

# Travel Insurance Prediction

TEAM ID: 591-581

Done By:

KAMPA VENKATA HANUMA TEJA

HARSHA SRI VIGNESH TADIGIRI

MEDIKONDA PREMCHAND

MAGULURI VENKATA SIVA RAMA KRISHNA

# **1. INTRODUCTION**

## **1.1 Project Overview:**

### **Objective:**

The goal of this project is to predict whether a person will buy travel insurance based on different factors like age, income, family size, and travel history.

### **How It Works:**

**Collect Data:** Gather information about people who have or haven't bought travel insurance in the past. This includes details like age, income, family size, and more

**Build a Smart System:** Use a smart computer program (machine learning) to analyse the collected information. This program learns from the data to predict if someone is likely to buy travel insurance.

**Help Travelers Make Decisions:** The system will be like a helpful guide. If someone is planning a trip, they can input their details, and the system will give them an idea of whether getting travel insurance is a good idea for them.

### **Key Components:**

**Dashboard:** A space where users can input their details and get predictions.

**Sprints:** The project is divided into short periods (sprints) to make sure things are done step by step.

### **Why It Matters:**

Travel insurance is important, and this project makes it easier for people to decide if it's right for them. It uses smart technology to understand patterns from the past and provide useful advice for the future.

## **1.2 Purpose:**

### **1. Informed Decision-Making:**

The primary purpose is to empower individuals with information to make informed decisions about whether to purchase travel insurance for their trips.

## 2. Personalized Insights:

By analysing various factors such as age, income, family size, and travel history, the project aims to provide personalized insights. This helps users understand if travel insurance aligns with their specific circumstances and needs.

## 3. Risk Mitigation:

The project serves as a tool for risk mitigation. By predicting the likelihood of an individual purchasing travel insurance, it contributes to better preparation for unforeseen events during travel.

## 4. User-Friendly Guidance:

Through a user-friendly interface, the project offers a straightforward way for users to input their details and receive clear guidance. This simplifies the decision-making process for users who might be unsure about the necessity of travel insurance.

## 5. Data-Driven Approach:

Leveraging machine learning and data analysis, the project adopts a data-driven approach to understand patterns in historical travel insurance purchases. This ensures a reliable basis for predictions and recommendations.

## 6. Accessibility and Convenience:

By allowing user registration through various means like email, Facebook, and Gmail, the project aims to be accessible and convenient for a diverse range of users, making it easy for them to engage with the system.

## 7. Incremental Progress (Sprints):

The project is organized into sprints, short periods of focused work, to ensure step-by-step progress. This approach helps in delivering a functional and effective system in a manageable and iterative manner.

## 8. Promoting Financial Security:

Ultimately, the purpose is to contribute to individuals' financial security during travel by providing insights that enable them to make well-informed decisions regarding the purchase of travel insurance.

## 9. Enhancing User Confidence:

Through accurate predictions and guidance, the project aims to enhance user confidence in the decision-making process, promoting a sense of security and preparedness for travel-related scenarios.

In summary, the purpose is to create a user-friendly, data-driven system that assists individuals in making informed decisions about travel insurance, thereby enhancing their travel experience and financial security.

## 2. Literature Survey

### 2.1 Existing Problems:

#### 1. Decision Uncertainty:

Problem: Travelers often face uncertainty about whether to invest in travel insurance.

Solution: Your project can provide data-driven insights, helping users make informed decisions based on their individual circumstances.

#### 2. Financial Risk Mitigation:

Problem: Travelers may be concerned about unexpected expenses due to medical emergencies or trip cancellations.

Solution: The project can assist users in understanding the financial risks associated with travel and guide them in mitigating these risks through appropriate insurance coverage.

#### 3. Tailored Recommendations:

Problem: Generic advice may not suit every traveller's needs.

Solution: By analysing individual factors like age, income, and travel history, your project can tailor recommendations, ensuring relevance to each user.

#### 4. Increasing Travel Safety:

Problem: Travelers might not be aware of potential risks associated with their destinations.

Solution: The project can consider travel history and destinations, offering insights into potential risks and encouraging users to prioritize safety measures.

#### 5. Efficient Resource Utilization:

Problem: Limited resources may lead to uncertainty about where to allocate travel-related funds.

Solution: By understanding users' financial capacities and priorities, your project can aid in efficient resource allocation, promoting a well-balanced travel plan.

#### 6. User Education:

Problem: Users might lack awareness of the importance and types of travel insurance.

Solution: The project can serve as an educational tool, explaining the benefits of travel insurance and guiding users through the decision-making process.

#### 7. Data-Driven Travel Planning:

Problem: Without data-driven insights, travellers may struggle to plan effectively.

Solution: Your project can leverage historical data to identify patterns, trends, and potential pitfalls, enhancing the overall effectiveness of travel planning.

#### 8. Enhanced Preparedness:

Problem: Travelers may not be adequately prepared for unforeseen events during their trips.

Solution: By predicting the likelihood of events and recommending appropriate insurance coverage, your project can contribute to enhanced travel preparedness.

#### 9. User Empowerment:

Problem: Users may feel overwhelmed by the complexity of travel insurance decisions.

Solution: Your project aims to empower users by simplifying the decision-making process, providing clarity, and building confidence in their travel choices.

## 10. Adapting to Changing Circumstances:

Problem: Circumstances, such as health conditions or financial situations, can change.

Solution: The project can adapt to changing user circumstances, providing continuous support and recommendations based on the latest data.

