# I. Introduction

The main idea of the project is to show which among the K-NN, SVM, and Neural Networks is better classifier in the machine learning set and which performs the best on which type of data. The objective is to predict whether a patient has diabetes or not using these three Machine Learning models by factoring in independent variables such as Cholesterol, Smoking Habits, Sex, BMI, Income, Age etc.

The dataset we are planning to use for this project is a dataset available on Kaggle.com and is originally from The Behavioral Risk Factor Surveillance System (BRFSS) [1]. BRFSS is a health-related telephone survey that is collected annually by the CDC. Each year, the survey collects responses from over 400,000 Americans on health-related risk behaviors, chronic health conditions, and the use of preventative services. It has been conducted every year since 1984. Although a csv of the original dataset containing 441,455 records was available on Kaggle from the year 2015. We are choosing to go with this particular dataset because it has fewer missing values and it contains only the related 22 features as compared to the other one which has a lot of irrelevant fields. These features are either questions directly answered by the participants, or calculated binary values based on their responses. For example - we would identify a smoker with the value 1 and a non-smoker with the value 0 in our dataset. Here’s the [link](https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset) to the dataset.

Second dataset [2] we are planning to use is originally from the National Institute of Diabetes and Digestive and Kidney Diseases and contains about 769 records of women 21 years and older and has the following risk factors as its features – Glucose, Pregnancies, Blood Pressure, Skin Thickness, Insulin, BMI, Age, and Diabetes pedigree function. We believe that upon comparison of results of this dataset with the former dataset, we would get important insights into which Machine Learning Algorithm works better on the two different databases. We will also be able to draw a better correlation between different features and will be able to determine what features or risk factors are more vital in accurately determining the outcome that is whether the user is at risk of Diabetes or not. Here’s the [link](https://www.kaggle.com/datasets/mathchi/diabetes-data-set) to the second dataset.

An important distinction between these two datasets is that the second dataset consists of primarily physical health markers such as Insulin levels, BMI, glucose levels etc., whereas the first dataset also takes into account, the user’s complex lifestyle and choices such as their fruits and veggies consumption, smoking and drinking habits, even their education and income levels. This could provide us with a valuable insight in determining the impact a person’s lifestyle and their day-to-day routines have with regard to this particular disease and up to what extent.

We chose this particular topic because Diabetes is amongst the most common chronic diseases not just here in the US, but the whole world impacting millions of people each year and putting a significant burden on the health systems throughout. The Centers for Disease Control and Prevention (CDC) [3] has estimated that as of this year, 37.3 million Americans have diabetes and of the 96 million approximate adults in the country – more than 1 in 3 have prediabetes. What’s more alarming is that, more than 8 in 10 of them don’t know they have it. Diabetes is also prevalent in our home country of India, where about 77-Million people are currently dealing with it making it the second most affected in the world after China [4], which is also why this subject hit really close to our hearts.

We all know somebody who’s struggling with diabetes, and we know that this could have been avoided if only the person was aware of it, and decided to act on it. We know that it’s a bit impractical to expect people, to make time out of their busy schedules, and visit a doctor for a regular check-up, especially if there are no symptoms yet. But it’s fairly reasonable to expect them to fill a short 5-minute survey about their health, and use the magic of the Machine Learning algorithms to predict whether they are at a high risk of a disease, that is currently the seventh leading cause of death in this country. We could convey our findings to them, which can literally save their life. We do understand that this is not an alternate for a visit to the Doctor, but given the ease of use, and cost being almost zero, this could be the next best thing. We can also use ML to show correlation between different features, such as correlation between Alcohol consumption and High Cholesterol, which could help users identify critical causes and make better life choices.

Type 2 diabetes is a predictable and preventable disease because it usually develops later in life as a result of lifestyle (example- low physical activity, obesity status) and other (example- age, sex, race, family history) risk factors. So, keeping that in mind, we have selected a database that covers a wide range of important risk factors as features, for diabetes and other chronic illnesses, which are - blood pressure, cholesterol, smoking, obesity, age, sex, race, diet, exercise, alcohol consumption, BMI, Household Income, Marital Status, Sleep, Time since last check-up, Education, Health care coverage, Mental Health

The US Preventive Services Task Force (USPSTF) [5] states with certainty that screening for prediabetes and type 2 diabetes and referring patients with prediabetes to effective preventive interventions has a moderate net benefit. Lifestyle interventions that focus on diet, physical activity, or both as well as interventions through prescription drugs such as metformin have demonstrated great efficacy in preventing or delaying progression to diabetes in people with prediabetes. Our project does not aim to replace a screening by a medical professional, but aims to act as supporting tool, to guide the people who are most vulnerable to it, in a healthier direction.

