MTH 786 Final-project

Harshit Saxena

22 January 2024

1 Introduction

Machine Learning (ML) is a disruptive force in computer science, reinventing how computers learn from data and make informed decisions autonomously. By utilizing statistical approaches, ML allows systems to grow and improve performance through exposure to varied datasets, reducing the need for explicit programming. Organizations across industries are increasingly turning to machine learning (ML) to gain insights, automate decision-making, and optimize operational operations.

This project focuses on the use of machine learning in healthcare, with a dataset that looks into patient characteristics associated to diabetes. The major goal is to create binary regression/classification models that can predict the likelihood of diabetes depending on the input factors. In the healthcare environment, machine learning (ML) plays an important role in proactive health management by identifying patients at risk early on. Overview of dataset, see Figure 1.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0

Figure 1: Overview of Dataset

2 Problem Statement

Scope of this project involves analyzing a dataset containing various features related to patients (768 records) and their likelihood of being affected by diabetes. The objective is to implement binary regression/classification models to assess the association between specific characteristics and the presence of diabetes. Aim is to leverage machine learning algorithms to accurately predict whether a patient is affected by diabetes based on the provided variables.

Details of input variables of the dataset:

1. Pregnancies: Number of times pregnant

2. Glucose: Plasma glucose concentration (glucose tolerance test)
3. Blood Pressure: Diastolic blood pressure (mm Hg)
4. Triceps: Skinfold thickness (mm)
5. Insulin: 2-Hr serum insulin (mu U/ml)
6. Mass: Body mass index (weight in Kg/ (height in m)^2)
7. Pedigree: Diabetes pedigree function
8. Age (years)

Target variable is "Outcome," which is binary (0 or 1), indicating the absence or presence of diabetes, respectively.
Furthermore, the project will explores statistical characteristics of the features and the target variable, visualizing their frequency distributions, analyzing their correlations and accuracy of the classification models implemented.

3 Analysis / Pre-Processing

Data cleaning, is imperative to identify and rectify errors, inconsistencies and missing values, safeguarding the reliability of subsequent analyses. Subsequently, data transformation ensures that raw data is appropriately formatted for analysis, involving actions such as scaling variables and encoding categorical data. Collectively, these steps contribute to the quality of exploratory data analysis by establishing a foundation of accuracy, proper formatting and alignment with chosen analytical methods for meaningful insights and informed decision-making.

In order to perform pre-processing of the dataset, null value check was implemented to identify if present. See Figure 2. Sample code commented that would allow encoding of categorical values in a column to numerical if required as per dataset. See Figure 3

```
print(df.isnull().sum())
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Figure 2: Null value check

```
#Check unique values of Pregnanices columnn to see if numercial or not
#data['Pregnancies'].unique()
#In case values given as Y, N in the column - reassign 0, 1
#Pregnancies_mapping = {'Y': 0, 'N': 1}

#Use a for loop to update values in the 'Pregnancies' column
#for index, row in df.iterrows():
#    df.at[index, 'Pregnancies'] = Pregnancies_mapping.get(row['Pregnancies'], -1)

#Convert column type to int from object
#df['Pregnancies'] = df['Pregnancies'].astype(int)
```

Figure 3: Sample encoding code

Statistical Characteristics: Initiated the analysis by examining the statistical characteristics of the dataset, including measures of central tendency and dispersion for each feature. This provided insights into the distribution and variability of the data. For instance, mean and median values provided a sense of the data's central tendency, while standard deviation and interquartile range informed about the spread of the values. These statistics served as a foundation for understanding the baseline characteristics of the dataset. See Figure 4

In [6]:
```
df.describe()
```

Out[6]:

	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

In [7]:
```
#Separate features and target variable
X = df.drop("Outcome", axis=1)
y = df["Outcome"]

