HEART STROKE PREDICTION USING MACHINE LEARNING

Sai Deeraj. D

Date: 07-03-2022

Abstract:

Globally 1 in 4 adults over the age of 25 will have a stroke in their lifetime. 13.7 million people worldwide will have their first stroke this year and five and a half million will die as a result with statistical evidence stroke is becoming a predominant disease nowadays and the major cause of death as well as the reason for this condition to become severe is the lack of patient being unknown whether the person has this disease or whether it may occur in the future as they say "Prevention is better than cure"

In this report some machine models of different accuracies are built to predict whether a person is likely to get stroke in future. In prediction of heart stroke there are so many determinant factors that make significant contribution towards accuracy of a prediction. One such factor is body mass index, age, gender and many other crucial factors that comes inherent based on a perfect dataset that is prepared based on survey

1. Problem Statement

Providing a methodological and optimized solution with good accuracy, based on the root causes of stroke occurrences along with using the concept of machine learning to train the model accordingly and predict whenever required.

2. Market/Customer/Business Need Assessment

- To convert our prediction system into a business idea we need to first turn our predictive system into a sellable API

- When more of such disease prediction systems our developed from the company side multiple API's can be sold to hospitals.

- So that hospital's will spend appropriate money to buys our API's

- Our scope of the business depends on how much we expand our connections and services throughout the state or country. During COVID-19 there has been an exponential surge in the medical sector in times like these our API's should cost low. Our API should be cheap and reliable until global market reaches stability by this time our business might have made significant branching then we can raise the cost by making it more reliable

3. Target Specification and characterization

- Our main objective is big hospitals where patients come for a regular health checkup and our API’s are an integral part of diagnosing or predicting stroke

- We provide certain free service API calls that can be made to generate patient’s complete report

- Once the limit of free API calls are exceeded we extend our service through subscription. Else we partner with big medical hospitals and start our service by extended our branches

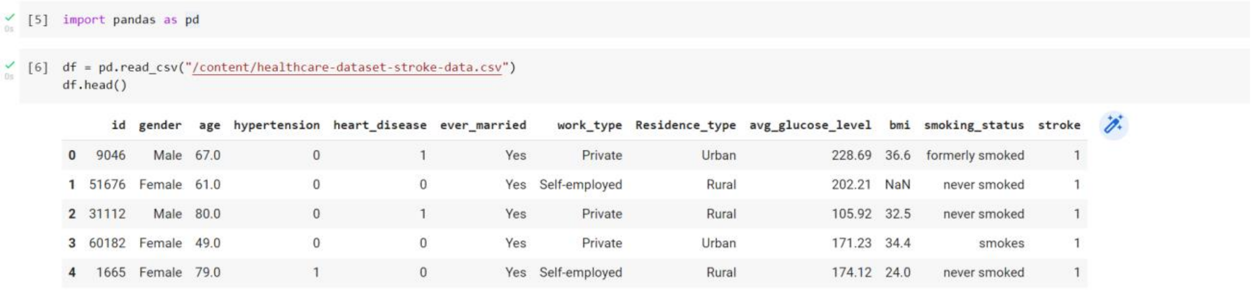
4. External Search(information sources)

Dataset used can be found here.

https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

The dataset shows clinical features that are essential for predicting stroke events, it contains 5111 instances with 11 features some missing values are there in the dataset but they are dealt within code for further processing

