Support Vector Machine

Author: Abhinav Reddy Chintalapuri

School of Engineering and Applied Science

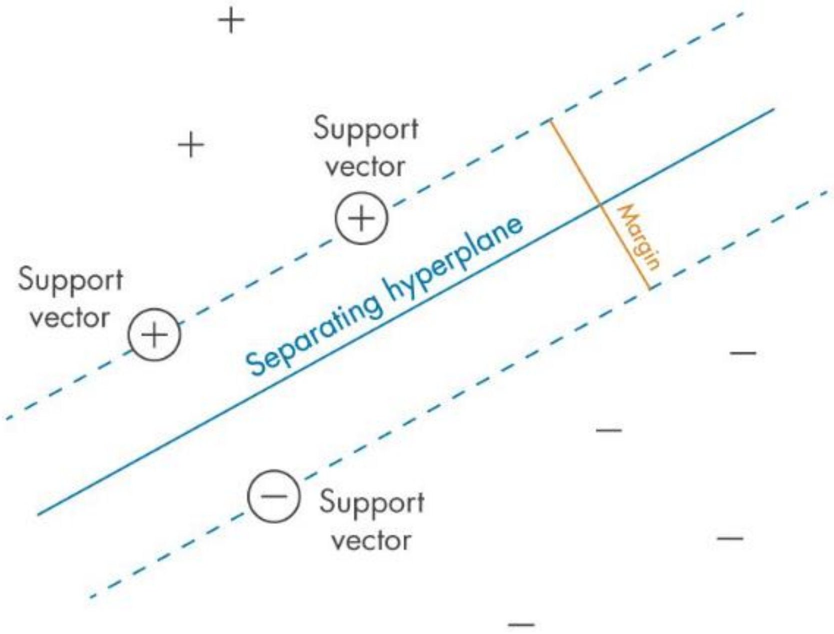
University at Buffalo, NY, 14260

achintal@buffalo.edu

1. Support Vector Machine

Support vector machines are supervised learning models which use certain learning algorithms that study data implemented for regression analysis and classification. The obtained training data the algorithm generates an optimum hyperplane that classifies upcoming instances.

The primary goal of an SVM is to specify a hyperplane that divides the points into two classes. The hyperplane is also known as the decision boundary or the separating hyperplane. When attempting to depict a hyperplane, consider a 2D dataset.



Experiment

Dataset:

The data used is the clinical records of patients with symptoms of diabetes. The attributes are used to build a model which predicts the likelihood of diabetes.

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex
count	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000
mean	60.576868	0.388128	621.634703	0.429224	37.684932	0.324201	263956.824384	1.357534	136.958904	0.662100
std	11.976087	0.488440	986.607994	0.496099	11.175674	0.469148	99863.329083	0.857723	4.033478	0.474078
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	0.500000	121.000000	0.000000
25%	51.000000	0.000000	115.000000	0.000000	30.000000	0.000000	217000.000000	0.900000	134.000000	0.000000
50%	60.000000	0.000000	257.000000	0.000000	38.000000	0.000000	257000.000000	1.100000	137.000000	1.000000
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	302500.000000	1.400000	140.000000	1.000000
max	95.000000	1.000000	7702.000000	1.000000	80.000000	1.000000	850000.000000	6.800000	148.000000	1.000000

The split is in the ratio of 80-20 for Train and test respectively.

```
X_train size :  (175, 12)
X_test size  :  (44, 12)
y_train size :  (175, 1)
y_test size  :  (44, 1)
```

