# *Travel Insurance Prediction*

By: Alwin Shaji Koshy - alwinshaji.koshy2021@vitstudent.ac.in
Nishan Mammen Sabu - nishanmammen.sabu2021@vitstudent.ac.in
Karthick - karthik.madhu2021@vitstudent.ac.in

## Predictive Modeling for Travel Insurance Purchase

Travel insurance is a critical component of travel planning, providing financial protection against unexpected events that can occur during a trip. These events may include medical emergencies, flight cancellations, lost or stolen luggage, and other travel-related mishaps. As global travel has increased, so too has the demand for travel insurance, with travelers seeking the most affordable and comprehensive coverage.
This predictive model utilizes machine learning, data mining, and statistical analysis techniques to identify patterns and trends in data. The model examines historical data on past travel insurance purchases, demographic information, and travel itineraries to predict the likelihood of an individual purchasing travel insurance.
The dataset for this project consists of customer database history, with the target variable being whether or not the customer purchased travel insurance. The goal of this project is to develop a predictive model that can accurately predict the likelihood of an individual purchasing travel insurance based on factors such as age, income, and number of family members.
Potential benefits of a predictive model for travel insurance purchase
A predictive model for travel insurance purchase can provide a number of benefits to both travelers and insurance companies. For travelers, it can help them to identify the factors that are most likely to influence their decision to purchase travel insurance. This information can be used to make more informed decisions about whether or not to purchase insurance, and to choose the right type of coverage for their needs.
For insurance companies, a predictive model can help them to better target their marketing efforts and to develop more effective pricing strategies. By understanding the factors that are most likely to influence a customer's decision to purchase travel insurance, insurance companies can tailor their marketing messages to reach the right customers and offer them the most competitive rates.
How the predictive model works
The predictive model for travel insurance purchase works by identifying patterns and trends in the data. The model analyzes historical data on past travel insurance

purchases, demographic information, and travel itineraries to identify factors that are associated with a higher or lower likelihood of purchasing travel insurance.

Once the model has identified these factors, it can be used to predict the likelihood of an individual purchasing travel insurance based on their own personal characteristics and travel plans. The model can be used to develop a score for each customer, indicating their overall likelihood of purchasing travel insurance.

How the predictive model can be used

The predictive model for travel insurance purchase can be used in a number of ways, both by travelers and insurance companies.

For travelers:

- Travelers can use the model to identify the factors that are most likely to influence their decision to purchase travel insurance. This information can be used to make more informed decisions about whether or not to purchase insurance, and to choose the right type of coverage for their needs.
- Travelers can also use the model to get a personalized score indicating their overall likelihood of purchasing travel insurance. This score can be used to compare different travel insurance options and to choose the policy that is right for them.

For insurance companies:

- Insurance companies can use the model to better target their marketing efforts. By identifying the factors that are most likely to influence a customer's decision to purchase travel insurance, insurance companies can tailor their marketing messages to reach the right customers and offer them the most competitive rates.
- Insurance companies can also use the model to develop more effective pricing strategies. By understanding the factors that are associated with a higher or lower likelihood of purchasing travel insurance, insurance companies can develop more accurate pricing models that reflect the true cost of providing coverage to different types of customers.

Conclusion

A predictive model for travel insurance purchase can provide a number of benefits to both travelers and insurance companies. By identifying patterns and trends in data, the model can help travelers to make more informed decisions about whether or not to purchase insurance, and to choose the right type of coverage for their needs. Insurance companies can also use the model to better target their marketing efforts and to develop more effective pricing strategies.

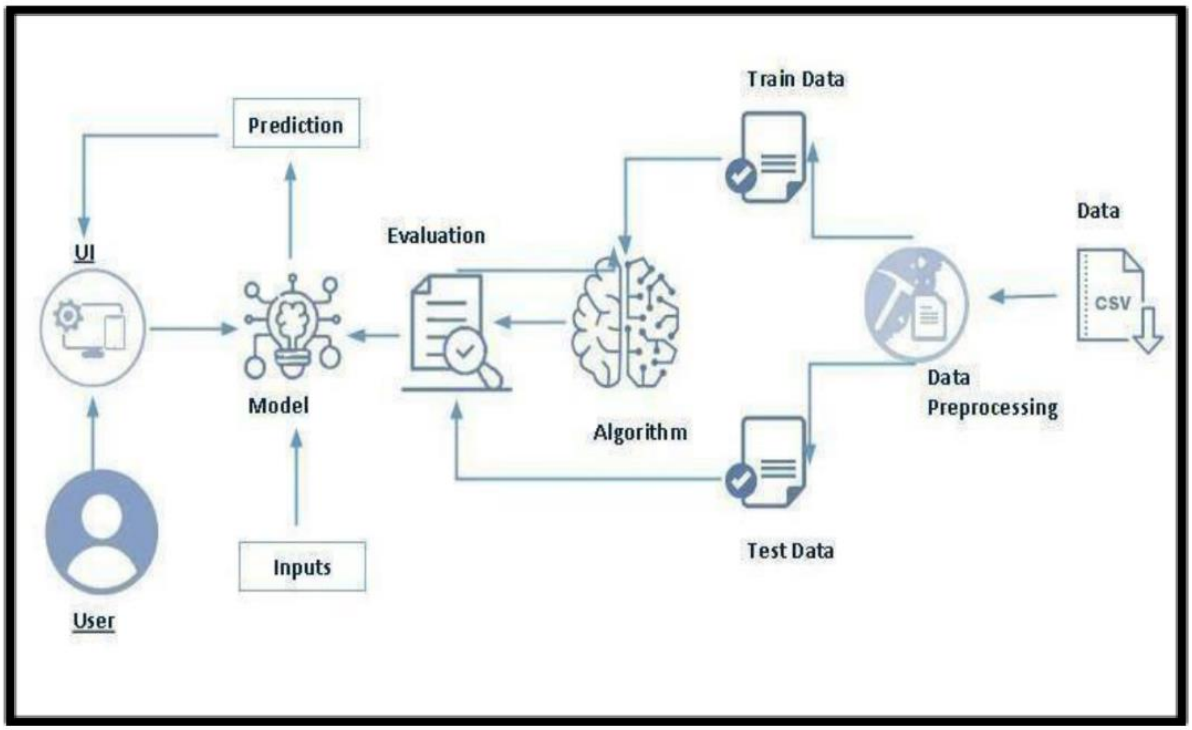
Potential applications of the predictive model

