Feature Selection and Classification of Type II Diabetes on High Dimensional Dataset

Priya Vinoth

Funding: No specific funding was received for this work.
Potential competing interests: No potential competing interests to declare.

Abstract

Information mining is a methodology of bringing huge models utilizing recorded information. It is normally utilized in different real applications to be express web records, double dealing distinctive confirmation talk attestation, human organizations, and so forth. Reenacted insight includes are utilized in information mining to imagine the future occasion subject to the models conveyed utilizing solid information. All the highlights got during information assortment may not be altogether important to the objective class of the model. Highlight choice is a system which picks the best subset of highlights in dataset to upgrade the demonstration of an information mining or AI estimation. As of now, observational assessment is driven on Naïve Bayesian classifier utilizing Pima Indian Type II Diabetes dataset with all the highlights what's more the subset of the highlights picked by predefined python libraries. The presentation of Naïve Bayesian classifier is assessed on all of things to come subset of the dataset to consider the effect of the high dimensionality on the presentation of Naïve Bayes Classifier.

Priya Mohan[*]

Department of Computer Science, Bharathiar University, Coimbatore-641046, Tamilnadu, India

[*]Correspondence: priya.vinoth13@gmail.com

Keywords: Feature selection; Classification; Naïve Bayes; Pima Indian Type II Diabetes.

I. Introduction

Request is a framework used in managed learning. It endeavours to get acquainted with a limit using getting ready data containing data features and obvious yield. This limit is then used to foresee the class name of any authentic data incorporate.

A segment of the striking gathering systems are determined backslide, Naive Bayes classifier, Kth-nearest neighbour classifier and reinforce vector machines.

Right when the dimensionality of the data feature space is immense, plan gets snared. Feature Selection is a strategy that is applied to diminish the dimensionality of data or shed unimportant features to improve the farsighted precision. The inspiration driving performing feature decision in portrayal is two-cover. The first is redesigning the show of the classifier by picking simply significant features and ousting tedious, uproarious or pointless features. The second is to solidify the amount of features in circumstances where the portrayal figuring can't scale up to the size of the rundown of capacities, either in time or space.

Three favourable circumstances of performing feature decision before exhibiting your data are:

- Reduces Overfitting: Less tedious data suggests less opportunity to choose decisions reliant on clatter. Improves Accuracy: Less tricky data suggests exhibiting accuracy improves. Reduces Training Time: Less data infers that counts train snappier.
- Fundamental advances: examining for the perfect subset by applying fitting chase methodologies surveying the model over the conveyed subset.

At this moment, is given on the impact of dimensionality on utilization and portrayal execution of the Naive Bayes classifier using the Pima Indian Type II Diabetes and how feature assurance is used to improve the introduction of classifiers.

II. Naive Bayes Classifier

The Naïve Bayes classifier is a simple probabilistic classifier which is based up the application of Bayes theorem with two elementary presumptions. It presumes that the absence or presence of any particular feature of the class is unrelated to the absence or presence of any other feature. Therefore, it presumes that no pair of features are dependent on each other and that each and every feature is given the same weight. These assumptions are usually incorrect in real world applications.

The Naive Bayes classifier is based on conditional probabilities & uses Bayes' theorem which attains the probability of an event occurring, given the probability of another event that has already occurred.

If A represents the dependent event and B represents the prior event, Bayes' theorem can be represented as follows:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

So as to obtain the probability of B given A, the algorithm counts the number of instances where A and B occur together & then divides it with the number of instances where A occurs alone.

One advantage of this classifier is that it only requires a small amount of training data to assess the parameters that are necessary for classification. Since only independent variables are assumed, the variances of the variables for every class need to be calculated. The Naive Bayes classifier can be used for both binary & multi class classification problems.

III. Proposed Architecture

In order to study the impact of high dimensionality on Naive Bayes classifier, we initially evaluate the model by performance measures such as accuracy, precision, recall, and support using kfold cross validation by considering all eight feature attributes of the Pima Indian Type II Diabetes dataset.

