FetalAI: USING MACHINE LEARNING TO PREDICT AND MONITOR FETAL HEALTH

**Members:**

Mohammad Anees A A

Christin Davis

Abhishek Madhuraj

Vaishnav Vijayan

**FetalAI: Using Machine Learning to predict and monitor Fetal Health**

Reduction of child mortality is reflected in several of the United Nations' Sustainable Development Goals and is a key indicator of human progress.The UN expects that by 2030, countries end preventable deaths of newborns and children under 5 years of age, with all countries aiming to reduce under 5 mortality to at least as low as 25 per 1,000 live births. Parallel to the notion of child mortality is of course maternal mortality, which accounts for 295 000 deaths during and following pregnancy and childbirth (as of 2017). The vast majority of these deaths (94%) occurred in low-resource settings, and most could have been prevented.In light of what was mentioned above, Cardiotocography (CTGs) are a simple and cost accessible option to assess fetal health, allowing healthcare professionals to take action in order to prevent child and maternal mortality. The equipment itself works by sending ultrasound pulses and reading its response, thus shedding light on fetal heart rate (FHR), fetal movements, uterine contractions and more. In this project, we have some characteristics of Fetal Health as a dataset. The target variable of this dataset is Fetal Health. Since it is a multi class classification, the classes are represented by *‘Normal’*, *‘Pathological’* and *‘Suspect’*.

**Technical Architecture**



**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analyzed by the model which is integrated.
* Once model analyzes the input the prediction is showcased on the UI

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

**1.)Define Problem / Problem Understanding**

* Specify the business problem
* Business requirements
* Literature Survey
* Social or Business Impact.

**2.)Data Collection & Preparation**

* Collect the dataset

**3.)Exploratory Data Analysis**

* Descriptive statistical
* Visual Analysis
* Feature Selection
* Scaling the data
* Checking if the dataset is balanced or not

**4.)Model Building**

* Splitting data into train and test
* Applying SMOTE for balancing the data
* Training the model after applying SMOTE
* Training the model in multiple algorithms
* Testing the model

**5.)Performance Testing**

* Create dataframe of model performance
* Bar plot for model performance

**6.)Model Deployment**

* Save the best model
* Integrate with Web Framework

**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.
* model.pkl is our saved model. Further we will use this model for flask integration.
* Training folder contains a model training file.

**Milestone 1: Define Problem / Problem Understanding**

**Activity 1: Specify the business problem**

Reduction of child mortality is reflected in several of the United Nations' Sustainable Development Goals and is a key indicator of human progress.The UN expects that by 2030, countries end preventable deaths of newborns and children under 5 years of age, with all countries aiming to reduce under‑5 mortality to at least as low as 25 per 1,000 live births.Parallel to notion of child mortality is of course maternal mortality, which accounts for 295 000 deaths during and following pregnancy and childbirth (as of 2017). The vast majority of these deaths (94%) occurred in low-resource settings, and most could have been prevented.In light of what was mentioned above, Cardiotocograms (CTGs) are a simple and cost accessible option to assess fetal health, allowing healthcare professionals to take action in order to prevent child and maternal mortality. The equipment itself works by sending ultrasound pulses and reading its response, thus shedding light on fetal heart rate (FHR), fetal movements, uterine contractions and more.

**Activity 2: Business requirements**

A Fetal Health classification project can have a variety of business requirements in the healthcare sector depending on the specific goals and objectives of the project.

The business requirement for fetal health classification typically arises in the healthcare industry, specifically in the obstetrics and gynecology (OB/GYN) department. The classification of fetal health is necessary to ensure the well-being of the unborn baby and to make informed decisions regarding pregnancy management.

There are several reasons why a healthcare provider may need to classify fetal health. For example, fetal health classification can be used to identify fetuses at risk of preterm labor, intrauterine growth restriction, or other medical conditions that could require early intervention or special care. Additionally, fetal health classification can help healthcare providers monitor the health of the fetus during pregnancy, identify any abnormalities or complications that may arise, and make informed decisions regarding delivery.

In order to classify fetal health, healthcare providers typically use a variety of tools and techniques, including ultrasound, fetal monitoring, and other diagnostic tests. Machine learning and artificial intelligence algorithms can also be used to help classify fetal health based on various parameters such as heart rate, movement, and other physiological measures. These techniques can help healthcare providers to make more accurate and timely diagnosis and treatment decisions, leading to better health outcomes for both the mother and baby.

**Activity 3: Literature Survey**

A literature survey for a Fetal Health 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 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:**

* **Promoting Informed Decision-Making:-** By providing accurate and up-to-date information on Fetal Health, can help expectant parents make informed decisions about their pregnancy and childbirth. For example, if a serious health issue is detected in the fetus, parents can decide whether to continue with the pregnancy or consider other options.
* **Reducing Infant Mortality:-** Access to information about fetal health can help expectant parents identify and treat potential health issues before they become life-threatening to the unborn child. This can help reduce the infant mortality rate and ensure that more babies are born healthy.
* **Improving Prenatal Care:-** When expectant parents know about the health of their fetus, they can work with their healthcare provider to create a tailored prenatal care plan that addresses any issues that may be present. This can lead to better outcomes for both the mother and the child.