In addition to the benefits mentioned above, a predictive model for travel insurance purchase could also be used in a number of other ways. For example, the model could be used to:

- Develop personalized travel insurance recommendations for customers.
- Identify customers who are at a higher risk of experiencing travel-related problems.
- Develop targeted marketing campaigns for travel insurance products.
- Evaluate the effectiveness of different travel insurance marketing strategies.
- Conduct research on the factors that influence travel insurance purchase decisions.

Overall, a predictive model for travel insurance purchase has the potential to revolutionize the way that travel insurance is marketed, sold, and priced. By providing a more accurate understanding of the factors that influence customers' purchase decisions, the model can help both travelers and insurance companies to make better decisions.

## Technical Architecture:-



## Project Flow:-

● User interacts with the UI to enter the input.

● Entered input is analysed by the model which is integrated.

● Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

● Define Problem / Problem Understanding

  ○ Specify the business problem

  ○ Business requirements

  ○ Literature Survey

  ○ Social or Business Impact.

● Data Collection & Preparation

  ○ Collect the dataset

  ○ Data Preparation

● Exploratory Data Analysis

  ○ Descriptive statistical

  ○ Visual Analysis

● Model Building

  ○ Training the model in multiple algorithms

  ○ Testing the model

● Performance Testing

  ○ Testing model with multiple evaluation metrics

  ○ Comparing model accuracy

● Model Deployment

  ○ Save the best model

  ○ Integrate with Web Framework

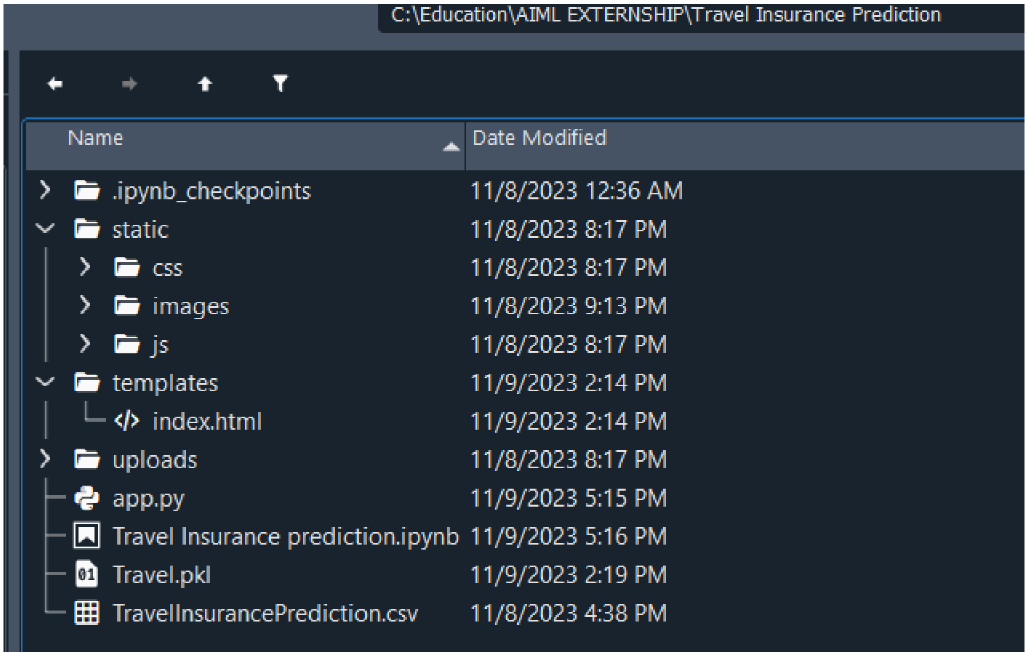
● Project Demonstration & Documentation

  ○ Record explanation Video for project end to end solution

  ○ Project Documentation-Step by step project development procedure

## Project Structure:-

Initially the project should be made using jupyter notebook and and after which the web would be developed using spyder interface.
The files for this would be in the order:-




Here Travel.pkl is the saved model by us in order to predict the desired output.

## Milestone 1: Define the Problem/ Problem Understanding

### Activity 1: Specify the business problem

Predicting Customer Interest in a New Travel Insurance Package

A travel company is offering a new travel insurance package that includes COVID cover. The company wants to identify which customers are most likely to be interested in purchasing the new package. To do this, the company will build a predictive model using data from its database history.

The data set includes information on almost 2000 previous customers, including their demographics, travel history, and whether or not they purchased travel insurance in 2019. The goal of the predictive model is to identify the factors that are most strongly correlated with a customer's purchase of travel insurance.

