



Mental Health Prediction

SmartInternz

[www.smartinternz.com](http://www.smartinternz.com/)

**Mental Health Prediction Using Machine Learning**

Mental Health diorders affect millions of people worldwide, impacting their emotional, psychological, and social well-being. Early detection and intervention are crucial to providing timely support and improving quality of life. However, many individuals remain undiagnosed or untreated due to lack of awareness, stigma, or limited access to mental health services. In recent years, the increasing prevalence of stress, anxiety, depression, and other mental health conditions has highlighted the need for proactive measures. Traditional diagnostic methods rely heavily on self-reported symptoms and clinical evaluations, which can be subjective or delayed. This is where predictive analytics can play a transformative role.

A Mental Health Prediction System leverages machine learning and data-driven approaches to assess risk factors, behavioral patterns, and psychological indicators that may signal potential mental health issues. By analyzing historical and real-time data—such as patient health records, lifestyle habits, social interactions, and biometric signals—the system can identify early warning signs and predict the likelihood of mental health disorders.

# 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 & Hyperparameter Tuning
  + Testing model with multiple evaluation metrics
  + Comparing model accuracy before & after applying hyperparameter tuning
* 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

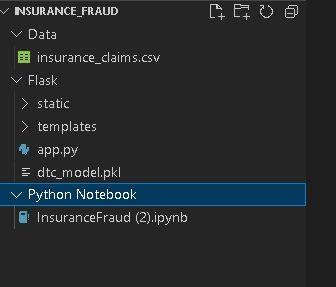
# Prior Knowledge:

You must have prior knowledge of following topics to complete this project.

* ML Concepts
  + Supervised learning: <https://www.javatpoint.com/supervised-machine-learning>
  + Unsupervised learning: <https://www.javatpoint.com/unsupervised-machine-learning>
* Decision tree: [https://www.javatpoint.com/machine-learning-decision-tree-classification-](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm) [algorithm](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm)
* Random forest: <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
* KNN: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
* Xgboost: [https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/) [understand-the-math-behind-xgboost/](https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/)
* Evaluation metrics: [https://www.analyticsvidhya.com/blog/2019/08/11-important-model-](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/) [evaluation-error-metrics/](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/)
* Flask Basics : <https://www.youtube.com/watch?v=lj4I_CvBnt0>

# Project Structure:

Create the Project folder which contains files as shown below



* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* Dtc\_model.pkl is our saved model. Further we will use this model for flask integration.
* Data Folder contains the Dataset used
* The Notebook file contains procedure for building th model.

# Milestone 1: Define Problem / Problem Understanding

## Activity 1: Specify the business problem

Refer Project Description

## Activity 2: Business requirements

A drug classification project can have a variety of business requirements, depending on the specific goals and objectives of the project. Some potential requirements may include:

* + Accurate and up-to-date information: The project should use the most recent and reliable data to classify drugs, in order to ensure that the information is accurate and relevant to current medical practices.
  + Flexibility: The classification system should be flexible and able to adapt to new drugs and changing information as it becomes available.
  + Compliance: The project should comply with all relevant laws and regulations, such as FDA guidelines for classifying drugs.
  + User-friendly interface: The classification system should be easy to use and understand for both medical professionals and patients.

## Activity 3: Literature Survey (Student Will Write)

A literature survey for a drug classification project would involve researching and reviewing existing studies, articles, and other publications on the topic of drug classification. The survey would aim to gather information on current classification systems, 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 drug classification projects, and any relevant data or findings that could inform the design and implementation of the current project.

## Activity 4: Social or Business Impact.

Social Impact :- Improved patient care: By providing accurate and up-to-date information on drugs, a drug classification project can help healthcare professionals make more informed decisions about treatment options, leading to improved patient care.

Business Model/Impact :- New drug development: By providing information on the properties and interactions of different drugs, a drug classification project can assist in the development of new treatments and therapies.

# Milestone 2: Data Collection & Preparation

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset.

## 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.

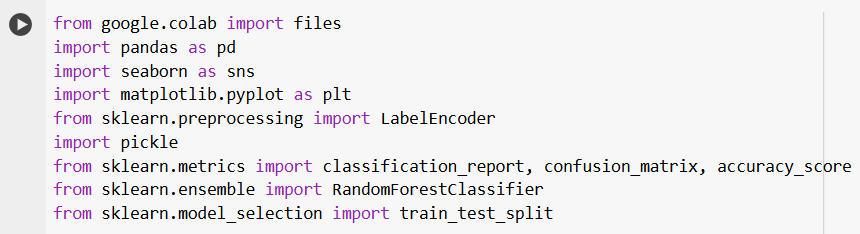
Link: <https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey>

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

**Note:** There are a number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.

## Activity 1.1: Importing the libraries

Import the necessary libraries as shown in the image.



## Activity 1.2: Read the Dataset

Our dataset format is .csv. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.

N 

## Activity 2: Data Preparation

Now that we have an understanding of the dataset, the next step is to pre-process the collected data. The raw dataset may contain noise or inconsistencies, making it unsuitable for training a machine learning model directly. To improve model performance, it's important to clean the data effectively. This pre-processing stage typically involves:

* Handling missing values
* Handling outliers

## Activity 2.1: Handling missing values

* + - For checking the null values, df.isna().any( ) function is used. To sum those null values we use .sum() function.