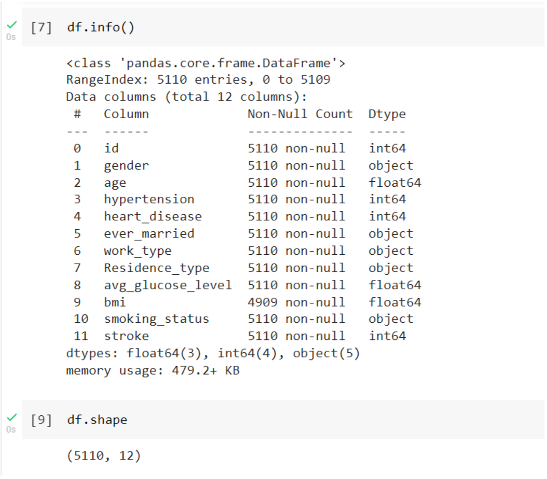
The stroke prediction dataset contains the following entities:



```
[5] import pandas as pd
```

```
[6] df = pd.read_csv("/content/healthcare-dataset-stroke-data.csv")
    df.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1



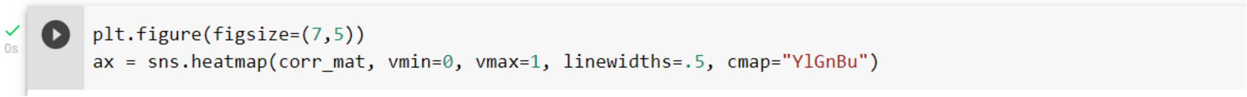
```
[7] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5110 non-null   int64  
 1   gender             5110 non-null   object 
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64  
 4   heart_disease      5110 non-null   int64  
 5   ever_married       5110 non-null   object 
 6   work_type          5110 non-null   object 
 7   Residence_type     5110 non-null   object 
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object 
 11  stroke             5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

```
[9] df.shape
```

