Assiut University
Faculty of Computer and Information
Department Computer Science

Report for Supervised Classification Model for Diabetes

Made By:

George Yohana Adeeb

(4th Software Engineering)

Abstract

In this project, we were asked to experiment with a real-world dataset, and to explore how machine learning algorithms can be used to find the patterns in data. We were expected to gain experience using a common data mining and machine learning libraries and were expected to submit a report about the dataset and the algorithms used. After performing the required tasks on a dataset of my choice, herein lies our final report.

- ## Problem Definition:

Diabetes-Mellitus refers to the metabolic disorder that happens from misfunction in insulin secretion and action. It is characterized by hyperglycemia. The persistent hyperglycemia of diabetes leads to damage, malfunction, and failure of different organs such as kidneys, eyes, nerves, blood vessels and heart. In the past decades several techniques have been implemented for the detection of diabetes. The diagnosis of diabetes is very important now a days. Here, there are various techniques, their classification and implementation using various types of software tools and techniques. The diagnosis of diabetes can be done using K-fold cross validation and classification, Vector support machine, Random Forest method, Data Mining Algorithm, etc.

- # Summary Of Work:

  - **Exploratory Data Analysis**
  - **Data Pre-processing**
  - **Model Selection and Optimization**
  - **Evolution / Final Submission**

- # Dataset:

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective is to predict based on diagnostic measurements whether a patient has diabetes. All patients here are females at least 21 years old of Pima Indian heritage.

As we see from our output, the attributes of the dataset are as follows:

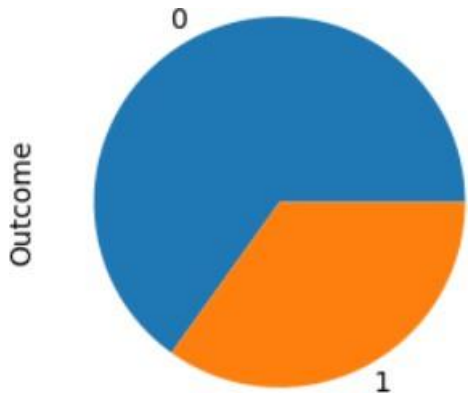
(All numeric valued) (768 row)

1. Number of times pregnant
2. Plasma glucose concentration 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/ (height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

Class Distribution: (class value 1 is interpreted as "tested positive for diabetes").

- # Problems Detected in the Data:

1. The number of examples is small to train the model. (768 examples)
2. Imbalanced data: refers to those types of datasets where the target class has an uneven distribution of observations.



(Figure 1)

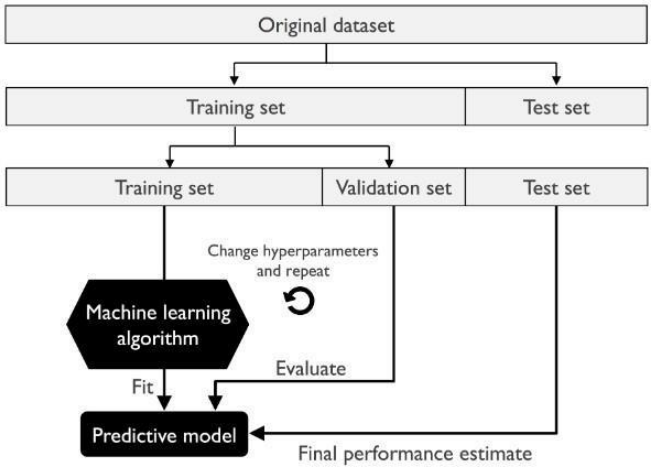
3. The extent of the numbers varies in the features.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

(Figure 2)

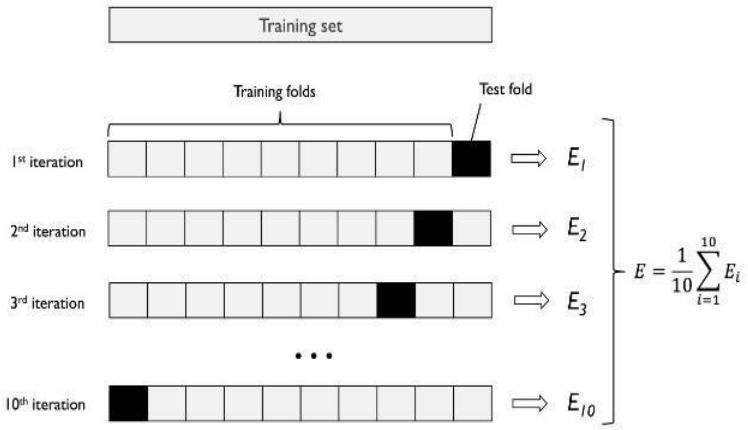
- **Data Splitting Methods:**

1. Holdout Method



(Figure 3)

2. K-Fold Cross-Validation method



(Figure 4)

- # Data Pre-processing:

1. Data Sampling

One approach to addressing the problem of class imbalance is to randomly resample the training dataset. The two main approaches to randomly resampling an imbalanced dataset are to delete examples from the majority class, called **under-sampling**, and to duplicate examples from the minority class, called **over-sampling.**

For better performance, we will use the **random over sampler** technique to train as many examples as possible.