## 2.2 References:

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

<https://www.researchgate.net/>

<https://www.google.co.in/>

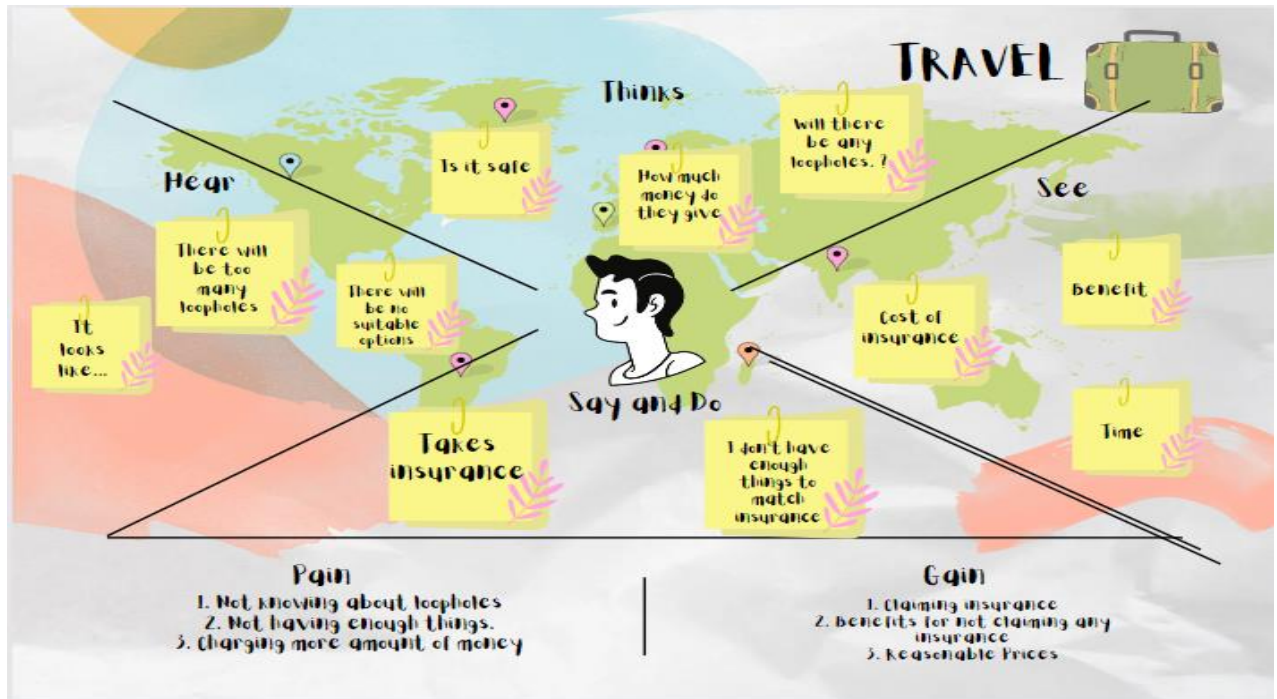
<https://chat.openai.com/>

## 2.3 Problem Statement Definition:

Travel insurance is very important, it's like a safety net for unexpected events during your journey, covering things like medical emergencies, lost luggage, or flight cancellations. As global travel increases, so does the need for affordable and thorough coverage. To make the process smarter, we're using a predictive model that analyses data on past travel insurance purchases, customer info, and travel plans. This model aims to predict if someone will buy travel insurance by looking at factors like age, income, and family size. The goal is to create a tool that accurately forecasts who's likely to get travel insurance based on these details.

### 3. Ideation and Proposed Solution

#### 3.1 Empathy map Canvas:



#### 3.2 Brainstorming:



## **4. Requirement Analysis**

### **4.1 Functional Requirement:**

#### **1. User Dashboard:**

Description: A dashboard where users can input their details and receive predictions.

Acceptance Criteria: Users can input relevant information, and the system provides accurate predictions.

#### **2. Prediction Model:**

Description: The system should include a machine learning model for predicting the likelihood of a user purchasing travel insurance.

Acceptance Criteria: The model accurately predicts user behaviour based on historical data.

#### **3. Sprint-Based Development:**

Description: The project should be developed in sprints, with each sprint delivering specific functionalities.

Acceptance Criteria: Functionalities are incrementally developed and released as per the sprint schedule.

#### **4. User Interface (UI):**

Description: A user-friendly interface that simplifies the process of inputting data and receiving predictions.

Acceptance Criteria: Users find the interface intuitive and easy to navigate.



## 4.2 Non-Functional Requirement:

### 1. Scalability:

Description: The system should be scalable to handle an increasing number of users and data over time.

Acceptance Criteria: The system performance remains stable as the user base grows.

### 2. Model Interpretability:

Description: The machine learning model should be interpretable to provide transparency in predictions.

Acceptance Criteria: The model's decision-making process can be explained and understood.

### 3. Performance Optimization:

Description: Optimize the system's performance to provide quick and responsive predictions.

Acceptance Criteria: Users experience minimal delays when interacting with the system.

### 4. Reliability:

Description: The system should be reliable, providing accurate predictions consistently.

Acceptance Criteria: The model's accuracy remains high over time, and the system is dependable for users.

### 5. Usability:

Description: The system should be user-friendly, catering to users with varying levels of technical expertise.

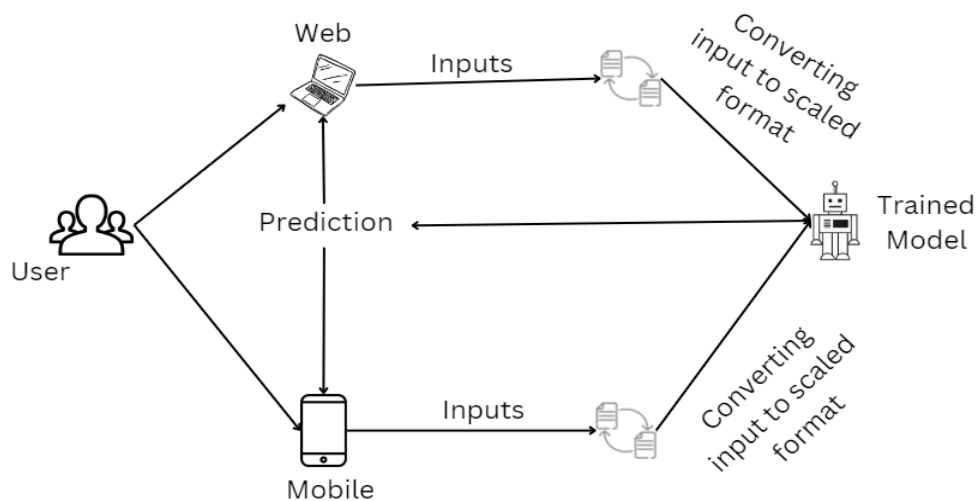
Acceptance Criteria: Users find the system easy to use, with clear instructions and guidance.