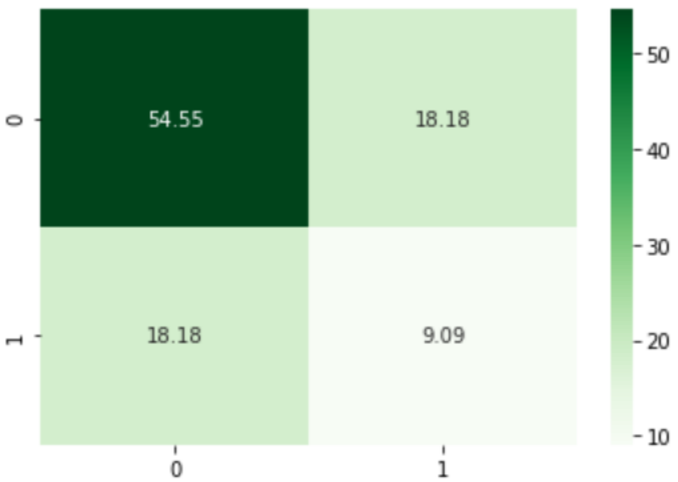
Accuracy for gamma value 0.05

```
Accuracy Score: 0.7727
SVC f1-score  : 0.5455
SVC precision : 0.5000
SVC recall    : 0.6000
              precision    recall  f1-score   support

           0       0.88      0.82      0.85        34
           1       0.50      0.60      0.55        10

    accuracy                           0.77        44
   macro avg       0.69      0.71      0.70        44
weighted avg       0.79      0.77      0.78        44
```

Accuracy for gamma value 0.06

```
Accuracy : 0.6364
SVC f1  : 0.3333
SVC recall    : 0.3333
SVC precision : 0.3333

              precision    recall  f1-score   support

           0       0.75      0.75      0.75        32
           1       0.33      0.33      0.33        12

    accuracy                           0.64        44
   macro avg       0.54      0.54      0.54        44
weighted avg       0.64      0.64      0.64        44
```

Confusion matrix:



Conclusion:

The model yielded accuracy of 77% in predicting possibility of diabetes in the patients.

References:

https://en.wikipedia.org/wiki/Support_vector_machine

Prof. Dr.Changyu Chen Slides

Data from kaggle

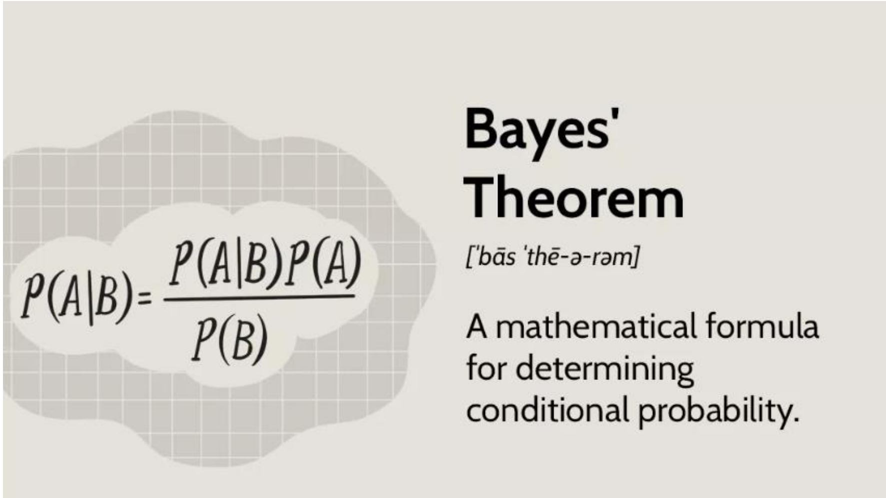
Towards Data Science: SVM Model

2. Naïve Bayes Classification:

The Naive Bayes Classifier is a popular supervised machine learning technique that is based on Bayes' theorem. This classic and feasible method performs well with large datasets and sparse matrices, such as pre-processed text data that generates millions of vectors based on the number of words in a dictionary. It performs admirably in document categorization projects, excels in text data projects such as sentiment data analysis, and excels in predicting categorical data in projects such as credit card fraud classification.

Bayes Theorem:

In probability theorem, bayes theorem gives the probability of an event according to the past outcomes of the event.