Once the model is trained, it can be used to predict the likelihood of each customer purchasing the new travel insurance package. This information can then be used by the company to target its marketing efforts and to develop more effective pricing strategies.

There are a number of factors that could influence a customer's decision to purchase travel insurance, including:

- Demographics: Age, income, family size, and travel experience can all play a role in a customer's decision to purchase travel insurance. For example, younger travelers may be more likely to purchase travel insurance, as they may be less familiar with the risks associated with travel.
- Travel history: The type of travel that a customer is planning can also influence their decision to purchase travel insurance. For example, travelers who are planning to travel to a remote destination or to engage in risky activities may be more likely to purchase travel insurance.
- Past behavior: If a customer has purchased travel insurance in the past, they are more likely to purchase it again in the future. This is because they have already experienced the benefits of travel insurance and are aware of the risks associated with travel.

In addition to these factors, the COVID-19 pandemic has also had a significant impact on travel insurance purchase decisions. Many travelers are now more likely to purchase travel insurance that includes COVID cover, as they are concerned about the risks associated with contracting the virus while traveling.

The predictive model that the travel company builds will be able to take all of these factors into account to predict the likelihood of each customer purchasing the new travel insurance package. This information can then be used by the company to make better decisions about how to market and price the new package.

Potential benefits of the predictive model

The predictive model can provide a number of benefits to the travel company, including:

- Improved targeting: The company can use the model to target its marketing efforts to the customers who are most likely to be interested in purchasing the new travel insurance package. This can lead to increased sales and a better return on investment for the company's marketing campaigns.
- More effective pricing: The company can use the model to develop more effective pricing strategies for the new travel insurance package. This can help the company to maximize its profits and to ensure that the package is affordable for its customers.
- Better understanding of customer needs: The model can provide the company with a better understanding of the factors that influence its customers' purchase decisions. This information can be used to improve the company's products and services and to develop new offerings that meet the needs of its customers.

Conclusion

By building a predictive model to predict customer interest in the new travel insurance package, the travel company can improve its targeting, pricing, and understanding of customer needs. This can lead to increased sales, a better return on investment, and a more successful product launch.

## Activity 2 :- Business Requirements

A travel insurance prediction system is a tool that can be used by insurers to predict the likelihood of a customer purchasing travel insurance. This information can then be used by insurers to target their marketing efforts, develop more effective pricing strategies, and better understand their customers' needs.

The business requirements for a travel insurance prediction system are essential to ensure that the system meets the needs of the travel insurance industry. Some potential requirements may include:

Accurate prediction: The primary requirement of a travel insurance prediction system is to generate accurate forecasts of the likelihood of a customer purchasing travel insurance. The accuracy of the prediction will directly impact the success of the system and the insurer's ability to make informed decisions.

Real-time access: The system must provide real-time access to the predictions, allowing insurers to make informed decisions quickly. This is critical as the travel industry is constantly evolving, and insurers must respond quickly to changes in the market. For example, if there is a sudden surge in demand for travel insurance to a particular destination, the insurer needs to be able to quickly identify the customers who are most likely to purchase insurance and target their marketing efforts accordingly.

Scalability: The system must be scalable, allowing it to handle large volumes of data as the travel industry grows. This requirement is essential as the system must be able to handle the increasing demand for travel insurance. For example, if the insurer launches a new marketing campaign that generates a large number of leads, the system must be able to quickly process the data and generate predictions for all of the leads.

User-friendly interface: The classification system should be easy to use and understand for all customers. This is important because the system may be used by customers to self-select for travel insurance, or by insurers to provide personalized recommendations to customers. For example, the system could be used to develop a "travel insurance risk assessment" tool that customers can use to assess their own risk and determine whether or not they need to purchase travel insurance.

In addition to the above requirements, the travel insurance prediction system should also be:

- Flexible: The system should be flexible enough to accommodate changes in the travel insurance industry. For example, the system should be able to handle new types of data, such as social media data or travel booking data.
- Secure: The system should be secure to protect the privacy of customer data.
- Cost-effective: The system should be cost-effective to develop and maintain.

By addressing these business requirements, a travel insurance prediction system can be developed that can provide significant benefits to insurers and customers alike.

Potential benefits of a travel insurance prediction system

A travel insurance prediction system can provide a number of benefits to insurers and customers alike, including:

For insurers:

- Improved targeting: The system can be used to target marketing efforts to the customers who are most likely to purchase travel insurance. This can lead to increased sales and a better return on investment for the insurer's marketing campaigns.
- More effective pricing: The system can be used to develop more effective pricing strategies for travel insurance products. This can help the insurer to maximize its profits and to ensure that the products are affordable for customers.
- Better understanding of customer needs: The system can provide the insurer with a better understanding of the factors that influence its customers' purchase decisions. This information can be used to improve the insurer's products and services and to develop new offerings that meet the needs of its customers.

For customers:

- Personalized recommendations: The system can be used to provide personalized recommendations to customers about whether or not they need to purchase travel insurance. This can help customers to make informed decisions about their travel insurance needs.
- More affordable premiums: By helping insurers to develop more effective pricing strategies, the system can lead to more affordable premiums for customers.
- Improved customer experience: By providing real-time access to predictions and a user-friendly interface, the system can improve the customer experience.

Overall, a travel insurance prediction system has the potential to revolutionize the travel insurance industry. By providing insurers with accurate and timely predictions, the system can help insurers to make better decisions about marketing, pricing, and product development. For customers, the system can provide personalized recommendations and more affordable premiums.