## 5. Project Design

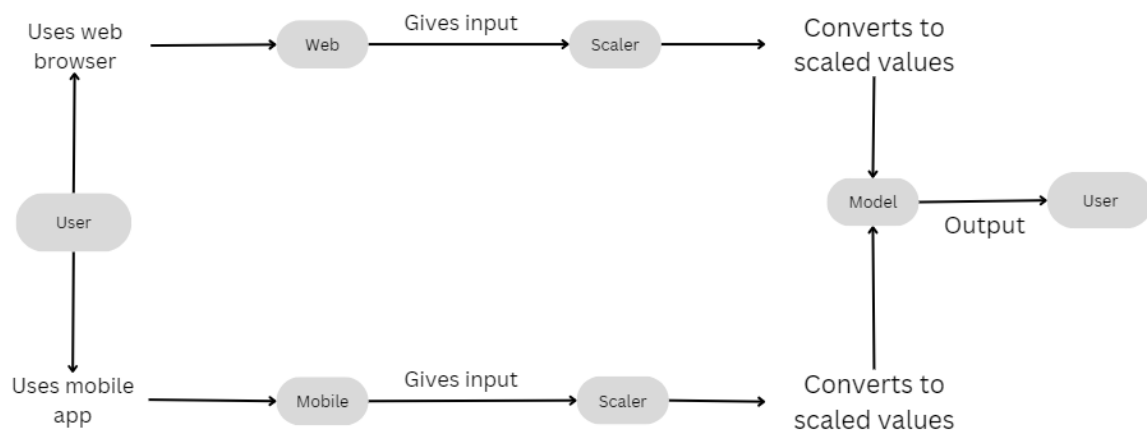
### 5.1 Data Flow Diagrams:

In our project the information/input will be given directly to the model through user interface (UI) of an application(mobile/web), since the model will already be trained by a dataset it will preprocess the data and gives required output by using machine learning algorithms as per the training.

Example: [\(Simplified\)](#)



Data flow diagram:

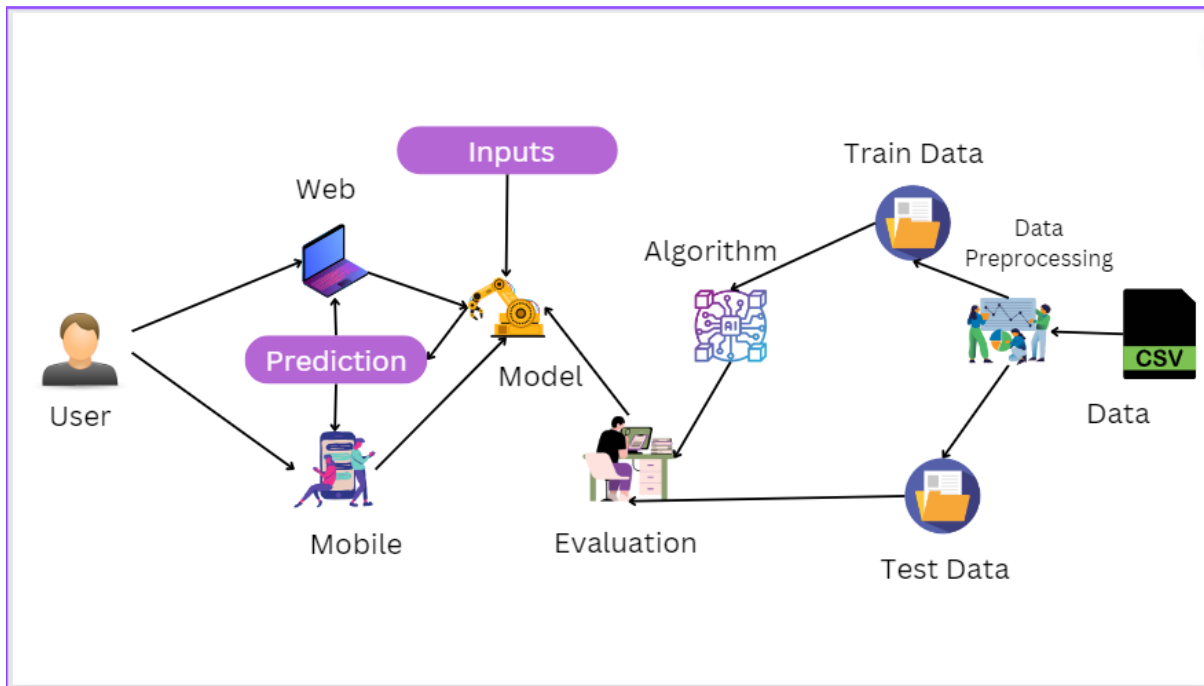


## User Stories

Use the below template to list all the user stories for the product.