**Business Model/Impact:**

* **Improved Patient Outcomes:** By detecting potential health issues in fetuses early, healthcare providers can develop treatment plans that help ensure better outcomes for both the mother and the child. This can lead to improved patient satisfaction and retention rates.
* **Increased Revenue:** Healthcare providers who offer fetal health testing and monitoring services may be able to generate additional revenue streams from expectant parents who are willing to pay for these services. Additionally, if a health issue is detected in the fetus, additional tests, procedures, and treatments may be required, which can generate additional revenue for the healthcare provider.
* **Research and Development:** Knowing about fetal health can also drive research and development in the healthcare industry. For example, new diagnostic tests and treatments can be developed based on data from fetal health monitoring and testing.

**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**

* There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.
* 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/andrewmvd/fetal-health-classification**](https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification)
* As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analyzing 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 might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

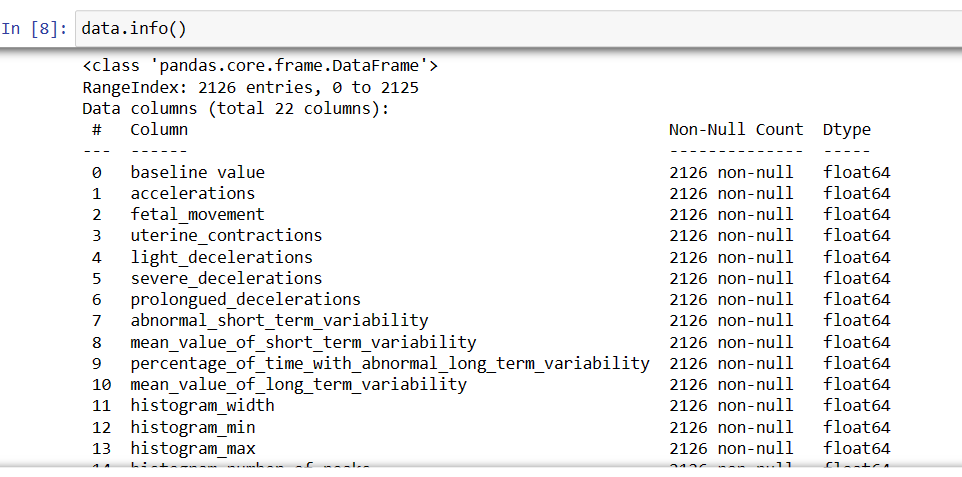
In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



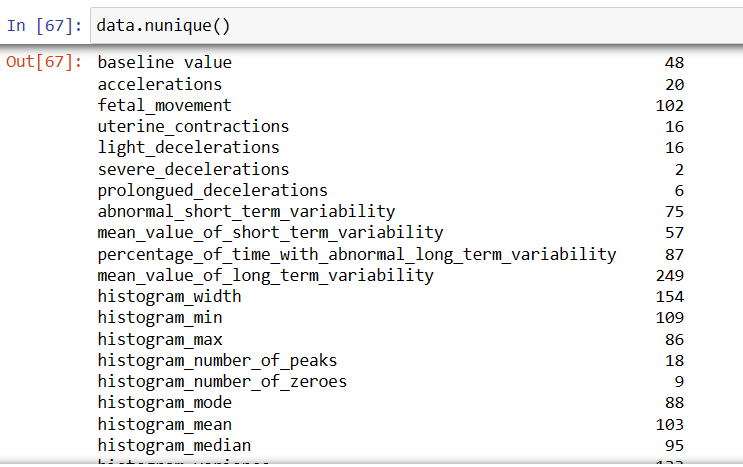
**Milestone 3: Exploratory Data Analysis**

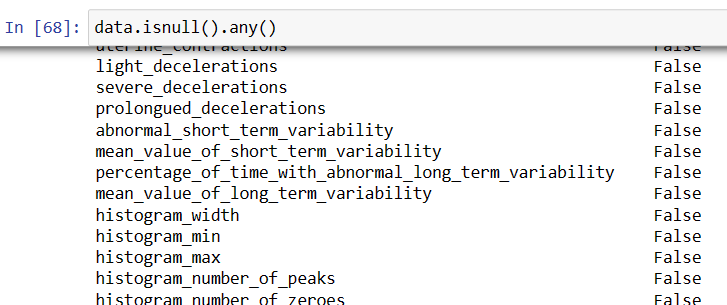
**Activity 1: Descriptive statistical analysis**

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.









