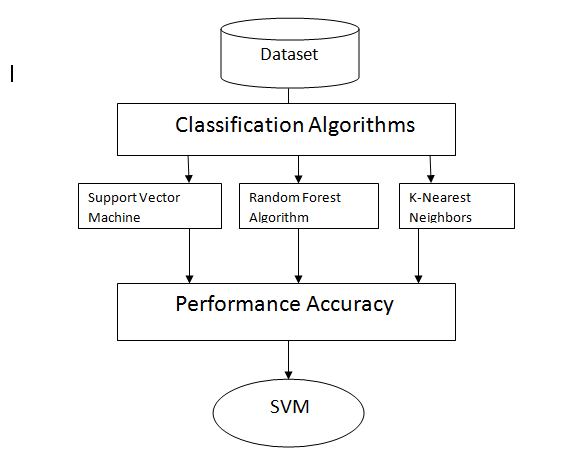
**Liver Patient Analysis**

**Project Description:**

Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this paper is to analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This paper focuses on the related works of various authors on liver disease such that algorithms were implemented using Weka tool that is a machine learning software written in Java. Various attributes that are essential in the prediction of liver disease were examined and the dataset of liver patients were also evaluated. This paper compares various classification algorithms such as Random Forest, Logistic Regression and Separation Algorithm with an aim to identify the best technique. Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver diseaserecommended to the user.

**Technical Architecture:**



**Project Objectives:**

By the end of this project you will:

* Know fundamental concepts of Python.
* Know how to install necessary packages and setting up the environment.
* Know how to build a web application using the Flask framework.

**Project Flow:**

* User interacts with the UI (User Interface) to predict the values
* The list of predicted output is showcased on the UI

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

* Data Collection
* Download dataset / Create dataset
* Data Preprocessing
* Import the Libraries
* Import the dataset
* Data Visualization
* Taking care of missing data
* Label Encoding
* Splitting Data into Train and Test
* Model Building
* Training and Testing the model
* Save the Model
* Application Building
* Build an HTML Page
* Build the python flask app
* Run the application

**Prerequisites:**

* **In order to develop this project we need to install following softwares/packages:**

**Anaconda Navigator :**

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook,

QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder

To install Anaconda navigator and to know how to use Jupyter Notebook & Spyder using Anaconda watch the video

Link:  [HYPERLINK "https://www.youtube.com/watch?v=5mDYijMfSzs&feature=emb\_logo"Click here to](https://www.youtube.com/watch?v=5mDYijMfSzs&feature=emb_logo) Watch video

**To build a Liver patient analysis you must require the following packages**

* **Install Flask**
* Flask is a Web application framework written in python
* For installation of Flask
* Open your anaconda prompt and Type “pip install Flask”

**Prior Knowledge:**

One should have knowledge on the following Concepts:

* **HTML Basics**

<https://www.youtube.com/watch?v=UB1O30fR-EE>

* **Bootstrap Basics**

[**https://www.youtube.com/watch?v=5GcQtLDGXy8**](https://www.youtube.com/watch?v=5GcQtLDGXy8)

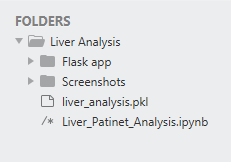
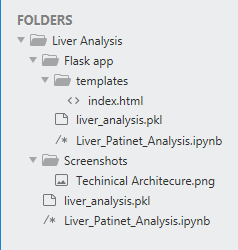
* **Flask Basics**

<https://www.youtube.com/watch?v=lj4I_CvBnt0>

It is recommended to watch above videos to understand the concept before you start your project.