## Activity 3:- Literature Survey

A literature survey for a travel insurance prediction project would involve researching and reviewing existing studies, articles, and other publications on the topic of travel insurance purchase prediction. The survey would aim to gather information on current prediction models, their strengths and weaknesses, and any gaps in knowledge that the project could address. The literature survey would also look at the methods and techniques used in previous travel insurance prediction projects, and any relevant data or findings that could inform the design and implementation of the current project.

The literature survey would be essential for ensuring that the travel insurance prediction project is well-informed and that it addresses a significant gap in the existing knowledge base. By reviewing the literature, the project team can identify the most promising approaches to travel insurance prediction and develop a project that is likely to be successful.

Here are some specific areas that the literature survey could cover:

- Current travel insurance prediction models: What are the most commonly used prediction models in the field? What are their strengths and weaknesses?
- Factors that influence travel insurance purchase decisions: What are the most important factors that influence whether or not a customer purchases travel insurance?
- Data sources for travel insurance prediction: What types of data are most commonly used for travel insurance prediction? How can this data be collected and cleaned?
- Methods and techniques for travel insurance prediction: What methods and techniques have been used in previous travel insurance prediction projects? What are the advantages and disadvantages of each method?
- Gaps in knowledge: Are there any areas where the current knowledge base on travel insurance prediction is lacking? Are there any questions that the project could address that would be of significant value to the field?

By conducting a comprehensive literature survey, the project team can develop a travel insurance prediction project that is well-grounded in the existing knowledge base and that has the potential to make a significant contribution to the field.

# Activity 4:- Social Or Business Impact

## Social and Business Impact of Travel Insurance Prediction Model

The development of a travel insurance prediction model has the potential to have a significant impact on both the social and business aspects of the travel industry.

### Social Impact

On the social side, a travel insurance prediction model can help to ensure that travelers are adequately protected against unforeseen events that may occur during their trip. By analyzing historical data on past travel habits, insurers can develop customized insurance policies that meet the specific needs of individual travelers. This can provide travelers with peace of mind knowing that they are covered in the event of a medical emergency, flight cancellation, or other travel-related mishap.

### Business Impact

For businesses, a travel insurance prediction model can have a number of benefits. First, it can help to increase revenue by enabling insurers to optimize their marketing campaigns and offer tailored insurance policies to meet the specific needs of different segments of the travel market. By analyzing historical data and trends, insurers can accurately forecast the likelihood of travelers purchasing insurance. This information can then be used to target marketing campaigns more effectively and to develop insurance policies that are more likely to appeal to potential customers.

Second, a travel insurance prediction model can help businesses to reduce costs. For example, insurers can use the model to identify customers who are at a higher risk of filing a claim. This information can then be used to develop risk-based pricing strategies that can help to reduce the cost of insurance for low-risk customers.

Overall, the development of a travel insurance prediction model has the potential to benefit both travelers and businesses alike. By providing travelers with peace of mind and helping businesses to increase revenue and reduce costs, the model can have a positive impact on the travel industry as a whole.

# Milestone 2: Data Collection and Preparation

## Activity 1:- Collect the Dataset

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

Link: https://www.kaggle.com/datasets/tejashvi14/travel-insurance-prediction-data

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualisation techniques and some analysing techniques.

## Activity 1.1:- Importing the libraries

Importing the Libraries

```
In [1]:
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

from sklearn.metrics import confusion_matrix,classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

import pickle
import shap
```

## Activity 1.2: Read the Dataset

```
In [2]:
df = pd.read_csv("TravelInsurancePrediction.csv")
```

```
In [3]:
df
```

Out[3]:

| | Unnamed: 0 | Age | EmploymentType | GraduateOrNot | AnnualIncome | FamilyMembers | ChronicDiseases | FrequentFlyer | EverTravelledAbroad | TravelInsurance |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 31 | Government Sector | Yes | 400000 | 6 | 1 | No | No | 0 |
| **1** | 1 | 31 | Private Sector/Self Employed | Yes | 1250000 | 7 | 0 | No | No | 0 |
| **2** | 2 | 34 | Private Sector/Self Employed | Yes | 500000 | 4 | 1 | No | No | 1 |
| **3** | 3 | 28 | Private Sector/Self Employed | Yes | 700000 | 3 | 1 | No | No | 0 |
| **4** | 4 | 28 | Private Sector/Self Employed | Yes | 700000 | 8 | 1 | Yes | No | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1982** | 1982 | 33 | Private Sector/Self Employed | Yes | 1500000 | 4 | 0 | Yes | Yes | 1 |
| **1983** | 1983 | 28 | Private Sector/Self Employed | Yes | 1750000 | 5 | 1 | No | Yes | 0 |
| **1984** | 1984 | 28 | Private Sector/Self | Yes | 1150000 | 6 | 1 | No | No | 0 |

## Activity 2: Data preparation

Now we pre-process the Data.