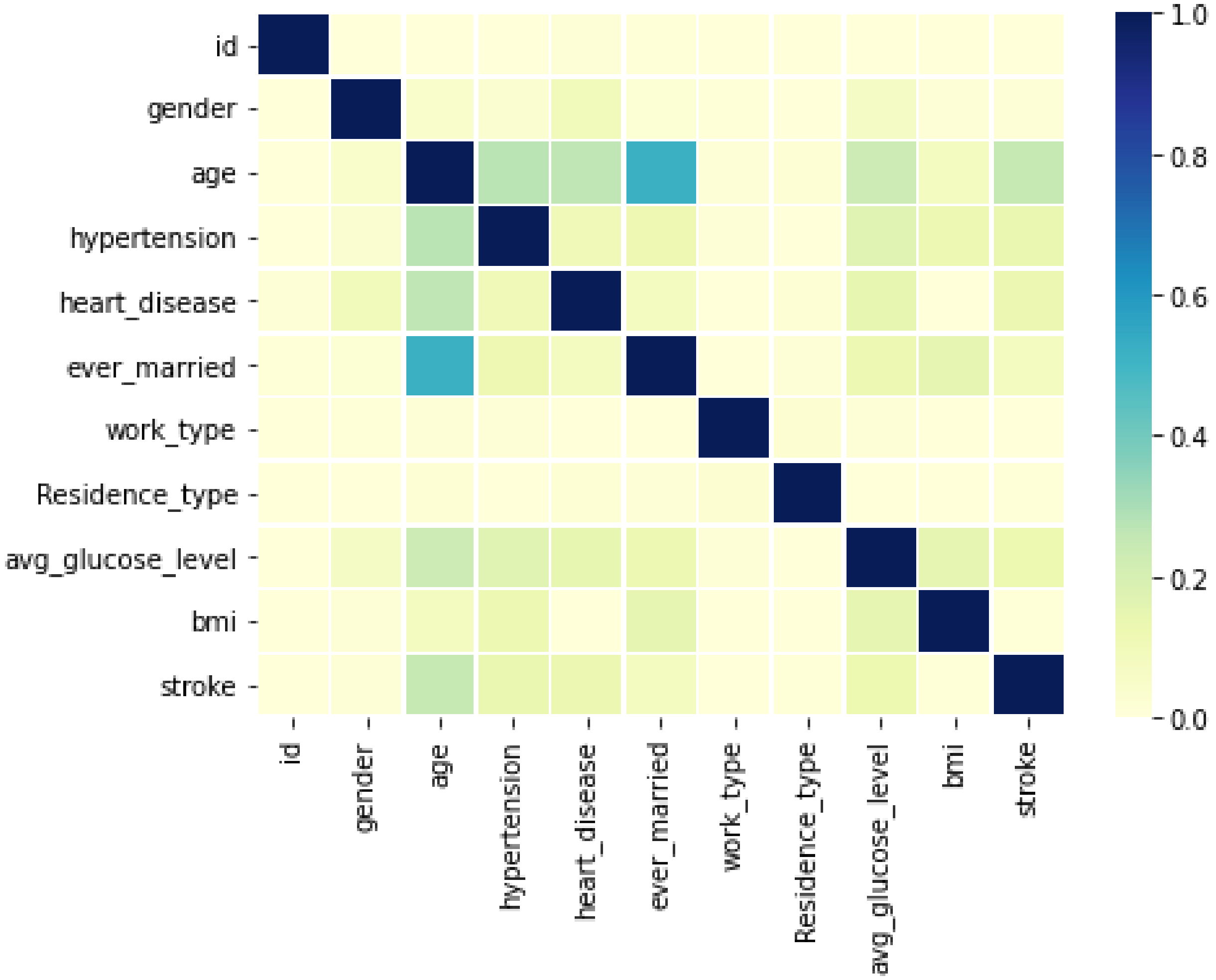
```
(5110, 12)
```

5. Benchmarking

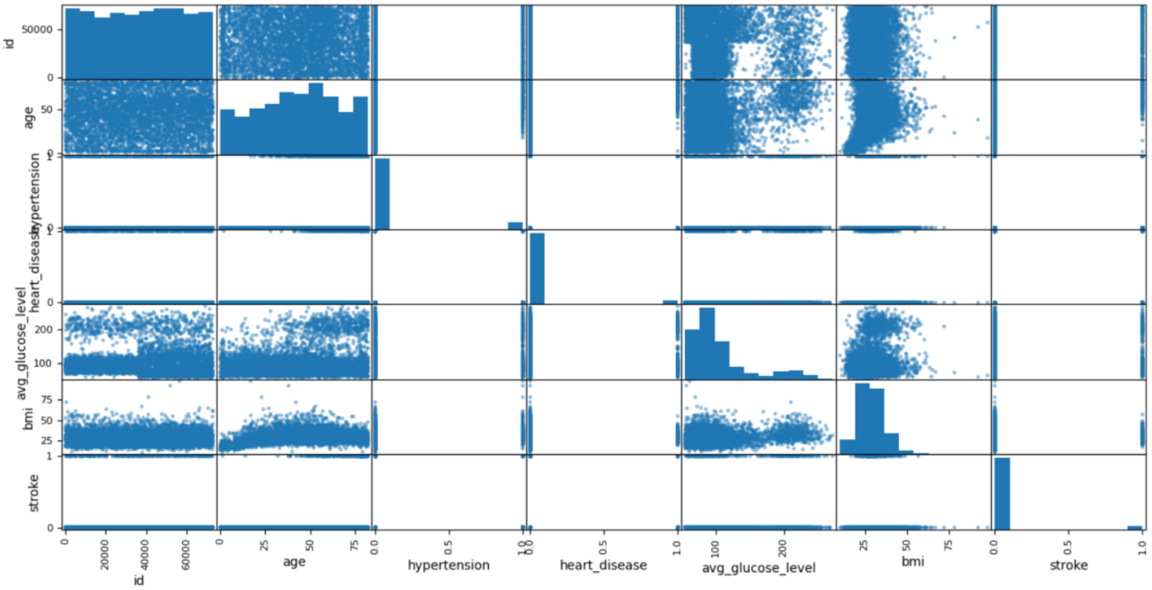


```
plt.figure(figsize=(7,5))
ax = sns.heatmap(corr_mat, vmin=0, vmax=1, linewidths=.5, cmap="YlGnBu")
```

Correlation matrix of the data:

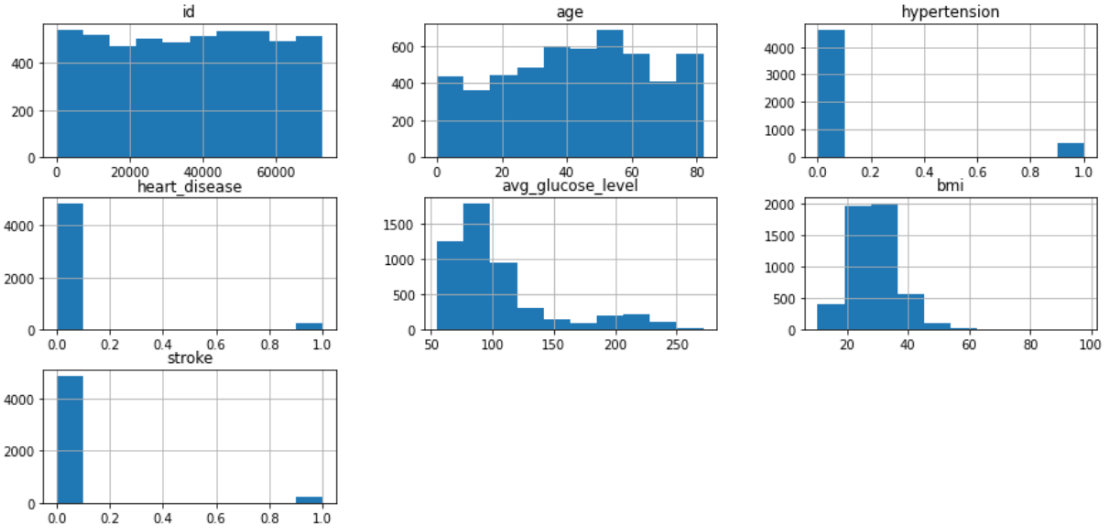


Scatter plot



The above data gives us an idea on how shows the relationship between two quantitative variables measured for the same individuals as well as that with different instances. Each individual in the data appears as a point on the graph. Each parameters scatter plot density shows how much each of them are related

Histogram visualization of parameters associated :



6. Applicable Patents

An international journal published on national institute of health website named: Cardiovascular/stroke risk predictive calculators: a comparison between statistical and machine learning models - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7487379/ - the journal mentioned is related to our study but very different from what we have implemented in our report

7. Applicable Regulations

The patents mentioned above might claim the technology used if the algorithms are not developed and optimized individually and for our requirements. Using a pre-existing model is off the table if it incurs a patent claim.

- Must provide access to the 3rd party websites to audit and monitor the authenticity and behavior of the service.
- Enabling open-source, academic and research community to audit the Algorithms and research on the efficacy of the product.
- Laws controlling data collection : Some hospitals might have a policy that data should be collected with patients consent and it should not be exposed publicly.

8. Applicable Constraints:

- The use of cloud platforms to store the data gathered within the hospital
- Continuous data collection and maintenance
- Lack of technical knowledge by the user

9. Business Opportunity

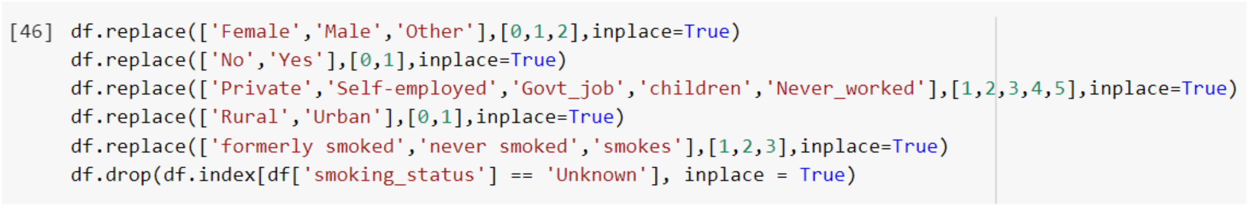
- There is a huge scope for this business to enter anytime at anyplace into the medical sector easily and grow partners and profits within a short amount of time
- For diagnosing whether a patient is having a risk of stroke in near future takes lot of analysis
  and time to conclude that whether he/she may or may not have stroke but in most cases as we
  see today are sudden attacks which may lead to death instantaneous or may lead to severity.
- But if a person is pre checked automatically whenever he/she visits hospital then there is a good chance that this disease may be suppressed

- As a business companies of this sort are always welcomed in medical sector but we need to partner with big companies to grow our business
- More importantly if our API's are working accurately and there is no need for doctors diagnosis and if death rates are dropping then with well awareness for our business as a startup will attract investors to invest in.