**Project Structure:**

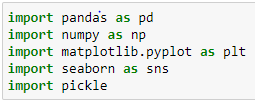
Create a Project folder that contains files as shown below

* We are building a Flask Application that needs HTML pages stored in the templates folder and a python script Liver\_Patient\_Analysis.ipynb for server-side scripting
* Templates folder contains index.html

**Milestone 2: Data Preprocessing**

**Activity1: Import Libraries**

****

**Pandas:** Pandas is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

**Numpy:** NumPy is a **Python** library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

**Matplotlib:** Matplotlib is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

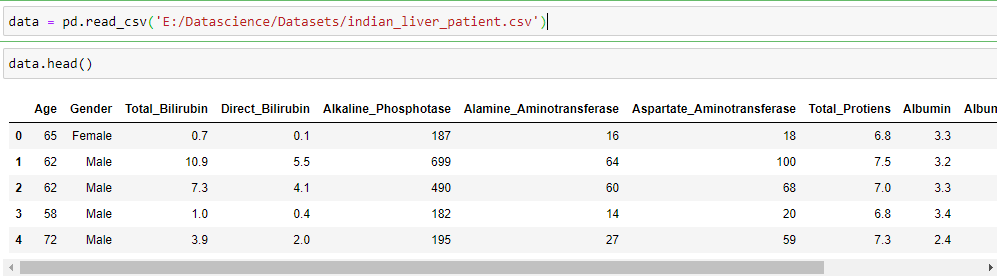
**Seaborn:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

**Pickle:** Python pickle module isused for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickledoes **is** that it “serializes” the object first before writing it to file. Picklingis a way to convert a python object (list, dict, etc.)

The first step is usually importing the libraries that will be needed in the program.

The required libraries to be imported to Python script are:

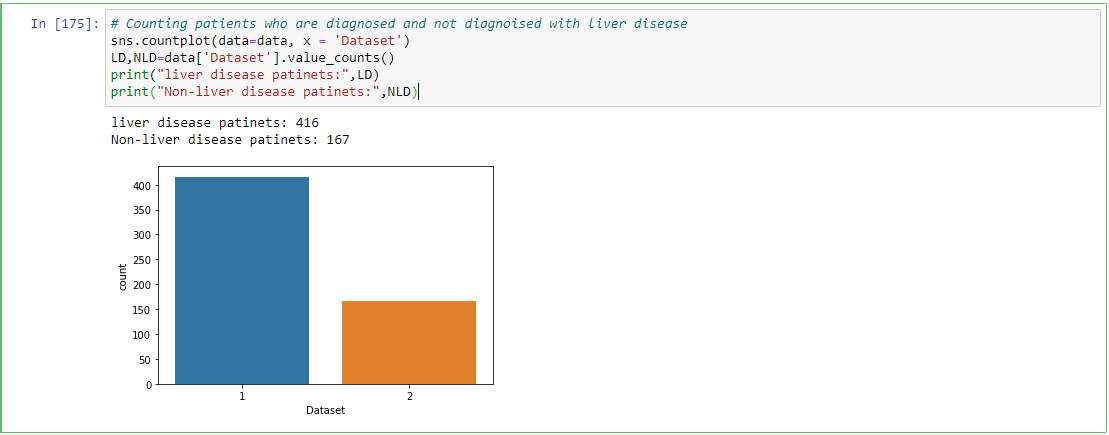
**Activity2: Import the Dataset**

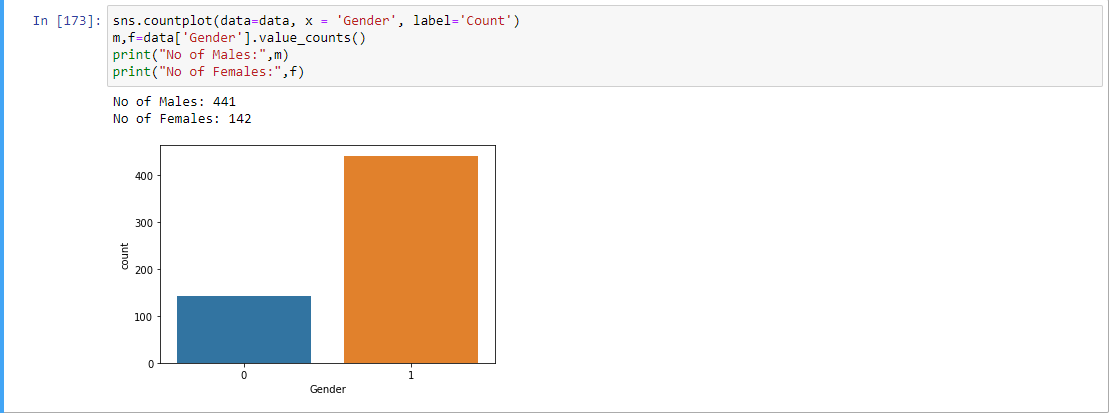


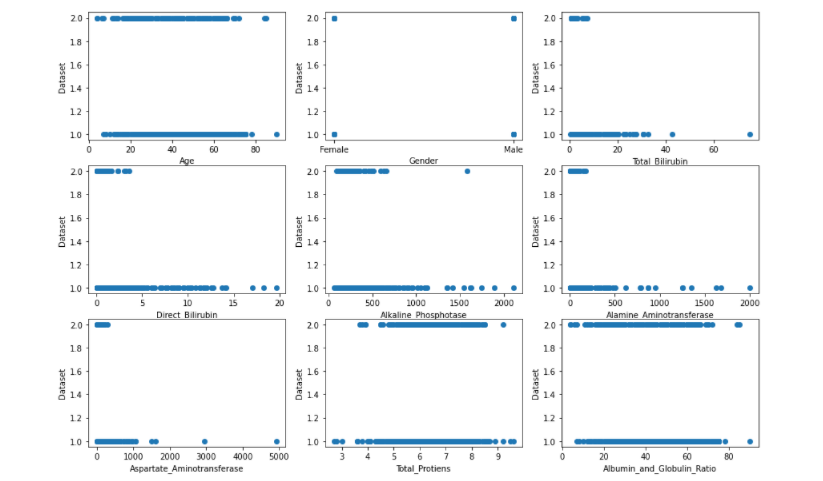
**Head():** Head() function in Pandas, by default, shows you the top 5 rows of data in the DataFrame

**Activity3: Data Visualization**

**Data visualization** gives us a clear idea of what the information means by giving it visual context through maps or graphs. This makes the data more natural for the human mind to comprehend and therefore makes it easier to identify trends, patterns, and outliers within large data sets.





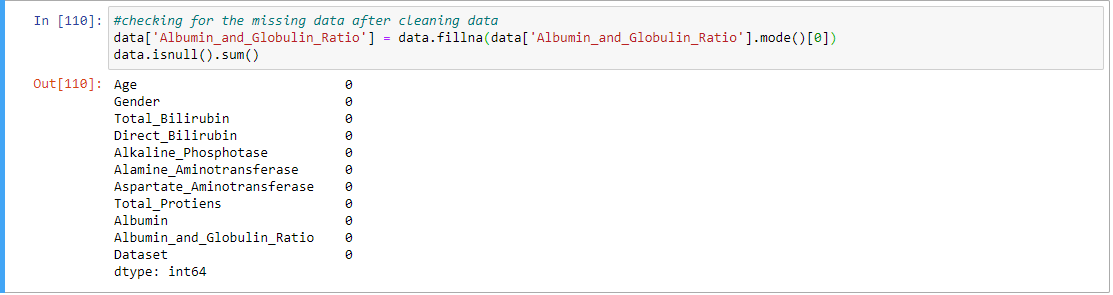


Here we are visualizing data between all inputs data with output data. So that we can better understand what happening in DataFrame.

**Activity4: Taking care of missing data**

Datasets may have missingvalues, and this can cause problems for many machinelearning algorithms. As such, it is good practice to identify and replace missing values for each column in your input data prior to modeling your prediction task.

