



Name	id
Nora Tarek al-suhaibi	444002730
Raya abdullah alsaab	444000732

Task1 Report

Title: Diabetes Model Documentation.

The Diabetes Dataset includes medical data for predicting diabetes in female patients over 21. It features indicators like number of pregnancies, BMI, insulin levels, and age, with a target variable showing diabetes status. This dataset is widely used to develop models for early diabetes detection.

Introduction :

- The Diabetes Dataset contains information about individuals diagnosed with diabetes, including demographic attributes, medical history, and clinical measurements.
- This dataset serves as a valuable resource for studying diabetes management, risk factors, and predictive modeling for disease outcomes.

Objectives:

•Predicting the Probability of Diabetes:

Logistic regression estimates the probability that an individual has diabetes given the input features. It outputs a value between 0 and 1, representing the likelihood of diabetes, which can be used to make a binary classification (diabetic or non-diabetic).

• Identifying Important Risk Factors:

The model helps in identifying which factors are most associated with the presence of diabetes. The coefficients of the logistic regression model indicate the strength and direction of the relationship between each feature and the likelihood of having diabetes.

• Improving Early Diagnosis and Prevention:

By accurately predicting the risk of diabetes, healthcare providers can identify high-risk individuals early on and recommend preventive measures or lifestyle changes to reduce the risk.

Dataset Features:

- Pregnancies: To express the Number of pregnancies.
- Glucose: To express the Glucose level in blood.
- BPressure: To express the Blood pressure measurement.
- SThickness: To express the thickness of the skin.
- Insulin: To express the Insulin level in blood.
- BMI: To express the Body mass index.
- DiabetesPedigreeFunction: To express the Diabetes percentage.
- Age: To express the age.

Dataset Target:

- Outcome: To express the final result 1 is YES o is NO.

Problem Statement:

- Define the problem : predicting the outcomes if a person has diabetes or not.
- Highlight the importance and potential applications of such predictions.

Importing Necessary Libraries:

pandas, numpy, and seaborn for data manipulation and visualization.

train_test_split and StandardScaler from sklearn for data splitting and scaling.

GaussianNB and LogisticRegression for building classification models.

Loading and Inspecting the Dataset:

Reads the dataset from diabetes.csv using `pd.read_csv`.

Displays the dataset and checks for any missing values using `isnull().sum()`.

Uses `dataset.info()` to get information about data types and non-null counts.

Data Visualization:

Uses seaborn to create a count plot for the 'Outcome' column to visualize the distribution of classes.

Data Splitting:

Features (x) are extracted from columns 1 to 7, and the target (y) is the 'Outcome' column.

The data is split into training and testing sets with a 75-25 split ratio (`test_size=0.25`) and a random seed for reproducibility (`random_state=42`).

Data Scaling:

Uses `StandardScaler` to standardize the features for better model performance.

Model Training and Evaluation:

A `GaussianNB` model is trained and evaluated on the training and testing sets.

A `LogisticRegression` model is then trained and evaluated similarly.

Performance metrics like training accuracy, testing accuracy, confusion matrix, classification report, and accuracy score are computed.

Evaluation Results:

Gaussian Naive Bayes Model:

Performance on testing data may be lower due to Naive Bayes' assumptions (e.g., feature independence) that may not hold true for this dataset.

Logistic Regression Model:

Confusion Matrix, Classification Report, and Accuracy Score provide detailed insights into the model's performance.

Identified Problems and Recommendations:

Data Imbalance:

The dataset may suffer from class imbalance, as indicated by the count plot of the 'Outcome'. Imbalanced datasets can negatively impact model performance, leading to biased predictions toward the majority class.

Solution: Consider techniques like oversampling the minority class, undersampling the majority class, or using Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset.

Feature Selection and Data Quality:

The choice of features might affect the model's accuracy. Currently, features from columns 1 to 7 are used without any feature selection or dimensionality reduction.

Solution: Perform feature importance analysis, principal component analysis (PCA), or other feature selection methods to optimize the features used.

Scaling Issues:

Although scaling has been applied, checking for outliers and normalizing data distributions could improve model performance further.

Solution: Perform outlier detection and consider normalizing features to a common range.

Model Hyperparameter Tuning:

The Logistic Regression model uses default hyperparameters, which might not be optimal.

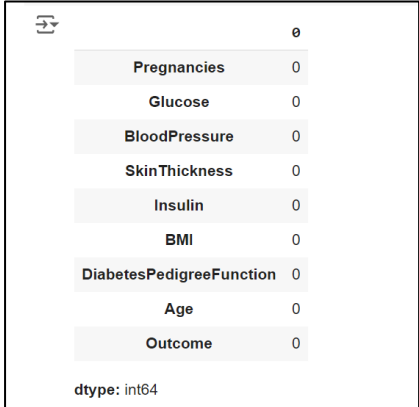
Solution: Use techniques like GridSearchCV or RandomizedSearchCV to tune hyperparameters for better performance.