The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

- Handling missing values
- Handling categorical data
- Handling Outliers

## Activity 2.1: Handling the missing values

In [4]:
```
df.shape
```

Out[4]: (1987, 10)

In [5]:
```
df.describe()
```

Out[5]:

| | Unnamed: 0 | Age | AnnualIncome | FamilyMembers | ChronicDiseases | TravelInsuran |
|---|---|---|---|---|---|---|
| **count** | 1987.000000 | 1987.000000 | 1.987000e+03 | 1987.000000 | 1987.000000 | 1987.0000 |
| **mean** | 993.000000 | 29.650226 | 9.327630e+05 | 4.752894 | 0.277806 | 0.3573 |
| **std** | 573.741812 | 2.913308 | 3.768557e+05 | 1.609650 | 0.448030 | 0.4793 |
| **min** | 0.000000 | 25.000000 | 3.000000e+05 | 2.000000 | 0.000000 | 0.0000 |
| **25%** | 496.500000 | 28.000000 | 6.000000e+05 | 4.000000 | 0.000000 | 0.0000 |
| **50%** | 993.000000 | 29.000000 | 9.000000e+05 | 5.000000 | 0.000000 | 0.0000 |
| **75%** | 1489.500000 | 32.000000 | 1.250000e+06 | 6.000000 | 1.000000 | 1.0000 |
| **max** | 1986.000000 | 35.000000 | 1.800000e+06 | 9.000000 | 1.000000 | 1.0000 |

In [6]:
```
df.corr()
```

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

Out[6]:

| | Unnamed: 0 | Age | AnnualIncome | FamilyMembers | ChronicDiseases | TravelIı |
|---|---|---|---|---|---|---|
| **Unnamed: 0** | 1.000000 | -0.004917 | -0.025031 | -0.041506 | -0.006858 | |
| **Age** | -0.004917 | 1.000000 | -0.020101 | 0.027409 | 0.007359 | |
| **AnnualIncome** | -0.025031 | -0.020101 | 1.000000 | -0.015367 | -0.001149 | |
| **FamilyMembers** | -0.041506 | 0.027409 | -0.015367 | 1.000000 | 0.028209 | |
| **ChronicDiseases** | -0.006858 | 0.007359 | -0.001149 | 0.028209 | 1.000000 | |
| **TravelInsurance** | 0.006196 | 0.061060 | 0.396763 | 0.079909 | 0.018190 | |

In [7]:
```
df.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   EmploymentType       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
```

Checking for null values

In [8]:
```
df.isnull().any()
```

Out[8]:
```
Unnamed: 0             False
Age                    False
EmploymentType         False
GraduateOrNot          False
AnnualIncome           False
FamilyMembers          False
ChronicDiseases        False
FrequentFlyer          False
EverTravelledAbroad    False
TravelInsurance        False
dtype: bool
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
Unnamed: 0             0
Age                    0
EmploymentType         0
GraduateOrNot          0
AnnualIncome           0
FamilyMembers          0
ChronicDiseases        0
FrequentFlyer          0
EverTravelledAbroad    0
TravelInsurance        0
dtype: int64
```

## Activity 2.2: Handling Categorical Values

As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding.

To convert the categorical features into numerical features we use encoding techniques. There are several techniques but in our project we are using manual encoding with the help of list comprehension.

• In our project, categorical features are Graduate or not , Frequent Flyer and ever travelled abroad, etc. With list comprehension encoding is done.

In [10]:

```
#Label Encoding
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df.GraduateOrNot = le.fit_transform(df.GraduateOrNot)
df.FrequentFlyer = le.fit_transform(df.FrequentFlyer)
df.EverTravelledAbroad = le.fit_transform(df.EverTravelledAbroad)
```

# Milestone 3: Exploratory Data Analysis

## Activity 1: Descriptive Statistical

In [17]:
```
df.describe()
```

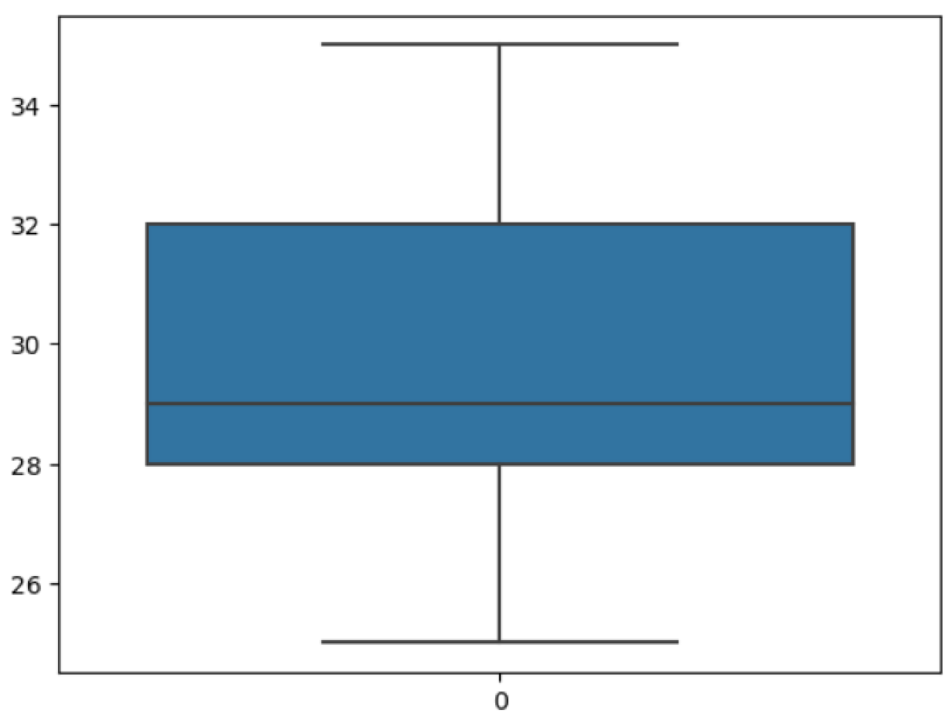
Out[17]:

| | Unnamed: 0 | Age | GraduateOrNot | AnnualIncome | FamilyMembers | ChronicDiseases | FrequentFlyer | EverTravelledAbroad | TravelInsurance |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 1987.000000 | 1987.000000 | 1987.000000 | 1.987000e+03 | 1987.000000 | 1987.000000 | 1987.000000 | 1987.000000 | 1987.000000 |
| **mean** | 993.000000 | 29.650226 | 0.851535 | 9.327630e+05 | 4.752894 | 0.277806 | 0.209864 | 0.191243 | 0.357323 |
| **std** | 573.741812 | 2.913308 | 0.355650 | 3.768557e+05 | 1.609650 | 0.448030 | 0.407314 | 0.393379 | 0.479332 |
| **min** | 0.000000 | 25.000000 | 0.000000 | 3.000000e+05 | 2.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 496.500000 | 28.000000 | 1.000000 | 6.000000e+05 | 4.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 993.000000 | 29.000000 | 1.000000 | 9.000000e+05 | 5.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 1489.500000 | 32.000000 | 1.000000 | 1.250000e+06 | 6.000000 | 1.000000 | 0.000000 | 0.000000 | 1.000000 |
| **max** | 1986.000000 | 35.000000 | 1.000000 | 1.800000e+06 | 9.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |

## Activity 2: Visual Analysis

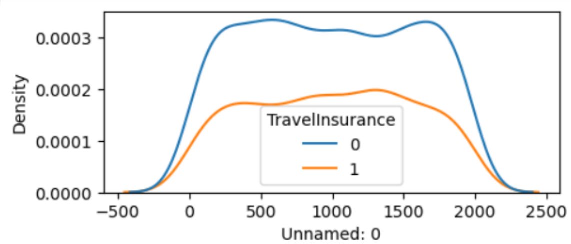
In [13]:
```
sns.boxplot(df["Age"])
```

Out[13]: <Axes: >



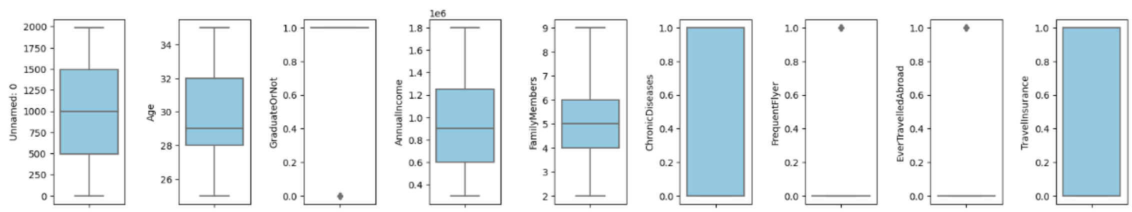
In [19]:

```python
for feature in nums:
    fig, ax = plt.subplots(figsize = (5,2))
    sns.kdeplot(data = df , hue = 'TravelInsurance' , x = feature, ax = ax)
    plt.show()
```



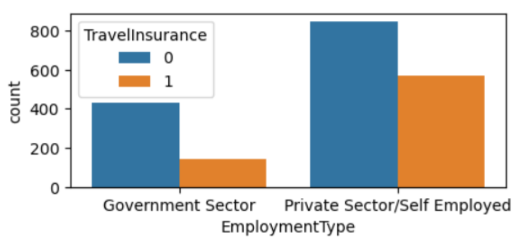

In [20]:

```python
plt.figure(figsize = (20,12))
for i in range(0, len(nums)):
    plt.subplot(4,11,i+2)
    sns.boxplot(y = df[nums[i]] , color = 'skyblue' , orient = 'v')
plt.tight_layout()
```




In [21]:

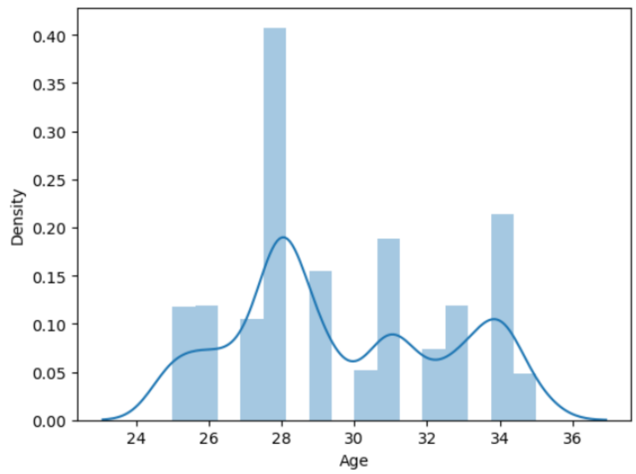
```python
target = 'TravelInsurance'
for feature in cats:
    fig, ax = plt.subplots(figsize = (5,2))
    sns.countplot(x = feature , hue = target , data = df, ax = ax)
    plt.show()
```



In [23]:
```python
sns.distplot(df["Age"])
```

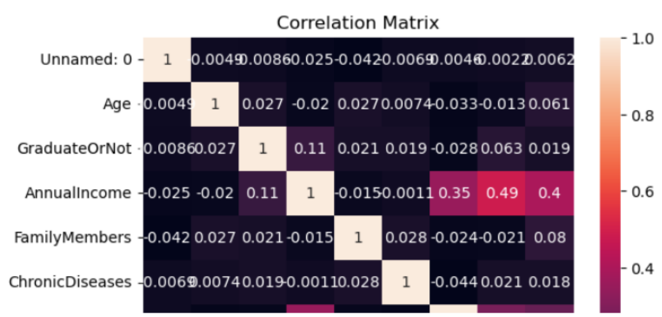
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

Out[23]: `<Axes: xlabel='Age', ylabel='Density'>`




In [25]:
```python
sns.heatmap(df.corr() , annot = True)
plt.title('Correlation Matrix')
plt.show()
```

```
The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only
valid columns or specify the value of numeric_only to silence this warning.
```




## Splitting data into train and test

In [27]:
```python
x = df.drop(['TravelInsurance'] , axis = 1)
y = df['TravelInsurance']
```

In [29]:
```python
x_train , x_test , y_train , y_test = train_test_split(x , y , test_size = 0.3 , random_state = 42)
```

# Milestone 4: Model Building

## Activity 1: Training the model in multiple algorithms

## Activity 1.1: Decision Tree model

In [40]:
```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(x_train , y_train)
y_pred = dtc.predict(x_test)
eval_classification(dtc)
```

```
Accuracy: 0.7085427135678392
Precision: 0.6871232541043861
Recall: 0.6834594594594594
F1-score: 0.6850654862963861
ROC AUC: 0.6834594594594595
```

## Activity 1.2: Random Forest Model

Random Forest Classifier

In [41]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train , y_train)
y_pred = rfc.predict(x_test)
eval_classification(rfc)
```

```
Accuracy: 0.8123953098827471
Precision: 0.8195305018870049
Recall: 0.7725585585585586
F1-score: 0.7856153490996767
ROC AUC: 0.7725585585585586
```

## Activity 1.3: KNN model

In [42]:
```
from sklearn.neighbors import KNeighborsClassifier
knc = KNeighborsClassifier()
knc.fit(x_train , y_train)
y_pred = knc.predict(x_test)
eval_classification(knc)
```

```
Accuracy: 0.7487437185929648
Precision: 0.748010509370349
Recall: 0.6961621621621621
F1-score: 0.7053966206969154
ROC AUC: 0.696162162162162
```

## Activity 1.4: Gradient Boosting model

In [43]:

```
from sklearn.ensemble import GradientBoostingClassifier
gbc = GradientBoostingClassifier()
gbc.fit(x_train , y_train)
y_pred = gbc.predict(x_test)
eval_classification(gbc)
```

```
Accuracy: 0.8274706867671692
Precision: 0.8625003926989413
Recall: 0.7772072072072072
F1-score: 0.7955159903296498
ROC AUC: 0.7772072072072073
```

## Activity 1.5: Naïve Bayes Model

In [44]:

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train , y_train)
y_pred = gnb.predict(x_test)
eval_classification(gnb)
```

```
Accuracy: 0.7621440536013401
Precision: 0.7703188496405127
Recall: 0.7077477477477477
F1-score: 0.7185308648533787
ROC AUC: 0.7077477477477478
```

## Activity 2: Testing the model

In [45]:

```
print(rfc.predict([[40,1,0,200000,5,1,0,1,1]]))
```

```
[0]
```

## Milestone 5: Performance Testing :- Testing with multiple evaluation metrics and comparing them

In [39]:

```python
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

def eval_classification(dtc):
    # Predict labels for the test data if not already done
    y_pred = dtc.predict(x_test)

    # Calculate evaluation metrics
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='macro')
    recall = recall_score(y_test, y_pred, average='macro')
    f1 = f1_score(y_test, y_pred, average='macro')

    # Calculate ROC AUC
    roc_auc = roc_auc_score(y_test, y_pred)

    # Display evaluation results
    print("Accuracy:", accuracy)
    print("Precision:", precision)
    print("Recall:", recall)
    print("F1-score:", f1)
    print("ROC AUC:", roc_auc)
```

```python
eval_classification(dtc)
```

```
Accuracy: 0.7085427135678392
Precision: 0.6871232541043861
Recall: 0.6834594594594594
F1-score: 0.6850654862963861
ROC AUC: 0.6834594594594595
```

In [41]:

```python
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train , y_train)
y_pred = rfc.predict(x_test)
eval_classification(rfc)
```

```
Accuracy: 0.8123953098827471
Precision: 0.8195305018870049
Recall: 0.7725585585585586
F1-score: 0.7856153490996767
ROC AUC: 0.7725585585585586
```

```python
eval_classification(knc)
```

```
Accuracy: 0.7487437185929648
Precision: 0.748010509370349
Recall: 0.6961621621621621
F1-score: 0.7053966206969154
ROC AUC: 0.6961621621621622
```

```python
eval_classification(gbc)
```

```
Accuracy: 0.8274706867671692
Precision: 0.8625003926989413
Recall: 0.7772072072072072
F1-score: 0.7955159903296498
ROC AUC: 0.7772072072072073
```

```
y_pred = gnb.predict(x_test)
eval_classification(gnb)
```

```
Accuracy: 0.7621440536013401
Precision: 0.7703188496405127
Recall: 0.7077477477477477
F1-score: 0.7185308648533787
ROC AUC: 0.7077477477477478
```

