Diabetes prognosis in female patients using machine learning classifiers

1. Introduction

This assignment solves the supervised learning problem in which we predict whether or not a female will develop diabetes based on various features provided in a dataset. Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function and Age are the eight factors that influence diabetes outcomes in patients. The statistics for the given dataset are as shown below, calculating mean, variance, and so on.

	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

The following is an important observation made from the dataset:

- There are missing values in the dataset (0) for many features which we are imputing with the mean values of each feature respectively.
- The outcome class is imbalanced with more negative values than positive values. Class imbalance can be fixed with the use of SMOTE oversampling technique (not used as it requires an external package to be imported).
- If the number of pregnancies, glucose, BMI, and age exceed the mean, the chances of developing diabetes increase due to the presence of a strong correlation, as observed by the WEKA(Data Mining) tool.

2. Data Preparation

Steps followed for data preparation:

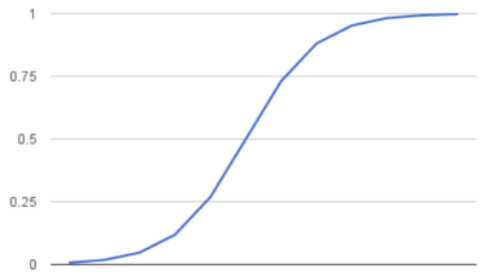
1. **Data Imputation with Mean:** Here, missing values or 0 values were replaced by the feature's mean. Blood Pressure, Skin Thickness, Insulin, and BMI were computed, but Pregnancies were excluded because they can have a value of 0.
2. **Data Splitting:** Raw data was split into train and test datasets using the train test split class (train_test_split) imported from sklearn model selection with a standard split of 80% train set and 20% test set.
3. **Feature Scaling:** Preprocessed data may have attributes with a mix of scales for a variety of values. The scaling procedure entails re-scaling the feature such that all of the data is normalized, resulting in a conventional distribution with a mean of 0 and a standard deviation of 1.
4. **Feature Selection:** An additional step was performed to consider only the important elements of the dataset that have a significant influence on the class's outcome. This was done to prevent the data from being overfitted. Used Correlation Attribute Evaluator of Weka (Data Mining Tool) to get the dominating features such as Glucose, BMI, Age, Pregnancies and Diabetes Pedigree Function which had the highest ranking.

3. Classifiers

Model selection gets influenced by factors such as underfitting, overfitting, generalization error, and validation. The two classifiers used are - Logistic Regression and Support Vector Machine. These two classifiers help distinguish diabetic patients from non-diabetic patients based on the dataset provided.

Logistic Regression Classifier

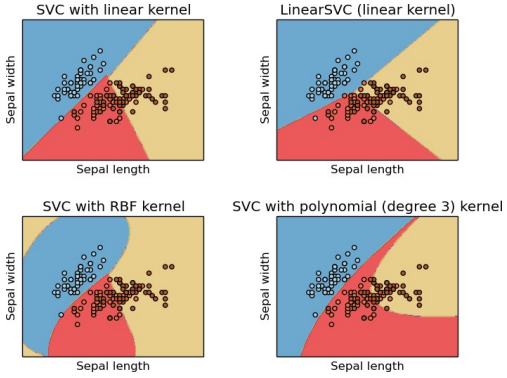
LR classifier support binary classification problems and thus provide high prediction accuracy. Logistic regression is based on statistical methodologies and is susceptible to overfitting. A LR classifier is a statistical approach for assessing a dataset in which one or more independent factors influence the outcome. A binary variable, also known as the dependent variable, is used to assess the outcome. The purpose of logistic regression is to determine the model that best describes the connection between a dependent variable and a set of independent variables. The outcome must be either categorical or discrete. It can be Yes or No, 0 or 1, true or False, and so on, but instead of presenting the exact values as 0 and 1, it presents the probabilistic values that fall between 0 and 1. Logistic Regression is used to categorize observations using many forms of data and can quickly discover the most efficient factors for classification. Instead of fitting a regression line, we fit a "S" shaped logistic function that predicts two maximum values in logistic regression (0 or 1).