To assess the impact of dimensionality, we reduce the numbers of features of the dataset from eight, to six, to four, to two and each subset is separately classified by using Naive Bayes classification Algorithm and the model is evaluated with the above mentioned performance measures.

We use the language Python as it is one of the most flexible languages and is great for working with machine learning algorithms. Python contains special libraries for machine learning namely scipy and humpy which great for linear algebra and getting to know kernel methods of machine learning.

The classes in the sklearn.feature_selection module of Python is used here for feature selection/dimensionality reduction on sample sets, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets.

Statistical tests can be used to select those features that have the strongest relationship with the output variable.

The scikit-learn library provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features.

class sklearn.feature_selection.SelectKBestSelect is used to select features according to the k highest scores.

The architectural diagram of the proposed methodology is shown in**Fig. 1.**

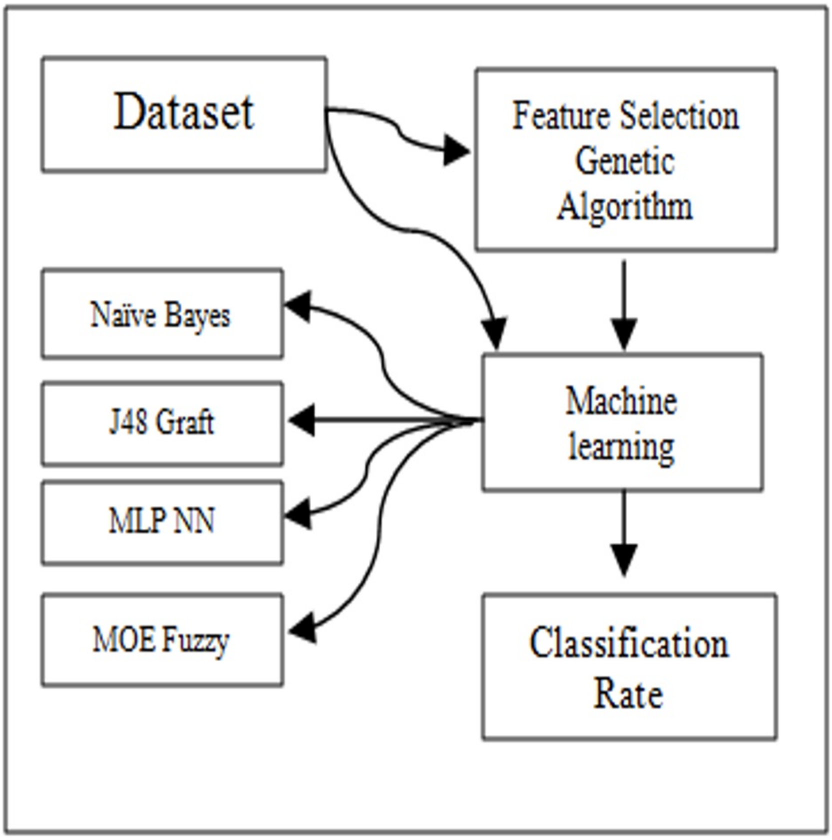

Fig. 1.

IV. Classification Algorithms Used in this Work

Classification is an important task in machine learning and data mining, which aims to classify each instance in the data into different groups. The feature space of a classification problem is a key factor influencing the performance of a classification learning algorithm.