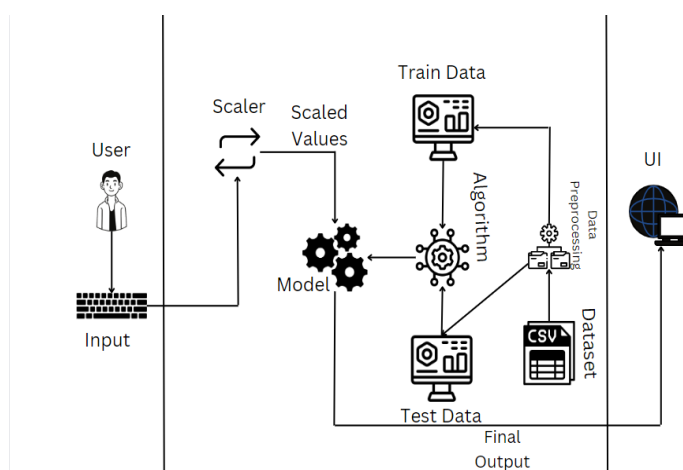
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)/ (Web User)	Age	USN-1	As a user, I can check whether a person will take travel insurance or not based on his/her age	I can get required information	High	Sprint-1
	Employment Type	USN-2	As a user, I can check whether a person will take travel insurance or not based on his/her Employment Type	I can get required information	High	Sprint-1
	Graduate or not	USN-3	As a user, I can check whether a person will take travel insurance or not by knowing if he/she is a graduate or not	I can get required information	High	Sprint-1
	Annual Income	USN-4	As a user, I can check whether a person will take travel insurance or not based on his/her annual income	I can get required information	High	Sprint-1
	Family Members	USN-5	As a user, I can check whether a person will take travel insurance or not based on his/her Family members	I can get required information	High	Sprint-1
	Chronic Diseases	USN-6	As a user, I can check whether a person will take travel insurance or not based on his/her Chronic Diseases.	I can get required information	Low	Sprint-2
	Frequent Flyer	USN-7	As a user, I can check whether a person will take travel insurance or not by seeing his/her travel enthusiasm.	I can get required information	Low	Sprint-2
	Ever travelled Abroad	USN-8	As a user, I can check whether a person will take travel insurance or not by knowing whether he/she ever travelled abroad or not.	I can get required information	High	Sprint-1
	Application	USN-9	As a user, I can check whether a person will take travel insurance or not by using a simple website or an application	I can use an app/web browser.	High	Sprint-1

## 5.2 Solution Architecture:



## 6. Project Planning and Scheduling

### 6.1 Technical Architecture:



## 6.2 Sprint Planning and Estimation:

### **Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

Use the below template to create product backlog and sprint schedule

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Members
Customer (Mobile user)/ (Web User)	Age	USN-1	As a user, I can check whether a person will take travel insurance or not based on his/her age	I can get required information	High	4
	Employment Type	USN-2	As a user, I can check whether a person will take travel insurance or not based on his/her Employment Type	I can get required information	High	4
	Graduate or not	USN-3	As a user, I can check whether a person will take travel insurance or not by knowing if he/she is a graduate or not	I can get required information	High	4
	Annual Income	USN-4	As a user, I can check whether a person will take travel insurance or not based on his/her annual income	I can get required information	High	4
	Family Members	USN-5	As a user, I can check whether a person will take travel insurance or not based on his/her Family members	I can get required information	High	4
	Chronic Diseases	USN-6	As a user, I can check whether a person will take travel insurance or not based on his/her Chronic Diseases.	I can get required information	Low	4
	Frequent Flyer	USN-7	As a user, I can check whether a person will take travel insurance or not by seeing his/her travel enthusiasm.	I can get required information	Low	4
	Ever travelled Abroad	USN-8	As a user, I can check whether a person will take travel insurance or not by knowing whether he/she ever travelled abroad or not.	I can get required information	High	4
	Application/ Website	USN-9	As a user, I can check whether a person will take travel insurance or not by using a simple website or an application	I can use an app/web browser.	High	4

## 6.3 Sprint Delivery Schedule:

### Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	25	5 Days	25 Oct 2023	30 Oct 2023	25	28 Oct 2023
Sprint-2	25	5 Days	31 Oct 2023	04 Nov 2023	25	04 Nov 2023
Sprint-3	25	5 Days	05 Nov 2023	10 Nov 2023	25	09 Nov 2023
Sprint-4	25	5 Days	11 Nov 2023	15 Nov 2023	25	15 Nov 2023

## 7. Coding and Solutioning

### 7.1 Feature 1:

Taking Input: In this project we are predicting for the travel insurance will be taken or not by the user, since to find that we need to take some inputs from the user like Age, Employment Type, Graduate or Not, Annual Income, Family Members, etc... Since for the prediction the model needs those data from the user and finally the model will predict and displays it in webpage. So for to take input we used a form.

Code:

```
<div class="container">
  <form id="myForm" action="predict" method="post" enctype="multipart/form-data">
    <label for="Age">Age: </label>
    <input type="number" id="Age" name="Age" placeholder="Enter your Age" required>

    <label for="EmploymentType">Employment Type: </label>
    <select name="EmploymentType" id="EmploymentType" required>
      <option value="">Choose One</option>
      <option value="GovernmentSector">Government Sector</option>
      <option value="PrivateSector/SelfEmployed">Private Sector/Self Employed</option>
    </select>

    <label for="GraduateOrNot">Graduate Or Not: </label>
    <select name="GraduateOrNot" id="GraduateOrNot" required>
      <option value="">Choose One</option>
      <option value="Yes">Yes</option>
      <option value="No">No</option>
    </select>

    <label for="AnnualIncome">AnnualIncome: </label>
    <input type="number" id="Number" name="AnnualIncome" placeholder="Enter your Annual Income" maxlength="10" required>

    <label for="FamilyMembers">FamilyMembers: </label>
    <input type="number" id="family" name="FamilyMembers" required>

    <label for="ChronicDiseases">ChronicDisease: </label>
    <select name="ChronicDiseases" id="ChronicDiseases" required>
      <option value="">Choose One</option>
      <option value="Yes">Yes</option>
      <option value="No">No</option>
    </select>
  </form>
</div>
```

## 7.2 Feature 2:

Dropdown feature for Yes/No: To make the input form very easy we introduced a dropdown feature for Yes/No and for Employment Type.