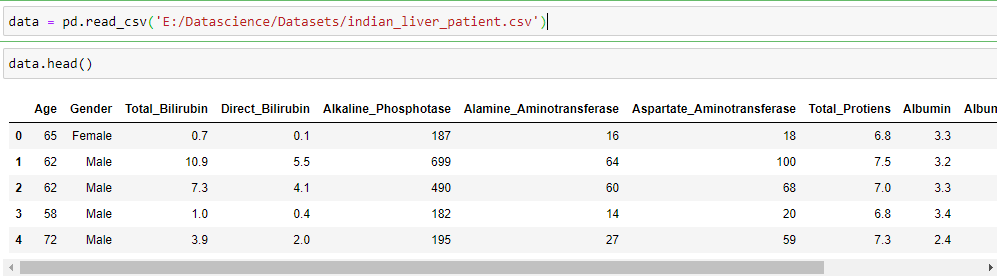
****



**Activity 5: Label Encoding**

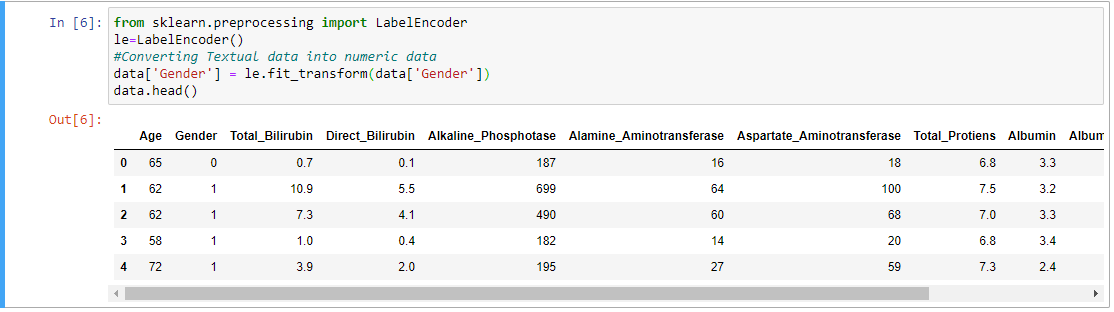
Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning

**Before Label Encoder**

****

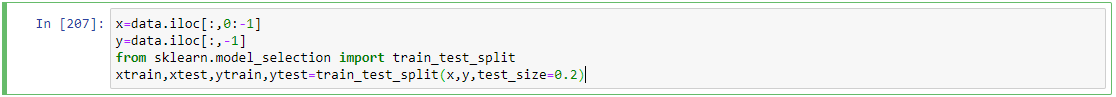
Here we are applying label encoding on Gender column, So it will converted into numerical data.

**After Label Encoder**

****

**Activity 6: Splitting data into train and test**

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Separating data into training and testing sets is an important part of evaluating data mining models. By using similar data for training and testing, you can minimize the effects of data discrepancies and better understand the characteristics of the model.

****

**Scikit-learn** **:** Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction. We can also called sklearn as short.

**Sklearn.model\_selection:** train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

**Test\_size():** This parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset. If you're specifying this parameter, you can ignore the next parameter.

**Milestone 3: Model Building**

The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions.

**Activity 1: Training and Testing model**

In a dataset, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built. Data points in the training set are excluded from the test (validation) set. The models generated are to predict the results unknown which is named as the test set.

Here we are used three different types machine learning models.

**Model 1: Support Vector Machine**

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. Support Vectors are simply the co-ordinates of individual observation. The goal of a support vector machine is not only to draw hyperplanes and divide data points, but to draw the hyperplane the separates data points with the largest margin, or with the most space between the dividing line and any given data point.

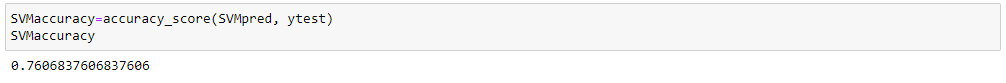
**When should we use this Algorithm:** SVM can be used for classification (distinguishing between several groups or classes) and regression (obtaining a mathematical model to predict something). They can be applied to both linear and non linear problems. Until 2006 they were the best general purpose algorithm for machine learning.