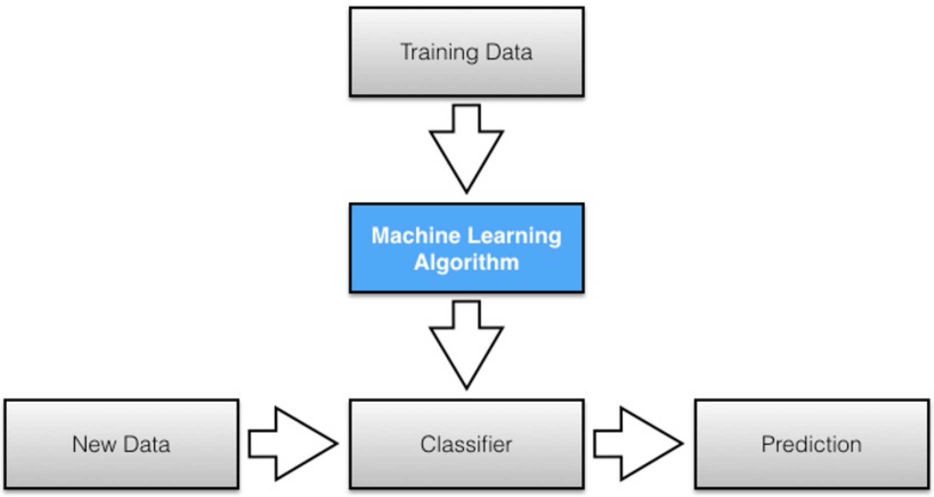

Fig. 2. Prediction Model Analysis

The Classification algorithms used in this research paper are

- Support vector machine (SVM)
- Naïve Bayes (NB)
- Random forest(RF)
- Logistic Regression(LR)
- K-Nearest Neighbor(KNN)
- Gradient Boosting Classifier(GBC)

V. Relative Work

In the healthcare domain, machine learning algorithms are widely used to predict the occurrence of a disease at an early stage. The researchers had tried to use a variety of classifiers to predict the diseases and have obtained good accuracy results. They classified and analyzed the performance using the universally accepted dataset from the UCI repository. The results were evaluated using the parameters like accuracy, sensitivity, and specificity. They performed the classification in 2 different cases, one with pre-processed data and the other without pre-processing.

VI. Framework for the Proposed Comparative Performance Study

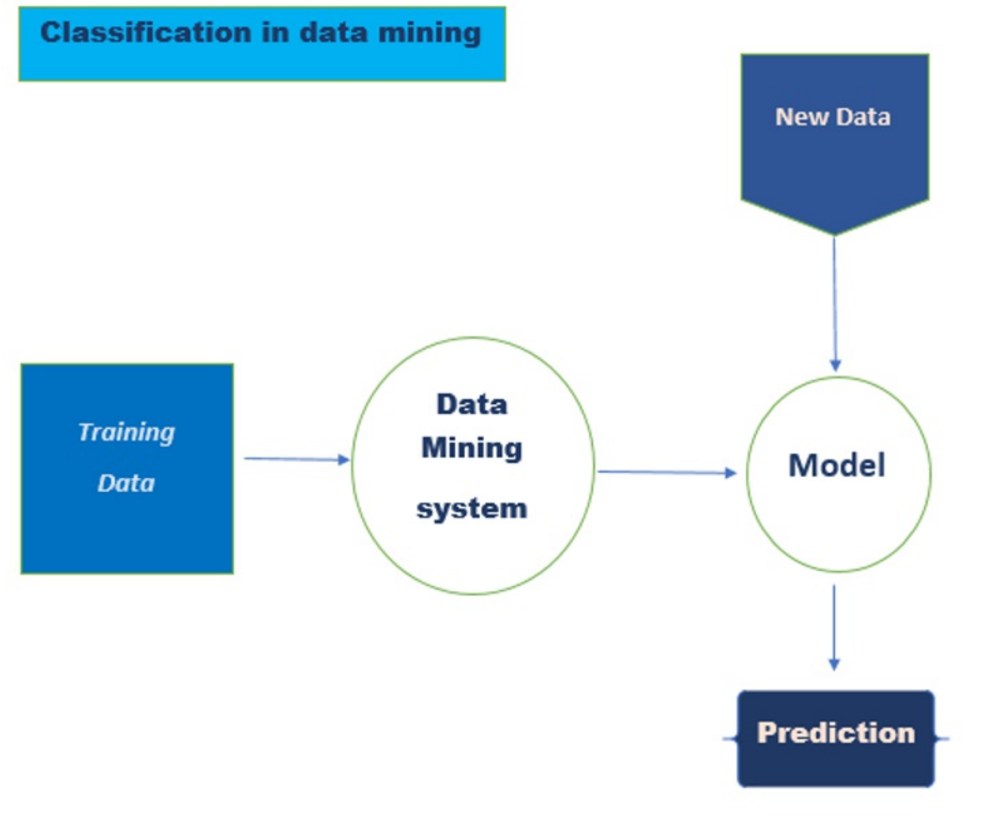

Fig. 3. Data mining System with classification Algorithms

VII. Methodology Followed

1. Loading of libraries

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

2. Loading of data

```
pima = pd.read_csv("C:/Users/Anu/Documents/diabetes.csv")
pima.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI \
0	6	148	72	35	0	33.6
1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1
4	0	137	40	35	168	43.1

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

Additional details about the attributes

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration after 2 hours in an oral glucose tolerance test
- Blood Pressure: Diastolic blood pressure (mm Hg)
- Skin Thickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index
- Diabetes Pedigree Function: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 or 1)

3. We then summaries our dataset

pima.shape()

```
(768, 9)
```

pima.describe()

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin \
count	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479
std	3.369578	31.972618	19.355807	15.952218	115.244002
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000
75%	6.000000	140.250000	80.000000	32.000000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000
mean	31.992578	0.471876	33.240885	0.348958
std	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.078000	21.000000	0.000000
25%	27.300000	0.243750	24.000000	0.000000
50%	32.000000	0.372500	29.000000	0.000000
75%	36.600000	0.626250	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

pima.groupby("Outcome").size()

```
Outcome
0    500
1    268
dtype: int64
```

4. To perform feature selection, we use the inbuilt Python module “SelectKBest” which is often used for feature selection (or dimensionality reduction) on very high-dimensional datasets. This class selects features according to the k highest scores.

from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import chi2

X = pima.iloc[:,0:8]

Y = pima.iloc[:,8]

select_top_4 = SelectK Best(score_func=chi2, k = 4)

fit = select_top_4.fit(X, Y)

features = fit.transform(X)

features[0:5]

```
[[148.    0.   33.6  50. ]
 [ 85.    0.   26.6  31. ]
 [183.    0.   23.3  32. ]
 [ 89.   94.   28.1  21. ]
 [137.  168.   43.1  33. ]]
```

So, the top performing features are Glucose, Insulin, BMI, Age

X_features = pd.DataFrame(data = features, columns = ["Glucose","Insulin","BMI","Age"])

X_features.head()

	Glucose	Insulin	BMI	Age
0	148.0	0.0	33.6	50.0
1	85.0	0.0	26.6	31.0
2	183.0	0.0	23.3	32.0
3	89.0	94.0	28.1	21.0
4	137.0	168.0	43.1	33.0

Y = pima.iloc[:,8]

Y.head()

```
0    1
1    0
2    1
3    0
4    1
Name: Outcome, dtype: int64
```

5. Standardization

It changes the attribute values to Guassian distribution with mean as 0 and standard deviation as 1. It is useful when the algorithm expects the input features to be in Guassian distribution.

from sklearn.preprocessing import StandardScalerrescaledX = StandardScaler().fit_transform(X_features)

X = pd.DataFrame(data = rescaledX, columns= X_features.columns)

X.head()

	Glucose	Insulin	BMI	Age
0	0.848324	-0.692891	0.204013	1.425995
1	-1.123396	-0.692891	-0.684422	-0.190672
2	1.943724	-0.692891	-1.103255	-0.105584
3	-0.998208	0.123302	-0.494043	-1.041549
4	0.504055	0.765836	1.409746	-0.020496

6. Binary Classification

from sklearn.model_selection import train_test_splitX_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state