Code:

```
<label for="EmploymentType">Employment Type: </label>
<select name="EmploymentType" id="EmploymentType" required>
  <option value="">Choose One</option>
  <option value="GovernmentSector">Government Sector</option>
  <option value="PrivateSector/SelfEmployed">Private Sector/Self Employed</option>
</select>

<label for="GraduateOrNot">Graduate Or Not: </label>
<select name="GraduateOrNot" id="GraduateOrNot" required>
  <option value="">Choose One</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>

<label for="ChronicDiseases">ChronicDisease: </label>
<select name="ChronicDiseases" id="ChronicDiseases" required>
  <option value="">Choose One</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>

<label for="FrequentFlyer">Frequent Flyer: </label>
<select name="frequentFlyer" id="FrequentFlyer" required>
  <option value="">Choose One</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>

<label for="EverTravelledAbroad">Ever Travelled Abroad: </label>
<select name="EverTravelledAbroad" id="EverTravelledAbroad" required>
  <option value="">Choose One</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>
```

### 7.3 Feature 3:

Avoid Type Errors: So, when inputs are given by default those will be in string format which is not perfect for to predict since the model only needs the Integer input. So, for simple conversion we used the type converter at the time of taking the input.

Code:

```
Age = int(request.form['Age'])

EmployementType = request.form['EmployementType']
if EmployementType == 'PrivateSector/SelfEmployed':
    EmployementType = 1
else:
    EmployementType = 0

GraduateOrNot = request.form['GraduateOrNot']
if GraduateOrNot == 'Yes':
    GraduateOrNot = 1
else :
    GraduateOrNot = 0

AnnualIncome = float(request.form['AnnualIncome'])

FamilyMembers = int(request.form['FamilyMembers'])

chronicDiseases = request.form['ChronicDiseases']
if chronicDiseases == 'Yes':
    chronicDiseases = 1
else:
    chronicDiseases = 0

FrequentFlyer = request.form['frequentFlyer']
if FrequentFlyer == 'Yes':
    FrequentFlyer = 1
else:
    FrequentFlyer = 0
```



## 8. Performance Testing

### 8.1 Performance Metrics

Decision Tree:

```
#Accuracy score
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print("Accuracy Score = ", accuracy_score(y_test, d_pred))
print("Confusion Matrix = ", confusion_matrix(y_test, d_pred))
print("Classification Report = ")
print( classification_report(y_test, d_pred))
```

Accuracy Score = 0.7738693467336684  
Confusion Matrix = [[328 47]  
[ 88 134]]  
Classification Report =

	precision	recall	f1-score	support
0	0.79	0.87	0.83	375
1	0.74	0.60	0.67	222
accuracy			0.77	597
macro avg	0.76	0.74	0.75	597
weighted avg	0.77	0.77	0.77	597

Logestic Regression:

```
#Accuracy score
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print("Accuracy Score = ", accuracy_score(y_test, l_pred))
print("Confusion Matrix = ", confusion_matrix(y_test, l_pred))
print("Classification Report = ")
print( classification_report(y_test, l_pred))
```

Accuracy Score = 0.7487437185929648  
Confusion Matrix = [[344 31]  
[119 103]]  
Classification Report =

	precision	recall	f1-score	support
0	0.74	0.92	0.82	375
1	0.77	0.46	0.58	222
accuracy			0.75	597
macro avg	0.76	0.69	0.70	597
weighted avg	0.75	0.75	0.73	597

## Random Forest:

```
#Accuracy score
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print("Accuracy Score = ", accuracy_score(y_test, r_pred))
print("Confusion Matrix = ", confusion_matrix(y_test, r_pred))
print("Classification Report = ")
print( classification_report(y_test, r_pred))
```

Accuracy Score = 0.7922948073701842  
Confusion Matrix = [[335 40]  
[ 84 138]]  
Classification Report =

	precision	recall	f1-score	support
0	0.80	0.89	0.84	375
1	0.78	0.62	0.69	222
accuracy			0.79	597
macro avg	0.79	0.76	0.77	597
weighted avg	0.79	0.79	0.79	597

## KNN Classifier:

```
#Accuracy score
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print("Accuracy Score = ", accuracy_score(y_test, k_pred))
print("Confusion Matrix = ", confusion_matrix(y_test, k_pred))
print("Classification Report = ")
print( classification_report(y_test, k_pred))
```

Accuracy Score = 0.7621440536013401  
Confusion Matrix = [[335 40]  
[102 120]]  
Classification Report =

	precision	recall	f1-score	support
0	0.77	0.89	0.83	375
1	0.75	0.54	0.63	222
accuracy			0.76	597
macro avg	0.76	0.72	0.73	597
weighted avg	0.76	0.76	0.75	597

## Gradient Boosting:

```
#Accuracy score
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print("Accuracy Score = ", accuracy_score(y_test, g_pred))
print("Confusion Matrix = ", confusion_matrix(y_test, g_pred))
print("Classification Report = ")
print( classification_report(y_test, g_pred))
```

Accuracy Score = 0.830820770519263  
Confusion Matrix = [[367 8]  
[ 93 129]]  
Classification Report =

	precision	recall	f1-score	support
0	0.80	0.98	0.88	375
1	0.94	0.58	0.72	222
accuracy			0.83	597
macro avg	0.87	0.78	0.80	597
weighted avg	0.85	0.83	0.82	597

Naïve Bayes:

```
#Accuracy score
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print("Accuracy Score = ", accuracy_score(y_test, n_pred))
print("Confusion Matrix = ", confusion_matrix(y_test, n_pred))
print("Classification Report = ")
print(classification_report(y_test, n_pred))
```

Accuracy Score = 0.7286432160804021  
Confusion Matrix = [[324 51]  
[111 111]]  
Classification Report =

	precision	recall	f1-score	support
0	0.74	0.86	0.80	375
1	0.69	0.50	0.58	222
accuracy			0.73	597
macro avg	0.72	0.68	0.69	597
weighted avg	0.72	0.73	0.72	597