* + - Dropping the irrelevant columns like timestamp, comments, etc, and filling the missing values with the help of .fillna() function

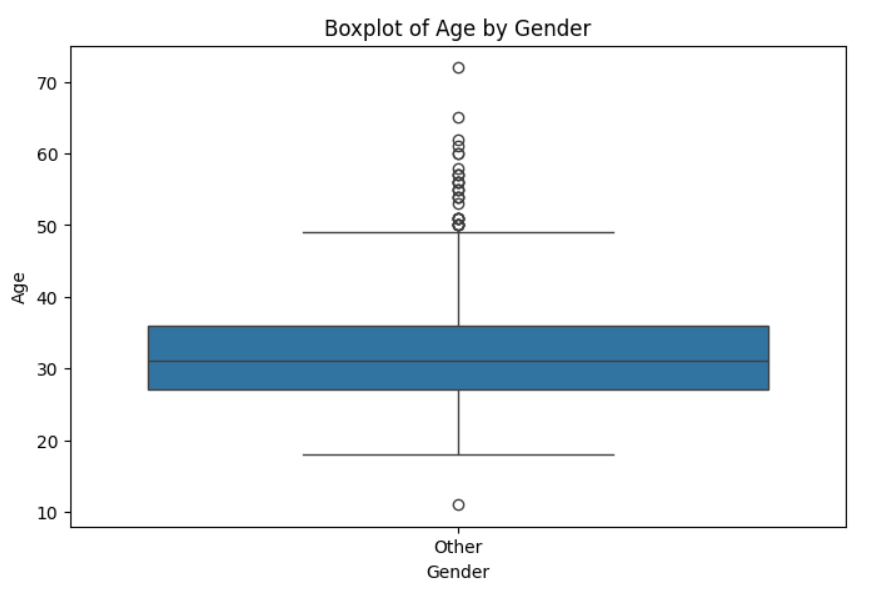
## 

## 

## Activity 2.2:

## Handling Outliers

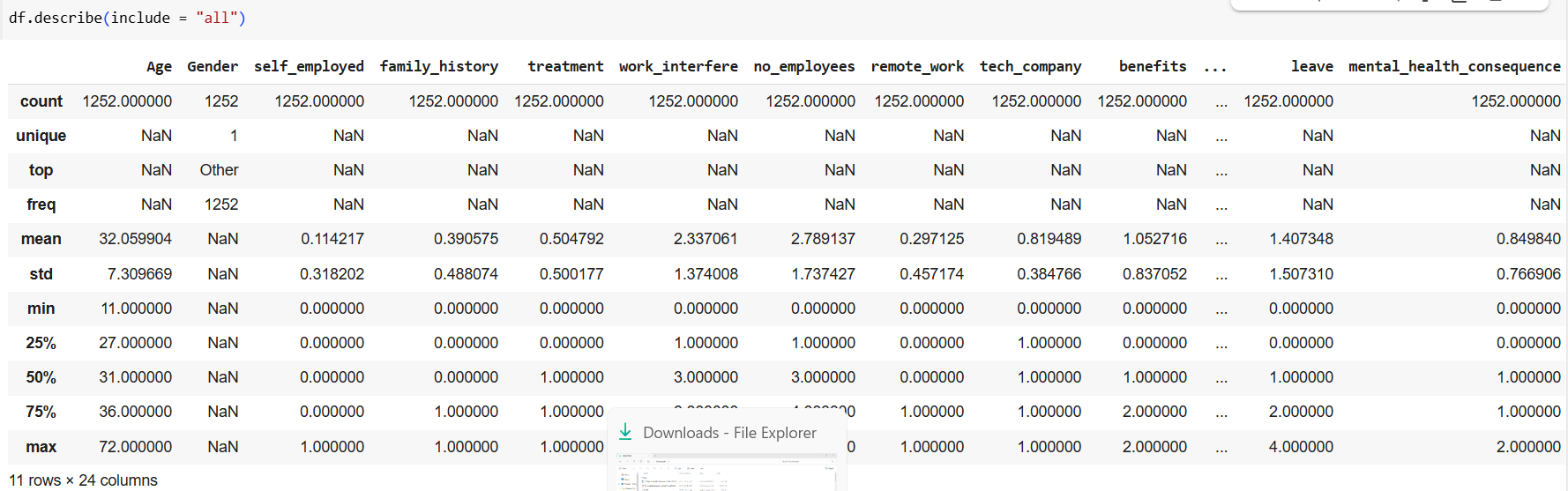
With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of age feature with some mathematical formula.

* From the below diagram, we could visualize that policy\_annual\_premium feature has outliers. Boxplot from seaborn library is used here.
* While outliers were identified and quantified, no treatment (removal or transformation) was applied at this stage. The decision to retain outliers was made to:
* Preserve the original data distribution for initial exploratory analysis
* Avoid premature data modification that might affect subsequent analyses
* Allow for comprehensive evaluation of the raw dataset

**Milestone 3: Exploratory Data Analysis**

## Activity 1: Descriptive statistical

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.