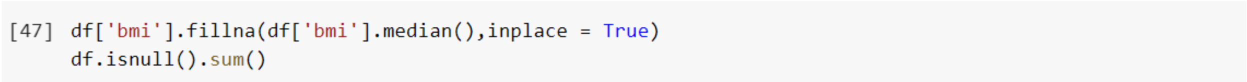
10. Concept Generation

This product requires the tool of machine learning models to be written from scratch in order to suit our needs. . Tweaking these models for our use is less daunting than coding it up from scratch. A well trained model can either be repurposed or built. But building a model with the resources and data we have is dilatory but possible. The customer might want to spend the least amount of time giving input data. . This accuracy will take a little effort to nail, because it's imprudent to rely purely on Classic Machine Learning algorithm .

Step1: Data cleaning remove - all unwanted values assign numeric values to string datatypes this helps in classification then for parameters that have missing values assign them with zero this method does not affect accuracy much.

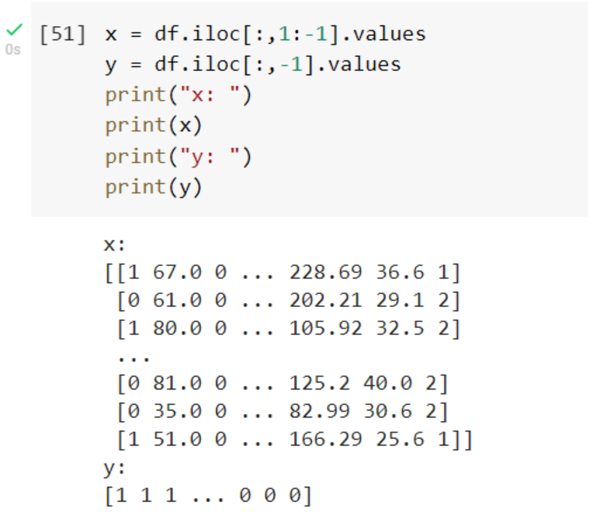


```
[46] df.replace(['Female','Male','Other'],[0,1,2],inplace=True)
     df.replace(['No','Yes'],[0,1],inplace=True)
     df.replace(['Private','Self-employed','Govt_job','children','Never_worked'],[1,2,3,4,5],inplace=True)
     df.replace(['Rural','Urban'],[0,1],inplace=True)
     df.replace(['formerly smoked','never smoked','smokes'],[1,2,3],inplace=True)
     df.drop(df.index[df['smoking_status'] == 'Unknown'], inplace = True)
```



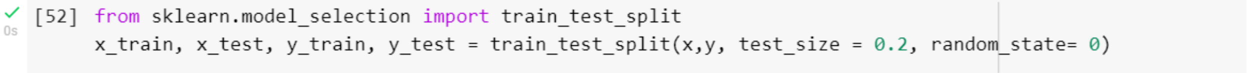
```
[47] df['bmi'].fillna(df['bmi'].median(),inplace = True)
     df.isnull().sum()
```

Step2: Divide inputs and outputs assign them to variables such as x and y



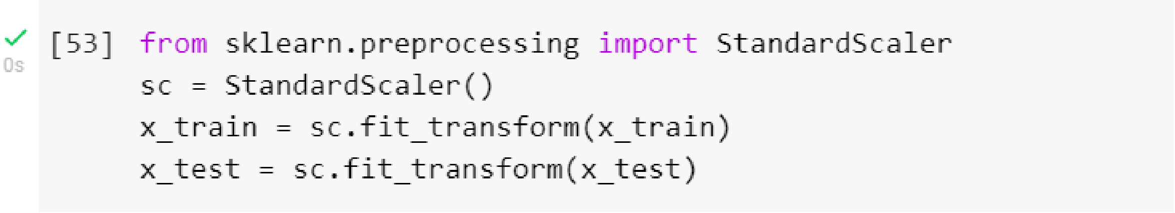
```
✓   [51] x = df.iloc[:,1:-1].values
0s        y = df.iloc[:,-1].values
          print("x: ")
          print(x)
          print("y: ")
          print(y)

          x:
          [[1 67.0 0 ... 228.69 36.6 1]
           [0 61.0 0 ... 202.21 29.1 2]
           [1 80.0 0 ... 105.92 32.5 2]
           ...
           [0 81.0 0 ... 125.2 40.0 2]
           [0 35.0 0 ... 82.99 30.6 2]
           [1 51.0 0 ... 166.29 25.6 1]]
          y:
          [1 1 1 ... 0 0 0]
```

Step3: Now split the data into training set(contains inputs and outputs from the dataset to train the model)
and testing set(contains inputs and outputs from the dataset to test the model) I am doing them using
80% of the dataset for training and remaining 20% of the dataset for testing



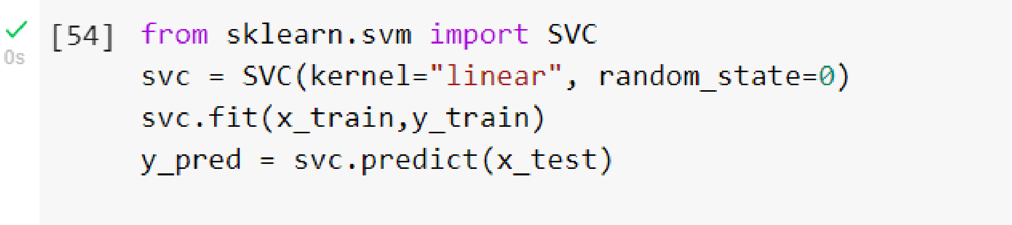
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2, random_state= 0)
```

Step4: Perform standardization for increased accuracy



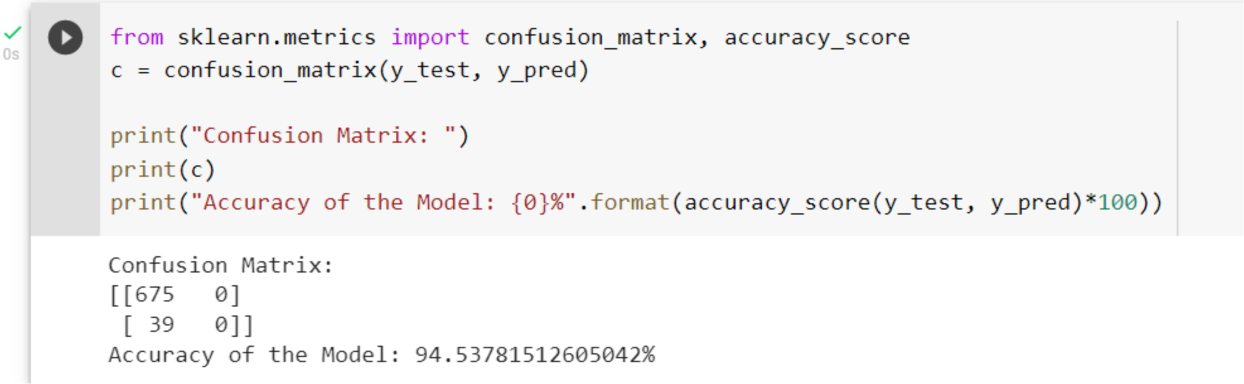
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)
```

Step5: Build model - I have used **Support vector classification model** with kernel as linear classifier



```
from sklearn.svm import SVC
svc = SVC(kernel="linear", random_state=0)
svc.fit(x_train,y_train)
y_pred = svc.predict(x_test)
```

The Accuracy of the initial model is given below:



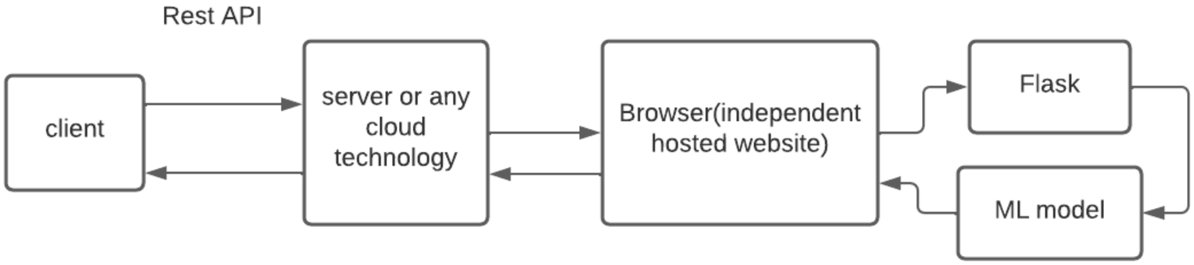
```
from sklearn.metrics import confusion_matrix, accuracy_score
c = confusion_matrix(y_test, y_pred)

print("Confusion Matrix: ")
print(c)
print("Accuracy of the Model: {0}%".format(accuracy_score(y_test, y_pred)*100))
```

```
Confusion Matrix:
[[675   0]
 [ 39   0]]
Accuracy of the Model: 94.53781512605042%
```

Produces 94% accuracy

11. Concept Development

The concept can be developed by first building a website for the business.
1. Importing libraries
2. Load the machine learning model
3. Build functions to preprocess and to predict whether the person will be having a stroke in future
4. Initialize the flask object
5. Set the route and the function that returns something to the user’s browser
6. Run and test the API
For running the model enter data of the patient or load the dataset of patient’s record
For each invoke of the API keep a count and charge accordingly


Rest API
client
server or any cloud technology
Browser(independent hosted website)
Flask
ML model

12. Final Report Prototype

The product takes the following functions to perfect and provide a good result.

Back-end

Model Development: This must be done before releasing the service. A lot of manual supervisedmachine learning must be performed to optimize the automated tasks.

- Performing EDA to realize the dependent and independent features.
- Algorithm training and optimization must be done to minimize overfitting of the model and hyperparameter tuning.

Webapp development: For releasing the service a platform is required and that is an interactive webpage and all the functionalities that are required to make a website responsive must be done in the backend. Building a secure payment gateway, counting the number of api calls made by an organization etc must be recorded and implemented.

Front End

Different user interface: The user must be given many options to choose form in terms ofparameters. This can only be optimized after a lot of testing and analysis all the edge cases.
Interactive visualization the data extracted from the trained models will return raw and inscrutable data. This must be present in an aesthetic and an “easy to read” style.
Feedback system: A valuable feedback system must be developed to understand the customer’s needs that have not been met. This will help us train the models constantly.
Attractive and responsive web pages: website should be made attractable with features such as API description, what it does and supporting accuracies with terms and conditions must be built.

13. Product details - How does it work?

A webpage is designed which shows description about our api plans and an option for free trail if free trail is pressed login details or sign up is popped. Once registered if want to try free trial then a link will be generated with prompt to enter patient’s details with different dropdown menus for selection if not a medical dataset can be uploaded to find results once done an excel sheet is exported with the results. This process is referred to as an api call and keeps on going till free trial is expired or api calls are exceeded. Now purchase option is popped up for the user or organization once purchased the link is provided with unlimited API calls and each time a call is made
it travels through a data pipeline to process the data and again export back the result.

14. Conclusion:

An ml model is built to predict whether a person will have heart stroke in the future and the model was found to produce 94% accuracy. This model is used as a supportive with doctors to confirm their hypothesis and produce faster results. With integration of this api to health reports the patient can be alarmed before stroke hits the patient unexpectedly thus reducing death rate caused by heartstroke

15. References/Source of Information:

https://www.kaggle.com/fedesoriano/stroke-prediction-dataset
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7487379/
https://towardsdatascience.com/deploying-a-heart-failure-prediction-model-using-flask-and-heroku-55fdf51ee18e