In our project we aim to take a deep dive into the following questions:

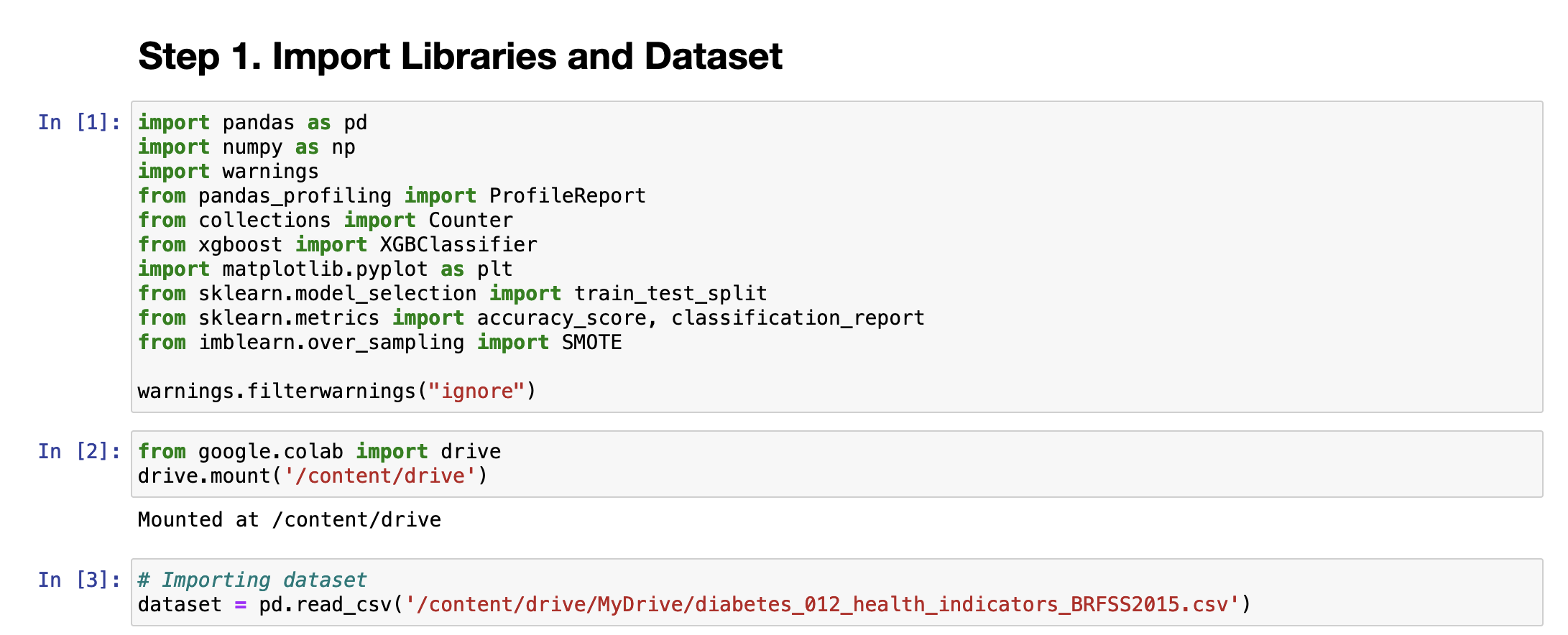
1. Can we accurately predict whether an individual has diabetes using K-NN, SVM, and Neural Networks model?
2. Which of these 3 models is most accurate in its prediction for the two given datasets?
3. How to use python libraries to visualize which factors are closely correlated to each other?
4. Can we use just a subset of the features, thereby making the survey shorter, and still accurately predict whether an individual has diabetes?

# II. Overview

For this project we have used two sets of databases and we are going to perform each ML algorithm on both of these sets. Going forward we are going to refer to The Behavioral Risk Factor Surveillance System (BRFSS) dataset as ‘Dataset I’ and the other dataset from National Institute of Diabetes and Digestive and Kidney Diseases as ‘Dataset II’.