Experiment:

Python was used throughout this project. NumPy, Pandas, Scikit-Learn, Seaborn, and Matplotlib were used to achieve the same results.

Dataset:

The dataset describes the attributes which are responsible for diabetes in a person.

```
RangeIndex: 646 entries, 0 to 645
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               646 non-null    int64  
 1   Glucose                   646 non-null    int64  
 2   BloodPressure             646 non-null    int64  
 3   SkinThickness             646 non-null    int64  
 4   Insulin                   646 non-null    int64  
 5   BMI                       646 non-null    float64
 6   DiabetesPedigreeFunction  646 non-null    float64
 7   Age                       646 non-null    int64  
 8   Outcome                   646 non-null    int64  
```

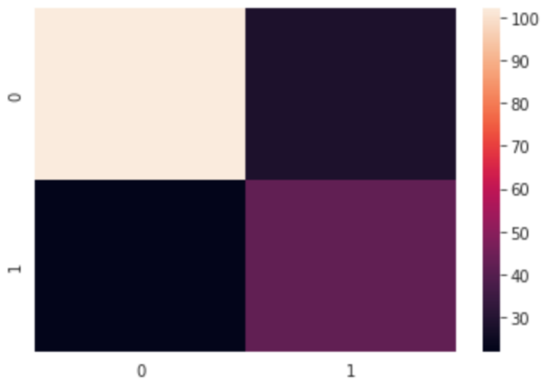
Results:

The model yielded an accuracy of 72% in predicting the occurrence of diabetes.

```
Classification :
              precision    recall  f1-score   support

         0.0       0.82      0.78      0.80       130
         1.0       0.60      0.66      0.63        64

    accuracy                           0.74       194
   macro avg       0.71      0.72      0.72       194
weighted avg       0.75      0.74      0.74       194


 F1:
 0.626865671641791

 Recall :
 0.65625

 Precision :
 0.6
```

Confusion Matrix:

<AxesSubplot:>

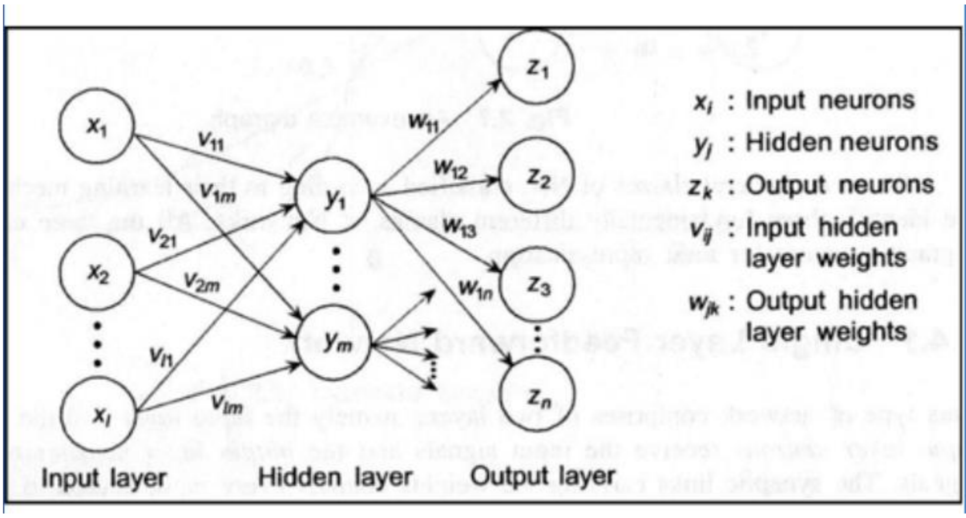


References:

- Wikipedia
- Lecture slides
- Kaggle Dataset
- Geeks for Geeks-naive-bayes-classifiers

3. Back-propagation

Back propagation: The core feature of neural network training is backpropagation. It is a method of optimizing neural network weights by providing the inputs of the past error rate data in the recent iteration. By doing so, you can make the model more reliable because the generalization increases as a result of error reduction.



