INTRODUCTION:

- Diabetes is the common disease faced by the all types and all age group people. Due to this disease the body doesn’t produce a required amount of hormones.

- The cells in our body needs glucose for growth that hormone is sort of essential. If somebody has polygenic disease, very little or no hormone is secreted. during this state of affairs, plenty of glucose is accessible within the blood stream however the body is unable to use it primarily there square measure 2 styles of polygenic disease ,which are Type-1 and Type-2.

INTRODUCTION

- Type-1 polygenic disease happens once the body's immune system is attacked and therefore the beta cells (these cells manufacture insulin) of exocrine gland square measure destroyed. This ends up in hormone deficiency.

- The only treatment to Type-1 polygenic disease is hormone. On the opposite hand,Type-2 polygenic disease is caused by relative hormone deficiency. exocrine gland inType-2 polygenic disease still produces hormone however it should not be effective or may not manufacture spare quantity of hormone to manage blood glucose . Type-2 polygenic disease is that the commonest type of polygenic disease, which usually develops at age forty and older.

Abstract:

Diabetes may be a one in all the leading explanation for visual defect, kidney failure, amputations, coronary failure and stroke. When we eat, our body turns food into sugars, or glucose. At that time, our exocrine gland is meant to unleash internal secretion. internal secretion is a "key” to open our cells, to permit the aldohexose to enter and permit us to utilize the aldohexose for energy.

However with diabetes, this method does not work. many major things will get it wrong – inflicting the onset of diabetes. Type 1 and type 2 diabetes are the most common forms of the disease. This paper focuses on development in machine learning that have created important impacts within the detection diabetes. Using different algorithms, we will be generating a proper output.

Literature Survey

Ref No.	Title of paper	Abstract	Outcome	Methodology	Research gap
1	Diabetes Detection Using Machine Learning Classification Methods , **2021.**	The main objective of this research is to predict the possible presence of diabetes - specifically in females- at an early stage using different machine learning techniques.	this model produced an accuracy of 82% based on the random forest classifier model.	Random Forest Classifier is one of the simplest and most diverse algorithms used for both classification and regression tasks, it uses multiple individual decision trees to operate as a single one..	Less Accuracy Only 82% accuracy
2	Investigating Health-Related Features and Their Impact on the Prediction of Diabetes Using Machine Learning , **2021.**	To investigate the prediction of diabetic patients and compare the role of HbA1c and FPG as input features. By using five different machine learning classifiers, and using feature elimination through feature permutation and hierarchical clustering	Identified several other features like hypertension, weight, and physical activity levels that had an indirect role in diabetic prediction. The LDL/HDL tests were also found to be correlated with diabetic conditions.	Comparison of correlation before and after hierarchical clustering based on Spearman's ranking	small-scale data

Ref No.	Title of paper	Abstract	Outcome	Methodology	Research gap
3	A Decision Support System for Diabetes Prediction Using Machine Learning and Deep Learning Techniques, **2019.**	employed a fully Convolutional Neural Network (CNN) to predict and detect the diabetes patients	The results showed that RF was more effective for classification of the diabetes in experiments which produced accuracy of 83.67%. The prediction accuracy for SVM reached 65.38% while DL method produced 76.81%	CNNs perform a series of operations on the input and transform it to produce the desired output. This output from previous layers can be taken as input to the next block.	No proper feature extraction and accuracy
4	Diabetes Prediction using Machine Learning Algorithms, **2019.**	Diabetes prediction model for better classification of diabetes with includes few external factors responsible for diabetes along with factors like glucose, BMI, age, Insulin, etc	Logistic Regression gives highest accuracy of 96%. Application of pipeline gave AdaBoost classifier as best model with accuracy of 98.8%.	various machine learning algorithms like Support Vector Classifier, Random Forest Classifier, Decision Tree Classifier, Extra Tree Classifier, Ada Boost algorithm, Perceptron, Logistic Regression, K-Nearest Neighbour, Gaussian Naïve Bayes, Bagging algorithm, Gradient Boost Classifier are used.	The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables. It not only provides a measure of how appropriate a predictor(coefficient size)is, but also its direction of association (positive or negative)

Ref No.	Title of paper	Abstract	Outcome	Methodology	Research gap
5	ANALYSIS AND PREDICTION OF DIABETES USING MACHINELEARNING, **2019.**	To determine new patterns and then to interpret these patterns to deliver significant and useful information for the users.	In this study the proposed method provides high accuracy with accuracy value of 90.36% and decision Stump provided less accuracy than other by providing 83.72% accuracy.	we have employed different classifiers like Decision Trees, KNN and Naïve Bayes.	in this study only limited base classifier used
6	Analysis and Prediction of Diabetes Mellitus using Machine Learning Algorithm , **2018.**	To cluster and predict symptoms in medical data, various data mining techniques were used by different researchers in different time	In this study the proposed method provide high accuracy with accuracy value of 90.36% and decision Stump provided less accuracy than other by providing 83.72% accuracy.	In the proposed system most known predictive algorithms applied are SVM, Naïve Net,DecisionStump, and Proposed Ensemble method (PEM)	on this study also only a single data set used in this study only limited base classifier used

Ref No.	Title of paper	Abstract	Outcome	Methodology	Research gap
7	Important Feature Selection & Accuracy Comparisons of Different Machine Learning Models for Early Diabetes Detection, **2018.**	introduce an approach to automatically predict type 2 diabetes mellitus (T2DM) applying a neural network. The objective of this paper is to find which type of model that works best for predicting diabetes	Result of input these features to the MLP neural network classifier which achieved an accuracy of 85.15%.	A multilayer perceptron (MLP) is a class of feed-forward artificial neural network. We use this algorithm because MLPs are used in research for their ability to solve problems sarcastically, which often allows approximate solutions for extremely complex issues like fitness approximation.	The quantity of the data-set is not large enough to train appropriately and prediction with lower efficiency
8	Prediction of Diabetes Using Machine Learning Algorithms in Healthcare , **2018.**	Comparison of the different machine learning techniques used. This study reveals which algorithm is best suited for prediction of diabetes. Helps doctors in early prediction of diabetes using machine learning techniques	In this experiment, it can be seen that SVM and KNN gives highest accuracy for predicting diabetes. Both these algorithms provide 77% accuracy which is highest as compared to other algorithms used in this paper.	six machine learning algorithms are used to predict diabetes disease. These six algorithms are K Nearest Neighbours (KNN), Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Logistic Regression (LR) and Random Forest (RF).	Some limitations of this study are the size of dataset and missing attribute values.

Outcome of Literature Survey:

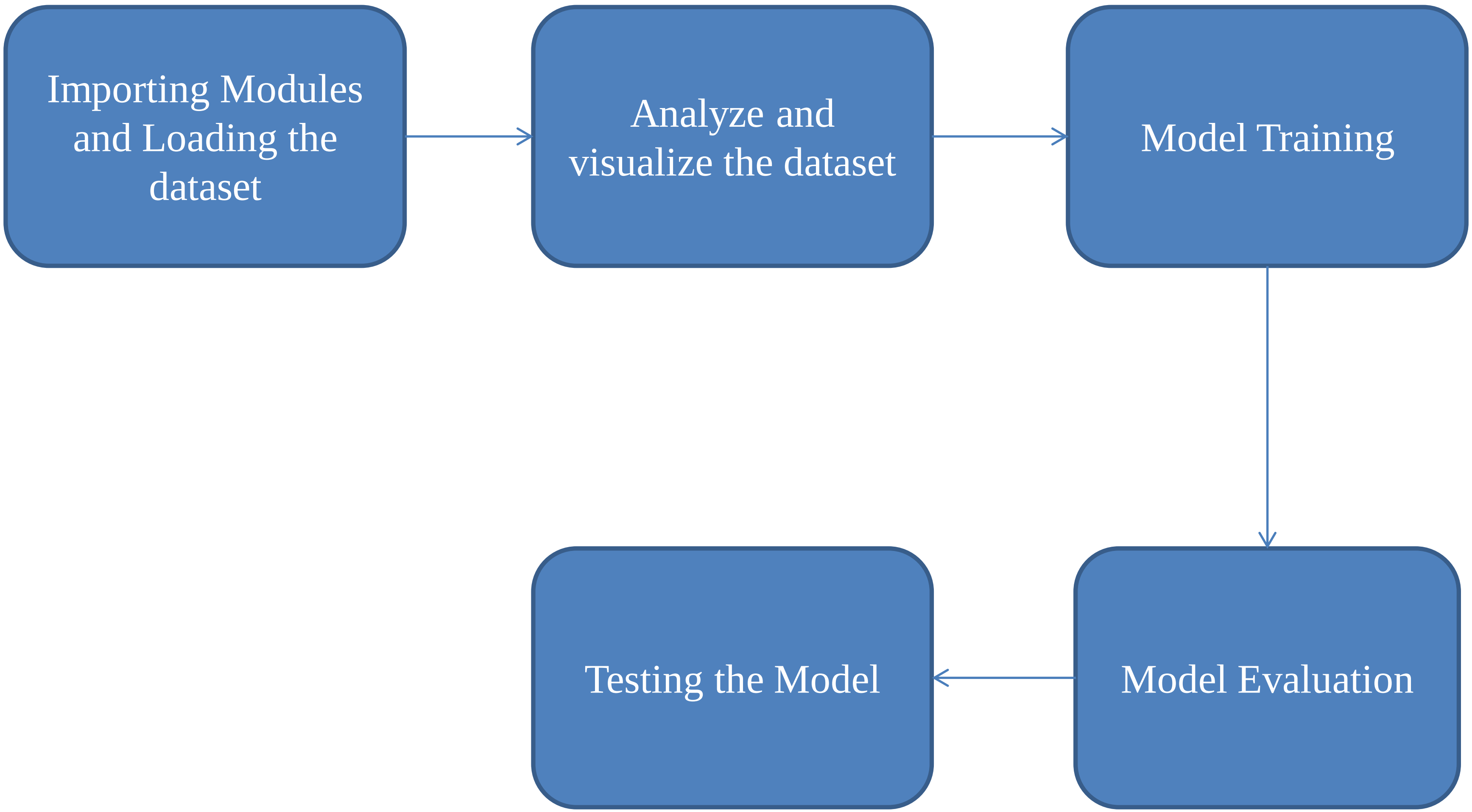
Based on previous works from last 4 years, we have identified :

1) Diabetes complications caused by several types
2) Type 1 and Type 2 identification using administrative data
3) Diabetes and its subtypes' misclassification are uncertain.

Research Objectives:

1. To Train the machine learning methods for prediction of multi type diabetes diseases.
2. To find the performance metrics of Algorithm.
3. Recognizing the failures of earlier attempts to diagnose subtypes of diabetes mellitus using machine learning techniques.
4. Creating a model or tool for sub-types of Diabetes Mellitus diagnosis that aids in detecting the disease early in individuals and may lower the risk of developing it.

General Block Diagram



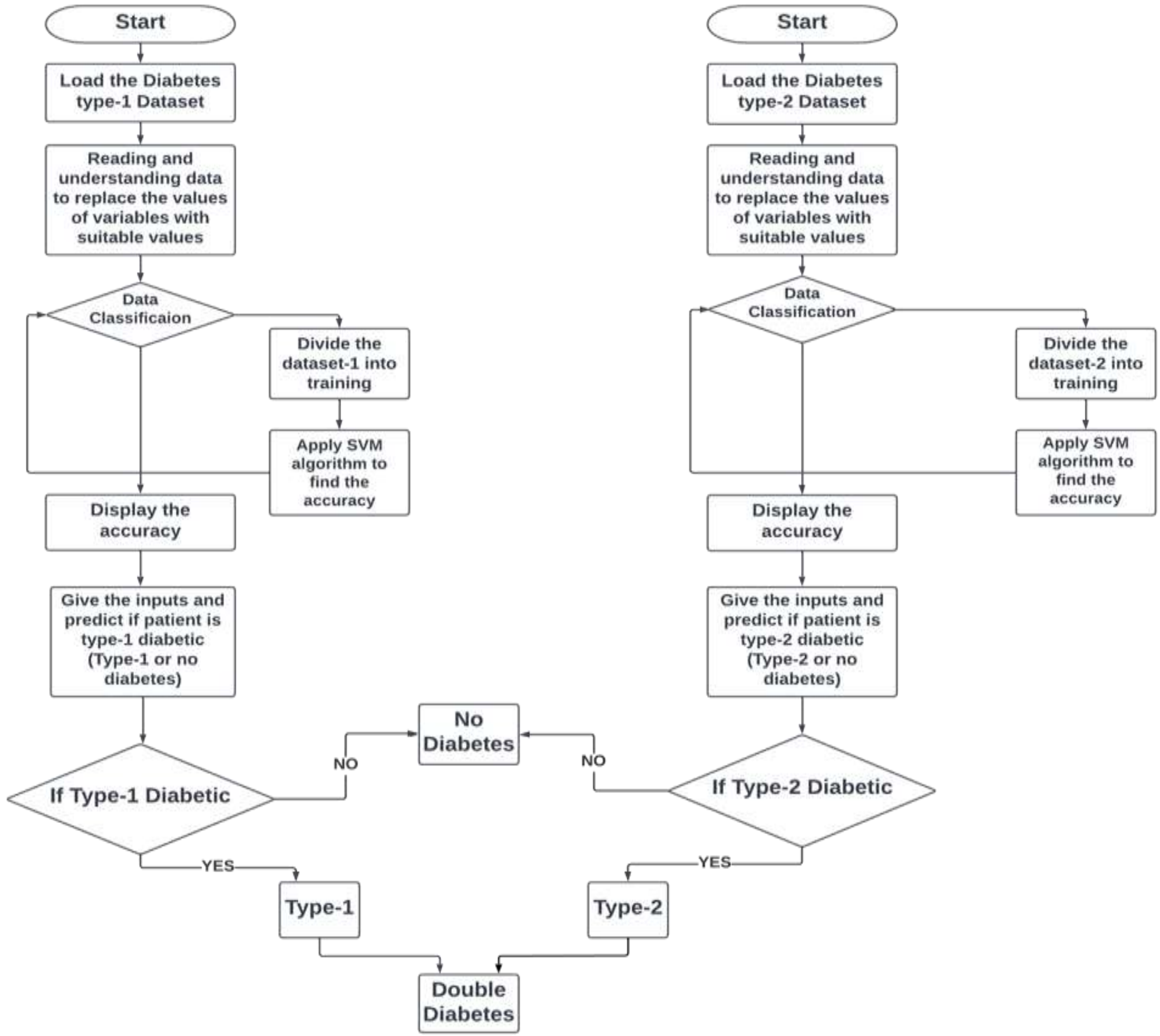
Software and Modules :

- Text Editor: Google Colaboratory
- Programming Language: Python

Modules:

1. NumPy
2. Pandas
3. Scikit-learn
4. Matplotlib
5. Seaborn

METHODOLOGY



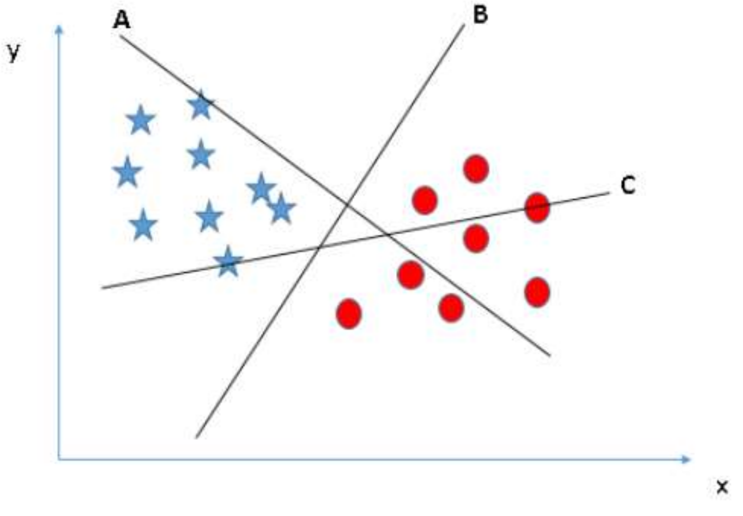
Start
Load the Diabetes type-1 Dataset
Reading and understanding data to replace the values of variables with suitable values
Data Classificaion
Divide the dataset-1 into training
Apply SVM algorithm to find the accuracy
Display the accuracy
Give the inputs and predict if patient is type-1 diabetic (Type-1 or no diabetes)
If Type-1 Diabetic
NO
YES
Type-1
Start
Load the Diabetes type-2 Dataset
Reading and understanding data to replace the values of variables with suitable values
Data Classification
Divide the dataset-2 into training
Apply SVM algorithm to find the accuracy
Display the accuracy
Give the inputs and predict if patient is type-2 diabetic (Type-2 or no diabetes)
If Type-2 Diabetic
NO
YES
Type-2
No Diabetes
Double Diabetes

Support Vector Machine:

- “Support Vector Machine” (SVM) is a supervised machine learning algorithm that can be used for both classification
- They have two main advantages: higher speed and better performance with a limited number of samples (in the thousands).
- Finds the linear hyper-plane that separates classes with the maximum margin.
- Lets understand SVM with different scenarios.

Identifying the right hyper-plane:

Scenario-1:



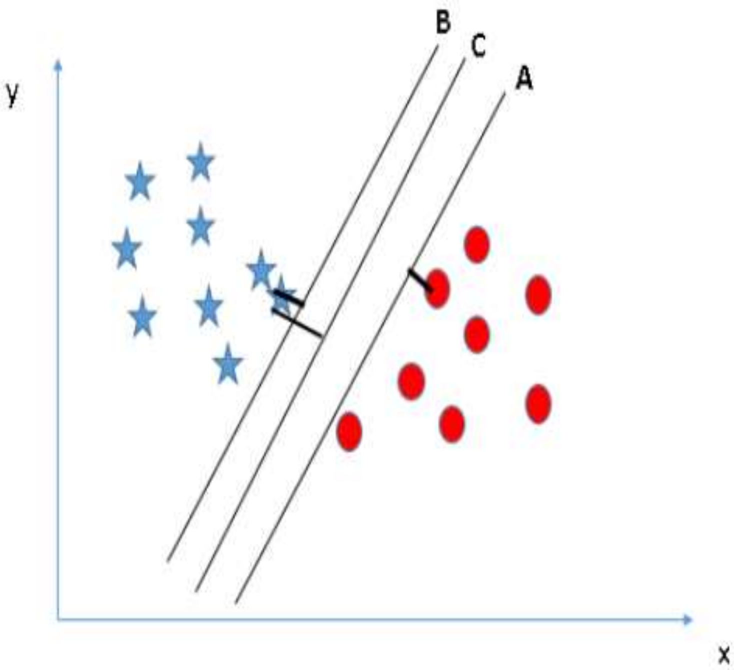
Result:

- Hyper-plane B

Reason: Select the hyper-plane which segregates the two classes better

Identifying the right hyper-plane:

Scenario-2:



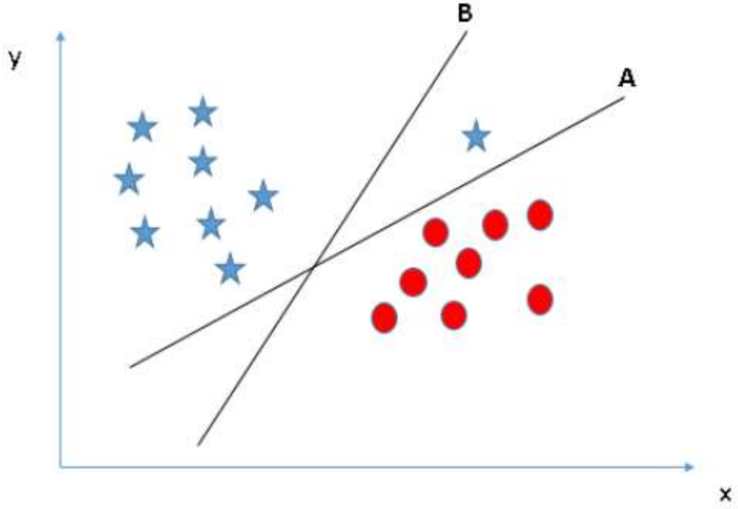
Result:

- Hyper-plane C

Reason: The margin for hyper-plane C is high as compared to both A and B

Identifying the right hyper-plane:

Scenario-3:

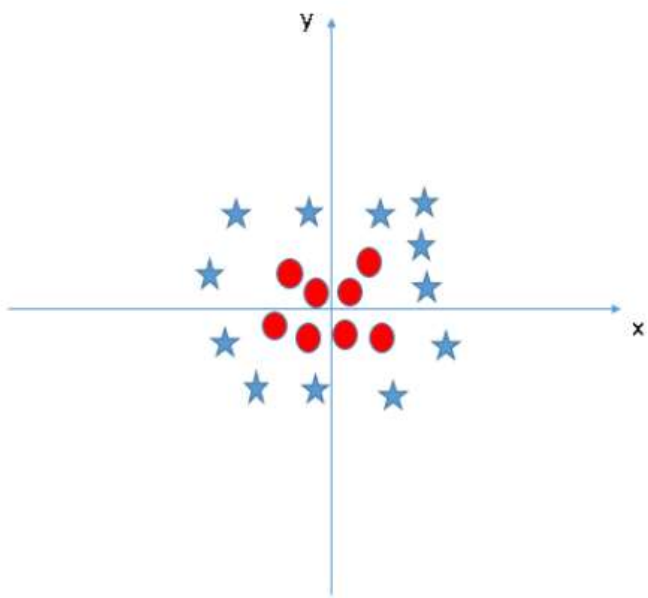


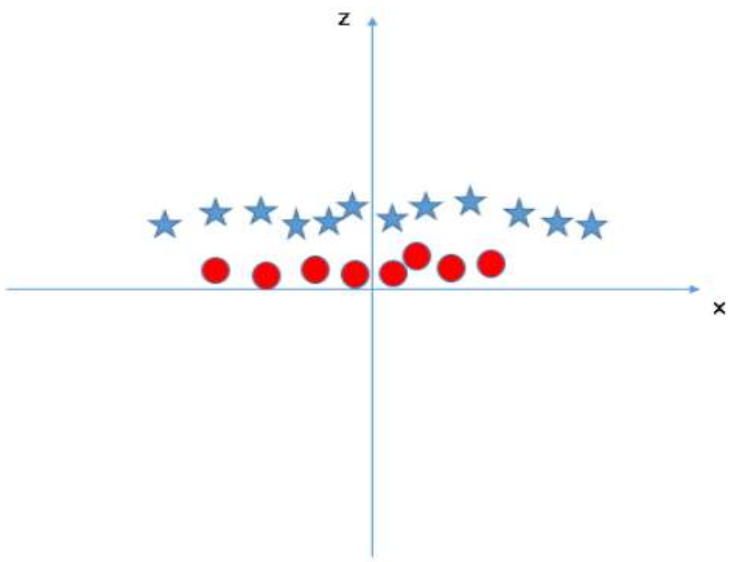
Result:

- Hyper-plane A

Reason: SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin.

Scenario-4 **Result:**





Formula : Z= X^2+Y^2

ANALYSIS

Features of Dataset-1

- Age
- Sex
- Height
- Weight
- BMI
- Adequate Nutrition
- Adequate Nutrition 1
- Autoantibodies
- Impaired Glucose Metabolism
- Insulin Taken
- How taken
- Family History of Type 1 Diabetes
- Family History of Type 2 Diabetes
- Hypoglycemia
- Pancreatic Disease affect

Features of Dataset-2

- Age
- Gender
- Family Diabetes
- High BP
- Physical Activeness
- BMI
- Smoking Habit
- Alcohol Habit
- Proper Sleep
- Sound Sleep
- Regular Medicine
- Junk Food
- Stress
- BP Level
- Pregnancies
- Pdiabetes
- Urination Frequency

Step-1: Importing Modules and Loading the Dataset

```
[ ] import pandas as pd
    import numpy as np
    from sklearn import svm
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.model_selection import train_test_split, cross_val_predict
    from sklearn.preprocessing import StandardScaler
    from sklearn.svm import SVC
    from imblearn.over_sampling import SMOTENC
    from sklearn.model_selection import GridSearchCV
    #from sklearn.metrics import classification_report, accuracy_score, recal
    import imblearn
    from sklearn.metrics import confusion_matrix
    import warnings
    warnings.filterwarnings('ignore')
    plt.style.use('fivethirtyeight')
```

```
data=pd.read_csv('Dataset Diabetes Type1 (Total)-11.csv')
```

Step-2: Understanding the Dataset-1 (Analyzing)

```
[ ]  data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 17 columns):
 #   Column                                  Non-Null Count  Dtype  
---  ------                                  --------------  -----  
 0   Age                                     306 non-null    object 
 1   Sex                                     306 non-null    object 
 2   Height                                  306 non-null    float64
 3   Weight                                  306 non-null    float64
 4   BMI                                     306 non-null    float64
 5   Adequate Nutrition                      306 non-null    object 
 6   Adequate Nutrition .1                   306 non-null    object 
 7   Education of Mother                     306 non-null    object 
 8   Autoantibodies                          306 non-null    object 
 9   Impaired glucose metabolism             306 non-null    object 
 10  Insulin taken                           306 non-null    object 
 11  How Taken                               306 non-null    object 
 12  Family History affected in Type 1 Diabetes 306 non-null    object 
 13  Family History affected in Type 2 Diabetes 306 non-null    object 
 14  Hypoglycemis                            306 non-null    object 
 15  pancreatic disease affected in child    306 non-null    object 
 16  Affected                                306 non-null    object 
dtypes: float64(3), object(14)
memory usage: 40.8+ KB
```

Step-2: Understanding the Dataset-2 (Analyzing)

```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 947 entries, 0 to 951
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Age               947 non-null    object 
 1   Gender            947 non-null    object 
 2   Family_Diabetes   947 non-null    object 
 3   highBP            947 non-null    object 
 4   PhysicallyActive  947 non-null    object 
 5   BMI               947 non-null    float64
 6   Smoking           947 non-null    object 
 7   Alcohol           947 non-null    object 
 8   Sleep             947 non-null    int64  
 9   SoundSleep        947 non-null    int64  
 10  RegularMedicine   947 non-null    object 
 11  JunkFood          947 non-null    object 
 12  Stress            947 non-null    object 
 13  BPLevel           947 non-null    object 
 14  Pregancies        947 non-null    float64
 15  Pdiabetes         947 non-null    object 
 16  UriationFreq      947 non-null    object 
 17  Diabetic          947 non-null    object 
dtypes: float64(2), int64(2), object(14)
memory usage: 140.6+ KB
```

Step-2: Understanding the Data (Analyzing)

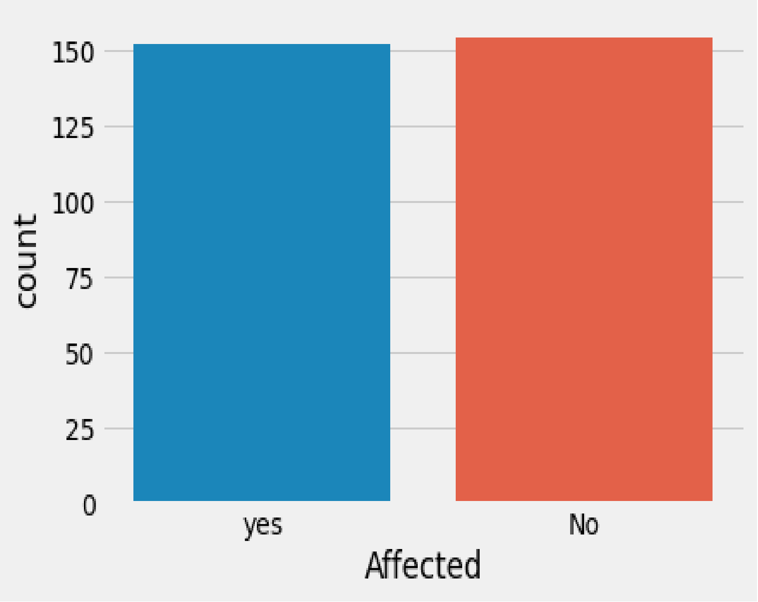
```
[ ]  data.describe(include='all')
```

	Age	Sex	Height	Weight	BMI
count	306	306	306.000000	306.000000	306.000000
unique	4	2	NaN	NaN	NaN
top	greater then 15	Male	NaN	NaN	NaN
freq	110	163	NaN	NaN	NaN
mean	NaN	NaN	1.349346	38.854575	20.860711
std	NaN	NaN	0.277601	16.689427	6.225745
min	NaN	NaN	0.440000	5.000000	10.077936
25%	NaN	NaN	1.220000	25.000000	16.928286
50%	NaN	NaN	1.420000	40.000000	19.813209
75%	NaN	NaN	1.560000	50.000000	23.711124
max	NaN	NaN	1.830000	87.000000	61.983471

Step-2: Understanding the Data (Visualize)

```
[ ] sns.countplot(x='Affected',data=data)
```

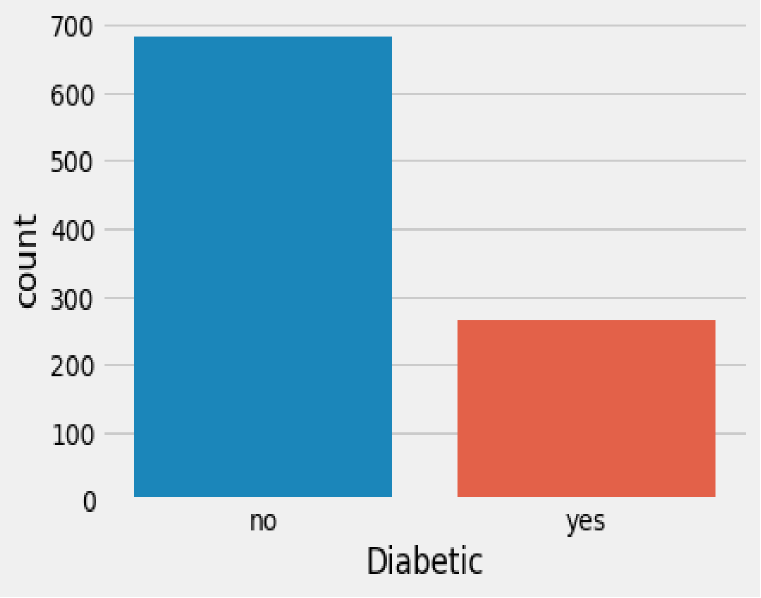
<matplotlib.axes._subplots.AxesSubplot at 0x7f5aea02be50>



Dataset-1

```
[ ] sns.countplot(x='Diabetic',data=data1)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f5ae7f13810>



Dataset-2

Step-2: Understanding the Data
(Replacing the values of Variables - Dataset-1)

```
[ ]  def preprocessing(df):
         df= df.copy()

         # Gender column Binary Encoding
         df['Sex'] = df ['Sex'].replace({'Female':0,'Male':1 })
         df['Age'] = df ['Age'].replace({'Less then 11': 10, 'Less then 15': 13, 'greater then 15': 50, 'Less then 5': 4})


         #Symptom Column Binary Encoding
         for column in df.columns.drop(['Age','Sex','Affected']):
          df[column]= df[column].replace({'No':0 , 'Yes': 1,'no':0,'none':1,'Injection':1})
```

Step-2: Understanding the Data
(Replacing the values of Variables - Dataset-2)

```
category_mapping = {
    'Age':{'less than 40':0, '40-49':1, '50-59':2, '60 or older':3},
    'Family_Diabetes':{'no':0, 'yes':1},
    'Gender':{'Female':0, 'Male':1},
    'Smoking':{'no':0, 'yes':1},
    'Pdiabetes':{'no':0, 'yes':1},
    'RegularMedicine':{'no':0, 'yes':1},
    'PhysicallyActive':{'one hr or more':0, 'more than half an hr':1, 'less than half an hr':2, 'none':3},
    'JunkFood':{'occasionally':0, 'often':1, 'very often':2, 'always':3},
    'BPLevel':{'low':0, 'normal':1, 'high':2},
    'highBP':{'no':0, 'yes':1},
    'Alcohol':{'no':0, 'yes':1},
    'UriationFreq':{'not much':0, 'quite often':1},
    'Stress':{'not at all':0, 'sometimes':1, 'very often':2, 'always':3},
    'Diabetic':{'no':0, 'yes':1},
}

for col in category_cols:
    data_clean[col] = data_clean[col].map(category_mapping[col])
```

Step-3: Splitting the dataset-1 to Train and Test

```
#train
y=df["Affected"]
X=df.drop("Affected", axis=1)

#test_train_split
X_train, X_test,y_train,y_test = train_test_split(X,y,train_size=0.7,shuffle=True,random_state=1)

#StandardScaler
scaler=StandardScaler()
scaler.fit(X_train)
X_train=pd.DataFrame(scaler.transform(X_train),index=X_train.index , columns=X_train.columns)
X_test=pd.DataFrame(scaler.transform(X_test),index=X_test.index, columns=X_test.columns)

return X_train,X_test,y_train,y_test
```

```
X_train,X_test,y_train,y_test= preprocessing(data)
```

Step-3: Splitting the dataset-2 to Train and Test

```
[ ] # split the data
    x = data_clean.drop('Diabetic', axis=1)
    Y = data_clean['Diabetic']
    x_train, x_test, Y_train, Y_test = train_test_split(x, Y, test_size=0.2, random_state=123, stratify=Y)
```

```
[ ] print(Y_train.value_counts())
    print(Y_test.value_counts())
```

```
0    545
1    212
Name: Diabetic, dtype: int64
0    137
1     53
Name: Diabetic, dtype: int64
```

Step-4: Applying Algorithm and Displaying the accuracy

Dataset-1

```
[ ]  model=SVC().fit(X_train,y_train)
     print('SVC(): trained')
```

```
SVC(): trained
```

```
[ ]  print("Accuracy of SVM: {:.2f}%".format(model.score(X_test,y_test) * 100))
```

```
Accuracy of SVM: 100.00%
```

Step-4: Applying Algorithm and Displaying the accuracy

Dataset-2

```
def grid_search(X_tr, X_te, y_tr, y_te, model, params, scoring='recall'):
    gs = GridSearchCV(estimator = model, param_grid = params, scoring = scoring, n_jobs=-1, cv=3)
    gs.fit(X_tr, y_tr)
    y_pred = gs.predict(X_te)
    print(f"{model}")
    print(f"Best parameter      : {gs.best_params_}")
    print(f"Test Accuracy Score : {accuracy_score(y_te, y_pred)}")
    print(f"Train Accuracy Score: {accuracy_score(y_tr, gs.predict(X_tr))}")
    print(f"Recall score        : {recall_score(y_te, y_pred)}")
    print(f"Classification Report \n{'-'*30}\n {classification_report(y_te, y_pred)}")
    return gs.best_params_
```

Step-4: Applying Algorithm and Displaying the accuracy

Dataset-2

```
model = SVC(random_state=123)
params = {
    'C' : [0.001, 0.01, 0.1, 1, 10],
    'kernel' : ['linear', 'poly', 'rbf', 'sigmoid'],
    'degree' : [2, 3, 4, 5]
}
svc_best = grid_search(x_train_smote, x_test, Y_train_smote, Y_test, model, params, scoring='accuracy')
```

```
SVC(random_state=123)
Best parameter      : {'C': 10, 'degree': 4, 'kernel': 'poly'}
Test Accuracy Score : 0.8526315789473684
Train Accuracy Score: 0.9100917431192661
Recall score        : 0.8867924528301887
Classification Report
------------------------------
              precision    recall  f1-score   support