Here, the columns details of the dataset

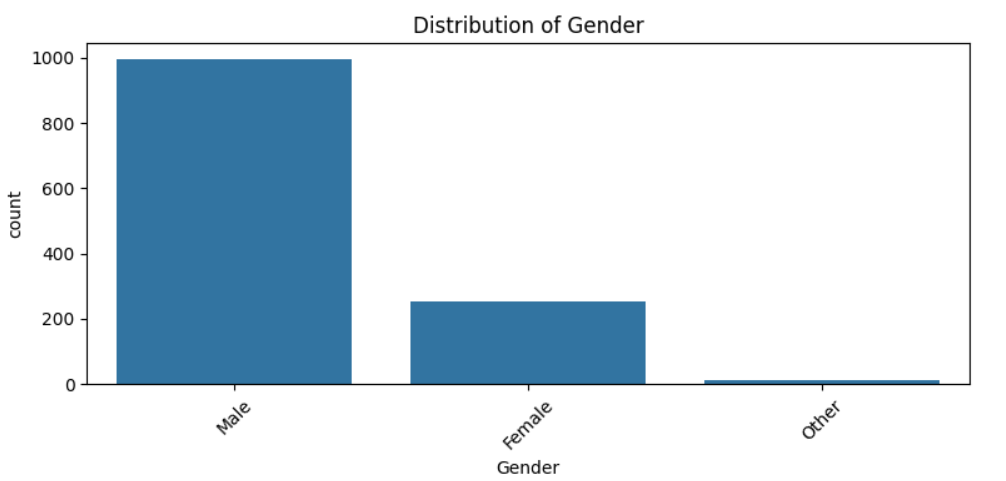
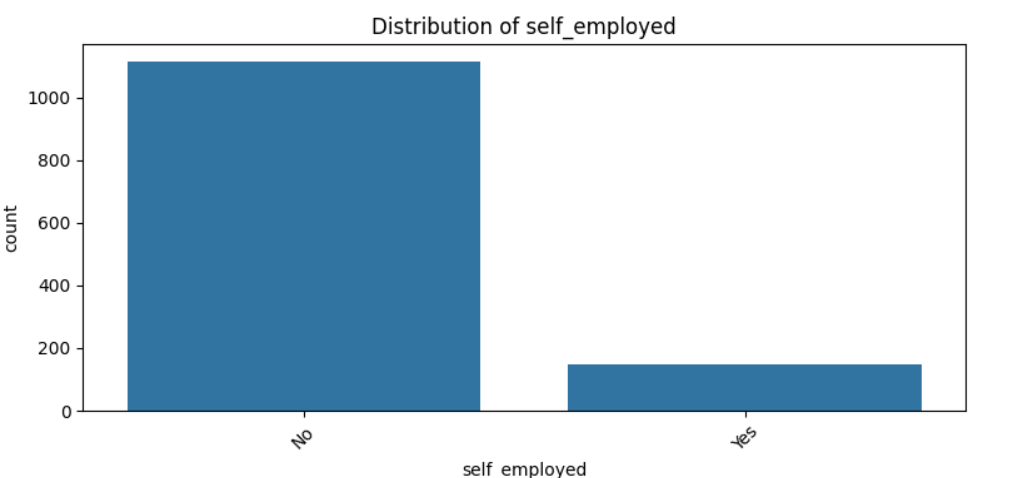
## 

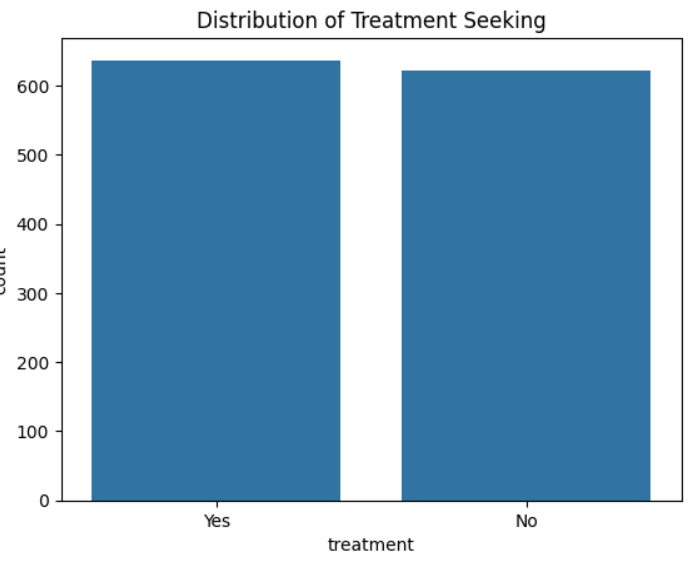
## Activity 2: Visual analysis

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

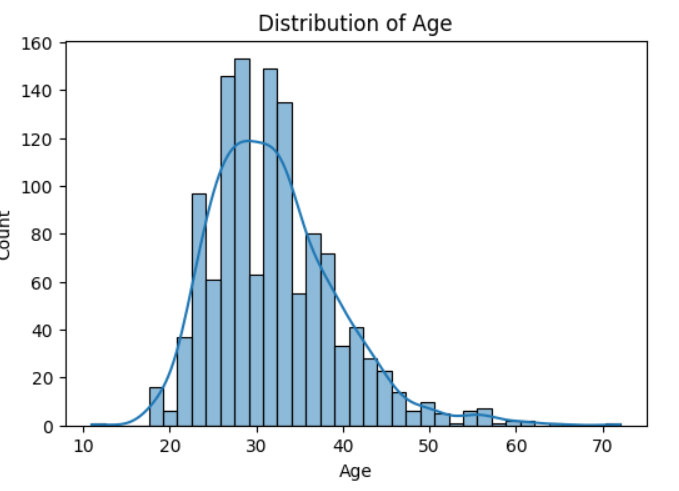
## Activity 2.1: Univariate analysis

Univariate analysis is understanding the data with a single feature. Here we have bar charts for different functions.





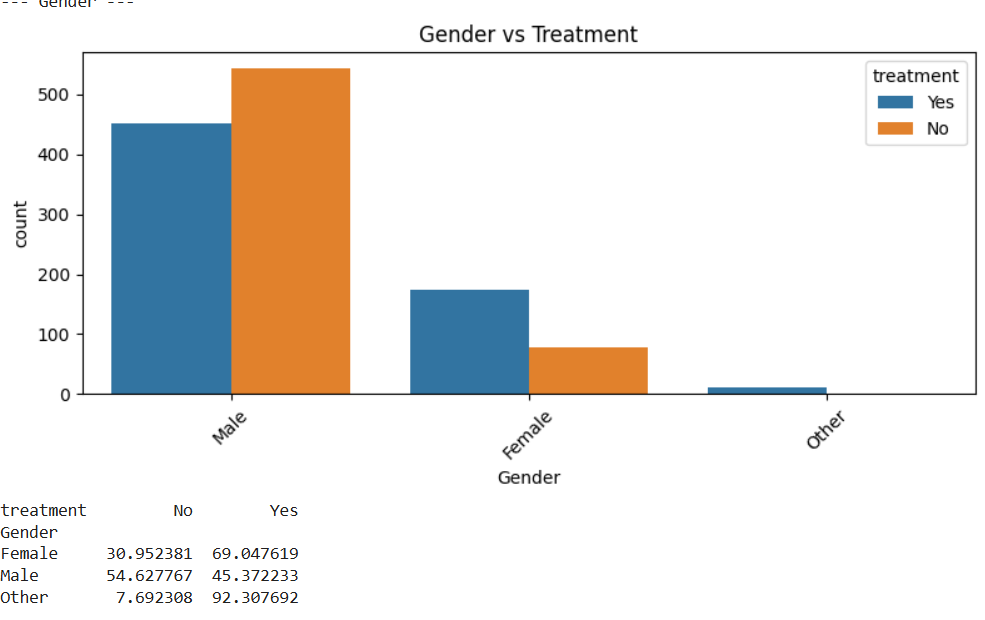
* The preceding visualizations were generated to analyze the distribution and relationships within the categorical variables of the dataset. Bar plots were employed to examine the frequency and proportions of different categories, providing insights into dominant trends, imbalances, or unexpected patterns
  + From the below histogram we can say that ‘age’ Feature is almost Normally distributed. Majority of Insurance claims are of age 20 to 35.

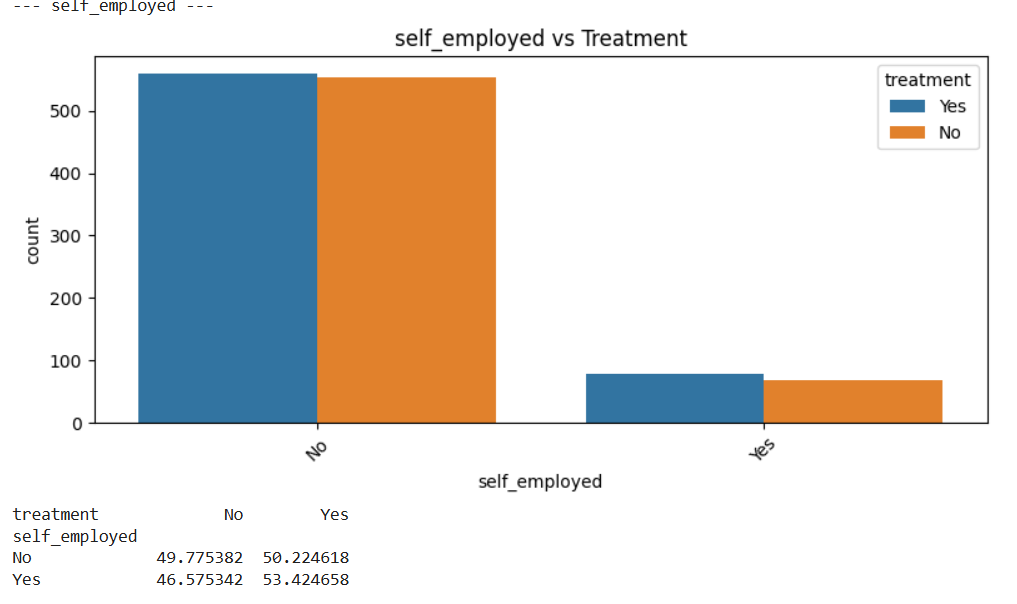


## Activity 2.2: Bivariate analysis

To find the relation between two features we use bivariate analysis. Here we can used barplot.

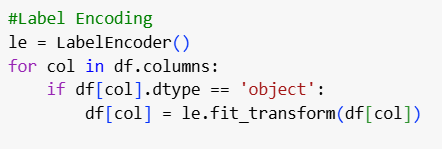
* + - Barplot is used here. As a 1st parameter we are passing Gender and as a 2nd parameter we are passing Treatement which we are treating as our target variable.
    - From the below plot you can understand that distribution of gender on Treatement.
    - We can concur from the following data that while more number of men were diagnosed with a mental illness they were less likely to get treatment when compared to women.
    - From the self employment vs treatment graph given below we can gather that the self employment is not affecting if the person is getting treatment or not





**Encoding the Categorical Features:**

* The categorical Features are can’t be passed directly to the Machine Learning Model. So we convert them into Numerical data based on their order. This Technique is called Encoding.
* Here we are importing Label Encoder from the Sklearn Library.
* Here we are applying fit\_transform to transform the categorical features to numerical features.



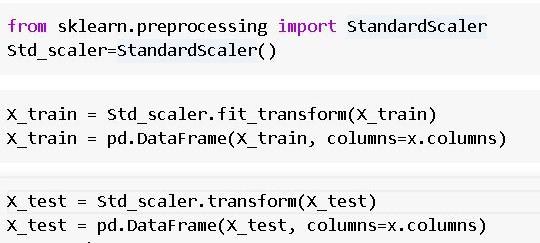
**Splitting data into train and test**

Now let’s split the Dataset into train and test sets. First split the dataset into x and y and then split the data set

The following code snippet prepares the dataset for a supervised machine learning task by separating features (X) from the target variable (y) and splitting them into training and testing sets: For splitting training and testing data we are using train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.

#### Scaling

* + - Scaling is a technique used to transform the values of a dataset to a similar scale to improve the performance of machine learning algorithms. Scaling is important because many machine learning algorithms are sensitive to the scale of the input features.
    - Here we are using Standard Scaler.
    - This scales the data to have a mean of 0 and a standard deviation of 1. The formula is given by: X\_scaled = (X - X\_mean) / X\_std



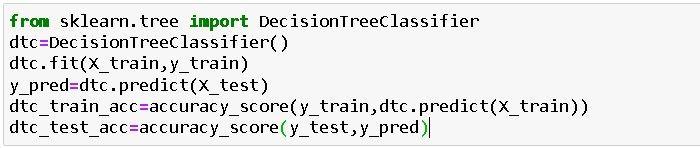
# Milestone 4: Model Building

## Activity 1: Training the model in multiple algorithms

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying three classification algorithms. The best model is saved based on its performance.

## Activity 1.1: Decision tree model

First Decision Tree is imported from sklearn Library then DecisionTreeClassifier algorithm is initialised and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. We can find the Train and Test accuracy by X\_train and X\_test.



## Activity 1.2: Random forest model

First Random Forest Model is imported from sklearn Library then RandomForestClassifier algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in a new variable. We can find the Train and Test accuracy by X\_train and X\_test.