## 9. Results

### 9.1 Output Screenshots

```
#Decision Tree
eval_classification(dtc)
```

Accuracy (Test Set) : 0.77  
precision (Test Set) : 0.74  
Recall (Test Set) : 0.60  
F1-Score (Test Set) : 0.67

```
# Logistic Regression
eval_classification(lr)
```

Accuracy (Test Set) : 0.75  
precision (Test Set) : 0.77  
Recall (Test Set) : 0.46  
F1-Score (Test Set) : 0.58

```
# Random forest
eval_classification(rfc)
```

Accuracy (Test Set) : 0.79  
precision (Test Set) : 0.78  
Recall (Test Set) : 0.62  
F1-Score (Test Set) : 0.69

```
# Knn
eval_classification(knn)
```

Accuracy (Test Set) : 0.76  
precision (Test Set) : 0.75  
Recall (Test Set) : 0.54  
F1-Score (Test Set) : 0.63

```
#gradient Boost
eval_classification(gbm)
```

Accuracy (Test Set) : 0.83  
precision (Test Set) : 0.94  
Recall (Test Set) : 0.58  
F1-Score (Test Set) : 0.72

```
# Naive Bayes
eval_classification(nb)
```

Accuracy (Test Set) : 0.73  
precision (Test Set) : 0.69  
Recall (Test Set) : 0.50  
F1-Score (Test Set) : 0.58

From all the observation from the above results we can conclude that gradient boosting is having the best performance.

## **10.     Advantages and Disadvantages**

### **10.1   Advantages:**

#### **1. Informed Decision-Making:**

Advantage: Users can make well-informed decisions about whether to purchase travel insurance based on personalized insights and predictions.

#### **2. Personalized Recommendations:**

Advantage: The system tailors recommendations based on individual factors such as age, income, family size, and travel history, providing personalized and relevant guidance.

#### **3. Risk Mitigation:**

Advantage: The project assists users in identifying and mitigating potential risks associated with travel, contributing to better preparation for unforeseen events.

#### **4. Financial Security:**

Advantage: Users can better understand and manage financial risks during travel, promoting a sense of financial security and peace of mind.

#### **5. Transparency in Predictions:**

Advantage: The machine learning model's interpretability ensures transparency in predictions, fostering user trust in the decision-making process.

#### **6. Continuous Adaptation:**

Advantage: The project can adapt to changing user circumstances and preferences, providing continuous support and recommendations as situations evolve.

#### **7. Enhanced Travel Safety:**

Advantage: The project considers factors such as travel history and destinations, offering insights into potential risks and promoting a safety-conscious approach to travel.

## 8. User-Friendly Interface:

Advantage: The system features a user-friendly interface that simplifies the process of inputting data and receiving predictions, ensuring accessibility for a broad user base.

## 9. Scalability:

Advantage: The system is designed to scale efficiently, accommodating an increasing number of users and data over time without compromising performance.

## 10. Usability and User Satisfaction:

Advantage: Users find the system easy to use, contributing to high satisfaction levels and positive user experiences.

### 10.2 Disadvantages

#### 1. Data Dependence:

Disadvantage: The accuracy of predictions heavily relies on the quality and relevance of historical data. Incomplete or biased data may lead to inaccurate predictions.

#### 2. Algorithmic Biases:

Disadvantage: Machine learning models may inherit biases present in the training data. This could result in discriminatory or unfair predictions, especially if historical data is biased.

#### 3. Model Complexity:

Disadvantage: Highly complex models may lack interpretability, making it challenging to explain predictions to users. This lack of transparency could reduce user trust.

#### 4. User Resistance:

Disadvantage: Some users may be hesitant to trust machine-generated predictions, especially when it comes to personal decisions like purchasing insurance.

## 5. Changing User Behaviour:

Disadvantage: User behaviour can evolve over time, and the model might struggle to adapt quickly to changes in preferences, priorities, or travel patterns.

## 6. Privacy Concerns:

Disadvantage: Collecting and analysing personal data for predictions raises privacy concerns. Users may be apprehensive about sharing sensitive information.

## 7. Limited Scope:

Disadvantage: The system's predictions are limited to the factors considered in the model. It may not account for all variables that could influence travel insurance decisions.

## 8. Unforeseen Events:

Disadvantage: The model may not anticipate entirely novel or unprecedented events that could impact travel decisions, as it relies on historical data patterns.

## 9. Model Evaluation Challenges:

Disadvantage: Evaluating the performance of the model can be challenging, and selecting appropriate metrics for assessment may not capture all aspects of its effectiveness.

## 10. Resistance to Change:

Disadvantage: Users, especially those accustomed to traditional decision-making processes, may resist adopting a data-driven approach, hindering the project's acceptance.

## 11. Conclusion

In conclusion, the travel insurance prediction project holds significant potential to revolutionize the way individuals approach travel planning and risk management. By leveraging machine learning and data-driven insights, the project aims to empower users with personalized recommendations, fostering informed decision-making regarding the purchase of travel insurance.

The advantages of the project are apparent, ranging from tailored guidance and risk mitigation to financial security and user empowerment. However, it is crucial to acknowledge the potential disadvantages and challenges, such as data dependence, algorithmic biases, and user resistance. These challenges highlight the importance of ethical considerations, user education, and ongoing monitoring to ensure the responsible and effective implementation of the predictive model.

As the project progresses, continuous adaptation to changing user behaviours and external factors will be key to maintaining its relevance and reliability. Striking a balance between predictive accuracy, model interpretability, and user privacy is essential for building trust and acceptance among users.

In navigating the complexities of travel planning, the travel insurance prediction project aims to provide a valuable tool that goes beyond generic recommendations. It aspires to be a user-friendly, scalable, and adaptable solution, contributing to enhanced travel safety, efficient resource allocation, and overall user satisfaction.