#Split the dataset into training and testing sets
train_size = int(0.8 * len(df))
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
```

Figure 4: Statistical Characteristics

Frequency Distributions: I proceeded to visualize the frequency distributions of individual features and the target variable. Histograms were employed to illustrate the distribution of input variables like Glucose, BMI, Age, Pregnancies and others. These visualizations allowed identification of any skewed or imbalanced distributions, providing valuable insights into the distributional properties of the dataset. See Figure 5

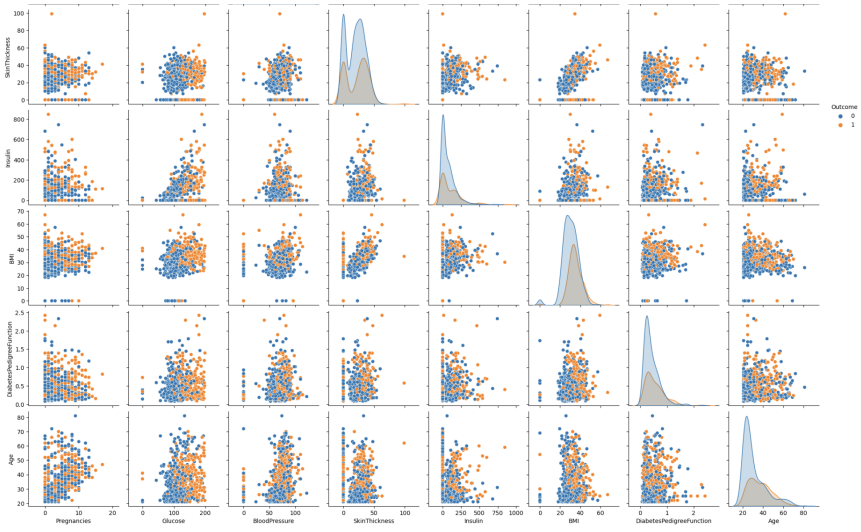

Figure 5: Frequency Distribution

Correlations: Understanding the relationships between features and the target variable is crucial. We computed correlation matrices and generated heatmaps to visualize the correlation coefficients. This step helped identify potential patterns and dependencies between variables. These insights laid the groundwork for feature selection and model building. See Figure 6

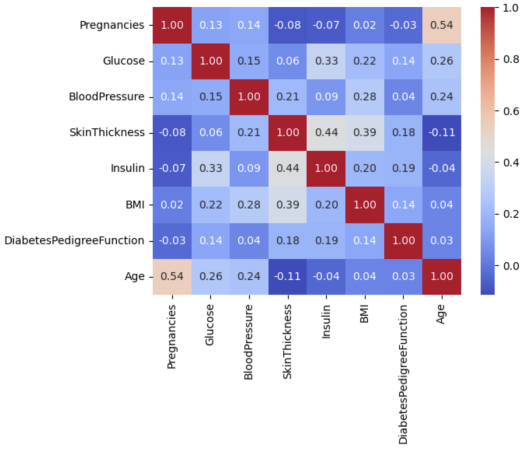

Figure 6: Frequency Distribution

4 Method

Logistic Regression: a widely-used algorithm for binary classification (dependent variable has only two possible outcomes), was applied due to its interpreability and simplicity. The model was trained on a subset of the dataset and evaluated using appropriate metrics.
This is a supervised statistical technique to find the probability of dependent variable(classes present in the variable). Gradient descent is employed in logistic regression to iteratively optimize model parameters and minimize the associated cost function, often the negative log-likelihood. The logistic functions (also known as the sigmoid functions) convert the probabilities into binary values which could be further used for predictions.

1. Sigmoid Function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

   Here, Z is the input to the sigmoid function.

2. Loss function (Gradient descent):

$$\theta_{j+1} = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

   Here, θ_{j+1} is the updated parameter, θ_j is the current parameter value, α is the learning rate, m is the number of training examples, $h_\theta(x^{(i)})$ is the sigmoid function applied to the linear combination of features $x^{(i)}$ with weights θ, and $y^{(i)}$ is the actual outcome for the i^{th} training example.

KNN Classification: The k-Nearest Neighbors (k-NN) classification algorithm involves predicting the class of a data point based on the classes of its nearest neighbors. Operating without the use of sklearn, the model assesses similarity using Euclidean distance, determining the k closest neighbors for each test sample.

Prediction is achieved through a majority voting mechanism among these neighbors. The dataset, encompassing parameters such as Pregnancies, Glucose, and Age, is split into training and testing sets for model evaluation. The analysis includes visualization of decision boundaries and emphasizes the adaptability of the hyperparameter 'k,' allowing for experimentation and optimization of the model's predictive performance.

1. Euclidean Distance:

$$EuclideanDistance = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Linear Regression: It seeks to establish a linear equation that best fits the observed data, minimizing the sum of squared differences between the predicted and actual values. In the case of simple linear regression, a single independent variable is considered, while multiple linear regression involves multiple predictors. The regression line's equation, often expressed as $y = mx + b$, signifies the relationship between the dependent variable y, independent variable x, slope m, and y-intercept b.