**Activity 2: Visual analysis**

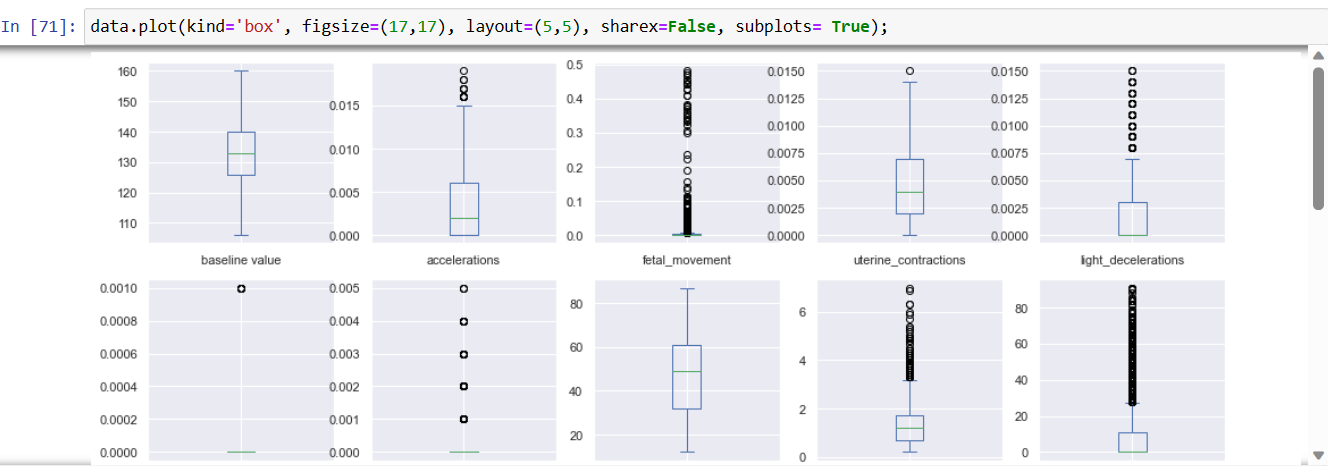
Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

**Activity 2.1: Univariate analysis**

In simple words, univariate analysis is understanding the data with a single feature. Here we have displayed different graphs such as histogram and boxplot.

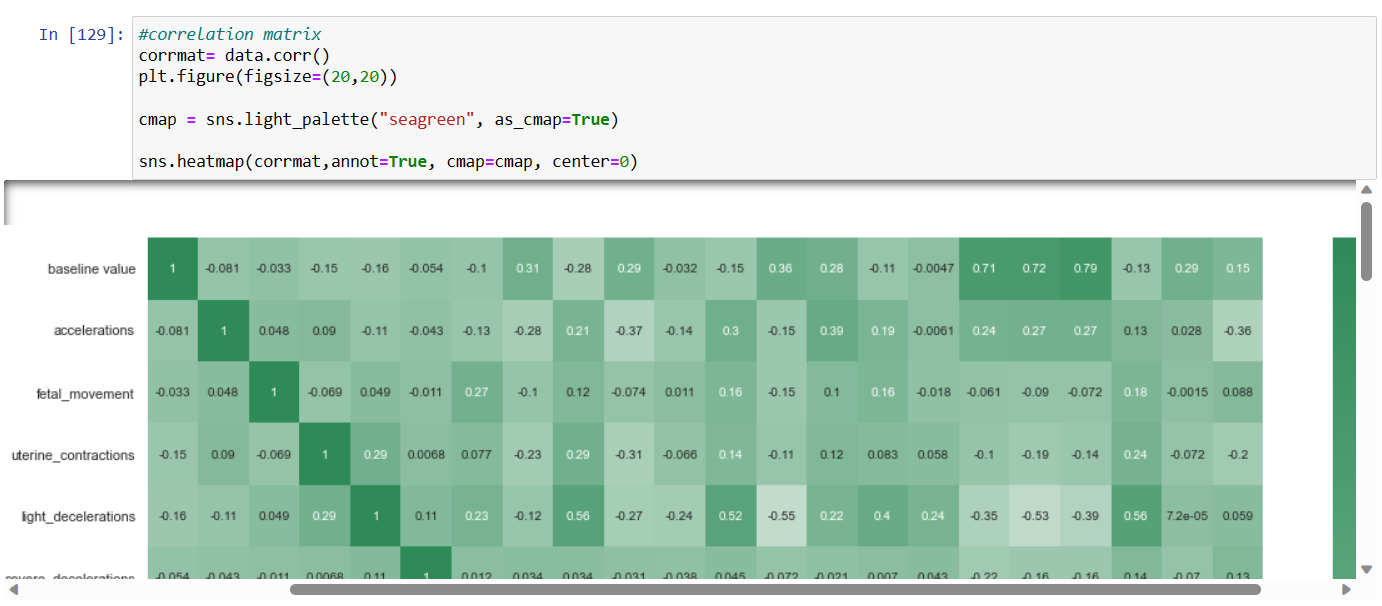
The Seaborn and matplotlib package provides a wonderful functions histogram and boxplot. With the help of histogram and boxplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.