# III. Importing Libraries and Dataset

Dataset I



Dataset II

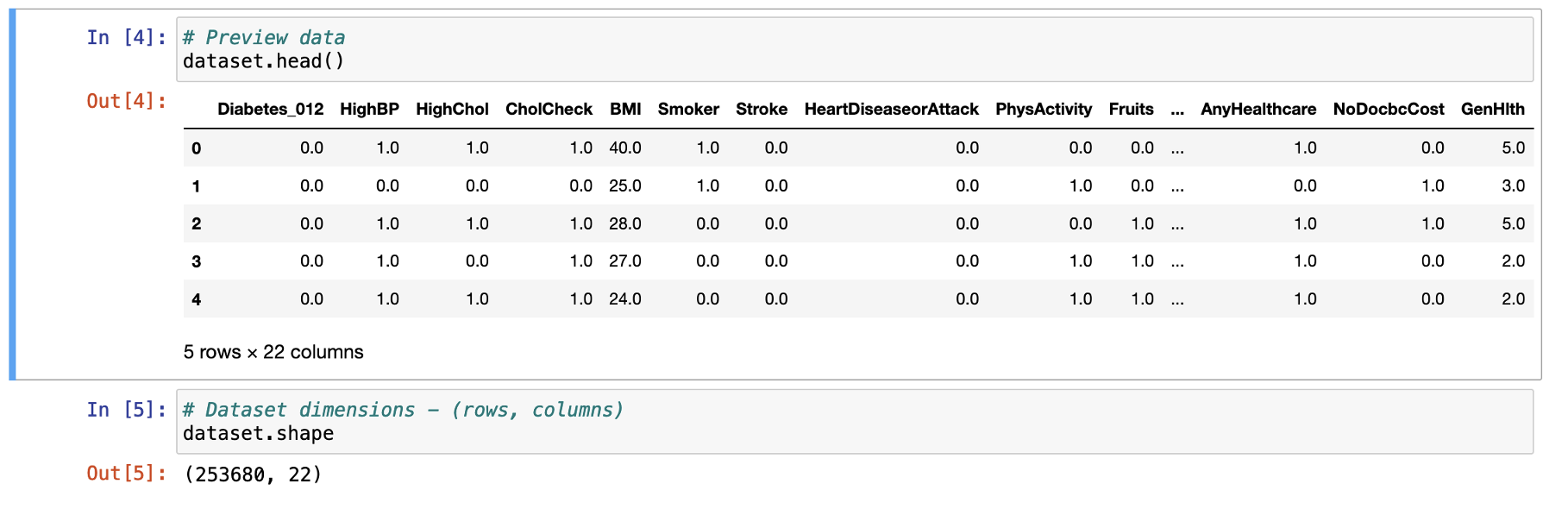


The very first step would be to import all the relevant libraries that we are going to use for this project followed by the dataset itself. Followed by that, we have imported our dataset using read\_csv function of the pandas library. Here’s a quick brief on all the python libraries we have used so far.

* Pandas - Pandas is short for “Python Data Analysis Library”. Pandas is the most commonly used library for data manipulation and analysis because it offers data structures and operations for manipulating numerical tables.
* Numpy – Numpy stands for ‘Numerical Python’ and it’s a powerful library used for working with multi-dimensional arrays, linear algebra, and matrices.
* Matplotlib – Matplotlib is an extension of pyplot library and is used for creating static, animated, and interactive visualizations with ease.
* Seaborn – Seaborn is a data visualisation library based on Matplotlib and is used to informative graphs and enhance visualizations.
* Warnings - Warnings is an extremely useful library as it notifies the user of some unusual activity in a program, where that particular condition doesn’t warrant raising an exception or terminating the program.
* ProfileReport – It’s a part of the pandas\_profiling library and its main utility is generating reports from the input dataframe.
* Counter - The Counter is a dict subclass that holds the data in an unordered collection, like hashtable objects.
* XGBClassifier – It’s a scikit-learn API compatible class for classification and is part of the XGBoost library which stands for eXtreme Gradient Boosting.
* Train\_test\_split – This is from the Sklearn model selection library and it is used to split the data into two parts, one for training and the other for testing.
* Accuracy\_score – As the name suggests, this helps us figure out the accuracy score of our data model.
* Classification\_report – It creates a text report displaying the main classification metrics.
* Smote - SMOTE stands for “Synthetic Minority Oversampling Technique” and is used for resampling.

# IV. Data Description

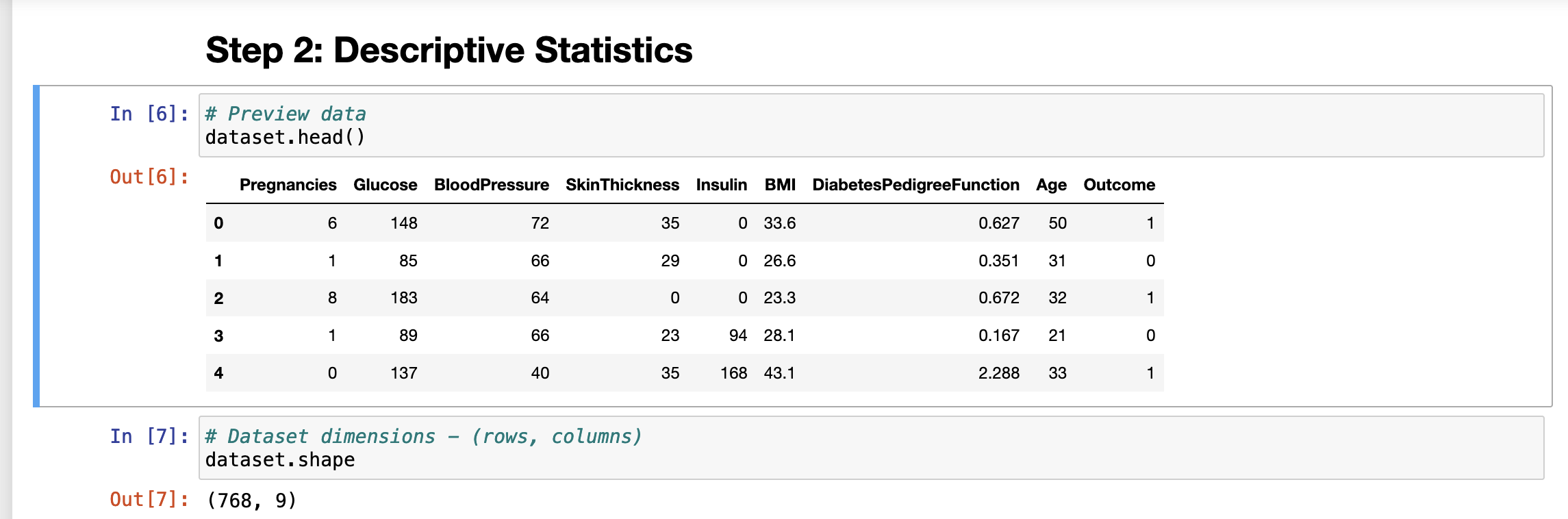
Dataset I

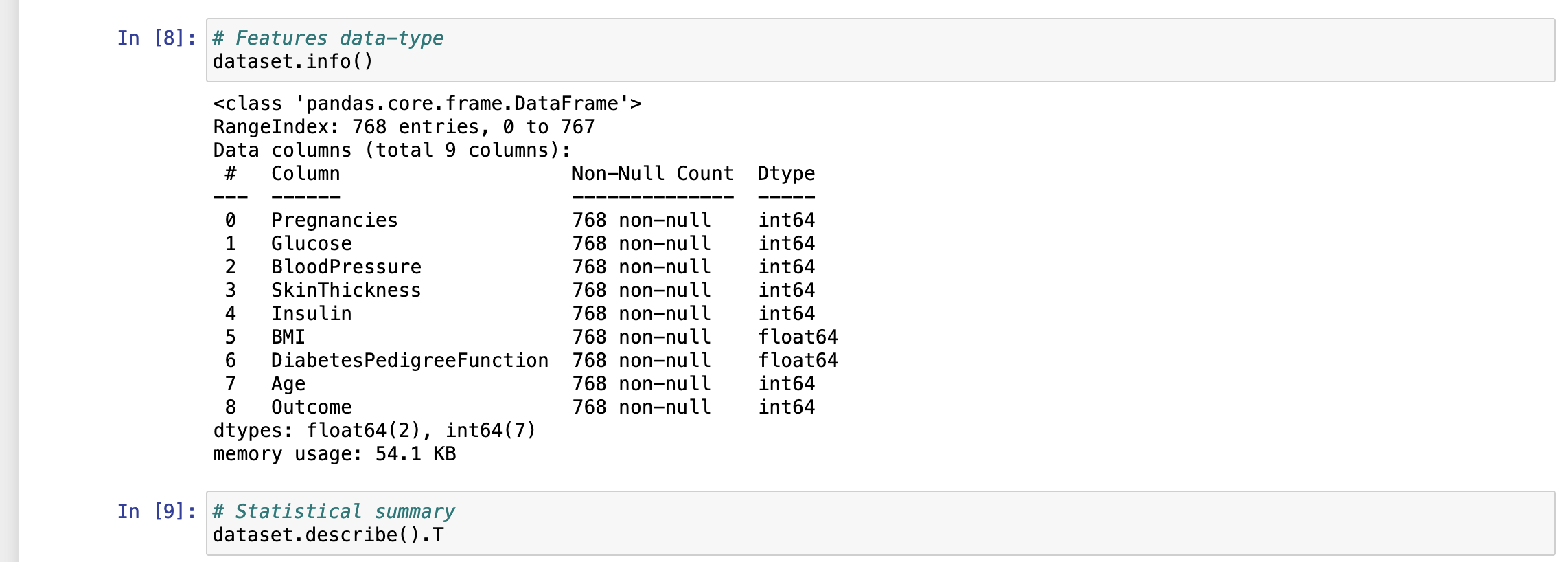






Dataset II







Head function returns top n rows of a DataFrame where n is a user input value. If its left blank, it assumes the default value to be 5. It’s a good way to have get an estimate of how your data looks like, since you are able to see all the columns and what kind of values they contain. Shape function returns the dimension of your array, or your dataset that is the number of rows and columns. In our case it returns (253680,22) for database I and (768, 9) for database II. For the sake of being able to compute our models in time, we have chopped off our database I to just 1000 rows. Info function prints a concise summary of the dataframe such as number of columns, column data types, memory usage etc. The next method used here is the describe method, which computes and displays a table containing statistical data such as mean, median, mode, standard deviation, percentiles etc. Followed by that we have used the isnull function, which basically check whether there in an empty or null value in our database which is used in combination with sum function, which does nothing but adds up all those null values.

We also have to ensure that we do not have any missing values in our database, we do that by using isnull() command. Here, we did not find any, so it’s a good idea to proceed with the current dataframe.

# III. Data Visualization

# Dataset I

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

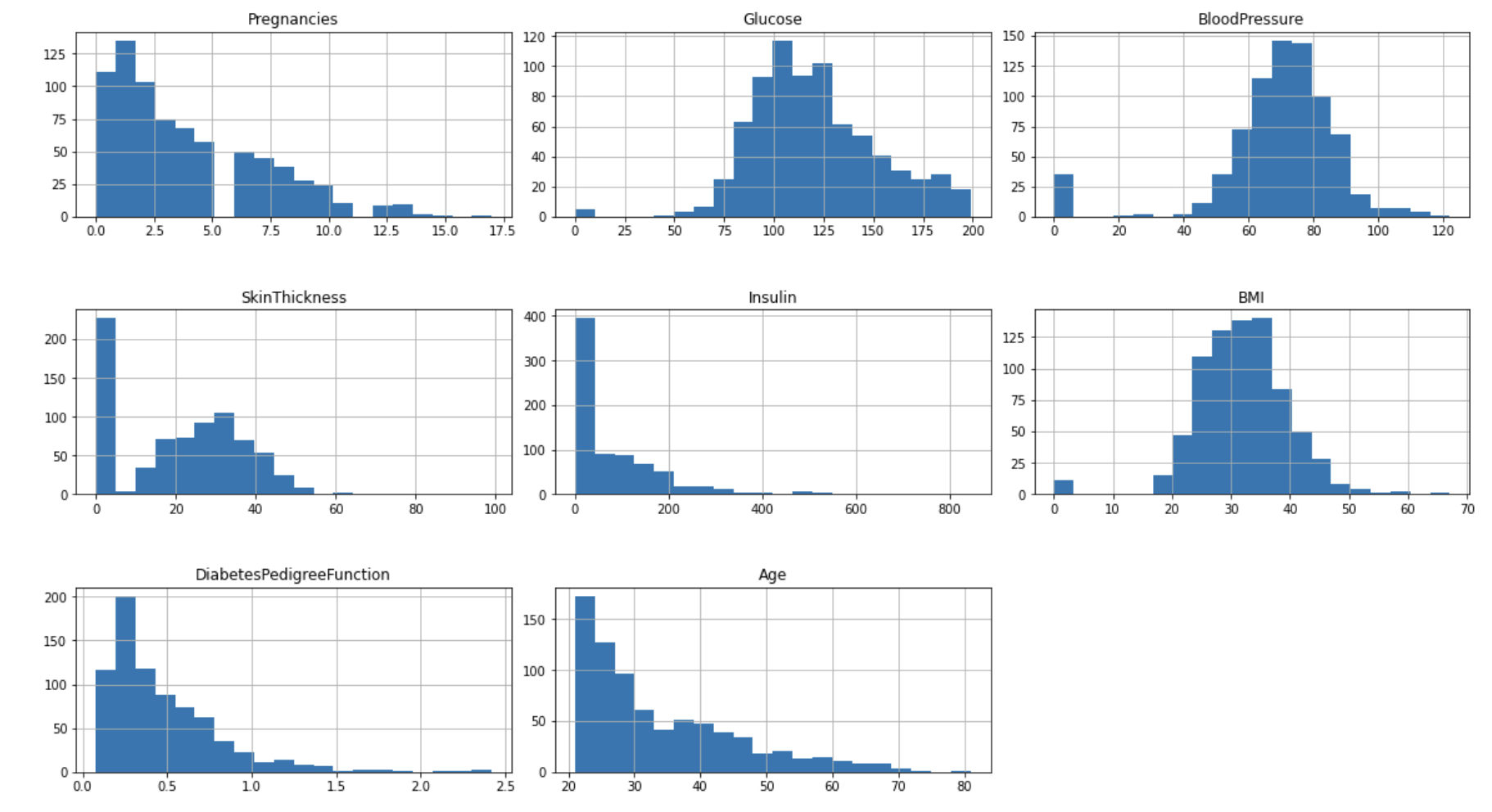
# 

# 

# 

# 

# Dataset II



Through making these charts, we can better understand the values in each and every feature of both the datasets and their range simply by having a glance.

# IV. Data Preprocessing

# After creating various graphs and visualizing the data, we analyze the correlation between different features using the heatmap, and select the features that are strongly correlated to the outcome variable. For dataset I, we have selected these features – BMI, Mental Health, AnyHealthcare, Physical Health, Blood Pressure, Cholesterol, Smoking Habits, Stroke History, Heart Attack, Education and Income whereas for Dataset II we have selected these features – Glucose, Blood Pressure, Skin Thickness, Insulin and BMI. After selecting our features, we split the dataset in two parts, where one part is 80% and would be used to train our dataset and the rest 20% would be used as the test dataset.

# V. Model Evaluation

# Dataset I

# 

# Dataset II

# 

**Logistic Regression**

Logistic Regression [6] is a Supervised Learning technique. It’s mainly used for predicting the categorical variable by mapping the independent input-output pairs.

The outcome is a categorical or discrete value like, true or false, right or wrong, 0 or 1 etc. Though, it doesn't give exact values. It gives a probabilistic value. One major distinction between Logistic Regression and Linear Regression is that Linear Regression predicts a regression line whereas Logistic Regression is used to solve classification problems.

Logistic Regression depends upon these 2 assumptions:

1) The dependent variable has to be categorical or discrete values.

2) The independent variable shouldn’t have multi-collinearity which means several other variables in the model should not be correlated.

Generally, Logistic Regression is used to produce these 2 kinds of results:

1) Binomial: Example: True or false.

2) Multinomial: Example: Pink or red or orange

3) Ordinal: Example: low, medium, high.

Logistic Regression can provide probabilities and categorize new data by utilizing continuous and discrete datasets. It can be used to classify the observations by using different types of data. Then, it can easily predict which variable is the most effective which is used for the categorization.

How it can help in predicting diabetes?

In our particular case, we have many features which have categorical data, such as, weather a person smokes or not, weather a person has a life insurance or not. We can use this model to draw out a correlation between these factors to the dependent variable, and then make a prediction based on how strong the correlation is. In dataset I, we got the accuracy score of Logistic Regression to be 76.0 and 72.07 in dataset II.

**KNN**

The k-nearest neighbors (KNN) is another supervised machine learning algorithm that can be used to solve both classification as well as regression problems. The KNN algorithm works on the assumption that similar data points exist in close proximity. KNN is really useful for the process of image recognition.

How KNN helps in predicting diabetes?

We are taking into account features like high blood pressure, glucose levels etc. Likelihood of diabetes has to be determined depending on the above-mentioned health parameters of patient. For this categorization, K-NN classification can be used. The K-NN algorithm will find similarity between features of the test set and outcome variable and come up with a prediction of weather a person has diabetes or not. In our case, dataset I produced an accuracy of 51 while dataset II produced an accuracy of 78.57 with this model.

**SVM**

Support Vector Machine [7] is another Supervised Learning algorithms, which is widely used for regression as well as classification problems. Although it has more real-life use cases for classification than regression in Machine Learning. The way it works is, it generates a best line or decision boundary that is the maximum distance between data points of both classes. This helps us in assigning future data their respective class or category with ease and confidence. This model selects outliers as vectors while creating the hyperplane. These outlier vectors cases are called support vectors, and hence the name Support Vector Machine.

There are two types of SVMs:

1) Linear SVM: used for linearly separable data which means dataset can be partitioned in two using a single straight line.

2)Non-linear SVM: used for non-linear separable data which means data cannot be partitioned in two classes using a single straight line.

Example:

Consider a unique fruit that has some features of an apple and an orange. The model which can accurately identify if the fruit an orange or an apple can be created by SVM. The model will first be trained by inputting many various images of apples and oranges so that it can differentiate and learn about the distinctive features of oranges and apples. The model can then be tested with the unique fruit. As the support vector generates a decision boundary between the two data (apples and oranges in this case) and choose extreme cases (support vectors), the extreme case of apples and oranges will be seem. With the help of the groundwork provided by the support vectors, the unique fruit can be categorized into an apple or an orange.

How does SVM help in predicting diabetes?

SVM model can create hyperplanes among different categorical features from our dataset and train on it. It can use the most extreme case, and create a decision boundary. With the help of the groundwork provided by decision boundary, predict which category would be more suitable when we input our test data.

SVM’s accuracy in database 1 came out to be 77.0 and 73.37 in database II.

# Naive Bayes

# It is another supervised learning model and is used for classification problems. It is based on Bayes theorem of conditional probability. This model is easy to setup and performs really well for large datasets. It’s not just easy to use, but also powerful enough to compete with and even outperform some sophisticated classification methods. A few use cases of Naive Bayes Algorithm include: Sentimental analysis, spam filtration and classification of articles.

# Naive Bayes gets its name because: Naive: It is called Naive as it presumes that occurrence of a specific feature is self-sustaining. Example: if the vegetable is identified on the basis of color, size, weight etc., then purple, spherical, 1oz is recognized as a Brinjal. Therefore, every feature independently gives an identity that it is a brinjal regardless of each other. Bayes: It is called Bayes as it follows Bayes' Theorem. In dataset I, the accuracy of this model is 41.5 and in dataset II, the accuracy is 71.42.

**Decision Tree**

Decision Tree [8] is a one more type of Supervised learning technique which is primarily used to solve classification problems, but in cases, can be used to solve Regression problems as well. The main idea of Decision Trees is to continuously to make decisions like yes or no based on certain rules to and split the dataset till the point each data point belonging to different class is isolated. This phenomenon creates a Tree like structure, hence the name. In this tree structure, internal nodes represent features, branches represent decision rules and each leaf node represents the outcome.

Reasons that make decision tree a viable classification model:

1) Since we humans also make decisions based upon our own certain set of rules, it mimics our thinking process while making a decision, and hence, it’s easy to understand.

2) The tree-like structure makes it easy for a user who’s reading the logic for the first time to understand what is going on.

In our case, decision tree model produced an accuracy of 65.0 on dataset I and 68.18 on dataset II.

# Random Forest

# Random forest is a Supervised Machine Learning Algorithm that works by combining results of multiple decision trees to reach one single result. It works well for both regression and classification problems. In random forest, instead of working on just one decision tree, it takes into consideration, each and every tree and aggregates them together to predict the most popular result. A singular decision tree is more prone to problems of bias and overfitting, as compared to multiple trees ensembled together in the random forest algorithm. Thus they predict results with more accuracy, especially in cases where individual decision trees are correlated with each other.

# Steps in random forest algorithm are:

# Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

# Step 2: Independent decision trees are constructed for every sample.

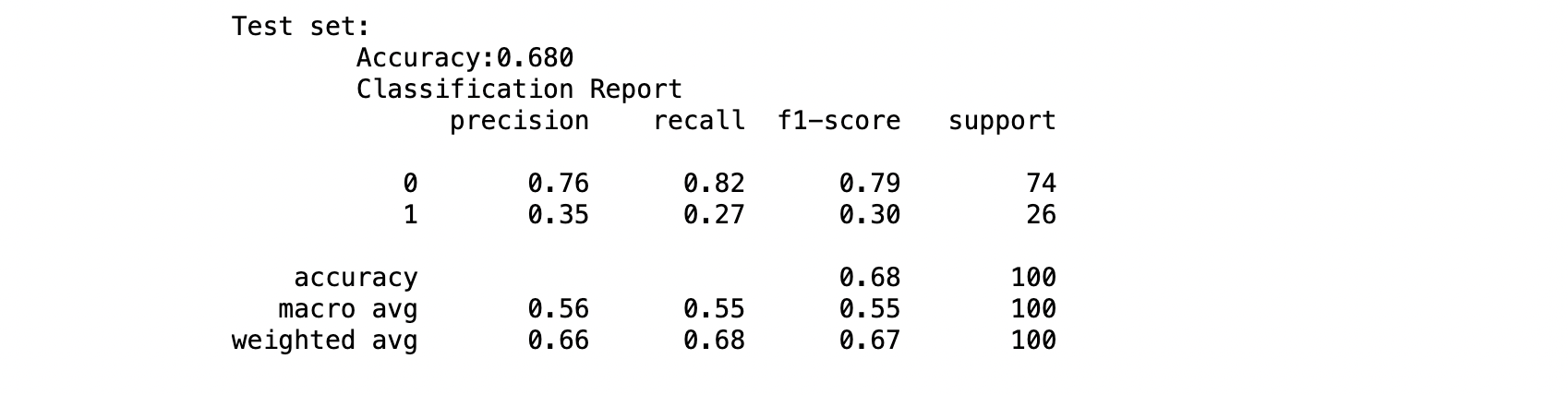
# Step 3: An output will be given by every decision tree.

# Step 4: Depending on Majority Voting or Averaging for Classification and regression respectively, concluding output is considered.

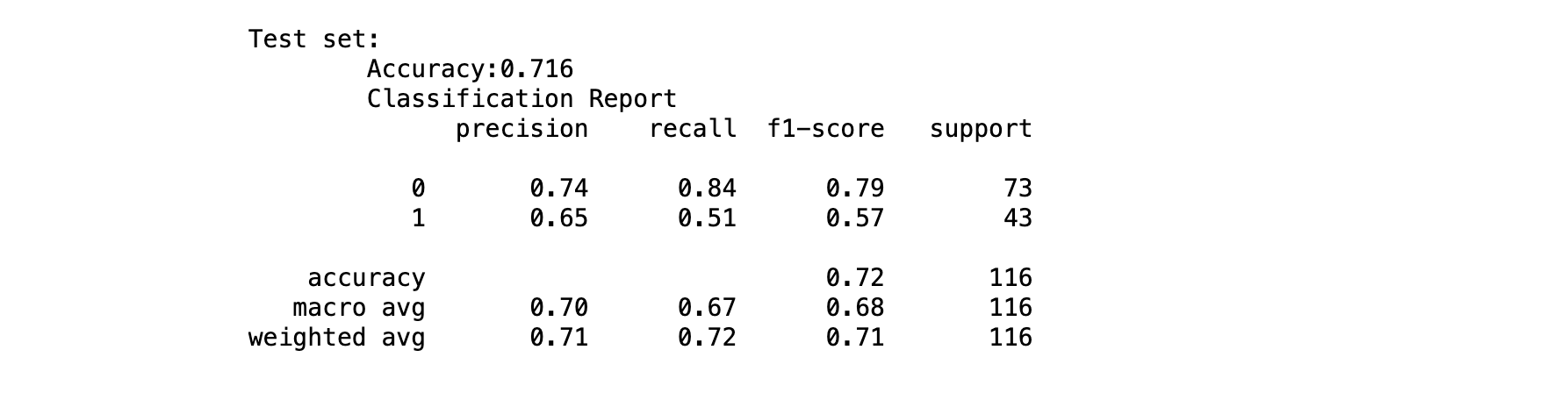
# In our case, dataset I predicted results with 65.0 accuracy and dataset II produced predictions with an accuracy score of 68.18

**Neural Network**

Dataset I



Dataset II



It is a sequence of algorithms which aspires to acknowledge prime correspondence within a set of data via an operation which imitates the function of a human brain.

Neural networks have the capacity of adjusting ro changing input. Hence, the network produces the best possible result. This can be done without redesigning the output norm. Neural network has it's roots in AI. This concept is rapidly obtaining approval and admiration in the trading systems.

The favorable outcome of neural networks for stock market price prognosis differs.

They are a sequence of algorithms which are very similar to the functions and the way animal brain operates, to identify relationships between a large amount of data. They, hence, duplicate the connections between the neurons and synapses found in the body of animals. Neural networks having many process layers are known as "deep" networks. These are used for deep learning algorithms.

They are used in financial services, risk assessment, fraud detection etc.

**References**

1. *Selected Subset of Features from BRFSS 2015- https://www.kaggle.com/code/alexteboul/diabetes-health-indicators-dataset-notebook/notebook*
2. *National Institute of Diabetes and Digestive and Kidney Diseases Dataset -https://www.kaggle.com/datasets/mathchi/diabetes-data-set*
3. *Centers for Disease Control and Prevention. National Diabetes Statistics Report website. https://www.cdc.gov/diabetes/data/statistics-report/index.html. Accessed April 12 2022.*
4. *Kannan, Ramya (2019-11-14). "India is home to 77 million diabetics, second highest in the world". The Hindu. Dated 2020-04-29.*
5. *Procedure Manual. US Preventive Services Task Force. Published May 2021. Accessed June 29, 2021. https://uspreventiveservicestaskforce.org/uspstf/about-uspstf/methods-and-processes/procedure-manual*
6. *https://www.javatpoint.com/logistic-regression-in-machine-learning*
7. *https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm*
8. *https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm*

Contributions:

1. First Page Proposal – Rahul M. Bhardwaj
2. Literature Review Draft – Akshay Singh
3. Google Collab Code – Rahul M. Bhardwaj
4. Documentation – Segment I to IV Data Preprocessing – Akshay Singh
5. Documentation – Segment V Model Evaluation- Harish Goud
6. Proofreading Documentation of Segment V – Akshay Singh & Rahul M. Bhardwaj
7. Conclusion – Rahul M. Bhardwaj

// To add - Google Collab vs Jupyter / others

// Conclusion

// Neural Network - rewrite

// add this joke at the end

*After assessing our models, you might find the predictions to be a bit harsh.*

*But it’s not like we can sugarcoat it.*