5 Results of the prediction task:

1. Logistic Regression: This model achieved an accuracy of 77.92%, indicating its effectiveness in predicting diabetes outcomes based on the given features. The result suggests a reasonable level of performance, demonstrating the model's ability to make accurate classifications nearly approximately 78% of the time. See figure 7

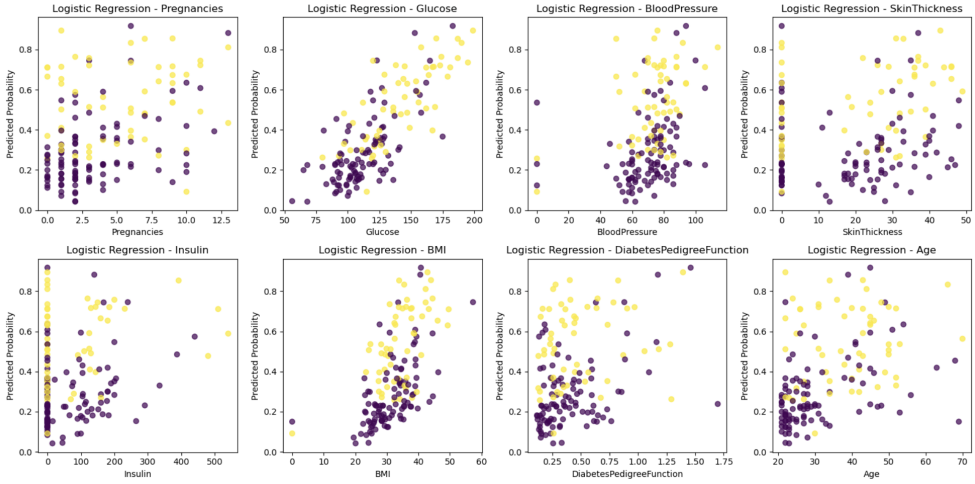

Figure 7: Logistic Regression Result

2. Linear Regression: This model yielded a mean square error of 0.161, reflecting the average squared difference between predicted and actual values. This low mean square error indicates a good fit of the model to the data, emphasizing its accuracy in capturing the underlying relationships and minimizing prediction errors. See figure 8

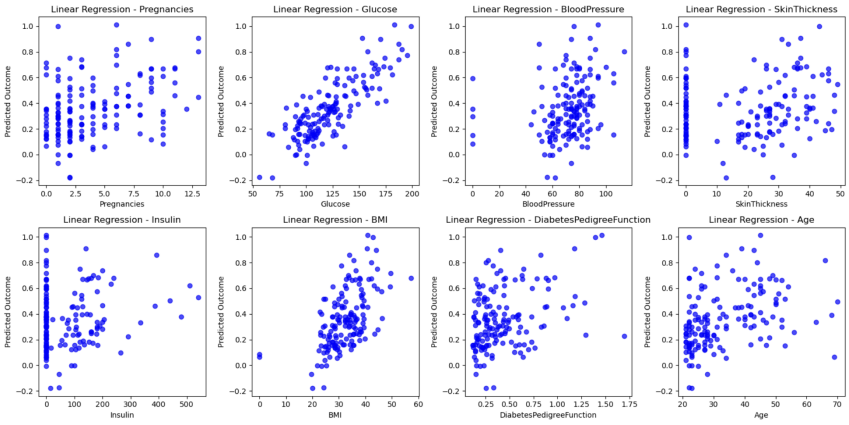

Figure 8: Linear Regression Result

3. KNN Classification: This model demonstrated an accuracy of 72% with value of K$\bar{7}$. The results affirm the effectiveness of leveraging the proximity of neighboring data points for making accurate predictions in this classification task. Accuracy of the model was increasing proportionally to the value of K. See figure 9

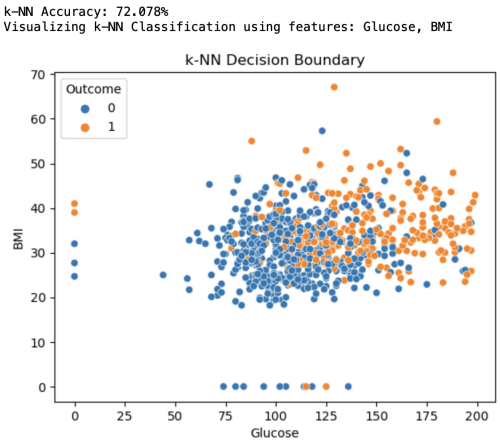

Figure 9: kNN Classfication - Glucose vs BMI

6 Conclusion

In addressing the challenge of predicting diabetes based on patient features, we explored multiple machine learning algorithms, namely logistic regression, linear regression, and k-nearest neighbors (KNN). The objective was to discern the associations between patient characteristics and diabetes outcomes, leveraging different methodologies for a comprehensive analysis.

Investigation into diabetes prediction unveiled insightful results across various models. Logistic regression achieved an accuracy of 77.2%, indicating its effectiveness in discerning diabetes outcomes. Simultaneously, linear regression demonstrated a robust fit with a low mean square error of 0.161, signifying its proficiency in capturing underlying relationships. The K-nearest neighbors (KNN) algorithm, with k=7, achieved an

accuracy of 72%. These diverse outcomes highlight the versatility of machine learning in healthcare predictions. While logistic regression offers valuable insights into feature-diabetes associations, linear regression excels in modeling dataset relationships. The inclusion of KNN enriches the analysis with proximity-based classifications. This project illuminates the a glimpse into the potential of machine learning in healthcare. Further model refinement and exploration of advance algorithms will enhance predictive capabilities in the healthcare sector including diabetes diagnosis and treatment.

References

[1] JK Kreiger. (2020). *Evaluating a Random Forest model.* Retrieved from `https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56`

[2] Abhigya. (2020). *Understanding Linear Regression.* Retrieved from `https://medium.com/analytics-vidhya/understanding-the-linear-regression-808c1f6941c0`

[3] Antony Christoper. (2021). *K-Nearest Neighbour.* Retrieved from `https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4`

[4] Abhigya. (2020). *Understanding Logistic Regression.* Retrieved from `https://medium.com/analytics-vidhya/understanding-logistic-regression-b3c672deac04`

[5] Saeed. (2021). *Data Visualization in Python with matplotlib, Seaborn, and Bokeh.* Retrieved from `https://machinelearningmastery.com/data-visualization-in-python-with-matplotlib-seaborn-and-bokeh/`