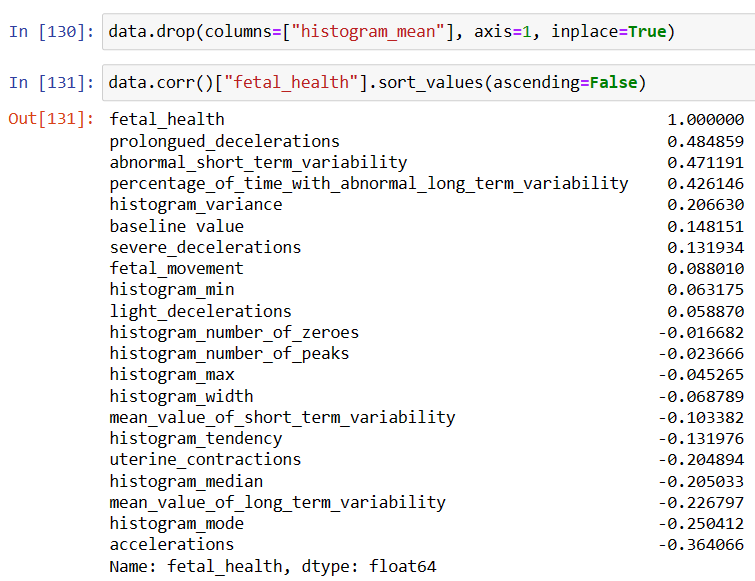


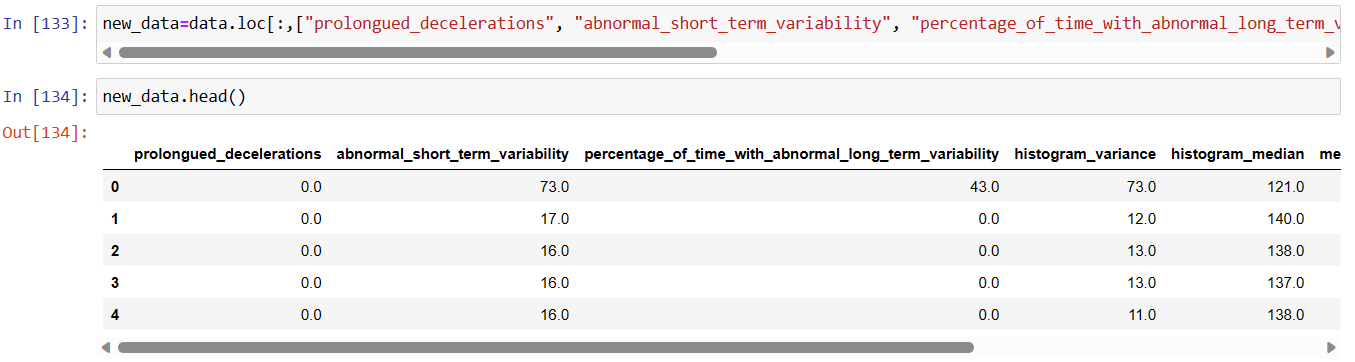
**Activity 2.2: Multivariate analysis**

In simple words, multivariate analysis is to find the relation between multiple features. Here we have used correlation matrix.