# Milestone 6: Model Deployment

## Activity 1: Save the best model

```
In [46]:
#dump selected model
pickle.dump(gbc,open('Travel.pkl','wb'))
```

## Activity 2: Integrate with Web Framework

| | |
|---|---|
| .ipynb_checkpoints | 11/8/2023 12:36 AM |
| static | 11/8/2023 8:17 PM |
| css | 11/8/2023 8:17 PM |
| images | 11/8/2023 9:13 PM |
| js | 11/8/2023 8:17 PM |
| templates | 11/9/2023 2:14 PM |
| index.html | 11/9/2023 2:14 PM |
| uploads | 11/8/2023 8:17 PM |
| app.py | 11/9/2023 5:15 PM |
| Travel Insurance prediction.ipynb | 11/9/2023 5:16 PM |
| Travel.pkl | 11/9/2023 2:19 PM |
| TravelInsurancePrediction.csv | 11/8/2023 4:38 PM |

app.py

```
from flask import Flask,render_template,request
import pickle
#import pandas as pd
#import numpy as np
```

```
# loading my mlr model
model=pickle.load(open('Travel.pkl','rb'))

# Flask is used for creating your application
# render template is use for rendering the html page

app= Flask(__name__)  # your application


@app.route('/')  # default route
def home():
    return render_template('index.html')
```

```python
@app.route('/predict',methods=['GET','POST']) # prediction route
def predict():
    Age = request.form['Age']
    EmploymentType = request.form['EmploymentType']
    if EmploymentType == 'Private Sector/Self Employed':
        EmploymentType = 1
    if EmploymentType == 'Government Sector':
        EmploymentType = 0

    AnnualIncome = request.form['AnnualIncome']

    FamilyMembers = request.form['FamilyMembers']
    ChronicDiseases = request.form['ChronicDiseases']
    if ChronicDiseases == 'Yes':
        ChronicDiseases = 1
    if ChronicDiseases == 'No':
        ChronicDiseases = 0
    FrequentFlyer = request.form['FrequentFlyer']
    if FrequentFlyer == 'Yes':
        FrequentFlyer = 1
    if FrequentFlyer == 'No':
        FrequentFlyer = 0

    EverTravelledAbroad = request.form['EverTravelledAbroad']
    if EverTravelledAbroad == 'Yes':
        EverTravelledAbroad = 1
    if EverTravelledAbroad == 'No':
        EverTravelledAbroad = 0

    total = [[int(Age), int(EmploymentType), float(AnnualIncome), FamilyMembers , int(ChronicDiseases), int(Frequ
    prediction = model.predict(total)
    if prediction == 1:
        prediction = 'Yes'
    if prediction == 0:
        prediction = 'No'

    return render_template("index.html", result = "Would the Travel insurance be favourable to the customer? " +
```

```python
# running your application
if __name__ == "__main__":
    app.run(port = 8000)

#http://localhost:5000/ or localhost:5000
```

Index.html

```html
<style>
body {
  background-image: url('../static/images/air.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
</style>
```

```html
<html>
<form action="/predict" method="POST">

<br>
<br>
<label >Travel Insurance Prediction</label>
<br>
<br>
<br>
Age
<br>
<input type="text" name="Age"></input>
<br>
<br>
<label >Employment Type</label>
<br>
<select name="EmploymentType">
  <option value="Private Sector/Self Employed">Private Sector/Self Employed</option>
  <option value="Government Sector">Government Sector</option>
</select>
<br>
<br>
Annual Income
<br>
<input type="text" name="AnnualIncome"></input>
<br>
<br>
Family Members
<br>
<input type="text" name="FamilyMembers"></input>
<br>
```

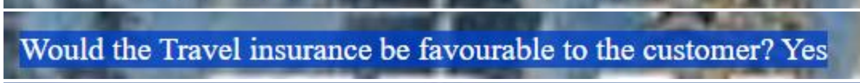
```
<br>
Annual Income
<br>
<input type="text" name="AnnualIncome"></input>
<br>
<br>
Family Members
<br>
<input type="text" name="FamilyMembers"></input>
<br>
<br>
<label >Chronic Diseases</label>
<br>
<select name="ChronicDiseases">
  <option value="Yes">YES</option>
  <option value="No">NO</option>
</select>
<br>
<br>
<label >Frequent Flyer</label>
<br>
<select name="FrequentFlyer">
  <option value="Yes">YES</option>
  <option value="No">NO</option>
</select>
<br>
<br>
<label >Ever Travelled Abroad</label>
<br>
<select name="EverTravelledAbroad">
  <option value="Yes">YES</option>
  <option value="No">NO</option>
</select>
<br>
<br>
<input type="submit"></input>

</form>

{{result}}
```

The output

```
In [1]: runfile('C:/Education/AIML EXTERNSHIP/Travel Insurance Prediction/app.py',
wdir='C:/Education/AIML EXTERNSHIP/Travel Insurance Prediction')
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Running on http://127.0.0.1:8000
Press CTRL+C to quit
```


Travel Insurance Prediction
Age
Employment Type
Private Sector/Self Employed
Annual Income
Family Members
Chronic Diseases
YES
Frequent Flyer
YES
Ever Travelled Abroad
YES
Submit
TRAVEL INSURANCE
27°C
Mostly clear
ENG
US
5:57 PM
11/9/2023


Would the Travel insurance be favourable to the customer? Yes


Would the Travel insurance be favourable to the customer? Yes


Travel Insurance Prediction
Age
Employment Type
Private Sector/Self Employed
Annual Income
Family Members
Chronic Diseases
YES
Frequent Flyer
YES
Ever Travelled Abroad
YES
Submit
Would the Travel insurance be favourable to the customer? Yes
TRAVEL INSURANCE
27°C
Mostly clear
ENG
US
5:58 PM
11/9/2023