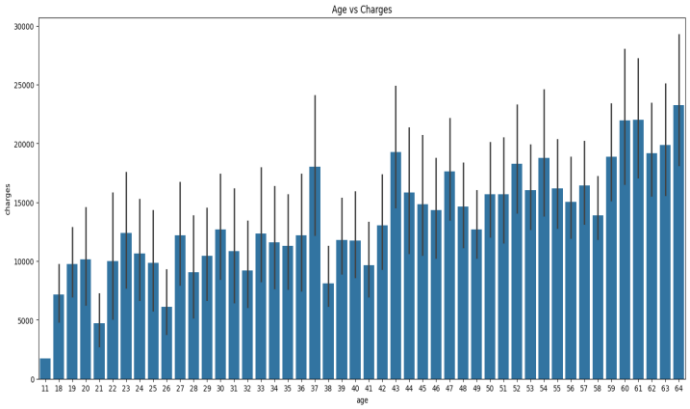
G. Visualising the age of customers.



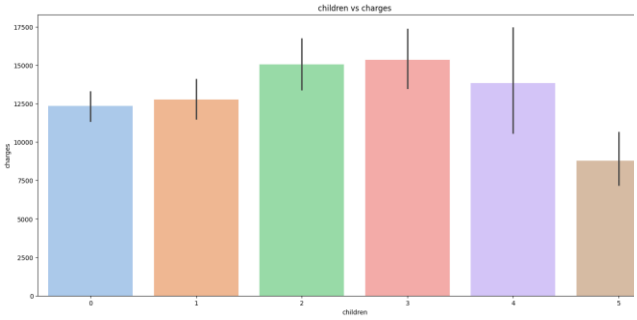
H. Representing the number of smokers and non-smokers.



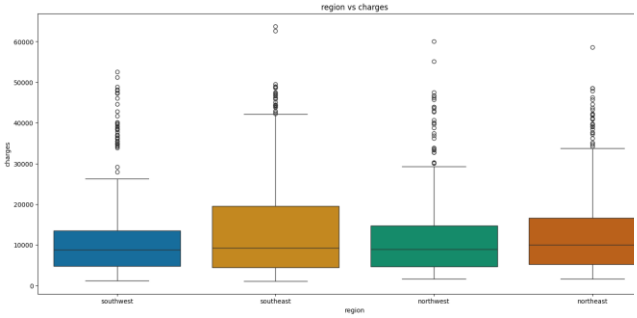
I. Distribution of people living in different regions (Note that there is not much of a difference in the charges incurred and region. Thus, including this parameter contributes to 'overfitting' of the model.).



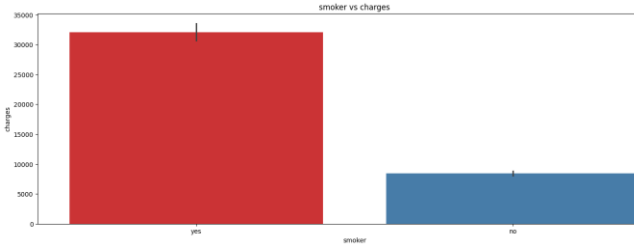
J. Bar graph relation of age vs charges (Note that age and charges are directly related).



K. Bar graph of children vs charges.



L. Bar graph of region vs charges (It is observable that there is not a concrete relation between these 2 factors).



M. Bar graph of smokers vs charges (Smokers pay more insurance charges than non-smokers).

2. Cost estimation using Sckit-learn and random forest regression analysis: -

Next step, the data is used to predict the cost of a person's health insurance. This is done using Random Forest approach. In Machine Learning, RFA is a method of training a model using randomly selected data. RFA contains numerous recursively spawned decision trees in random order, with a set amount of maximum decision trees known as estimators. This works by splitting the data into 3:1 ratio in a random manner, and using 75% data to train the model and using the rest 25% to test the efficiency of the data. The exact working of this model is briefed up below: -

a. At first, using the LabelEncoder function from Sklearn, the 2 tailed factors like gender (either male or female), smoking history(either yes or no) are assigned a binary value. This is for the ML model to interpret the factors.

b. Next, iLoc() function is used to select data using an integer-based position. This allows us to access specific rows and columns using integer-based indexing.

c. The dataset is divided in 3:1 ratio using Skearn's train-test split module. This splits the data into 3 parts training set and 1 part testing set. The training set is used to train the ML Model while the testing set is used to generate predicted values from the training model, compare it with the actual value and calculate the error %.