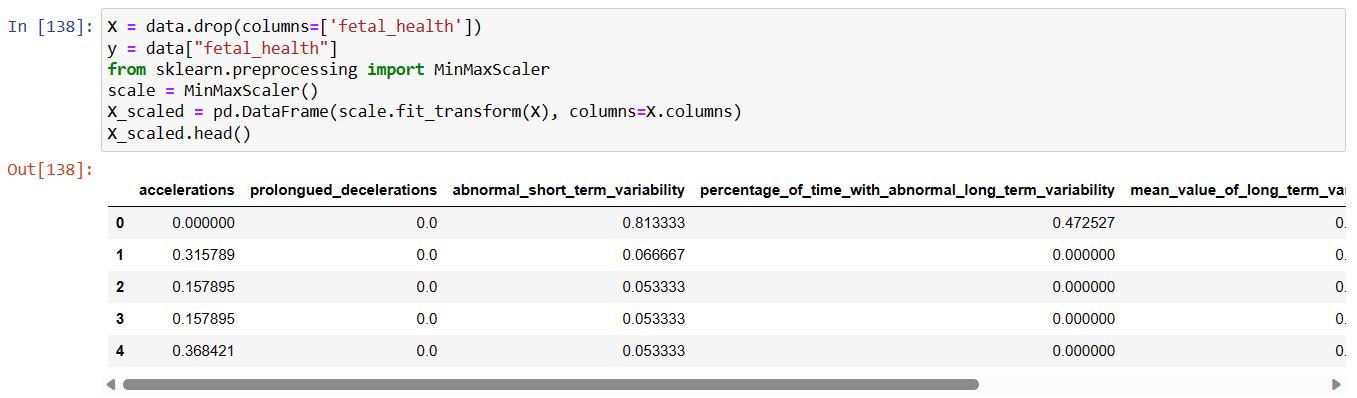


**Activity 3: Feature Selection**

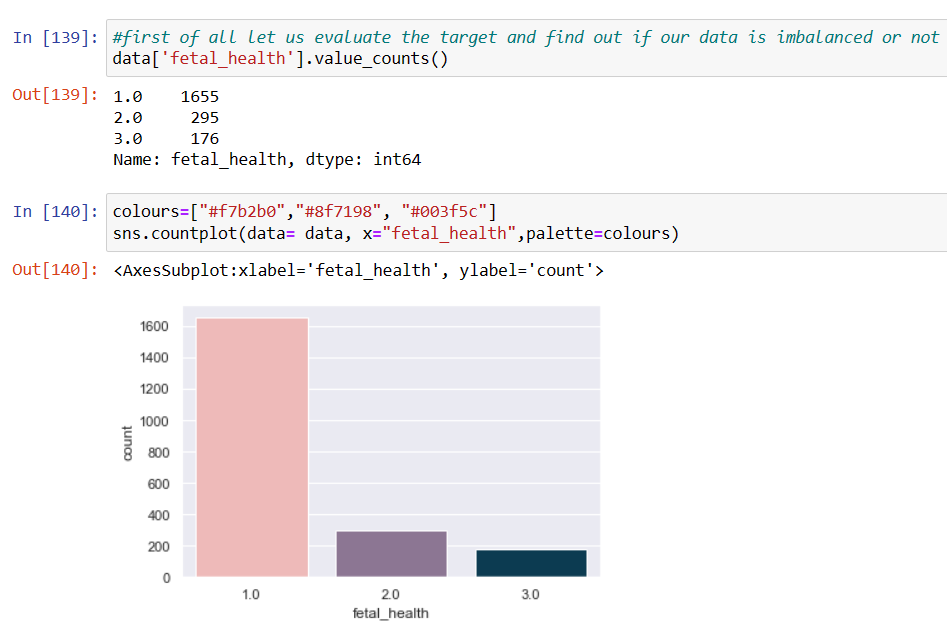




**Activity 4: Scaling the data**



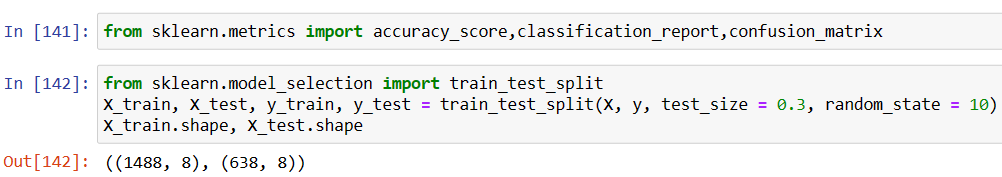
### **Activity 5: Checking if the dataset is balanced or not**



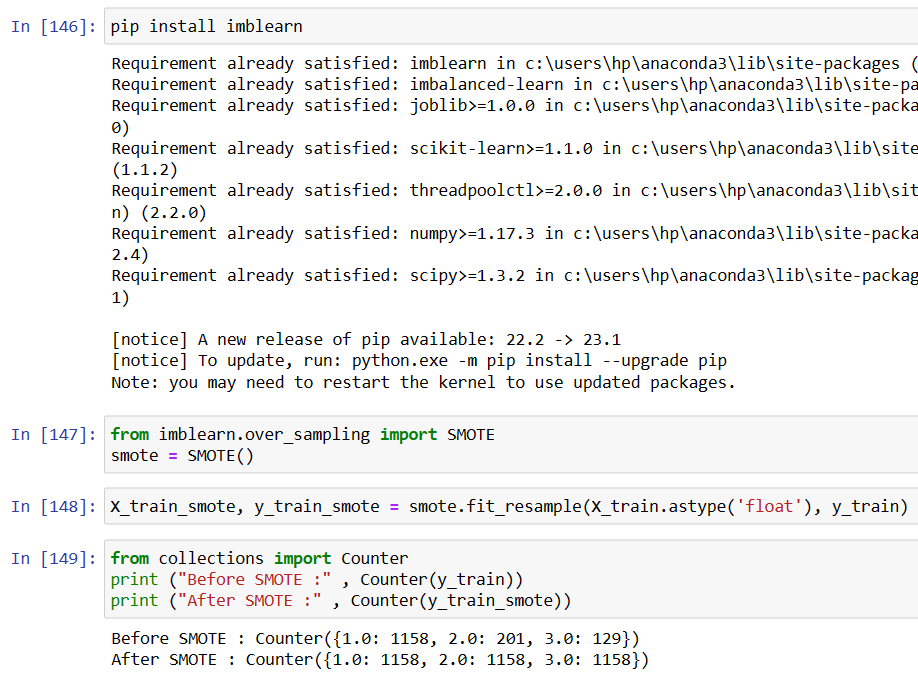
After checking, we get to know that the dataset is highly imbalanced. So in the later stages we have balanced the dataset before training the model.