2. Feature Scale

One approach to addressing the problem of the extent of the numbers varies in the features. It involves modifying values by one of two primary methods: Normalization or Standardization. Normalization takes the input values and modifies them to lie between [0,1].

Standardization methods modify the values so that they center at 0 and have a standard deviation of 1.

For better performance, we will use the **normalization** and especially **min-max normalization,** because there are no existence of negative numbers which leads to better performance of the model to train data.

- # Model Selection and Optimization:

1. ML Methods:

- Logistic Regression
- Support Vector Machine
- Random Forest Classifier
- Extra-Trees Classifier

We will evaluate the models that perform well in structured data. It's known that tree-based methods are state of the art when we are talking about structured data thus, we will train and evaluate Random Forests and Extra-Tree Classifier.

Linear models like logistic regression and support vector machines didn't perform well in this problem so we won't consider them in our final submission.

2. Hyper-Parameter Tunning:

We have used one of the methods used for hyperparameter tunning (Grid search technique) to find the right parameters for each ML method.

(max_iter / C: Regularization parameter / max_depth / n_estimators)

- # Performance Measures

Performance measures are used to evaluate the effectiveness of a machine learning model in making predictions on a given dataset.

1. **Accuracy:** is the most common performance measure and is simply the proportion of correct predictions made by the model over the total number of predictions. However, accuracy can be misleading in cases where the dataset is imbalanced, i.e., the number of samples in one class is much larger than in the other.

2. **Recall:** also known as sensitivity, measures the proportion of true positives (correctly predicted positive samples) over the total number of actual positive samples. Recall is useful when the goal is to identify all positive samples, as it ensures that the model is not missing any positive samples.

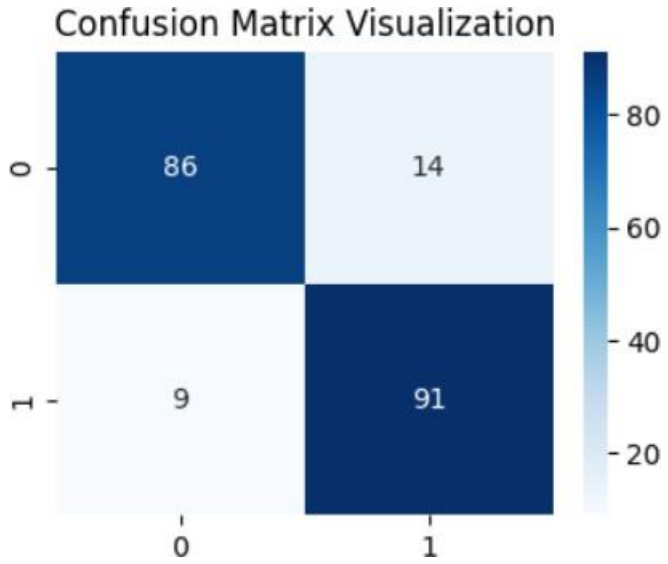
3. **Precision measures**: the proportion of true positives over the total number of predicted positive samples. Precision is useful when the goal is to minimize false positives, as it ensures that the positive predictions made by the model are accurate.

4. **The F1 score:** is a harmonic mean of precision and recall, and it provides a balanced measure of the model's performance. F1 score is particularly useful when the dataset is imbalanced, and the goal is to optimize both precision and recall.

- # Error Analysis

The confusion matrix is a useful tool for summarizing the performance of a machine learning model, while error analysis is a process of analyzing the errors made by the model to identify patterns and trends in the errors. The confusion matrix can be used as a starting point for error analysis, providing a detailed breakdown of the model's performance on each class in the dataset.

The confusion matrix resulting from the final model:



(Figure 5)

• Results

1. Logistic Regression

Holdout method

```
recall:  0.8
precision:  0.8010839020473706
f1_score:  0.7998198378540687
accuracy score  0.8
```

K-Fold cross validation method

	precision	recall	f1-score	support
0	0.79	0.78	0.79	3000
1	0.78	0.80	0.79	3000
accuracy			0.79	6000
macro avg	0.79	0.79	0.79	6000
weighted avg	0.79	0.79	0.79	6000

```
best params:  {'max_iter': 250}
CV accuracy scores: [0.732 0.754]
CV accuracy: (0.743, 0.0110000000000001)
```

2. Support Vector Classifier

Holdout method

```
recall:  0.7849999999999999
precision:  0.7864033765450709
f1_score:  0.7847363019699131
accuracy score  0.785
```

K-Fold cross validation method

	precision	recall	f1-score	support
0	0.76	0.76	0.76	1000
1	0.76	0.76	0.76	1000
accuracy			0.76	2000
macro avg	0.76	0.76	0.76	2000
weighted avg	0.76	0.76	0.76	2000

```
best params:  {'C': 10}
CV accuracy scores: [0.762 0.804]
CV accuracy: (0.783, 0.02100000000000002)
```

- # Results

3. Random Forest Classifier

Holdout method

```
recall:  0.85
precision:  0.8617197188921042
f1_score:  0.8487750781328762
accuracy score  0.85
```

K-Fold cross validation method

	precision	recall	f1-score	support
0	0.80	0.76	0.78	1500
1	0.77	0.81	0.79	1500
accuracy			0.79	3000
macro avg	0.79	0.79	0.79	3000
weighted avg	0.79	0.79	0.79	3000

```
best params:  {'max_depth': 10, 'n_estimators': 750}
CV accuracy scores: [0.804 0.86 ]
CV accuracy: (0.8320000000000001, 0.02799999999999997)
```

4. Extra-Trees Classifier

Holdout method

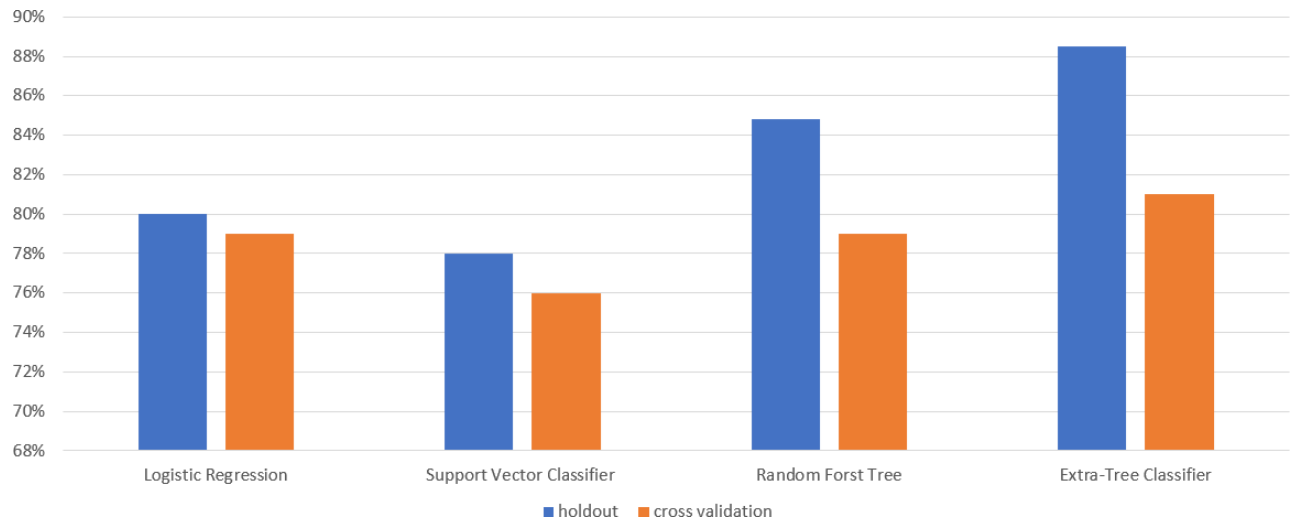
```
recall:  0.885
precision:  0.8859649122807018
f1_score:  0.8878048780487805
accuracy score  0.885
```

K-Fold cross validation method

	precision	recall	f1-score	support
0	0.83	0.78	0.80	2000
1	0.79	0.84	0.81	2000
accuracy			0.81	4000
macro avg	0.81	0.81	0.81	4000
weighted avg	0.81	0.81	0.81	4000

```
best params:  {'n_estimators': 500}
CV accuracy scores: [0.81 0.88 0.84 0.8  0.86 0.86 0.96 0.95
CV accuracy: (0.882, 0.055461698495448165)
```

- **Results Analysis (Based on F1 score)**



(Figure 6)

After Reviewing the results, we decided to use the model that use Extra-Tree method using holdout method for splitting the data and consider it our final submission.

- # Conclusion:

Literature Review:

https://www.neliti.com/publications/257905/classification-of-diabetes-mellitus-using-machine-learning-techniques

Our model resulted in a better performance compared with the model in the literature review because linear models like support vector classifier didn't perform better than tree-based methods in structured data.

- # Future Work:

1. Improve the model by bringing in new data using a data integration technique.
2. Built a large model with big data and merged it with a web application using it in hospitals to detect diabetes in women.