```
=22, test_size =0.2)
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB
models = []
models.append(("Cross-validation score of NB: ", GaussianNB()))
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=10, random_state=22)
    cv_result = cross_val_score(model, X_train, Y_train, cv = kfold, scoring = "accuracy")
    names.append(name)
    results.append(cv_result)
for i in range(len(names)):
print(names[i], results[i].mean())
```

7. Accuracy Calculation

```
gnb = GaussianNB()
gnb.fit(X_train, Y_train)
predictions = gnb.predict(X_test)
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
print("Accuracy of NB: ", accuracy_score(Y_test, predictions))
print("Classification score of NB: ")
print(classification_report(Y_test, predictions))
conf = confusion_matrix(Y_test, predictions)
label = ["0","1"]
sns.heatmap(conf, annot=True, xticklabels=label, yticklabels=label)
```

```
Cross-validation score of NB:  0.7604970914859862
Accuracy of NB:  0.7337662337662337
Classification score of NB:
             precision    recall  f1-score   support

          0       0.74      0.92      0.82       100
          1       0.72      0.39      0.51        54

avg / total       0.73      0.73      0.71       154
```

VIII. Results

When our system is actualized and we've tried our classifier on execution measurements, for example, exactness, accuracy, review, f-measure and backing, we see the accompanying outcomes:

1. Performance of the Naive Bayes Classifier without feature selection:

```
C:\Users\Anu\AppData\Local\Programs\Python\Python35-32>test4.py
Cross-validation score of NB:  0.7668693812797461
Accuracy of NB:  0.7077922077922078
Classification score of NB:
             precision    recall  f1-score   support

          0       0.73      0.87      0.79       100
          1       0.63      0.41      0.49        54

avg / total       0.70      0.71      0.69       154
```

2. Performance of the Naive Bayes Classifier after feature selection:

a. With 2 feature subset

```
C:\Users\Anu\AppData\Local\Programs\Python\Python35-32>test4.py
[[148.   0.]
 [ 85.   0.]
 [183.   0.]]
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
Cross-validation score of NB:  0.7345319936541512
Accuracy of NB:  0.7207792207792207
Classification score of NB:
             precision    recall  f1-score   support

          0       0.72      0.93      0.81       100
          1       0.72      0.33      0.46        54

avg / total       0.72      0.72      0.69       154
```

b. With 4 feature subset

```
C:\Users\Anu\AppData\Local\Programs\Python\Python35-32>test4.py
[[148.    0.    33.6  50. ]
 [ 85.    0.    26.6  31. ]
 [183.    0.    23.3  32. ]
 [ 89.   94.    28.1  21. ]
 [137.  168.    43.1  33. ]]
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
Cross-validation score of NB:  0.7604970914859862
Accuracy of NB:  0.7337662337662337
Classification score of NB:
             precision    recall  f1-score   support

          0       0.74      0.92      0.82       100
          1       0.72      0.39      0.51        54

avg / total       0.73      0.73      0.71       154
```

c. With 6 feature subset

```
C:\Users\Anu\AppData\Local\Programs\Python\Python35-32>test4.py
[[  6.  148.   35.    0.   33.6  50. ]
 [  1.   85.   29.    0.   26.6  31. ]
 [  8.  183.    0.    0.   23.3  32. ]
 [  1.   89.   23.   94.   28.1  21. ]
 [  0.  137.   35.  168.   43.1  33. ]
 [  5.  116.    0.    0.   25.6  30. ]
 [  3.   78.   32.   88.   31.   26. ]]
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
Cross-validation score of NB:  0.752432575356954
Accuracy of NB:  0.7207792207792207
Classification score of NB:
             precision    recall  f1-score   support

          0       0.74      0.87      0.80       100
          1       0.65      0.44      0.53        54

avg / total       0.71      0.72      0.71       154
```

IX. Conclusion

It is observed that the Naive Bayes classifier works exceptionally well with the reduced feature subset consisting of 4 features as compared to the original 8 feature attribute set. However, the accuracy of this classifier for the reduced feature subset consisting of 2 and 6 attributes, though similar, was lesser than the accuracy score of the 4-feature subset and greater than the original attribute set.

The observations have been summarised on Table1 given below.

Table 1. Observation table for classifier performance

Dimensionality	Classifier Accuracy Performance
3	0.6207
5	0.6337
7	0.6207
9	0.6077

This phenomenon is also termed as “Curse of Dimensionality” and can be observed in Fig. 4 by charting the data we collected.

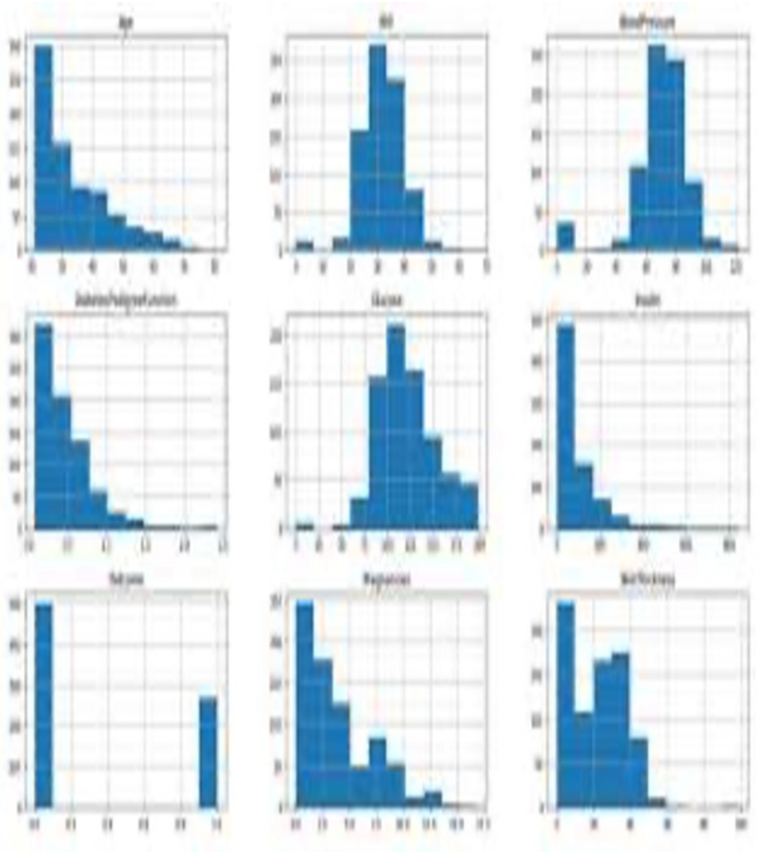

Fig. 4. The chart exhibits that as the dimensionality of a dataset expands, the classifier's presentation additionally increments till where it arrives at the ideal number of highlights. Expanding the dimensionality further without progressively number of preparing tests will bring about a huge decline in the presentation of the classifier.

Therefore, we come to the conclusion that the optimal number of feature attributes for the Pima Indian Type II Diabetes subset is four.

The method used for feature selection in our paper was the Univariate Selection method. Other models used for feature selection are Recursive Feature Elimination, Principal Component Analysis and Feature Importance.

Recursive Feature Elimination

The Recursive Feature Elimination (or RFE) works by recursively evacuating characteristics and building a model on those qualities that remain.

It utilizes the model precision to distinguish which qualities (and mix of characteristics) contribute the most to foreseeing the objective property.

Head Component Analysis

Head Component Analysis (or PCA) utilizes straight polynomial math to change the dataset into a packed structure.

For the most part this is known as an information decrease system. A property of PCA is that you can pick the quantity of measurements or head segment in the changed outcome.

Highlight Importance

Stowed choice trees like Random Forest and Extra Trees can be utilized to assess the significance of highlights. The method each of these models follow, are given in the table below:

Table 2. Method used and parameters considered by models used for feature selection

	Univariate	**RFE**	**PCA**	**Feature Importance**
Method Used	Statistical Test (Chi square used here)	Recursively removes attributes and builds a model	Linear algebra to transform dataset into a compressed form	Bagged decision trees. eg- Random Forest
Parameters Considered	Feature having the strongest relationship with output variable	Attribute contributing the most to predict target variable	Data reduction to the number of dimensions the user wants	Most important feature