**Milestone 4: Model Building**

## **Activity 1: Splitting data into train and test**

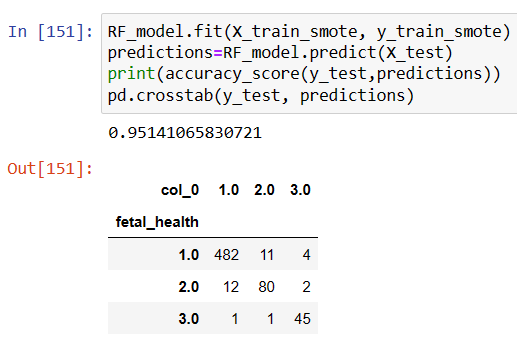


## **Activity 2: Applying SMOTE for balancing the data**



After applying SMOTE, the dataset is balanced. And now we will again train the model after balancing the dataset to check the accuracy.

## **Activity 3: Training the model after applying SMOTE**

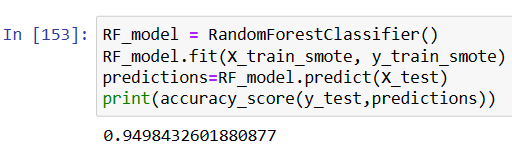


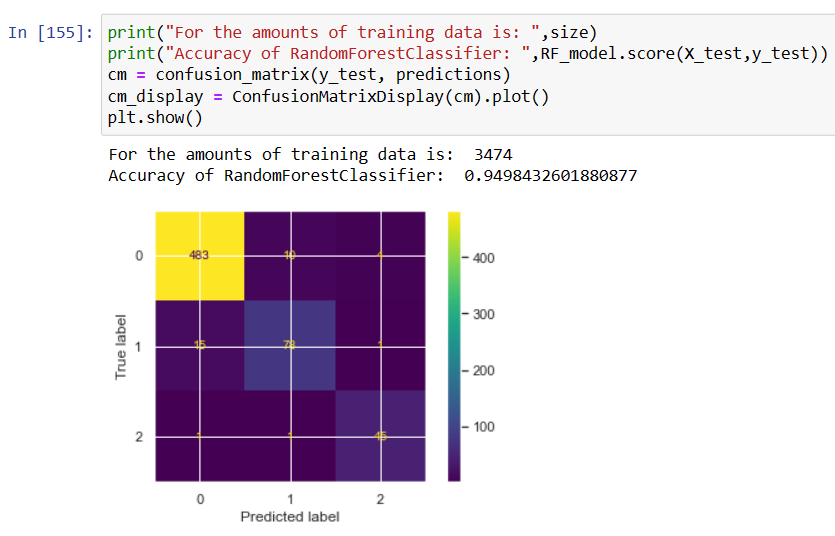
**Activity 4: Training the model in multiple algorithms**

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 4.1: Random forest model**

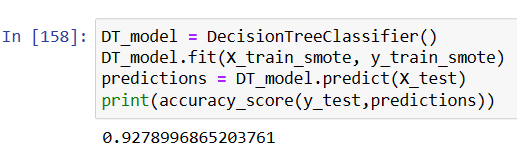
A function named randomForest is created and train and test data are passed as the parameters. Inside the function, the RandomForestClassifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

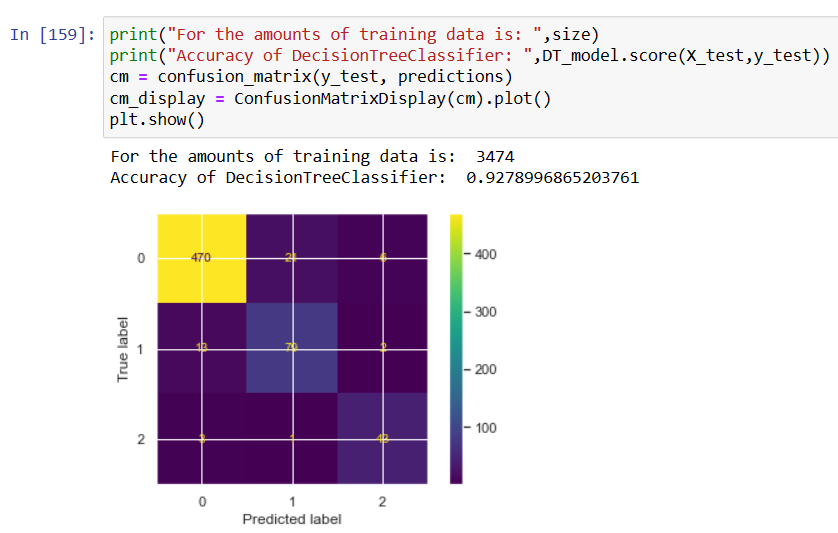




**Activity 4.2: Decision Tree**

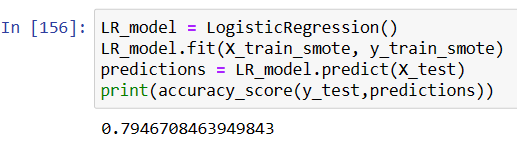
A function named decisionTree is created and train and test data are passed as the parameters. Inside the function, DecisionTreeClassifier algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