d. The independent variable sets are fed into standard scaler where it converts the data to a dataset with mean=1 and variance=0. This improves the efficiency of the algorithm since now multiple varying factors lie on a single scale, thus reducing variance bias.

e. Now, an empty model is created for fitting the data. Since the method used here is RFA, RandomForestRegressor(estimators=100,depth=4) is called. The estimators is used to specify the number of decision trees to be spawned and the depth specify the height of each decision tree.

f. Now, the training data is fed into this ML Model, and prediction result parameters are calculated. Using the testing data, 3 important parameters MSE, RMSE and R2 score is calculated. [MSE=Mean Square

Error, RMSE=Root Mean Square Error]. RMSE provides the offset values of the predicted result. To give an idea, if the predicted value of insurance cost is $7k and RMSE is $2000, then the value may lie anywhere between $9000 and $5000. This estimation is shaped like a bell curve with the $9k and $5k being the least probable and $7k being the most. R2 provides the accuracy of the ML Model by comparing the predicted value of testing data and it's original value, and calculating the error %.

g. Since the model is now done, it can predict the cost of insurance. The required details are entered by the user. Now, the inputs are converted into a standardized scaling, and fit into the ML Model, and the predicted value is shown as output.

In summary, this ML Model, accompanied with various python modules, is used for analyzing the given data set and drawing inferences between each variable from it while predicting the insurance values.

V. GAP

In the realm of insurance cost prediction using Python, a comprehensive exploration of related work is crucial to inform the project's development. This involves delving into existing Python libraries and frameworks such as Pandas, NumPy, scikit-learn, TensorFlow, and PyTorch, understanding their applications in data manipulation and machine learning. Examining similar open-source projects provides valuable insights into effective implementation strategies.

By harnessing the capabilities of Python, a language renowned for its versatility, ease of integration, and extensive libraries for data analysis and machine learning, this project seeks to provide a robust and adaptable solution. Python's rich ecosystem, including libraries like Pandas, NumPy, and scikitlearn, facilitates seamless data manipulation, preprocessing, and model development. Additionally, Python's readability and ease of collaboration make it an ideal choice for implementing a solution that can be integrated into existing insurance cost prediction.

VI. DRAWBACKS

Although the ML Model is fairly robust in predicting values, it falls short on certain aspects, such as: -

1. The r2 score from the testing data is estimated at 0.8081, which means the ML Model is 80% accurate in predicting the cost values. 80% is a relatively low accuracy% in a field where >95% is considered as the norm.

2. Estimated RMSE score is 5440 (approx.) which means that the difference between predicted value and actual value is >=5440, which is fairly high degree of offset.

3. In the input dataset fed into the ML Model, a lot of bias can be observed. For instance, people who smoke and have 0 kids only compromise 8% of the data and in that, it is observed that there is a net negative cost difference in contrast with the inference data i.e people who smoked paid lesser insurance costs than people who didn't. This creates bias due to the lack of data, which results in a low prediction accuracy%

```
# Prepare input data for prediction
customer_data = pd.DataFrame({
    'age': [22],
    'sex': [1],  # Assuming male is encoded as 1
    'bmi': [30.6],  # You need to provide BMI for the customer
    'children': [0],  # You need to provide the number of children for the customer
    'smoker': [0],  # Assuming the customer is a smoker
})

# Standardize input data
customer_data_scaled = sc.transform(customer_data)

# Predict insurance amount
predicted_insurance_amount = model.predict(customer_data_scaled)


print('The predicted insurance amount is: ',predicted_insurance_amount,' per annum')
print('On including offsets, your total cost estimates around: ',predicted_insurance_amount+5000, 'to',

✓ 0.0s
The predicted insurance amount is:  [6485.69456929]  per annum
On including offsets, your total cost estimates around:  [11485.69456929] to [1485.69456929]  per annum
```

A. People who don't smoke

```
# Prepare input data for prediction
customer_data = pd.DataFrame({
    'age': [18],
    'sex': [1],  # Assuming male is encoded as 1
    'bmi': [22.7],  # You need to provide BMI for the customer
    'children': [0],  # You need to provide the number of children for the customer
    'smoker': [1],  # Assuming the customer is a smoker
})

# Standardize input data
customer_data_scaled = sc.transform(customer_data)

# Predict insurance amount
predicted_insurance_amount = model.predict(customer_data_scaled)


print('The predicted insurance amount is: ',predicted_insurance_amount,' per annum')
print('On including offsets, your total cost estimates around: ',predicted_insurance_amount+5000, 'to',predicted_insurance_amoun

The predicted insurance amount is:  [4539.4521373]  per annum
On including offsets, your total cost estimates around:  [9539.4521373] to [-460.5478627]  per annum
```