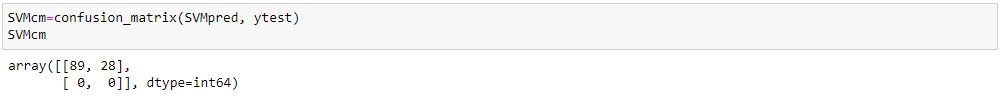
**svm.fit():** Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained. A model that is well-fitted produces more accurate outcomes.

**svm.predict():** The predict() function **accepts only a single argument** which is usually the data to be tested. It returns the labels of the data passed as argument based upon the learned or trained data obtained from the model. Thus, the predict() function works on top of the trained model and makes use of the learned label to map and predict the labels for the data to be tested

**Accuracy\_score():** When taking scientific measurements, it is important to be both accurate and precise. Accuracy represents how close a measurement comes to its true value. This is important because bad equipment, poor data processing or human error can lead to inaccurate results that are not very close to the truth

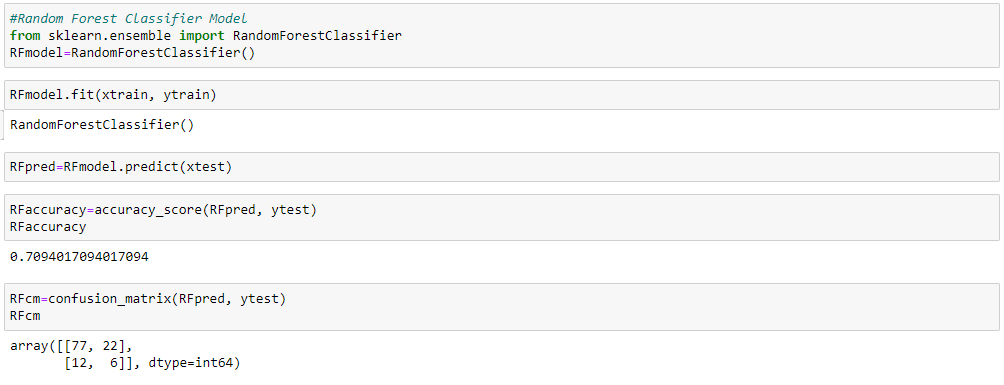


**Confusion\_matrix():** A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. The rows represent the predicted values of the target variable.

****

**Model 2: Random Forest Algorithm**

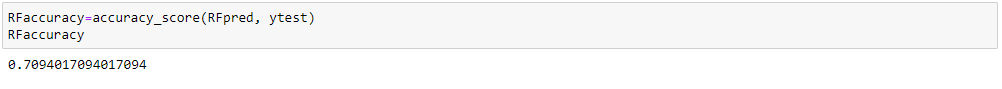
As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.



**RFmodel.fit():** Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained. A model that is well-fitted produces more accurate outcomes.

**RFmodel.predict():** The predict() function **accepts only a single argument** which is usually the data to be tested. It returns the labels of the data passed as argument based upon the learned or trained data obtained from the model. Thus, the predict() function works on top of the trained model and makes use of the learned label to map and predict the labels for the data to be tested

**Accuracy\_score():** When taking scientific measurements, it is important to be both accurate and precise. Accuracy represents how close a measurement comes to its true value. This is important because bad equipment, poor data processing or human error can lead to inaccurate results that are not very close to the truth