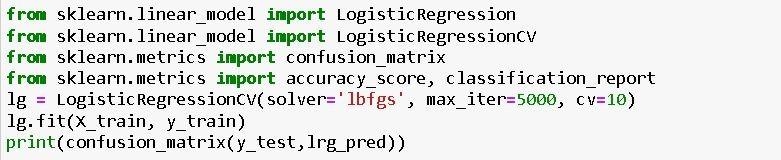
## Activity 1.3: KNN model

KNN Model is imported from sklearn Library then KNeighborsClassifier algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.



## Activity 1.4: Logistic Regression model

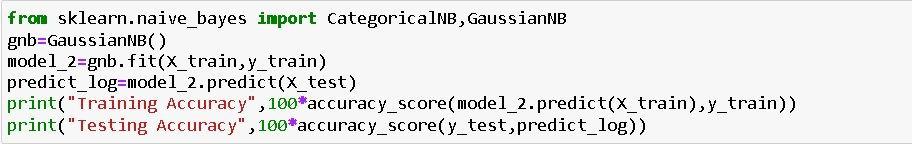
Logistic Regression Model is imported from sklearn Library then Logistic Regression algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in new variable. For evaluating the model, confusion matrix is done.



## Activity 1.5: Naïve Bayes model

Naïve Bayes Model is imported from sklearn Library then Naïve Bayes algorithm is initialised and training data is passed to the model with .fit() function. Test data is predicted with

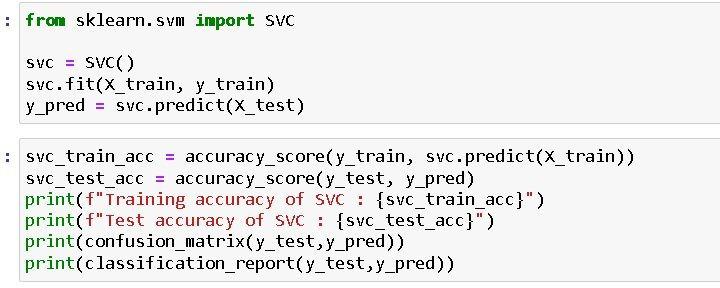
.predict() function and saved in new variable. We can find the Train and Test accuracy by X\_train and X\_test.



## Activity 1.6: SVM model

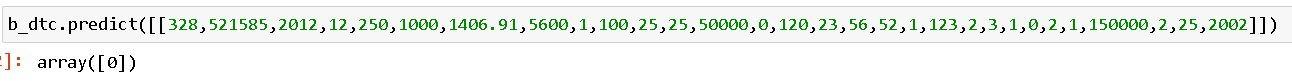
SVM Model is imported from sklearn Library then SVM algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with

.predict() function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.



## Activity 2: Testing the model

Here we have tested with Decision Tree algorithm. You can test with all algorithm. With the help of predict() function.



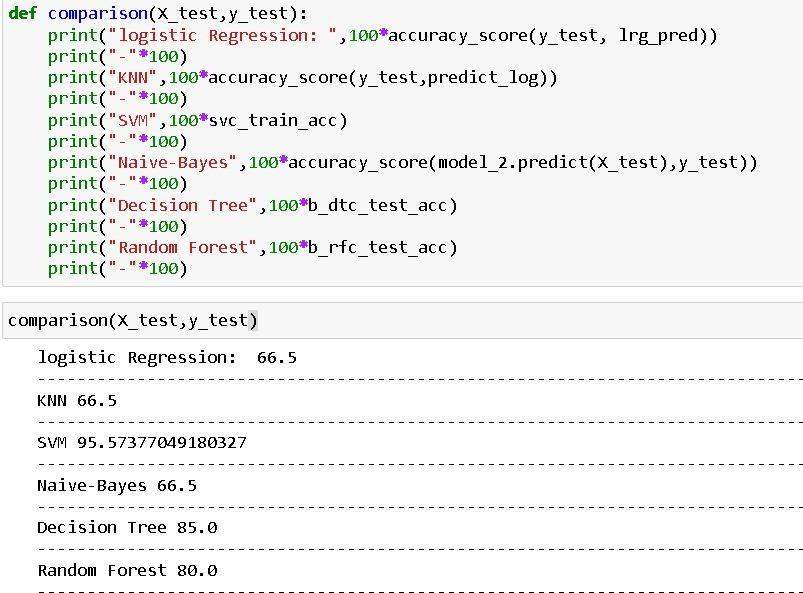
# Milestone 5: Performance Testing & Hyperparameter Tuning

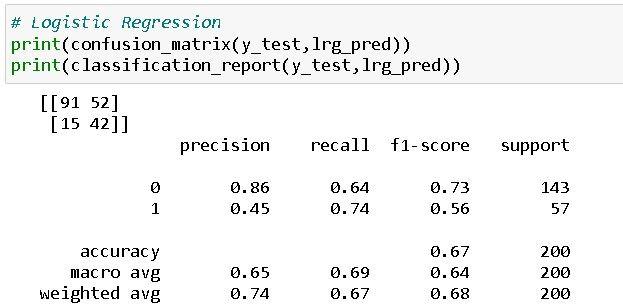
## Activity 1: Testing model with multiple evaluation metrics

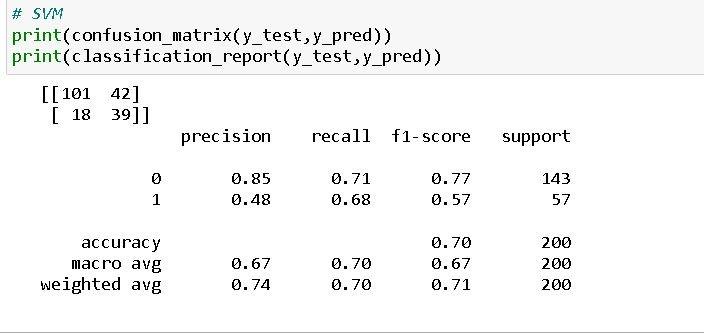
Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for classification tasks including accuracy, precision, recall, support and F1-score.

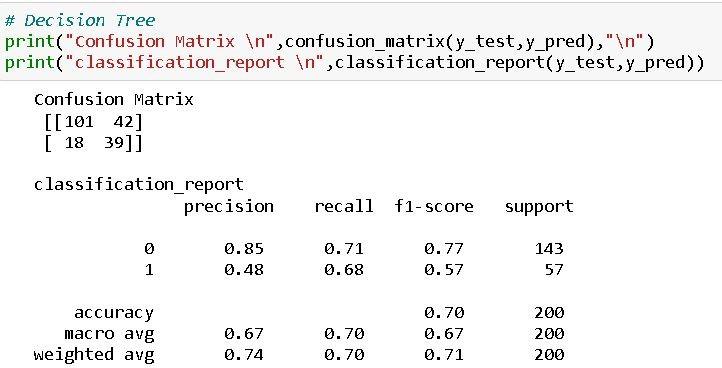
## Activity 1.1: Compare the model

For comparing the above four models, the compareModel function is defined.









After calling the function, the results of models are displayed as output. From the above models Decision Tree is performing well.

## Activity 2: Comparing model accuracy before & after applying hyperparameter tuning (Hyperparameter tuning is optional. For this project it is not required.)

Evaluating performance of the model From sklearn, cross\_val\_score is used to evaluate the score of the model. On the parameters, we have given rf (model name), x, y, cv (as 5 folds). Our model is performing well.

**Note:** To understand cross validation, refer to this [link](https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85)



# Milestone 6: Model Deployment

## Activity 1: Save the best model

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance.This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.

## Activity 2: Integrate with Web Framework

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server-side script
* Run the web application

## Activity 2.1: Building Html Page:

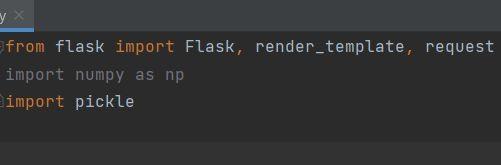
For this project create HTML file namely

### index.html

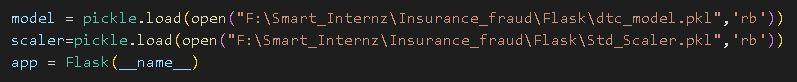
and save them in the templates folder. Refer this [link](https://drive.google.com/file/d/1kgOB42PCwDzr4KYB7euFZWluyarjq2CH/view?usp=share_link) for templates.

## Activity 2.2: Build Python code:

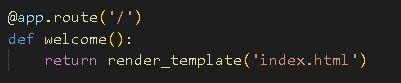
Import the libraries



Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module ( name ) as argument.



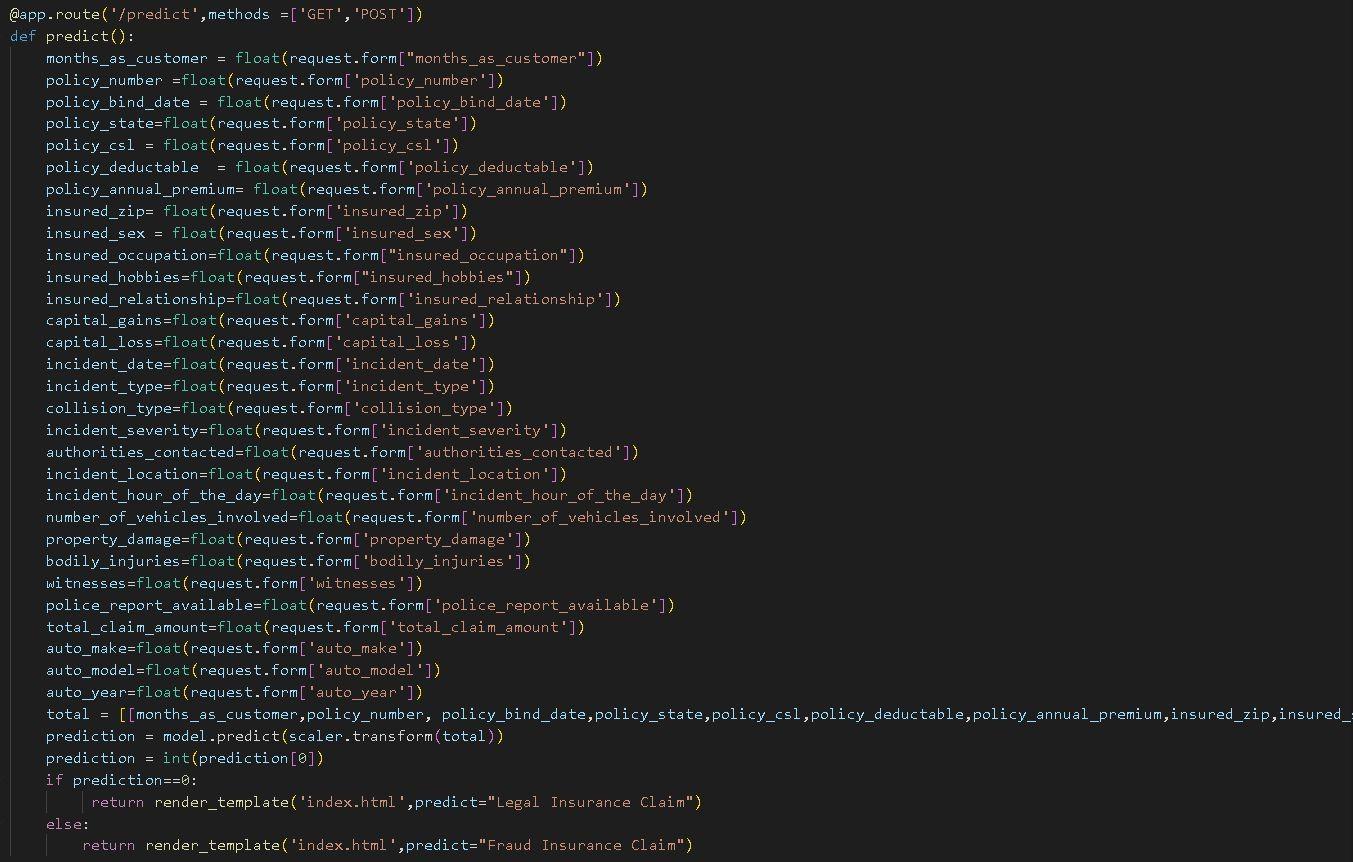
Render HTML page:



Here we will be using a declared constructor to route to the HTML page which we have created earlier.

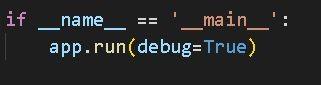
In the above example, ‘/’ URL is bound with the index.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Retrieves the value from UI:



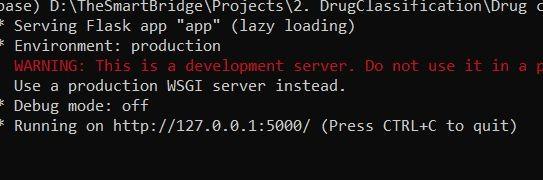
Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the submit.html page earlier.

Main Function:

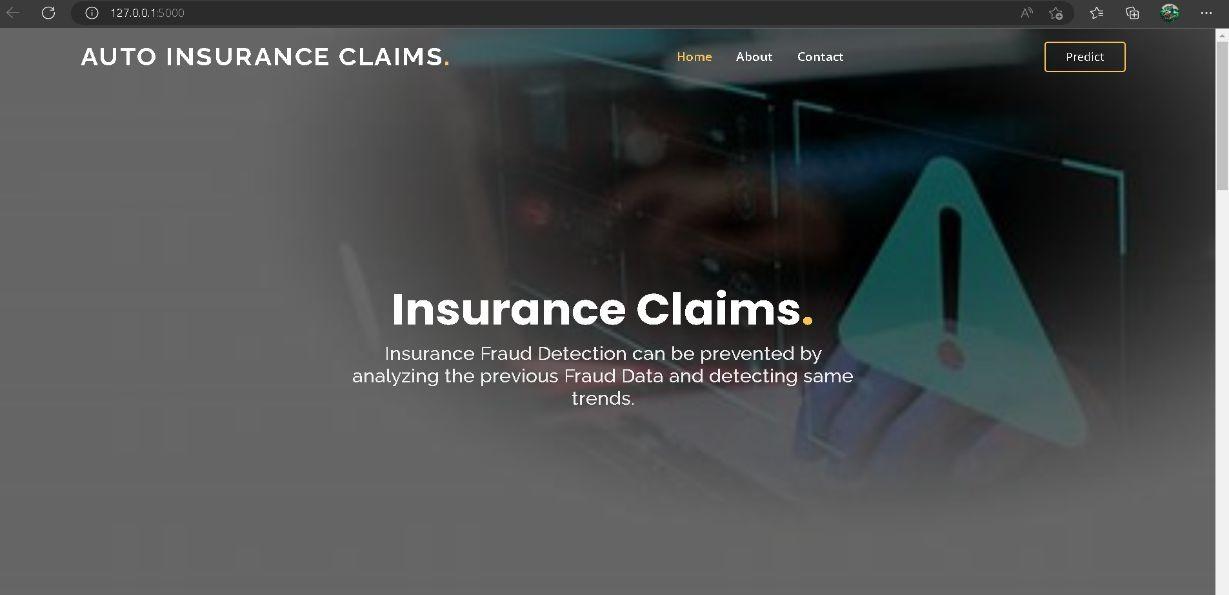


## Activity 2.3: Run the web application

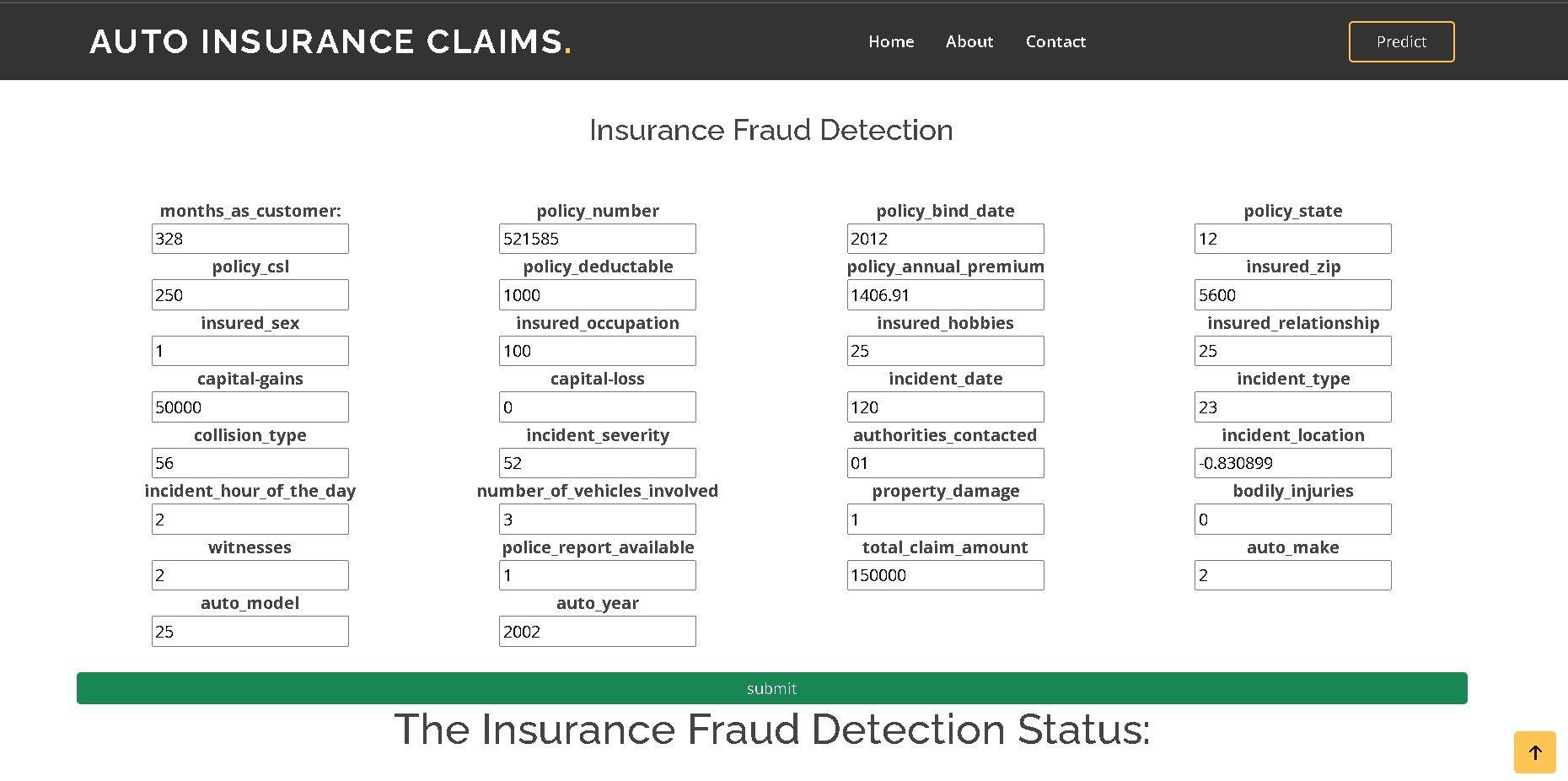
* + Open anaconda prompt from the start menu
  + Navigate to the folder where your python script is.
  + Now type “python app.py” command
  + Navigate to the localhost where you can view your web page.
  + Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

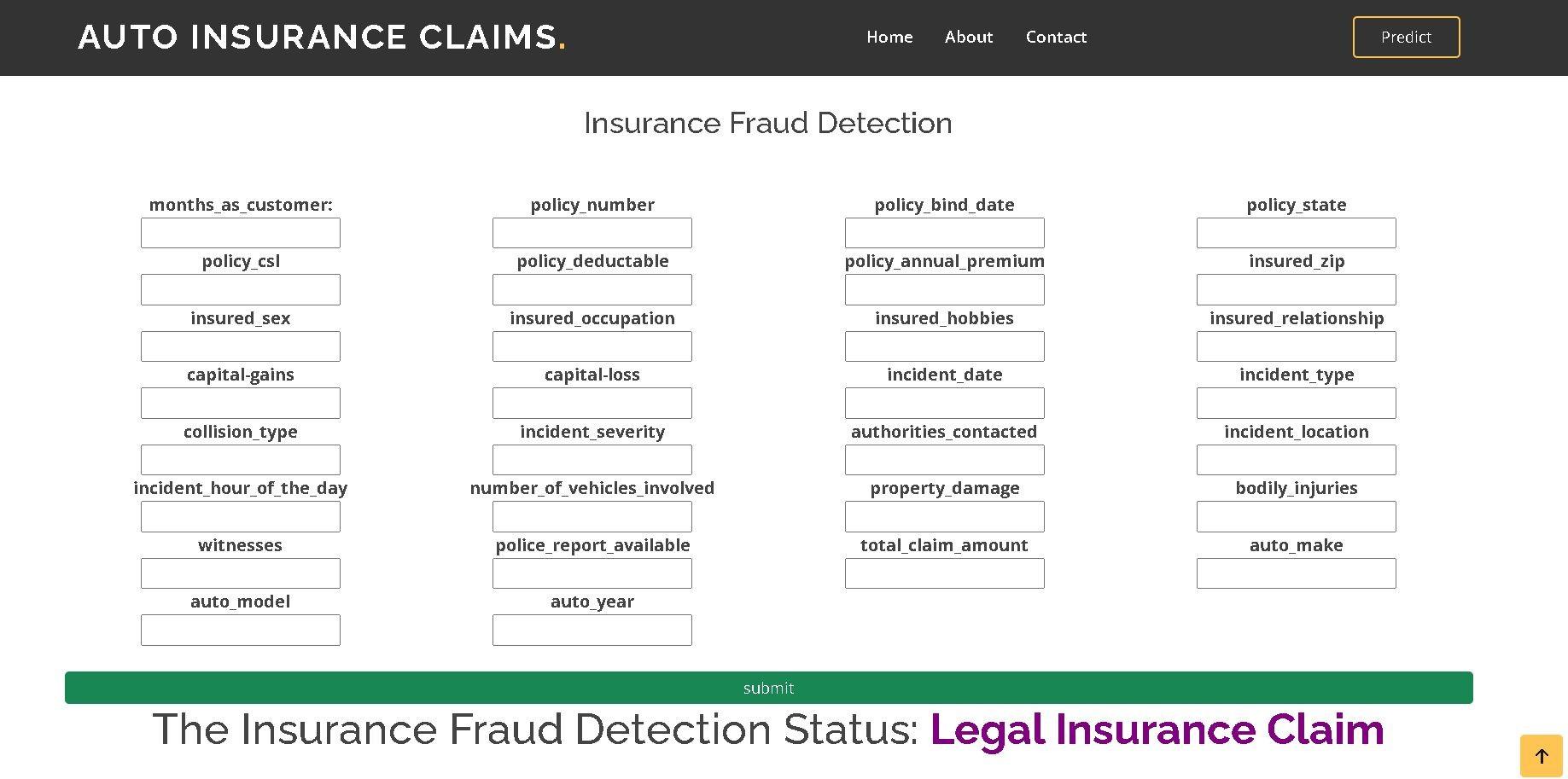


Now,Go the web browser and write the localhost url [(http://127.0.0.1:5000)](http://127.0.0.1:5000/) to get the below result









**Milestone 7: Project Demonstration & Documentation**

Below mentioned deliverables to be submitted along with other deliverables

## Activity 1:- Record explanation Video for project end to end solution

**Activity 2:- Project Documentation-Step by step project development procedure**

Create document as per the template provided