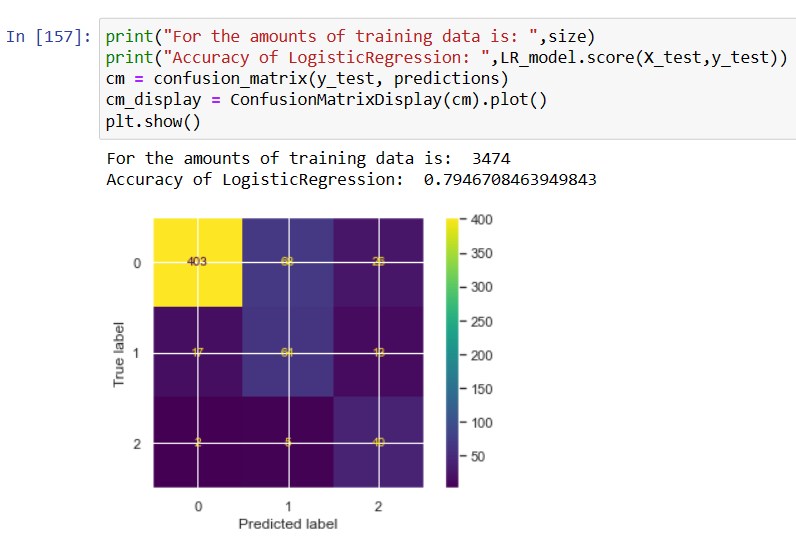




**Activity 4.3: Logistic Regression**

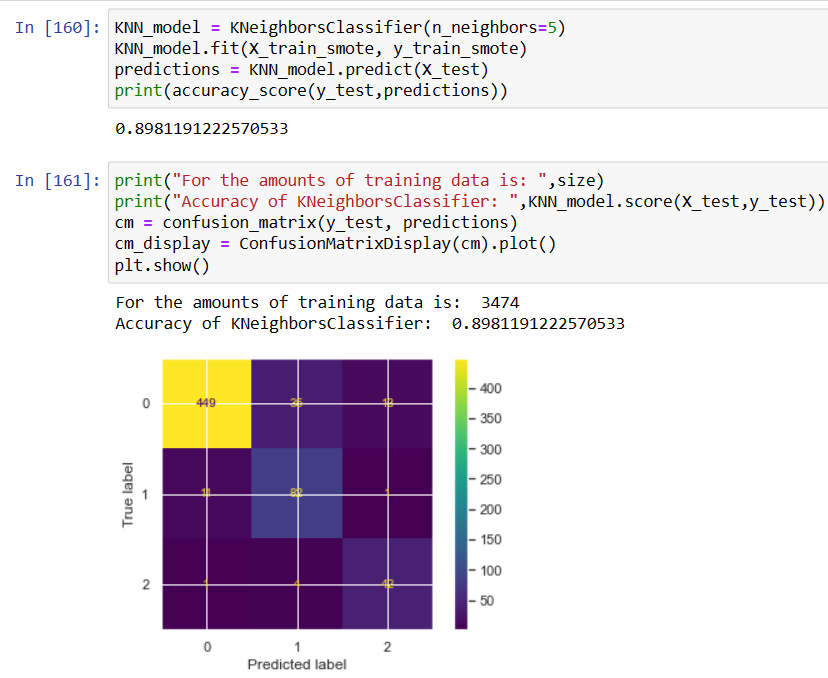
A function named LogisticRegression() is created and train and test data are passed as the parameters. Inside the function, LogisticRegression algorithm is initialized and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.



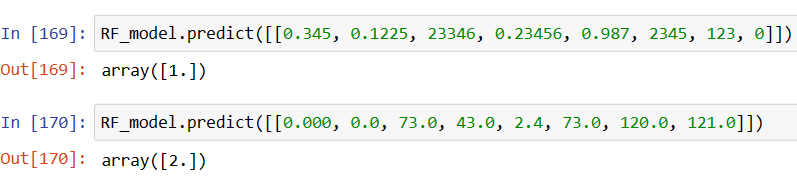


**Activity 4.4: K-Nearest Neighbors**

A function named KNeighborsClassifier() is created and train and test data are passed as the parameters. Inside the function, KNeighbors algorithm is initialized 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. For evaluating the model, a confusion matrix and classification report is done.

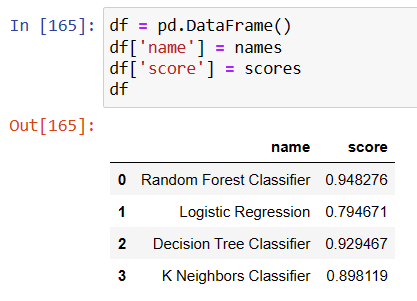


**Activity 5: Testing the model**

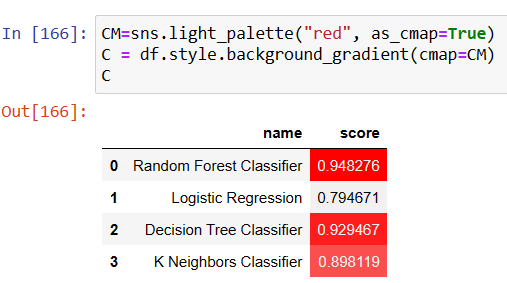


**Milestone 5: Performance Testing**

## **Activity 1: Create dataframe of model performance**



## **Activity 1.1: Adding colors to the dataframe**



**Activity 2: Bar plot for model performance**



After comparing the model with the help of bar plot. We came to a conclusion that Random Forest is showing the highest accuracy and is performing well.

**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 and saving its weights and configuration. 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 Pages**

For this project create three HTML files namely

• index.html

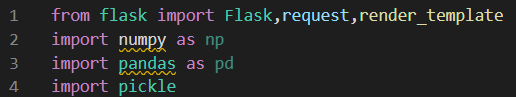
• inspect.html

• outputt.html

and save them in the templates folder.

**Activity 2.2: Build Python code**

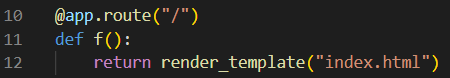
Import the libraries



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.

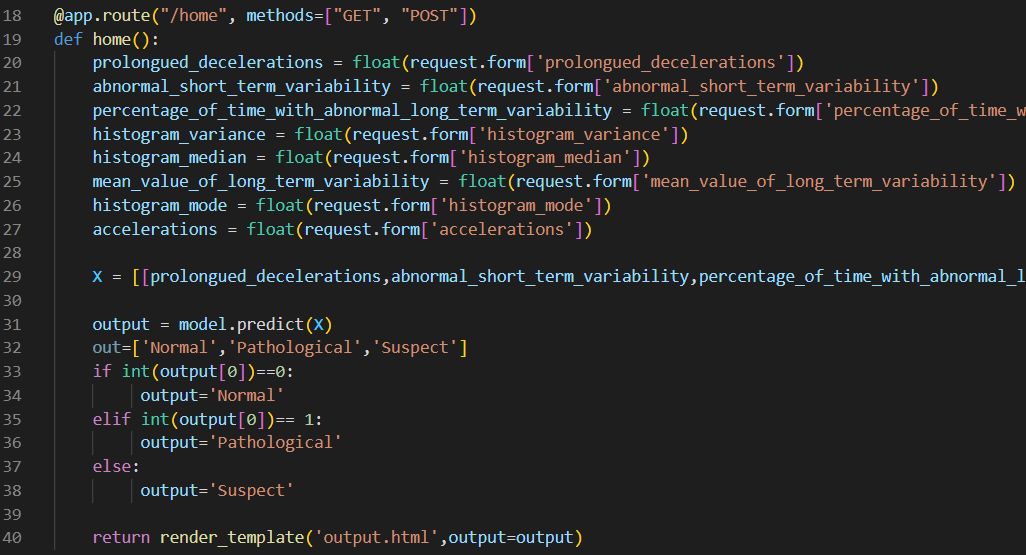


Render HTML page:



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 home.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**

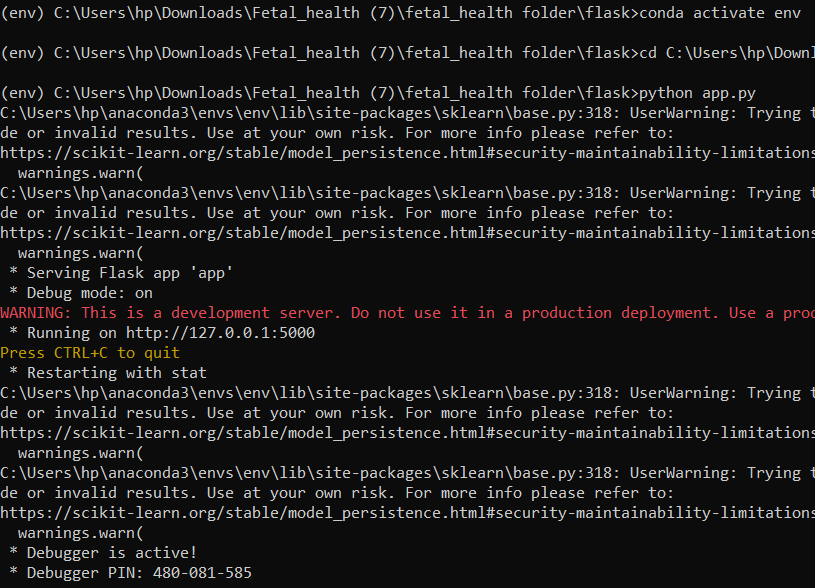
● Open anaconda prompt from the start menu

● 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) to get the below result.