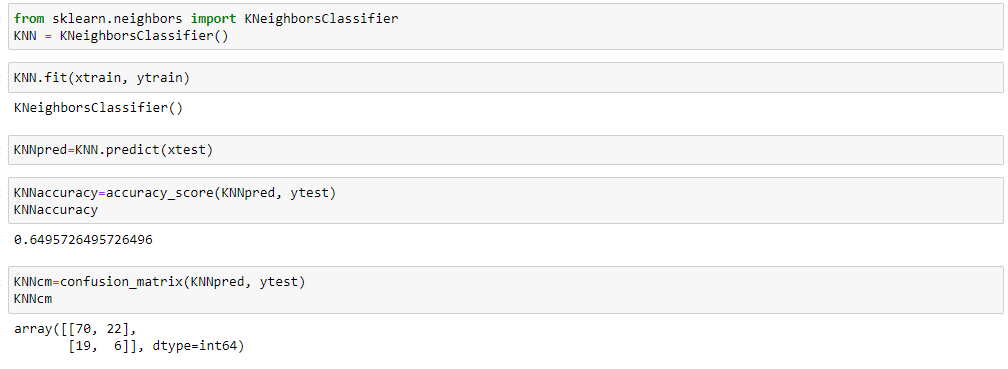


**Confusion\_matrix():** A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. The rows represent the predicted values of the target variable.



**Model 3: K-Nearest Neighbors** **Algorithm**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

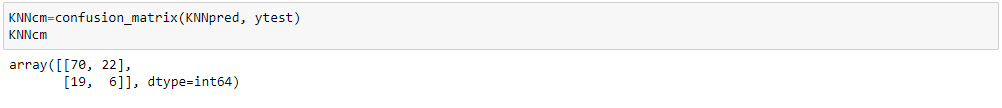


**KNN.predict():** The predict() function **accepts only a single argument** which is usually the data to be tested. It returns the labels of the data passed as argument based upon the learned or trained data obtained from the model. Thus, the predict() function works on top of the trained model and makes use of the learned label to map and predict the labels for the data to be tested

**Accuracy\_score():** When taking scientific measurements, it is important to be both accurate and precise. Accuracy represents how close a measurement comes to its true value. This is important because bad equipment, poor data processing or human error can lead to inaccurate results that are not very close to the truth



**Confusion\_matrix():** A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. The rows represent the predicted values of the target variable.



**Milestone 4: Save the Model**

The pickle module implements serialization protocol, which provides an ability to save and later load Python objects using special binary format. Unlike json , pickle is not limited to simple objects. It can also store references to functions and classes, as well as the state of class instances.

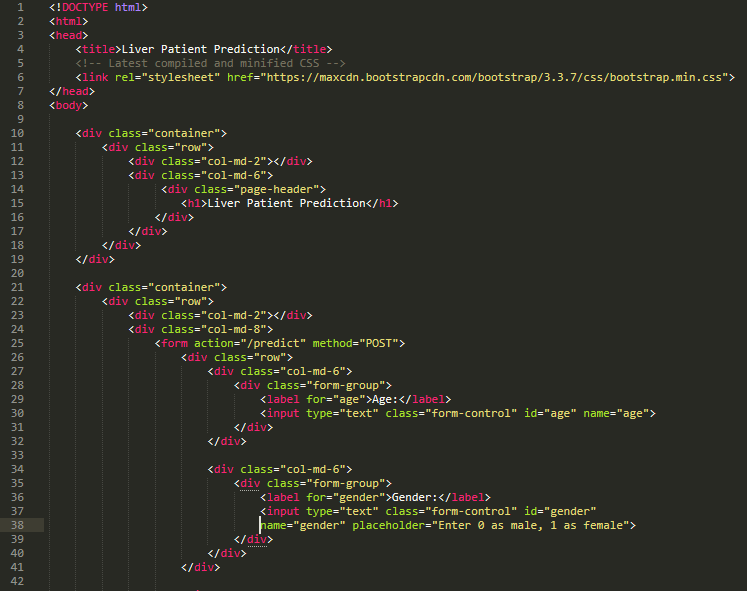
C:\Users\Acer\Desktop\SmartBridge\Liver Analysis\Screenshots\save_pickle1.PNG