As with any innovative project, addressing the identified challenges and remaining attuned to user feedback will be integral to its long-term success. Ultimately, the project represents a promising step toward a more data-informed, user-centric approach to travel insurance decisions, potentially reshaping the landscape of travel planning for a diverse range of users.

## **12. Future Scope**

The travel insurance prediction project has a promising future with various avenues for expansion and improvement. Here are some potential future scopes for the project:

### **1. Enhanced Predictive Models:**

Explore advanced machine learning techniques and algorithms to improve the accuracy and precision of predictive models. Incorporate deep learning or ensemble methods to capture complex relationships within the data.

### **2. Geographical and Seasonal Considerations:**

Expand the model to incorporate geographical and seasonal factors that may impact travel insurance decisions. Consider regional variations in travel risks and insurance preferences.

### **3. Integration with Travel Platforms:**

Collaborate with travel platforms, booking websites, or travel agencies to integrate the predictive model directly into their systems. This seamless integration can provide users with real-time recommendations during the travel booking process.

### **4. Multimodal Predictions:**

Extend the project to consider various modes of transportation and accommodation. Provide holistic recommendations that encompass not only flight-related risks but also ground transportation and lodging.

### **5. Collaboration with Insurance Providers:**

Collaborate with insurance providers to offer users personalized insurance quotes based on the predictive model's insights. This can streamline the process of purchasing insurance directly through the platform.



## 6. Mobile Application Development:

Create a dedicated mobile application to make the project more accessible and user-friendly. Mobile apps can provide on-the-go predictions and recommendations, enhancing the overall user experience.

## 7. Integration with Health Data:

Explore partnerships with health data providers to incorporate health-related data into the predictive model. This can help users assess health risks associated with travel and make more informed decisions.

## 8. Gamification Elements:

Introduce gamification elements to encourage user engagement. This could include reward systems for making informed travel decisions or achieving specific milestones in travel planning.

## 9. Research on Emerging Travel Trends:

Stay informed about emerging travel trends and integrate these insights into the predictive model. This ensures that the project remains relevant and aligned with evolving user preferences.

## 10. Natural Language Processing (NLP):

Implement natural language processing capabilities to enhance user interaction. This could include voice-activated commands, chatbots for assistance, and the ability to interpret user feedback in natural language.

## 13. Appendix

### 1. Data Sample:

A sample of the dataset used for training and testing the predictive model is provided below. The dataset includes information such as age, employment type, education status, annual income, family size, chronic diseases, travel enthusiasm, and international travel history.

### 2. Data Preprocessing Details:

The following steps were taken in data preprocessing:

- Handling missing values
- Encoding categorical variables
- Normalizing numerical values

### 3. Model Selection Criteria

The Gradient Boosting algorithm was selected as the preferred machine learning model due to its efficiency in binary classification tasks and robust performance in handling complex relationships within the data.

### 4. Hyperparameter Tuning:

The hyperparameters for the Gradient Boosting algorithm were tuned based on the following considerations: [List hyperparameter names and values].

### 5. Evaluation Metrics:

The model's performance was assessed using the following evaluation metrics:

- Accuracy
- Precision
- Recall
- F1 Score

## 6. Sample Predictions:

Sample predictions generated by the model, based on hypothetical user inputs, are presented in the following format:

- Input: [User Input Features]
- Prediction: [Predicted Outcome]

## 7. User Interface Mock-ups:

Visual representations (mock-ups) of the user interface design are included to illustrate user interactions with the system.

## **UI images**

## 8. Scalability Plan:

To ensure scalability, the system architecture includes [Brief description of scalability considerations].

## 11. Documentation Samples:

Snippets of documentation detailing the solution architecture, data sources, preprocessing steps, model details, and deployment procedures are provided in the complete documentation document.

## 13. Acknowledgments:

We acknowledge the contributions of [List any external resources, datasets, or individuals] to the development of the travel insurance prediction project.

Source Code:

## HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href= "{{url_for('static', filename='css/main.css')}}">
  <title>Travel Insurence Predictor</title>
</head>
<body>
  <div class="bg-image"></div>
  <div class="container">
    <form id="myForm" action="predict" method="post" enctype="multipart/form-
data">
      <label for="Age">Age: </label>
      <input type="number" id="Age" name="Age" placeholder="Enter your
Age" required>

      <label for="EmploymentType">Employment Type: </label>
      <select name="EmploymentType" id="EmploymentType" required>
        <option value="">Choose One</option>
        <option value="GovernmentSector">Government Sector</option>
        <option value="PrivateSector/SelfEmployed">Private Sector/Self
Employed</option>
      </select>

      <label for="GraduateOrNot">Graduate Or Not: </label>
      <select name="GraduateOrNot" id="GraduateOrNot" required>
        <option value="">Choose One</option>
        <option value="Yes">Yes</option>
        <option value="No">No</option>
      </select>

      <label for="AnnualIncome">AnnualIncome: </label>
      <input type="number" id="Number" name="AnnualIncome"
placeholder="Enter your Annual Income" maxlength="10" required>

      <label for="FamilyMembers">FamilyMembers: </label>
      <input type="number" id="family" name="FamilyMembers" required>

      <label for="ChronicDiseases">ChronicDisease: </label>
      <select name="ChronicDiseases" id="ChronicDiseases" required>
        <option value="">Choose One</option>
        <option value="Yes">Yes</option>
        <option value="No">No</option>
      </select>
    </form>
  </div>
</body>
</html>
```

```

<label for="FrequentFlyer">Frequent Flyer: </label>
<select name="frequentFlyer" id="FrequentFlyer" required>
  <option value="">Choose One</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>

<label for="EverTravelledAbroad">Ever Travelled Abroad: </label>
<select name="EverTravelledAbroad" id="EverTravelledAbroad" required>
  <option value="">Choose One</option>
  <option value="Yes">Yes</option>
  <option value="No">No</option>
</select>
<span><button type="submit" id="submit">Submit</button></span>
{% if prediction_text %}
<label id="label">{{ prediction_text }}</label>
{% endif %}
</form>
</div>
<div class="heading">
  <h1>Travel Insurance Predictor</h1>
  <p>Welcome to the Travel Insurance Predictor, your go-to destination for informed travel decisions!
Our website is designed to simplify the process of choosing travel insurance by leveraging advanced predictive
algorithms. By inputting key factors such as age, employment type, and more, users can receive accurate
predictions on whether they should consider taking out travel insurance for their upcoming journeys. Our goal is
to empower travelers with personalized insights, allowing them to make well-informed decisions that align with
their unique circumstances. Travel confidently with the Travel Insurance Predictor, ensuring peace of mind for
every adventure.</p>
</div>
<script src="{url_for('static', filename='js/script.js')}"></script>
</body>
</html>

```

## CSS:

```

*, *:before, *:after
{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body

```

```
{
  display: flex;
  justify-content: space-evenly;
  align-items: center;
  width: 100%;
  min-height: 100vh;
  margin: 0;
}

.bg-image {
  background-image: url('back.jpg');
  filter: blur(5px);
  /* -webkit-filter: blur(8px); */
  background-position: center;
  background-size: cover;
  background-repeat: no-repeat;
  background-attachment: fixed;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  z-index: -1;
}

.container
{
  display: flex;
  justify-content: center;
  align-items: center;
  width: 100%;
  max-width: 340px;
  padding-top: 10px;
  border-radius: 10px;
  background: rgba(221, 221, 214, 0.3);
  box-shadow: 5px 5px 10px rgba(0,0,0,5);
}

label
{
  display: block;
  font-size: 20px;
  margin-bottom: 5px;
}

input
{

```

```
    display: block;
    border: 2px solid black;
    outline: none;
    margin-bottom: 10px;
    padding: 5px;
    border-radius: 5px;
}

select
{
    display: block;
    border: 2px solid black;
    border-radius: 5px;
    font-size: 15px;
    outline: none;
    margin-bottom: 10px;
    padding: 5px;
}

option
{
    display: block;
    border: 2px solid black;
    border-radius: 5px;
    font-size: 15px;
    outline: none;
    margin-bottom: 10px;
    padding: 5px;
}

span
{
    display: flex;
    justify-content: center;
    align-items: center;
}

button
{
    cursor: pointer;
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
    font-size: 15px;
    padding: 10px;
    border: 2px solid black;
    border-radius: 5px;
    outline: none;
}
```

```
    margin-bottom: 5px;
}

#submit:hover + #label
{
    background-color: black;
    color: white;
    font-size: 20px;
    border: 2px solid black;
    border-radius: 5px;
    padding: 5px;
    margin-top: 10px;
}

#label
{
    background-color: black;
    color: white;
    font-size: 20px;
    border: 2px solid black;
    border-radius: 5px;
    padding: 5px;
    margin-top: 10px;
}

.heading
{
    width: 100%;
    max-width: 700px;
    padding: 10px;
}

h1
{
    color: #ffffff;
}

p
{
    color: #ffffff;
    font-size: 20px;
    text-align: justify;
    margin: 10px;
}
```



## App.py:

```
from flask import Flask, render_template, request
import numpy as np
import pickle
# from sklearn.preprocessing import MinMaxScaler

app = Flask(__name__)

# Load the model
with open("Travel.pkl", "rb") as model_file:
    model = pickle.load(model_file)

# Load the MinMaxScaler
with open("MinMaxScaler.pkl", "rb") as scaler_file:
    ms = pickle.load(scaler_file)

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

@app.route('/predict', methods=['POST'])
def predict():
    Age = int(request.form['Age'])

    EmploymentType = request.form['EmploymentType']
    if EmploymentType == 'PrivateSector/SelfEmployed':
        EmploymentType = 1
    else:
        EmploymentType = 0

    GraduateOrNot = request.form['GraduateOrNot']
    if GraduateOrNot == 'Yes':
        GraduateOrNot = 1
    else:
        GraduateOrNot = 0

    AnnualIncome = float(request.form['AnnualIncome'])

    FamilyMembers = int(request.form['FamilyMembers'])

    chronicDiseases = request.form['ChronicDiseases']
    if chronicDiseases == 'Yes':
        chronicDiseases = 1
    else:
        chronicDiseases = 0

    FrequentFlyer = request.form['frequentFlyer']
```

```

if FrequentFlyer == 'Yes':
    FrequentFlyer = 1
else:
    FrequentFlyer = 0

EverTravelledAbroad = request.form['EverTravelledAbroad']
if EverTravelledAbroad == 'Yes':
    EverTravelledAbroad = 1
else:
    EverTravelledAbroad = 0

# Create an array with the input features
input_features = np.array([Age, EmploymentType, GraduateOrNot, AnnualIncome,
FamilyMembers, chronicDiseases, FrequentFlyer, EverTravelledAbroad])

# Reshape the array for compatibility with MinMaxScaler
input_features_reshaped = input_features.reshape(1, -1)

# Scale the input features using the MinMaxScaler
input_features_scaled = ms.transform(input_features_reshaped)

# Make the prediction
prediction = model.predict(input_features_scaled)

# Determine the prediction text based on the model output
if prediction[0] == 1:
    prediction_text = "The Person will take the Insurance"
else:
    prediction_text = "The Person does not take the Insurance"

return render_template('index.html', prediction_text=prediction_text)

if __name__ == "__main__":
    app.run(debug=True)

```

Github-Link:

<https://github.com/smartinternz02/SI-GuidedProject-611657-1700563779.git>

Video-Demo-Link:

[https://drive.google.com/drive/folders/15JJtrKPvTK\\_cR42GfDr\\_bkHKPT5Gecph](https://drive.google.com/drive/folders/15JJtrKPvTK_cR42GfDr_bkHKPT5Gecph)