In the above image, the input is changed by hidden layers and an output is resulted. Each propagation is handled by set of weights (and biases). To reduce the error between the outputs it maps from the given inputs and the expected outputs, the network must change these weights during training. The weights are modified using the gradient descent optimization method for each iteration.

Loss function: $w(n+1) = w(n) - \epsilon(\partial L/\partial w)$, L - the loss function ,ϵ - the learning rate

Experiment:

Dataset: The dataset describes the attributes which are responsible for diabetes in a person.

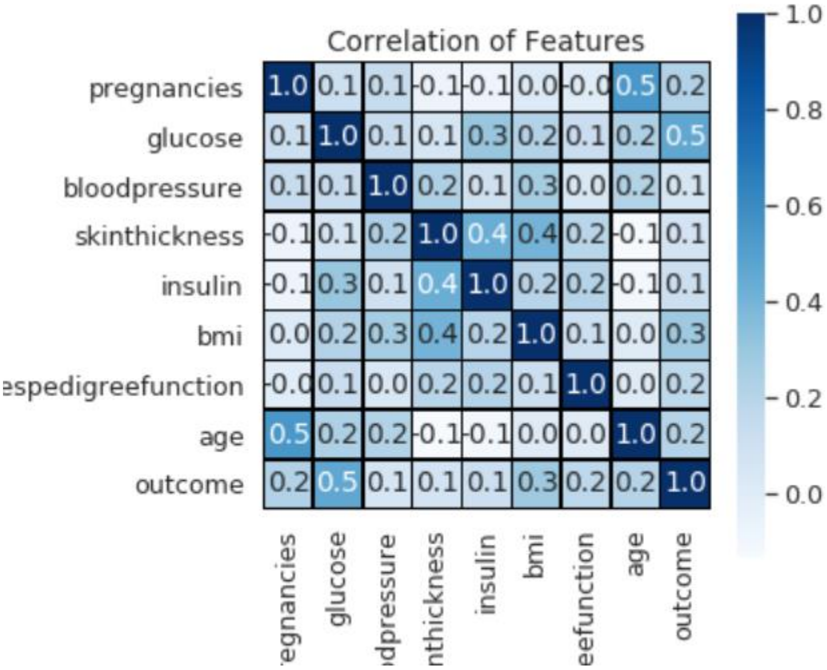
```
Data columns (total 9 columns):
Pregnancies                 648 non-null int64
Glucose                     648 non-null int64
BloodPressure               648 non-null int64
SkinThickness               648 non-null int64
Insulin                     648 non-null int64
BMI                         648 non-null float64
DiabetesPedigreeFunction    648 non-null float64
Age                         648 non-null int64
Outcome                     648 non-null int64
```

First few rows of the data:

pregnancies	glucose	bloodpressure	skinthickness	insulin	bmi	diabetespedigreefunction	age	outcome
0	162	76	56	100	53.2	0.759	25	1
6	111	64	39	0	34.2	0.260	24	0
2	107	74	30	100	33.6	0.404	23	0
5	132	80	0	0	26.8	0.186	69	0
0	113	76	0	0	33.3	0.278	23	1

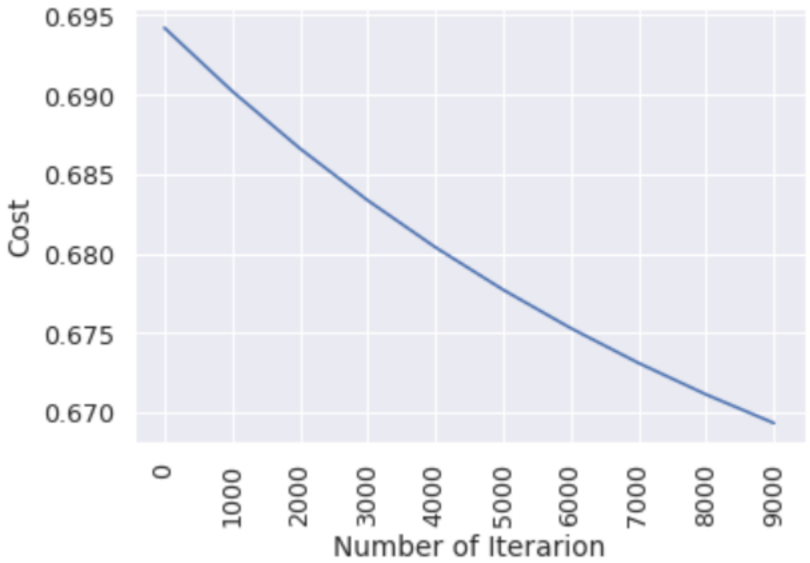
Feature selection:

Using correlation matrix, the features with highest correlation are removed.



Results:

The back-propagation is iterated for 10000 times and the optimal accuracy is obtained according to the cost after many iterations.
The cost decreased when iterations increased.



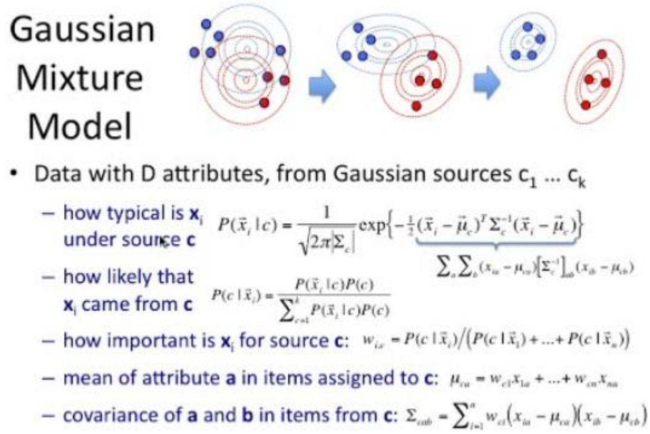

```
Train Accuracy: 64%
```

References:

- Wikipedia
- Lecture Slides
- Kaggle dataset
- Neptune website- [backpropagation-algorithm]

4. Gaussian Mixture Classification:

A Gaussian Mixture is a function composed of many Gaussian distributions, each identified by k 1,..., K, where K is the number of clusters in our dataset. A Gaussian mixture model (GMM) seeks a combination of multidimensional Gaussian probability distributions to best represent any input dataset.



Dataset:
The dataset used was supermarket customer data,

	ID	Year_Birth	Income	Kidhome	Teenhome	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	...	NumWebVisitsMonth
count	1595.000000	1595.000000	1571.000000	1595.000000	1595.000000	1595.000000	1595.000000	1595.000000	1595.000000	1595.00000	...	1595.000000
mean	5512.365517	1968.733542	51966.144494	0.452038	0.510972	48.858934	298.881505	26.253918	163.596865	36.91348	...	5.356113
std	3265.308270	12.104515	26372.245136	0.537829	0.541014	28.901938	332.447385	40.024696	227.083699	54.82033	...	2.411511
min	0.000000	1893.000000	1730.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	...	0.000000
25%	2671.500000	1959.000000	34884.500000	0.000000	0.000000	24.000000	23.000000	1.000000	15.000000	2.00000	...	4.000000
50%	5370.000000	1970.000000	51148.000000	0.000000	0.000000	49.000000	168.000000	8.000000	62.000000	11.00000	...	6.000000
75%	8369.500000	1977.000000	68316.500000	1.000000	1.000000	74.000000	505.000000	33.000000	216.500000	49.00000	...	7.000000
max	11191.000000	1996.000000	666666.000000	2.000000	2.000000	99.000000	1493.000000	199.000000	1725.000000	259.00000	...	20.000000

Results:
The clusters are based on customer spending and frequency of visit.

References:

- Wikipedia
- Lecture Slides
- Kaggle dataset
- Towards Datascience- gaussian-mixture-models