B. People who smoke

INFERENCE: It can be observed from the 2 images that a 22 year old male with 0 children who smokes pays about $6400/annum, while for the same person who doesn't smoke pays $6500/annum. This directly contradicts the positive correlation obtained earlier, and this
inaccuracy raised due to only 8% of data specifying for this specific parameter

VII. CONCLUSION

In conclusion, the implementation of an insurance cost prediction system using Python demonstrates the transformative potential of data-driven decision-making in the insurance industry. By leveraging machine learning techniques, this project aims to enhance the efficiency of cost assessments, improve fraud detection capabilities if the ML Model is tuned to process it, and ultimately contribute to a more streamlined and customer-centric insurance process.

As we navigate the complexities of insurance cost processing, it is imperative to acknowledge the ongoing evolution of this project. Continuous improvement, adaptability to changing patterns, and adherence to ethical considerations remain integral to the long-term success of the predictive model. The applications of this project extend beyond the realms of efficiency and cost reduction, encompassing areas of risk management, customer satisfaction, and strategic decision-making within the insurance domain.

Looking forward, the integration of such predictive models into real-world insurance ecosystems requires a holistic approach. Collaboration with industry experts, adherence to regulatory standards, and a commitment to transparency are crucial for ensuring the ethical and responsible use of predictive analytics in the insurance sector. As the project advances, its impact on cost processing efficiency and fraud mitigation is expected to contribute significantly to the broader landscape of insurance operations.

VIII. REFERENCES

[1] *Seaborn, Heatmap [Online], Available:https://seaborn.pydata.org/generated/seaborn.heatmap.html*

[2] *Sckit-Learn, Random Forrest Classifier [Online], Available: https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier. html*

[3] *Dataset: Arun Jangir, and Willian Aliveira. (2023). Healthcare Insurance [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/6678394*

[4] *Sahu, Ajay and Sharma, Gopal and Kaushik, Janvi and Agarwal, Kajal and Singh, Devendra, Health Insurance Cost Prediction by Using Machine Learning (February 22, 2023). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022, Available at SSRN: https://ssrn.com/abstract=4366801 or http://dx.doi.org/10.2139/ssrn.4366801*

[5] *Source Code: -*

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)
import os
print(os.listdir(r"C:\Users\vphar\Downloads\input"))

# for data visualizations
import matplotlib.pyplot as plt
import seaborn as sns

# reading the data
data1 =
pd.read_csv(r"C:\Users\vphar\Downloads\input\insurance.cs
v")

# checking the shape
print(data1.shape)

# checking the head of the dataset
data1.head()

# describing the data
data1.describe()

# checking if the dataset contains any NULL values
data1.isnull().any()
if data1.isnull().any().any():
    raise ValueError("Error: The CSV file contains null
values.")
else:
    print("No null valued detected. Proceeding with
further processing.....")
sns.pairplot(data1,hue="smoker")

# lmplot between age and charges
sns.lmplot(x='age', y='charges', data=data1)

# lmplot between age and charges
```

```
sns.lmplot(x='age', y='charges', data=data1)

# bubble plot to show relation bet age, charges and
children
plt.rcParams['figure.figsize'] = (15, 8)
plt.scatter(x = data1['age'], y = data1['charges'], s =
data1['children']*100, alpha = 0.2, color =
'red',label='children')
plt.title('Bubble plot', fontsize = 30)
plt.xlabel('Age')
plt.ylabel('Charges')
plt.legend()
plt.show()

# unique value counts in the sex category
data1['sex'].value_counts()

# pie chart
size = [676, 662]
colors = ['pink', 'lightblue']
labels = "Male", "Female"
plt.rcParams['figure.figsize'] = (8, 8)
plt.pie(size, colors = colors, labels = labels, shadow =
True)
plt.title('A pie chart representing share of men and
women ')
plt.legend()
plt.show()

# visualizing the ages of the customers
plt.rcParams['figure.figsize'] = (15, 5)
sns.distplot(data1['age'])
plt.title('Variations in age')
plt.xlabel('Age')
plt.ylabel('count')
plt.show()

# visualizing how many childrens the customers have
sns.countplot(data1['children'])
plt.title('Distribution of no.of Childrens')
plt.xlabel('NO. of Childrens')
plt.ylabel('count')
plt.show()