References

- P. C. Thirumal and N. Nagarajan, — Utilization of data mining techniques for diagnosis of diabetes mellitus — A case study, ‖ ARPN J. Eng. Appl. Sci., vol. 10, no., pp. 8–13, 2015.
- R. Valdez, P.W. Yoon, N. Qureshi, R.F. Green, M.J. Khoury, —Family history in public health practice: a genomic tool for disease prevention and health promotion,‖ Ann. Rev. public. health. no. 31, pp. 69-87, 2010.
- International Diabetes Federation, — Idf diabetes atlas 2017, ‖ 2017.
- S.M. Grundy, — Obesity, metabolic syndrome, and cardiovascular disease, ‖ J. Clin. Endocrinol. Meta., no. 89, pp. 2595-2600, 2004.
- M. Mashayekhi, F. Prescod, B. Shah, L. Dong, K. Keshavjee, A. Guergachi, — Evaluating the performance of the Framingham Diabetes Risk Scoring Model in Canadian electronic medical records,‖ Can. J. diabet., no. 39, pp. 152-156, 2015.
- Retrieved http://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, Accessed 27th Jul 2018.
- http://www.who.int/news-room/fact-sheets/detail/diabetes retrieved 27/07/2018.
- M. Opper, O. Winther, Gaussian processes for classification: mean-field algo-rithms, Neural Comput. 12 (20 0 0) 2655–2684

- Alam, T.M., Iqbal, M.A., Ali, Y., Wahab, A., Ijaz, S., Baig, T.I., Hussain, A., Malik, M.A., Raza, M.M., Ibrar, S. and Abbas, Z., 2019. A model for early prediction of diabetes. Informatics in Medicine Unlocked, 16, p.100204.
- Kaur, H. and Kumari, V., 2018. Predictive modelling and analytics for diabetes using machine learning approach. Applied Computing and Informatics.
- Sisodia, D. and Sisodia, D.S., 2018. Prediction of diabetes using classification algorithms. Procedia computer science, 132, pp.1578-1585
- Shu, T., Zhang, B. and Tang, Y.Y., 2017. An extensive analysis of various texture feature extractors to detect Diabetes Mellitus using facial specific regions.Computers in biology and medicine, 83, pp.69-83.
- Hayashi, Y. and Yukita, S., 2016. Rule extraction using Recursive-Rule extraction algorithm with J48 graft combined with sampling selection techniques for the diagnosis of type 2 diabetes mellitus in the Pima Indian dataset.Informatics in Medicine Unlocked, 2, pp.92-104.
- Santhanam, T. and Padmavathi, M.S., 2015. Application of K-means and genetic algorithms for dimension reduction by integrating SVM for diabetes diagnosis. Procedia Computer Science, 47, pp.76-83.
- Aslam, M.W., Zhu, Z. and Nandi, A.K., 2013. Featuregeneration using genetic programming with comparative partner selection for diabetes classification.Expert Systems with Applications, 40(13), pp.5402-5412.
- Yang, X. S. (2009). F-irefly algorithms for multimodal optimization. In 5th symposium on stochastic algorithms, foundations and applications, SAGA 2009 (pp 169–178).
- L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, Kluwer Academic Publishers. Manufactured in The Netherlands.
- Utkin, L.V., 2019. An imprecise extension of SVM-based machine learning models.Neurocomputing,331, pp.18-32.