A CERTIFIED TRAINING

ON AIML

Title- Travel Insurance Prediciton System

Submitted in partial fulfillment of the Required Credits for the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE(CSE)

SUBMITTED TO:

Ms. Naresh Kumar Marwal

Assistant professor
Department of science and technology
Jayoti vidyapeeth women's university
Jaipur, Rajasthan

SUBMITTED BY:

jv'n PRACHI DEVI

jv-u/20/4225

FACULTY OF EDUCATION AND METHDOLOGY

DEPARTMENT OF SCIENCE AND TECHNOLOGYJAYOTI VIDHYAPEETH WOMEN'S UNIVERSITY,JAIPUR, RAJASTHAN

India's First State Private Women's University

JAYOTI VIDYAPEETH WOMEN'S UNIVERSITY
ज्योति विद्यापीठ महिला विश्वविद्यालय

CERTIFICATE

The report is hereby approved as a bonafide and certified training work "**AIML**" carried out and presented by JV'n Prachi Devi(jv-u/20/4225)) in a manner to warrant its acceptance in partial fulfillment of the required credits for the degree of Bachelor of Technology. However, the undersigned do not necessarily endorse or take responsibility for any statement or opinion expressed or conclusion drawn there in, but only approve the report for the purpose for which it is submitted.

Jv'n Naresh Kumar Marwal
Supervisor
Department of Science & Technology
Jayoti Vidyapeeth Women's University
Jaipur,Rajasthan

Jv'n Nishu Sharma
Coordinator
Department of Science and Technology
Jayoti Vidyapeeth Women's University
Jaipur, Rajasthan

Jv'n Prof. Dr. Shobha Lal
Dean and Proctor
Faculty of Education and Methodology
Jayoti Vidyapeeth Women's University ,Jaipur, Rajasthan

ACKNOWLEGMENT

I express my gratitude to all those who helped me to prepare and complete my dissertation work entitled , I convey my deep gratitude and heart full thanks to **Honorable Chairperson Ma'am jv'n Mrs. Mithlesh Garg Jayoti Vidyapeeth Women's University** for her inspiration, Cooperation and encouragement for pursuing my dissertation. Her valuable Suggestion and guidance helped me a lot to complete my work in this instruction within a very short period.

I render my sincere respect and heart full gratitude to my mam Jv'n Ms. Nishu Sharma. I also thankful to all the faculty members, for their Valuable Suggestion towards completing the dissertation work. I am also grateful to my classmates, who helped me directly or indirectly in completing my Dissertation work successfully.

Last but not least, I am really grateful to my parents, who remained a constant Source of encouragement and inspiration during the completion of this work Successfully in Jayoti Vidyapeeth Women's University, Jaipur.

JV'n Prachi Devi
(jv-u/20/4225)

Department of Science and Technology
Jayoti Vidyapeeth Women's University, Jaipur

DECLARATION

"I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contain no material previously Published or written by another person nor materials which have been accepted for the award of any other degree or diploma of any university or institution of higher learning, except where due to acknowledgment has been made in the text.

Place- Jaipur **Jv'n Prachi Devi**
Date – july 26, 2022 **(jv-u/20/4225)**

Introduction

Insurance risk assessment has long been modeled by actuaries using techniques combining concepts from actuarial science, probability, statistics, finance and economy. During the last decades, important improvements in computer performance, advances in the field of machine learning and a boost of access to data-driven information has led to an increased interest in adopting trends and outcome, is now knows as a promising Path for insurance R&D, including for marketing, underwriting ,fraud detecting, pricing and valuation.

Travel insurance is a distinct line of business which can be found both as an individual and group health insurance exposure. Claims are characterized by low incidence frequency, but a heavily skewed severity tail for high claims given the potential for extremely high medical expenses for out of country emergencies. Grounds are therefore usually small, but high volatility of claims for more severe incidents requires precautions for modeling future claims. The lack of literature regarding theoretical applications to travel claims combinned with the ease of the data available specific to this line of business leads to great opportunities for developing statistical models. Furthermore, increased democratization of travelling is a source of motivation for refining methods of risk assessment.

Travel Insurance

Canadian insurance companies started offering travel coverage in the early 90's, while government reduced the medical coverage for out-of-country emergencies. Today, the government covers approximately 7% of travel medical claims. Given this minimal coverage, additional coverage is highly recommended when planning a trip abroad. Travel insurance coverage can be found as a standalone individual insurance product. It can also be part of a group insurance provided by an employer. The coverage includes medical claims and sometimes baggage loss or trip cancellation fees. This study focuses specifically on medical claims. In group insurance, employees are entitled to group insurance coverage, and the premium payment is usually split with their employer. Travel insurance can be part of the coverage which most commonly includes disability, life and other health insurances. Each insured employee is given a single certificate number, to which one or multiple additional insurers can be added. An additional insured is known as a dependant and could be a spouse or a child. Therefore, different types of protections are available, such as single, family, spouse or single parent. Insurance policy systems usually aggregate information like the date of birth, the employment status, the salary, the occupation, the maximum amount and duration of the coverage and the postal code, for both the main insured and the dependants. In this case, the list of insurers includes the list of all employees at risk of making a claim, whether or not a trip is planned, and no information regarding the trip is available prior to a claim.

Libraries which I used in this model

1. **Numpy**
2. **Pandas**
3. **Matplotlib**
4. **Seaborn**
5. **Skleanr**

Numpy

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

NumPy stands for Numerical Python.

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

Data Science: is a branch of computer science where we study how to store, use and analyze data for deriving information from it.

Pandas

Pandas is an open source Python package that is most widely

used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like Activen State's Active Python.

Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data

Matblotlib

Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

Matplotlib was created by John D. Hunter.

Matplotlib is open source and we can use it freely.

Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

Seaborn

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

Distplot stands for distribution plot, it takes as input an array and plots a curve corresponding to the distribution of points in the array.

In [3]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC

%matplotlib inline
```

Linear Model

This section introduces the theoretical concepts of Generalized Linear Models (GLMs), which are often regarded as the most appropriate models for an Incidence and Severity analysis. Indeed, they enable to relate explanatory variables to a response variable with multiplicative factors which can easily be interpreted and used for actuarial rating. Furthermore, they allow great flexibility for the probability distribution function of the response variable which is in most cases more appropriate than the normal distribution used in Ordinary least squares (OLS) regression. Indeed, a conventional linear regression model is defined as:

$$E(y|x) = x > \beta$$

Used Regression Models in project

Linear regression is suitable for problems where we want to predict a

certain numerical value, as opposed to a “yes or no” prediction where we use logistic regression.

For example, if an insurance company wants to predict whether an individual is likely to die early (a “yes or no” prediction) and gets to claim the insurance, they should use logistic regression.
But here, we want to predict the insurance cost of an individual—and we are going to use linear regression to do that.

Insurance Prediction with Machine Learning

The task of insurance prediction is something that adds value to every insurance company. They use data from their database about everyone they have contacted to promote their insurance services and try to find the most potential people who can buy insurance. This helps a company to target the most profitable customers and saves time and money for the Insurance Company.

Types of Regression Analysis Techniques

There are many types of regression analysis techniques, and the use of each method depends upon the number of factors. These factors include the type of target variable, shape of the regression line, and the number of independent variables.

Below are the different regression techniques:

1. Linear Regression

2. Logistic Regression

1. Linear Regression

Linear regression is one of the most basic types of regression in machine learning. The linear regression model consists of a predictor variable and a dependent variable related linearly to each other. In case the data involves more than one independent variable, then linear regression is called multiple linear regression models.

The below-given equation is used to denote the linear regression model:

$$y=mx+c+e$$

where m is the slope of the line, c is an intercept, and e represents the error in the model.

2. Logistic Regression

Logistic regression is one of the types of regression analysis technique, which gets used when the dependent variable is discrete. Example: 0 or 1, true or false, etc. This means the target variable can have only two values, and a sigmoid curve denotes the relation between the target variable and the independent variable.

Logistic function is used in Logistic Regression to measure the relationship between the target variable and independent variables. Below is the equation that denotes the logistic regression.

In addition to the above regression methods, there are many other types of regression in machine learning, including Elastic Net Regression, Jackknife Regression, Stepwise Regression, and Ecological Regression.

These different types of regression analysis techniques can be used to build the model depending upon the kind of data available or the one that gives the maximum accuracy. You can explore these techniques more or can go through the course of supervised learning on our website.

```
In [64]:
y_test.value_counts()
```

```
Out[64]:
0    383
1    214
Name: TravelInsurance, dtype: int64
```

```
In [70]:
model = LogisticRegression()
model.fit(x_train, y_train)
```

```
C:\Users\Dell\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
Out[70]:
LogisticRegression()
```

Chronic Diseases are more in Non frequent flyers

Encoding Object Data Values

```
In [63]:
data.dtypes[data.dtypes == 'object']
```

```
Out[63]:
Employment Type    object
dtype: object
```

```
In [64]:
data['Employment Type'].unique()
```

```
Out[64]:
array([0, 'Private Sector/Self Employed'], dtype=object)
```

```
In [65]:
data['Employment Type'].replace(['Government Sector', 'Private Sector/Sedatalf Employed'],[0, 1]
```

```
In [66]:
data['GraduateOrNot'].unique()
```

```
Out[66]:
array([1, 0], dtype=int64)
```

```
In [67]:
data['GraduateOrNot'].replace(['Yes', 'No'], [1, 0], inplace=True)
```

```
In [68]:
data['FrequentFlyer'].unique()
```

```
Out[68]:
array([0, 1], dtype=int64)
```

```
In [69]:
data['FrequentFlyer'].replace(['Yes', 'No'], [1, 0], inplace=True)
```

```
In [70]:
data['EverTravelledAbroad'].unique()
```

```
Out[70]:
array([0, 1], dtype=int64)
```

```
In [71]:
data['EverTravelledAbroad'].replace(['Yes', 'No'], [1, 0], inplace=True)
```

Scaling Age, Annual Income and Family Members

In [72]:
```
cols_to_scale = ['Age', 'AnnualIncome', 'FamilyMembers']
scale = MinMaxScaler()
scale.fit(data[cols_to_scale])
```

Out[72]:
```
MinMaxScaler()
```

In [73]:
```
scaled = scale.fit_transform(data[cols_to_scale])
```

In [74]:
```
for i in range(3):
    data[cols_to_scale[i]] = scaled[:,i]
```

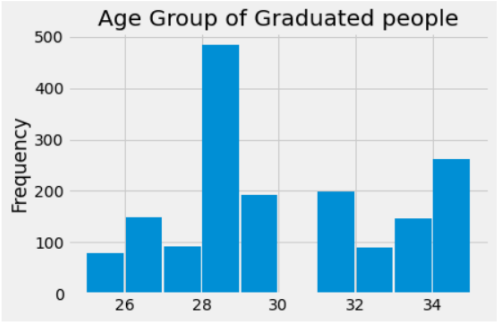
In [75]:
```
data.head()
```

Out[75]:

	Unnamed: 0	Age	Employment Type	GraduateOrNot	AnnualIncome	FamilyMembers	ChronicDiseases
0	0	0.6	0	1	0.066667	0.571429	1
1	1	0.6	Private Sector/Self Employed	1	0.633333	0.714286	0
2	2	0.9	Private Sector/Self Employed	1	0.133333	0.285714	1
3	3	0.3	Private Sector/Self Employed	1	0.266667	0.142857	1
4	4	0.3	Private Sector/Self Employed	1	0.266667	0.857143	1

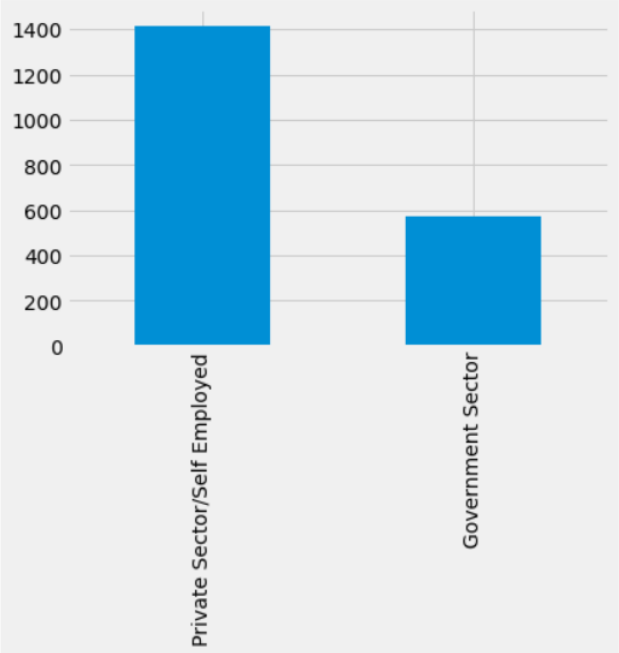
Age Group with respect to GraduateOrNot

In [14]:
```
plt.title('Age Group of Graduated people')
plt.style.use('fivethirtyeight')
data['Age'][data['GraduateOrNot'] == 'Yes'].plot(kind='hist', rwidth=0.95)
plt.show()
```

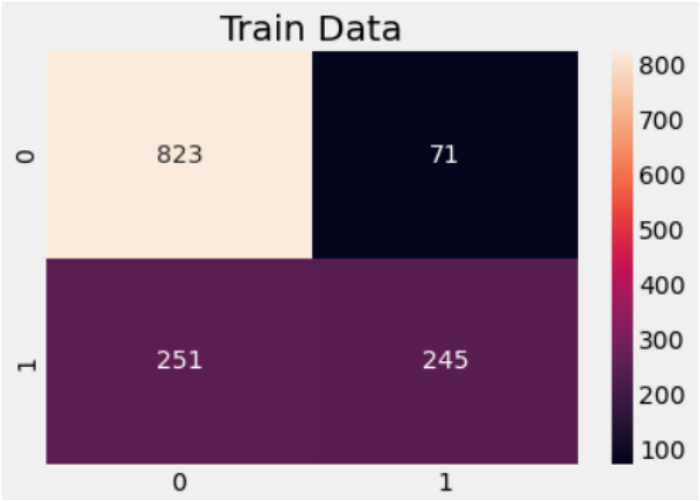


In [18]:

```
plt.style.use('fivethirtyeight')
data['Employment Type'].value_counts().plot(kind='bar')
plt.show()
```

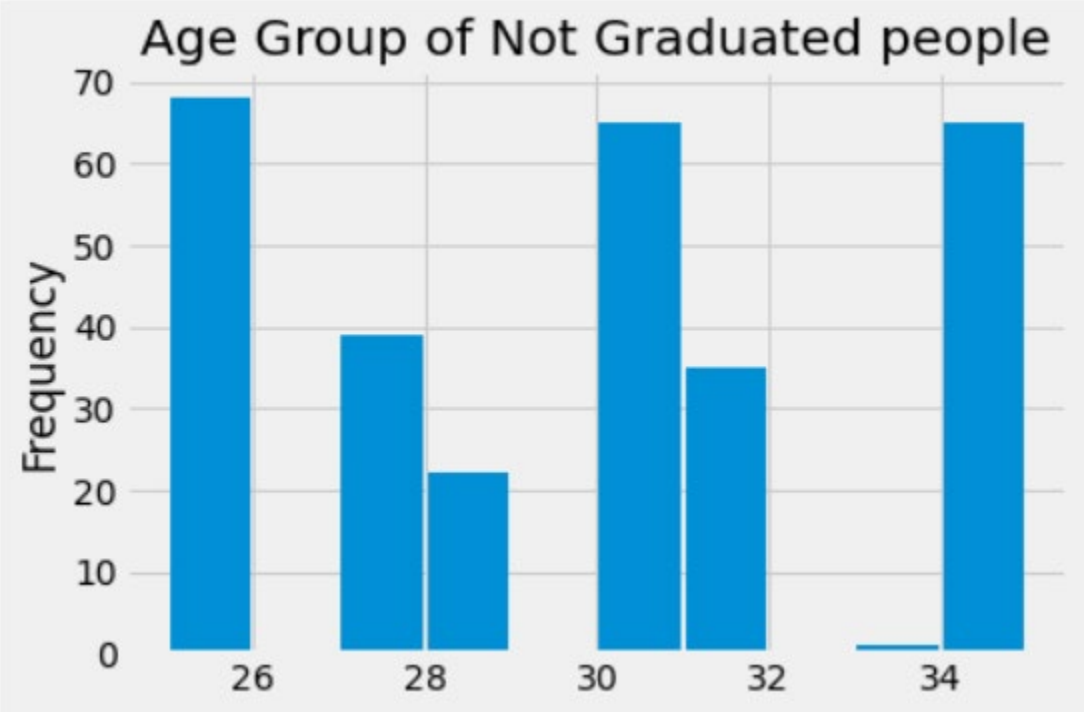


In [72]:
```
plt.title('Test Data')
sns.heatmap(confusion_matrix(y_test,y_pred_test), annot=True, fmt='g')
plt.show()
print('\n')
plt.title('Train Data')
sns.heatmap(confusion_matrix(y_train,y_pred_train), annot=True, fmt='g', )
plt.show()
```



In [5]:
```
data.info()
```

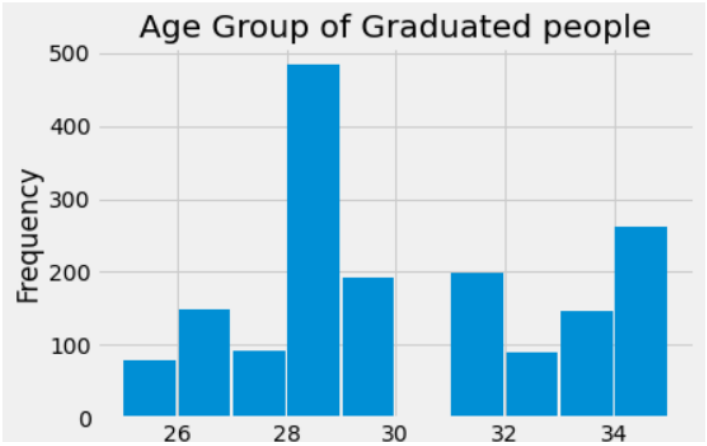
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1987 entries, 0 to 1986
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Unnamed: 0           1987 non-null   int64
 1   Age                  1987 non-null   int64
 2   Employment Type      1987 non-null   object
 3   GraduateOrNot        1987 non-null   object
 4   AnnualIncome         1987 non-null   int64
 5   FamilyMembers        1987 non-null   int64
 6   ChronicDiseases      1987 non-null   int64
 7   FrequentFlyer        1987 non-null   object
 8   EverTravelledAbroad  1987 non-null   object
 9   TravelInsurance      1987 non-null   int64
dtypes: int64(6), object(4)
memory usage: 155.4+ KB
```

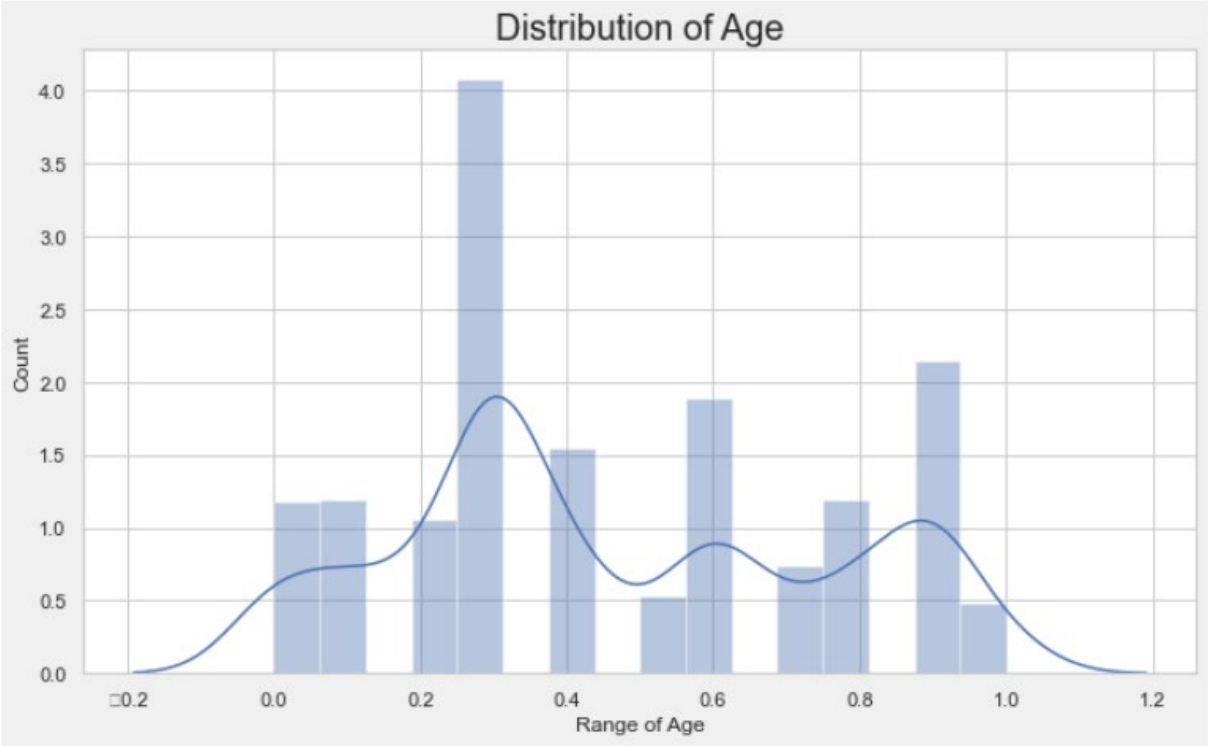


Age Group with respect to GraduateOrNot

In [14]:

```
1 plt.title('Age Group of Graduated people')
2 plt.style.use('fivethirtyeight')
3 data['Age'][data['GraduateOrNot'] == 'Yes'].plot(kind='hist', rwidth=0.95)
4 plt.show()
```




Distribution of Age
Count
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
□0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Range of Age

Code:-

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
%matplotlib inline

data=pd.read_csv("TravelInsurancePrediction.csv")
data

data.columns
data.isna().sum()
data.dtypes
data.head()
data['GraduateOrNot'].value_counts()
plt.figure(figsize=(10,8))
sns.heatmap(data.corr(),cbar=True,annot=True,cmap='Blues')
fig, ax = plt.subplots(figsize=(8,8))
sns.countplot(hue='Age',x='TravelInsurance',data=data)
plt.show()
fig, ax = plt.subplots(figsize=(8,8))
(data.groupby('Age').sum()['TravelInsurance']/data.groupby('Age').count
()['TravelInsurance']).plot(kind='bar')
plt.title('% of Insurance purchased breakdown by age')
plt.ylabel('Percentage')
plt.show()
plt.figure(figsize=(10, 6))
```

```
sns.set(style = 'whitegrid')
sns.distplot(data['Age'])
plt.title('Distribution of Age', fontsize = 20)
plt.xlabel('Range of Age')
plt.ylabel('Count')
data.hist(figsize=(12,8),bins=20)
plt.show()

data['AnnualIncome'].hist(color='green',bins=50,figsize=(8,4))
data['GraduateOrNot'].value_counts().plot(kind='bar')
plt.style.use('fivethirtyeight')
plt.show()
plt.title('Age Group of Graduated people')
plt.style.use('fivethirtyeight')
data['Age'][data['GraduateOrNot'] == 'Yes'].plot(kind='hist',
rwidth=0.95)
plt.show()
plt.title('Age Group of Not Graduated people')
plt.style.use('fivethirtyeight')
data['Age'][data['GraduateOrNot'] == 'No'].plot(kind='hist', rwidth=0.95)
plt.show()
data['Employment Type'].unique()
plt.style.use('fivethirtyeight')
data['Employment Type'].value_counts().plot(kind='bar')
plt.show()
data['AnnualIncome'].plot(kind='hist', rwidth=0.95)
plt.show()
emp_type = ['Government Sector', 'Private Sector/Self Employed']

for typ in emp_type:
    plt.title(f'Annual Income of {typ}')
    data['AnnualIncome'][data['Employment Type'] ==
typ].plot(kind='hist', rwidth=0.95)
    plt.show()
    print('\n')
```

```
data['FamilyMembers'].plot(kind='hist', rwidth=0.95)
plt.show()
data['FrequentFlyer'].unique()
data['FrequentFlyer'].value_counts().plot(kind='bar')
plt.show()
data['EverTravelledAbroad'].unique()
data['EverTravelledAbroad'].value_counts().plot(kind='bar')
plt.show()
data['EverTravelledAbroad'][data['FrequentFlyer'] ==
'Yes'].value_counts().plot(kind='bar')
plt.show()
data['EverTravelledAbroad'][data['FrequentFlyer'] ==
'No'].value_counts().plot(kind='bar')
plt.show()
data['TravelInsurance'].unique()
data['TravelInsurance'].value_counts().plot(kind='bar')
plt.show()
data['TravelInsurance'][data['EverTravelledAbroad'] ==
'Yes'].value_counts().plot(kind='bar')
plt.show()
plt.show()
data['TravelInsurance'][data['EverTravelledAbroad'] ==
'No'].value_counts().plot(kind='bar')
plt.show()
data['ChronicDiseases'].unique()
data['ChronicDiseases'].value_counts().plot(kind='bar')
plt.show()


data['ChronicDiseases'][data['FrequentFlyer'] ==
'Yes'].value_counts().plot(kind='bar')
plt.show()
data['ChronicDiseases'][data['FrequentFlyer'] ==
'No'].value_counts().plot(kind='bar')
plt.show()
```

```
plt.figure(figsize=(30, 10))
data.Age.value_counts(normalize=True)
data.Age.value_counts(normalize=True).plot.pie()
plt.show()
sns.displot(data['Employment Type'])
plt.show()
data.dtypes[data.dtypes == 'object']
data['Employment Type'].unique()
data['GraduateOrNot'].unique()
data['GraduateOrNot'].replace(['Yes', 'No'], [1, 0], inplace=True)
data['FrequentFlyer'].unique()
data['FrequentFlyer'].replace(['Yes', 'No'], [1, 0], inplace=True)
data['EverTravelledAbroad'].unique()
data['EverTravelledAbroad'].replace(['Yes', 'No'], [1, 0], inplace=True)
cols_to_scale = ['Age', 'AnnualIncome', 'FamilyMembers']
scale = MinMaxScaler()
scale.fit(data[cols_to_scale])

scaled = scale.fit_transform(data[cols_to_scale])
for i in range(3):
    data[cols_to_scale[i]] = scaled[:,i]
data.head()
x, y = data.drop('TravelInsurance',axis=1),data['TravelInsurance']
x.head()
y.head()
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
random_state=1)
x_train.shape, x_test.shape
y_train.value_counts()
y_test.value_counts()
model = LogisticRegression()
model.fit(x_train, y_train)
y_pred_test = model.predict(x_test)
y_pred_train = model.predict(x_train)
```

```
y_pred_test[y_pred_test >= 0.5] = 1
y_pred_test[y_pred_test < 0.5] = 0

y_pred_train[y_pred_train >= 0.5] = 1
y_pred_train[y_pred_train < 0.5] = 0
   plt.title('Test Data')
sns.heatmap(confusion_matrix(y_test,y_pred_test), annot=True, fmt='g')
plt.show()
print('\n')
plt.title('Train Data')
sns.heatmap(confusion_matrix(y_train,y_pred_train), annot=True,
fmt='g', )
plt.show()
print(classification_report(y_test, y_pred_test))
print(classification_report(y_train, y_pred_train))
```

https://docs.google.com/document/d/1aJCdLNJq13DPExjXRr2jGjN7etaqxU7K/edit