# checking how many people smoke
data1['smoker'].value_counts().plot.bar(color = 'pink')

# visualizing the regions from where the people belong
sns.countplot( data1['region'])
plt.title('Distribution of people living in different
regions')
plt.xlabel('Regions')
plt.ylabel('count')
plt.show()

# Age vs Charges
# the more the age the more will be insurance charge
(roughly estimated)
plt.figure(figsize = (18, 8))
sns.barplot(x = 'age', y = 'charges', data = data1)
plt.title("Age vs Charges")

# sex vs charges
# males have slightly greater insurance charges than
females in general
plt.figure(figsize = (18, 6))
sns.violinplot(x = 'sex', y = 'charges', data = data1)
plt.title('sex vs charges')

# children vs charges
# no. of childrens of a person has a very interesting
dependency on insurance costs
plt.figure(figsize = (18, 8))
sns.barplot(x = 'children', y = 'charges', data = data1,
palette ='pastel',hue='children')
plt.title('children vs charges')

# region vs charges
# From the graph we can see that the region actually does
not play any role in determining the insurance charges
plt.figure(figsize = (18, 8))
sns.boxplot(x = 'region', y = 'charges', data = data1,
palette = 'colorblind',hue='region')

plt.title('region vs charges')
# smoker vs charges
# from the graph below, it is visible that smokers have
more insurance charges than the non smokers
plt.figure(figsize = (18, 6))
sns.barplot(x = 'smoker', y = 'charges', data = data1,
palette = 'Set1',hue='smoker')
plt.title('smoker vs charges')

# Selecting only numeric columns
numeric_data1 = data1.select_dtypes(include=np.number)

# Plotting the correlation plot for the dataset
f, ax = plt.subplots(figsize=(10, 10))
corr = numeric_data1.corr()
sns.heatmap(corr, mask=np.zeros_like(corr, dtype=bool),
            cmap=sns.diverging_palette(30, 10,
as_cmap=True), square=True, ax=ax)

# removing unnecassary columns from the dataset
data = data1.drop('region', axis = 1)
print(data.shape)
data.columns

# label encoding for sex and smoker
# importing label encoder
from sklearn.preprocessing import LabelEncoder

# creating a label encoder
le = LabelEncoder()

# label encoding for sex
# 0 for females and 1 for males
data['sex'] = le.fit_transform(data['sex'])

# label encoding for smoker
# 0 for smokers and 1 for non smokers
data['smoker'] = le.fit_transform(data['smoker'])
# splitting the dependent and independent variable

x = data.iloc[:,:5]
y = data.iloc[:,5]

print(x.shape)
print(y.shape)
# splitting the dataset into training and testing sets

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size = 0.2, random_state = 30)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

# standard scaling
from sklearn.preprocessing import StandardScaler

# creating a standard scaler
sc = StandardScaler()

# feeding independents sets into the standard scaler
```

```
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)


# feature extraction
from sklearn.decomposition import PCA
pca = PCA(n_components = None)
x_train = pca.fit_transform(x_train)
x_test = pca.transform(x_test)

# REGRESSION ANALYSIS
# RANDOM FOREST
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

# creating the model
model = RandomForestRegressor(n_estimators = 40,
max_depth = 4, n_jobs = -1)

# feeding the training data to the model
model.fit(x_train, y_train)

# predicting the test set results
y_pred = model.predict(x_test)

# calculating the mean squared error
mse = np.mean((y_test - y_pred)**2, axis = None)
print("MSE :", mse)

# Calculating the root mean squared error
rmse = np.sqrt(mse)
print("RMSE :", rmse)

# Calculating the r2 score
r2 = r2_score(y_test, y_pred)
print("r2 score :", r2)

# Prepare input data for prediction
customer_data = pd.DataFrame({
    'age': [18],
    'sex': [1],  # Assuming male is encoded as 1
    'bmi': [22.7],  # You need to provide BMI for the
customer
    'children': [0],  # You need to provide the number of
children for the customer
    'smoker': [1],  # Assuming the customer is a smoker
})

# Standardize input data
customer_data_scaled = sc.transform(customer_data)

# Predict insurance amount
predicted_insurance_amount =
model.predict(customer_data_scaled)



print('The predicted insurance amount is:
',predicted_insurance_amount,' per annum')
print('On including offsets, your total cost estimates
around: ',predicted_insurance_amount+5000,
'to',predicted_insurance_amount-5000,' per annum')
```