Evaluation Metrics:

Relying solely on accuracy can be misleading, especially with imbalanced data.

Solution: Focus on additional metrics such as precision, recall, F1-score.

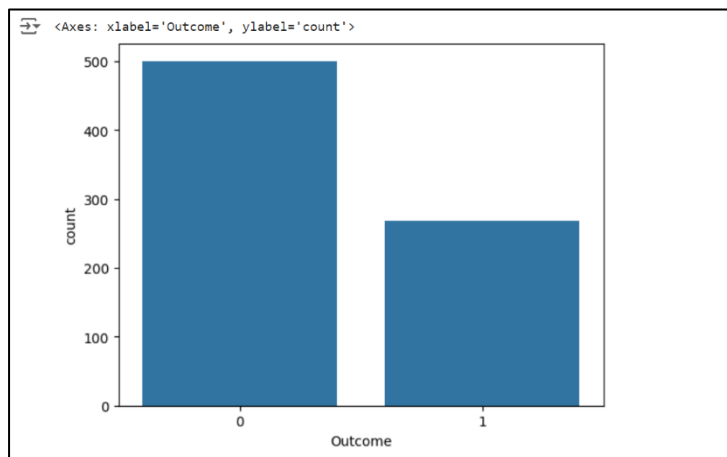
output about cood:



	0
Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

dtype: int64

It appeared to me that there are no null values in the data I was working on.



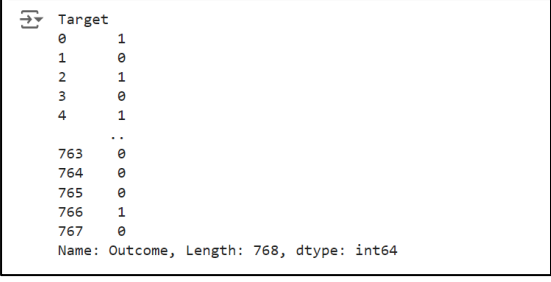
The graph in front of me shows the outcome results.

And we noticed that people who don't have diabetes are more likely to have diabetes than people who do.

Features						
	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	148	72	35	0	33.6	
1	85	66	29	0	26.6	
2	183	64	0	0	23.3	
3	89	66	23	94	28.1	
4	137	40	35	168	43.1	
..	
763	101	76	48	180	32.9	
764	122	70	27	0	36.8	
765	121	72	23	112	26.2	
766	126	68	0	0	30.1	
767	93	70	31	0	30.4	
DiabetesPedigreeFunction Age						
0	0.627	50				
1	0.351	31				
2	0.672	32				
3	0.167	21				
4	2.288	33				
..				
763	0.171	63				
764	0.340	27				
765	0.245	30				
766	0.349	47				
767	0.315	23				

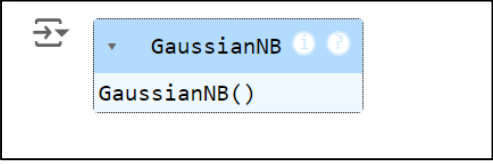
[768 rows x 7 columns]

In this output just show us all rows and 7 columns of data which are "**Features**".



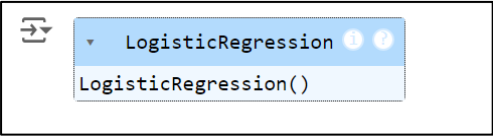
```
Target
0      1
1      0
2      1
3      0
4      1
..
763    0
764    0
765    0
766    1
767    0
Name: Outcome, Length: 768, dtype: int64
```

In this output all rows are shown to us but only one column which is "**Target**".



GaussianNB ⓘ ?
GaussianNB()

Show us the category used "**GaussianNB**" as well as the result of the "**object**" that will represent the training model.



LogisticRegression ⓘ ?
LogisticRegression()

This output shows

LogisticRegression" it displays information about the "model.

"LogisticRegression ()" is used to create an object from the model that can be used to train data and make predictions.

[[100 23] [28 41]]					
	precision	recall	f1-score	support	
0	0.78	0.81	0.80	123	
1	0.64	0.59	0.62	69	
accuracy			0.73	192	
macro avg	0.71	0.70	0.71	192	
weighted avg	0.73	0.73	0.73	192	
0.734375					

It shows me which numbers I predicted, which ones are correct and which ones are wrong, and also **"Precision"** as well as **"Recall"** appeared to me and we calculated it only for the numbers we predicted correctly, And also **"F1-score"** shows me inside it the calculation of the average between **"Precision"** and **"Recall"**.

It also shows us **"Accuracy"**, **"macro Avg"**, and **"weighted avg"**.

The overall result **"0.734375"** shows that the model has reasonable performance with an accuracy of **"73%"**.

References:

Link about data:

https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce?select=product_category_name_translation.csv