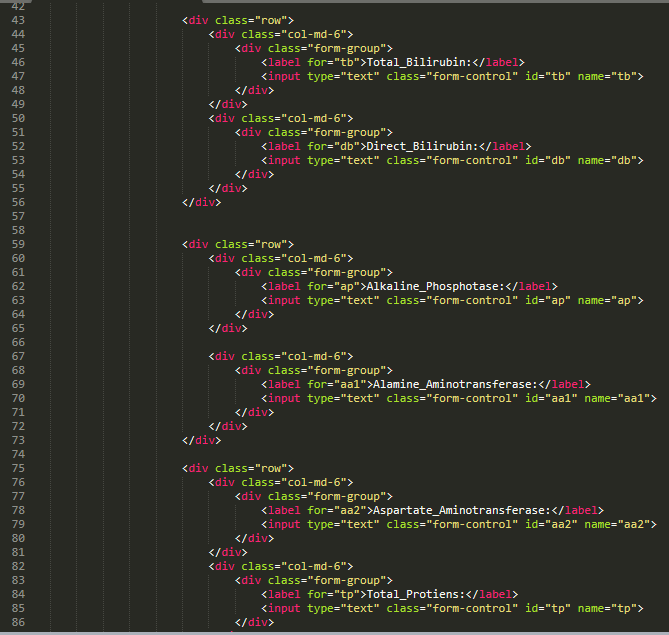
**Milestone 5: Application Building**

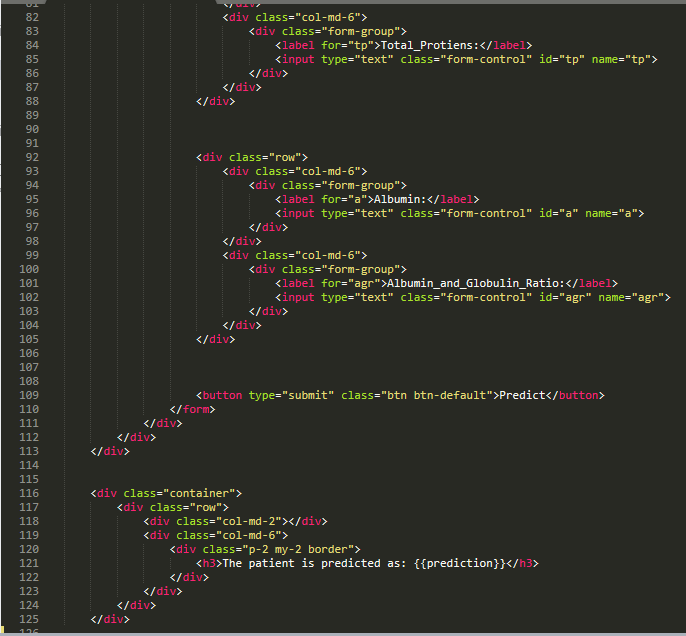
**Activity 1: Build an HTML Page**

We Build an HTML page to take the input values upon clicking the button for liver patient prediction, it has to redirect to url for “predict” which returns predicted value. The output is to be then displayed on browser. The HTML pages are put under templates folder.

* We use HTML to create the front end part of the web page.
* Here, we created one html pages- index.html
* index.html displays the UI design of liver patient prediction.

****



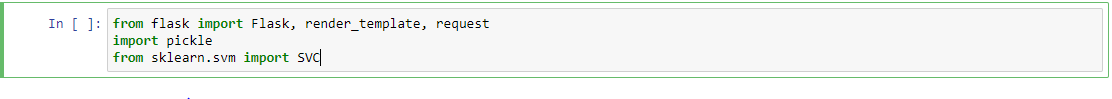
­

**Activity 2: Build the python flask app**

In the flask application, the input parameters are taken from the HTML page These factors are then given to the model to know if the Visa is approved or not and is sent back to the HTML page to notify the user.

In the flask application, whenever the user interacts with UI and press “Submit” button, It will give predicted value whether the patient belongs to liver disease or not.

**Activity 1: Importing Libraries**

****

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 Pickle library to load the model file.

**Activity 2: Routing to the html Page**



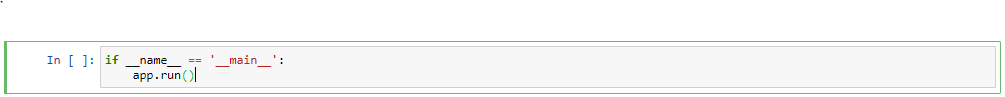
Here, declared constructor is used to route to the HTML page created earlier.