           0       0.95      0.84      0.89       137
           1       0.68      0.89      0.77        53

    accuracy                           0.85       190
   macro avg       0.82      0.86      0.83       190
weighted avg       0.88      0.85      0.86       190
```

RESULTS

Step-5: Prediction of Type-1 Diabetes

```
[ ]  input_data = (50,0,1.4,45,22.95918367,0,0,0,0,0,1,1,0,0,1,1)

     # changing the input_data to numpy array
     input_data_as_numpy_array = np.asarray(input_data)

     # reshape the array as we are predicting for one instance
     input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

     # standardize the input data
     std_data = scaler.transform(input_data_reshaped)
     print(std_data)

     prediction = classifier.predict(std_data)
     print(prediction)

     if (prediction[0] == 'no'):
       print('The person is not diabetic')
     else:
       print('The person is type-1 diabetic')
```

```
[[ 5.00000000e+01 -1.66014658e-17  1.40000000e+00  4.50000000e+01
   2.29591837e+01  5.81051303e-17  5.81051303e-17  1.24510993e-17
  -5.39547638e-17  2.49021987e-17  1.00000000e+00  1.00000000e+00
  -3.32029316e-17  3.32029316e-17  1.00000000e+00  1.00000000e+00]]
['yes']
The person is type-1 diabetic
```

Step-5: Prediction of Type-2 Diabetes

```
[ ]  input__data = (2,1,1,1,2,28,0,0,6,1,2,1,1,0,0,0,0)

     # changing the input_data to numpy array
     input_data_numpyarray = np.asarray(input__data)

     # reshape the array as we are predicting for one instance
     input__datareshaped = input_data_numpyarray.reshape(1,-1)

     # standardize the input data
     std__data = Scaler.transform(input__datareshaped)
     print(std__data)

     predictions = Classifier.predict(std__data)
     print(predictions)

     if (predictions[0] == 1):
       print('The person is not diabetic')
     else:
       print('The person is type-2 diabetic')
```

```
[[-4.42941635 -4.6078873  -2.31735064  0.59222614  3.58193294 69.13886886
  -1.51338628 -1.14854209  4.19641672  0.97642461  1.42047285  0.82746517
  -0.34633289 -0.52338689 -0.68773031 -0.11812488 -0.36638118]]
[0]
The person is type-2 diabetic
```

Step-6: Conclusion (Type-1 or Type-2)

```
[ ]  if prediction[0]=='no' and predictions[0]==1:
       print("Result --> Patient is not diabetic")
     if prediction[0]=='no' and predictions[0]==0:
       print("Result --> Patient is Type-2 diabetic")
     if prediction[0]=='yes' and predictions[0]==1:
       print("Result --> Patient is Type-1 diabetic")
     if prediction[0]=='yes' and predictions[0]==0:
       print("Result --> Pateint is Double diabetic")
```

```
Result --> Pateint is Double diabetic
```

Applications:

- Early prediction of diabetes.
- Correctly identify Diabetes subtypes.

Conclusion and Future Scope:

- Hence, we are able to predict the type of diabetes. By applying SVM algorithm for the both type-1 data set and type-2 data set; the accuracy of the SVM are 100%, 85%.

- SVM classifier can be optimized by tuning other parameters thus it can be beneficial to improve results. SVM classifier can be employed into any medical research for better outcomes.

- We will be trying different algorithms like KNN and ANN to improve the performance.

References:

[1]Abdulhadi, N., & Al-Mousa, A. **(2021, July).** Diabetes detection using machine learning classification methods. In *2021 International Conference on Information Technology (ICIT)* (pp. 350-354). IEEE.

[2]Ahmad, H. F., Mukhtar, H., Alaqail, H., Seliaman, M., & Alhumam, A. **(2021).** Investigating health-related features and their impact on the prediction of diabetes using machine learning. *Applied Sciences*, *11*(3), 1173..

[3]Yahyaoui, A., Jamil, A., Rasheed, J., & Yesiltepe, M. **(2019, November)**. A decision support system for diabetes prediction using machine learning and deep learning techniques. In *2019 1st International Informatics and Software Engineering Conference (UBMYK)* (pp. 1-4). IEEE.

[4]Mujumdar, A., & Vaidehi, V. **(2019).** Diabetes prediction using machine learning algorithms. *Procedia Computer Science*, *165*, 292-299.

[5]Saru, S., & Subashree, S. **(2019).** Analysis and prediction of diabetes using machine learning. *International journal of emerging technology and innovative engineering*, *5*(4)

References:

[6]Alehegn, M., Joshi, R., & Mulay, P. **(2018).** Analysis and prediction of diabetes mellitus using machine learning algorithm. *International Journal of Pure and Applied Mathematics*, *118*(9), 871-878.

[7]Rubaiat, S. Y., Rahman, M. M., & Hasan, M. K. **(2018, December).** Important feature selection & accuracy comparisons of different machine learning models for early diabetes detection. In *2018 International Conference on Innovation in Engineering and Technology (ICIET)* (pp. 1-6). IEEE.

[8]Sarwar, M. A., Kamal, N., Hamid, W., & Shah, M. A. (**2018**, September). Prediction of diabetes using machine learning algorithms in healthcare. In ***2018*** *24th international conference on automation and computing (ICAC)* (pp. 1-6). IEEE.

Thank You!