$$\ln\left(\frac{p}{(1-p)}\right) = b_0 + b_1 * x$$

Support Vector Machine Classifier

SVM works on linearly separating vectors such as diabetic and non-diabetic vectors, which aids in the identification of diabetic patients. SVM is based on geometrical features of the data and has a lower risk of overfitting. SVM is a supervised learning model with related learning algorithms that examine data used for classification and regression analysis in machine learning. Given a series of training examples, each of which is labeled as belonging to one of two categories, an SVM training algorithm creates a model that assigns new examples to one of the two categories, resulting in a non-probabilistic binary linear classifier. The SVM algorithm's purpose is to find the optimum line or decision boundary for categorizing n-dimensional space so that we may simply place fresh data points in the correct category in the future. A hyperplane is the optimal choice boundary.



$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\vec{w}\cdot\vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2,$$

We are given a training dataset of n points of the form: $(\vec{x}1, y1)$ ..... $(\vec{x}n, yn)$. Where yi are either 1 or -1, each of which indicates the class to which the point $(\vec{x}1)$ belongs. $(\vec{x}1)$ denotes a p-dimensional real vector. A hyperplane can be expressed as a collection of points $(\vec{x}1)$ satisfying: $\vec{w}.\, x - b = 0$, where b is a vector. Where $\vec{w}$ is the (not normalized) vector to the hyperplane. The extended SVM for scenarios when the data are not linearly separated is given as: Where the parameter λ specifies the tradeoff between increasing the margin-size and guaranteeing that the $\vec{x}1$ lies on the correct side of the margin.

4. Evaluation

We used numerous performance indicators from the scikit-learn library to evaluate the performance of each classifier. Some of the metrics we included in our solution are as follows:

Classification Report - Logistic Regression

	precision	recall	f1-score	support
0	0.83	0.91	0.87	107
1	0.73	0.57	0.64	47
accuracy			0.81	154
macro avg	0.78	0.74	0.75	154
weighted avg	0.80	0.81	0.80	154

Classification Report - Support Vector Machine

	precision	recall	f1-score	support
0	0.81	0.88	0.84	107
1	0.66	0.53	0.59	47
accuracy			0.77	154
macro avg	0.73	0.71	0.72	154
weighted avg	0.76	0.77	0.77	154

K-Folds Cross Validation: The training set is divided into k smaller sets, and the model is trained with data from k-1 of the folds. The resulting model is evaluated against the remaining data to generate performance measures such as accuracy, which is the average of all test result values. The classifier's hyperparameters are then tweaked using GridSearchCV to produce the optimum parameters for the model in prediction.

Confusion Matrix: We were able to determine the correctness and accuracy of classifier models using the Confusion Matrix. The actual classification values are shown in the figure as columns, and the anticipated classification values are shown as rows. The outcome in our challenge is binary (0 or 1), which means the classifier will predict whether the person has diabetes (1) or not (0). In our scenario, it's critical to reduce False negatives, because the model shouldn't overlook occasions when individuals who truly have diabetes were predicted with no diabetes.

		Actual 0	Actual 1
Predicted	0	True Negatives(TN)	False Negatives(FN)
	1	False Positives(FP)	True Positives(TP)

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{F1} = \frac{TP}{TP+\frac{FN+FP}{2}}$$

Accuracy: The number of right predictions (TP + TN) made by the model across all types of predictions.

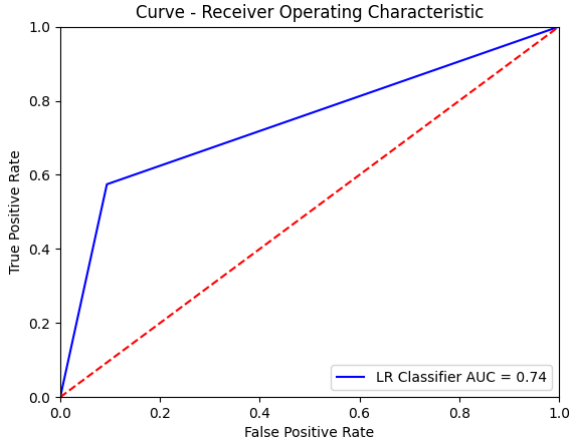
Precision: Precision is a metric that assesses the accuracy of positive predictions. In our case, it shows us how many of the individuals we diagnosed with diabetes actually have diabetes.

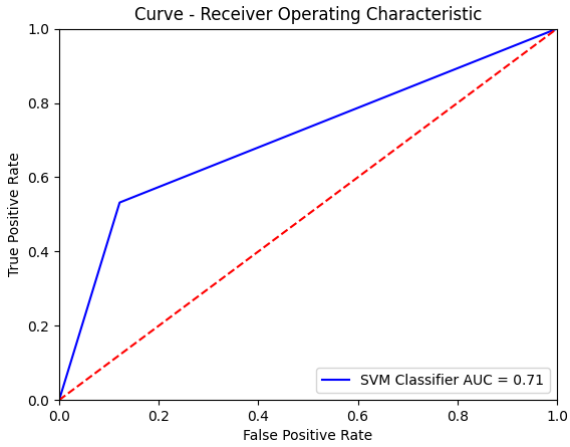
Recall: The ratio of positive cases accurately detected by the classifier is referred to as recall. In our case, it shows us what fraction of patients with diabetes were diagnosed as such by the classifier.

F1 Score: The harmonic mean of precision and recall is used to calculate the F1 score. In contrast to the standard mean, which provides equal weight to all values, the harmonic mean lends more weight to low values.

AUC and ROC Curve: The Receiver Operating Characteristic (ROC) curve compares the genuine positive rate (also known as recall) to the false positive rate. The false positive rate (FPR) is the proportion of negative events

that are misclassified as positive. It is equal to one minus the true negative rate, which is the proportion of negative cases that are accurately categorized as negative. The Area Under Curve (AUC) is a metric used to assess the performance of a classifier. The higher the ROC AUC score, the better the classifier predicts. The ROC Curve allows us to examine the model's performance over all conceivable thresholds.





5. Conclusion

Based on the obtained accuracies, we can see that the accuracies vary as we process the data with different contexts (see figure below on the right). Result obtained is:

LR classifier provides highest accuracy of **82.46% with raw data and feature scaling** and provides highest accuracy of **80.51% after we compute imputation with mean** on the feature set.

SVM Classifier gives the best accuracy of **75.73% using Grid Search and applying K-Fold Cross Validation** (K=5) equal sub samples for knowing the best model and parameters.

```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (numeric): 9 Outcome):
        Correlation Ranking Filter
Ranked attributes:
 0.4666    2 Glucose
 0.2927    6 BMI
 0.2384    8 Age
 0.2219    1 Pregnancies
 0.1738    7 DiabetesPedigreeFunction
 0.1305    5 Insulin
 0.0748    4 SkinThickness
 0.0651    3 BloodPressure
```

S.No.	Classifier Name	Accuracy % - With Raw Data	Accuracy % - With Feature Scaling Only	Accuracy % - With Mean Imputation and feature scaling	Accuracy % - Using Grid Search for knowing best model, parameters and applying K-Fold Cross Validation (K=5)	Standard Deviation %	Best parameters
1	Logistic Regression	82.46	82.46	80.51	75.39	± 5.03	{'C': 1, 'solver': 'newton-cg'}
2	Support Vector Machine	79.22	79.22	77.27	75.73	± 3.77	{'C': 10, 'kernel': 'linear'}

Dominating Features

The entire raw data set was analyzed with the "WEKA" Waikato Environment for Knowledge Analysis. Weka is a library of machine learning algorithms useful for data mining. We can see from the data acquired with Weka that the various parameters associated with the ranking (see figure above on the left).

As a result, we may conclude that Glucose, BMI, Age, Pregnancies and Diabetes Pedigree Function are the key determining factors in determining if a patient would get diabetes.

Subgroup

If a person's glucose level, BMI, age, pregnancy history, and Diabetes Pedigree Function are all high, he or she is more likely to have diabetes.