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

Here, “index.html” is rendered when submit button is clicked on the UI

**Activity 3: Main Function**

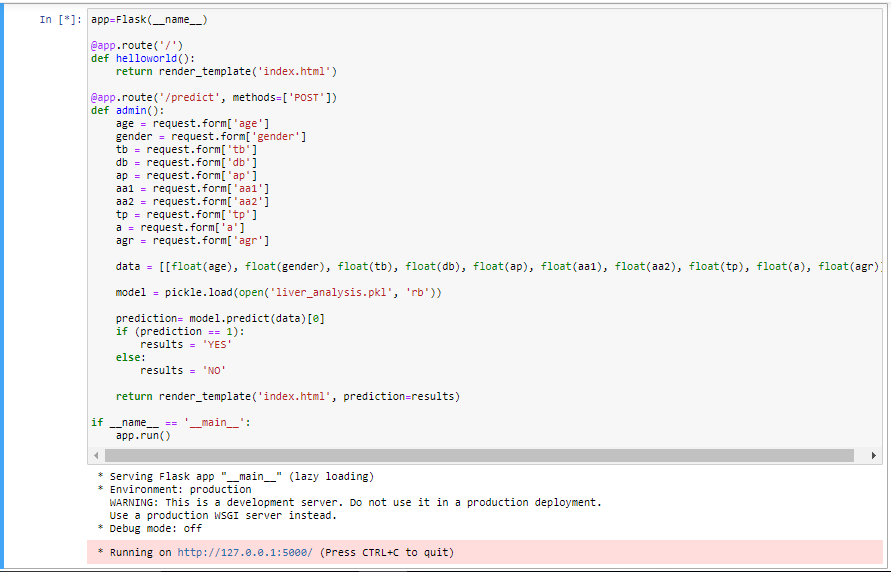
This is used to run the application in a local host.



The local host runs on port number 5000.

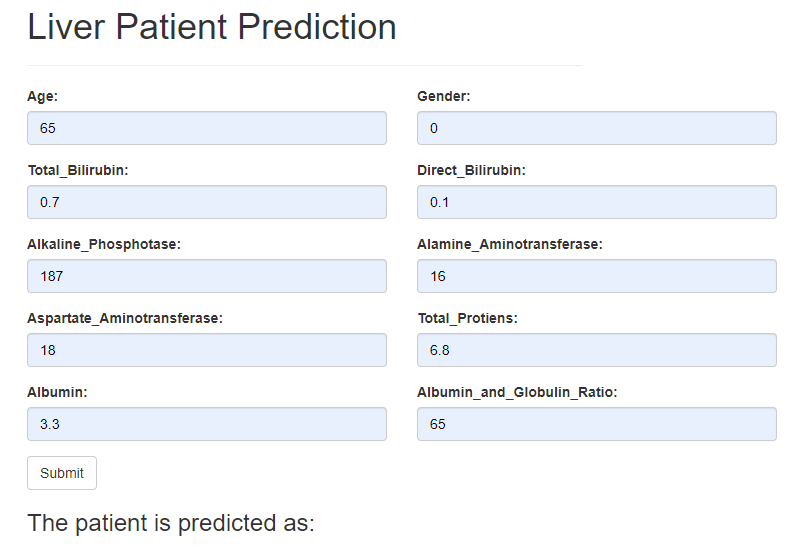
**Activity 4: Run the application**

* Open the anaconda prompt from the start menu.
* Navigate to the folder where your app.py resides.
* Now type “python app.py” command.
* It will show the local host where your app is running on [**http://127.0.0.1.5000/**](http://127.0.0.1.5000/)
* Copy that local host URL and open that URL in the browser. It does navigate me to where you can view your web page.



**Activity 5: Output Screenshot**

Before submit the data:

****

After submit the data:

