Peter the Great St.Petersburg Polytechnic University
Institute of Computer Science and Cybersecurity
Higher School of Artificial Intelligence Technologies

COURSE WORK

« Diabetic prediction mobile app»

On discipline «Seminar on the specialty»

Completed by students
gr. 5130203/20101: Abutalipov Alexandr
Gavshin Artem
Mankiev Danil
Pisarev Vladislav
Nguen Ba Phu

Supervisor: Espinola Rivera Holger Elias

Saint-Petersburg
2024

CHAPTER 1. PROBLEM STATEMENT

Diabetes is a chronic disease that affects millions of individuals globally, leading to severe health complications and increased healthcare costs. Early detection and management are crucial in preventing the progression of the disease and improving patients' quality of life. Despite the availability of diagnostic methods, there remains a need for accessible and efficient tools that can predict the risk of diabetes in individuals, particularly in regions with limited healthcare resources.

The primary goal of this project is to develop a mobile application that leverages machine learning algorithms to predict the risk of diabetes in users.

Project tasks:

1. Analyze the Pima Indians Diabetes dataset and prepare data for training and testing KNN and SVM models.
2. Make benchmark of two models and compare the performance using the metrics
3. Develop the user interface of a mobile application using the Kotlin programming language and the Jetpack framework Compose UI framework
4. Develop a backend in the Go programming language and connect the PostgreSQL database
5. Implement communication between the frontend and backend using the REST API and integrate the model into the application

CHAPTER 2. METHODOLOGY OF SOLUTION

The project used data from the Pima Indians Diabetes dataset. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Data preprocessing

1. Load the dataset and perform the analysis of the content
2. Check data for cleaning (missing values, duplicated rows)
3. Plot the histogram of frequencies for numerical features

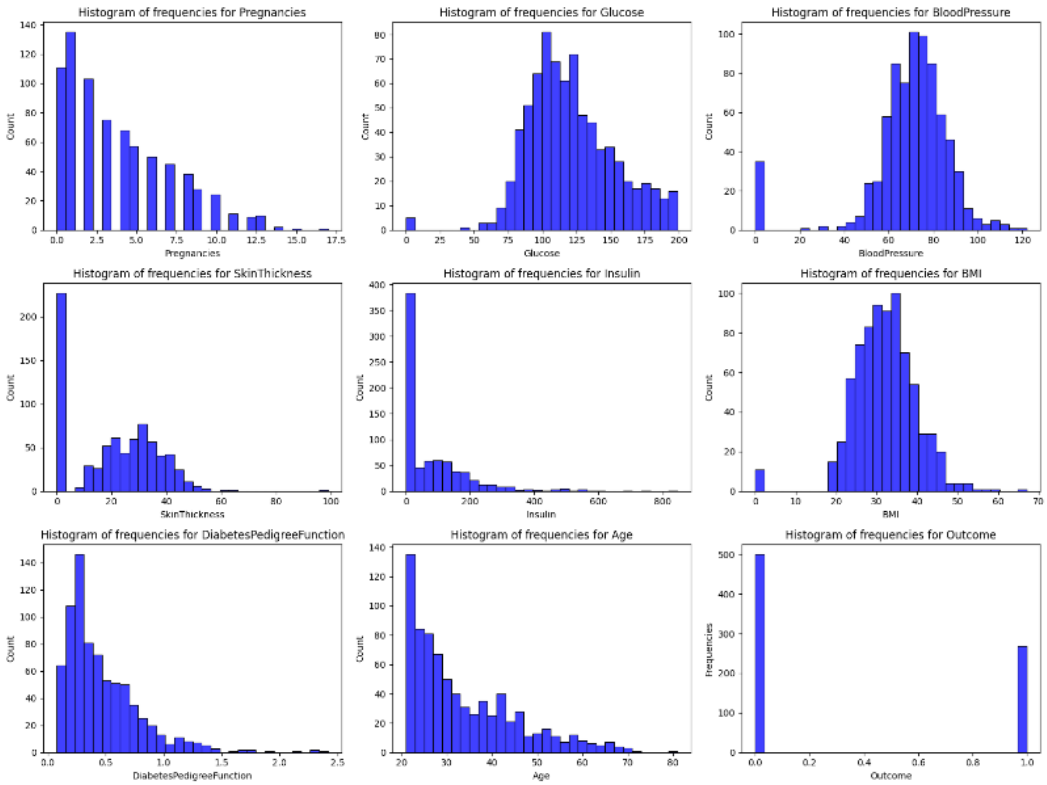

Figure 1. histogram of frequencies for numerical features

4. Perform correlation analysis

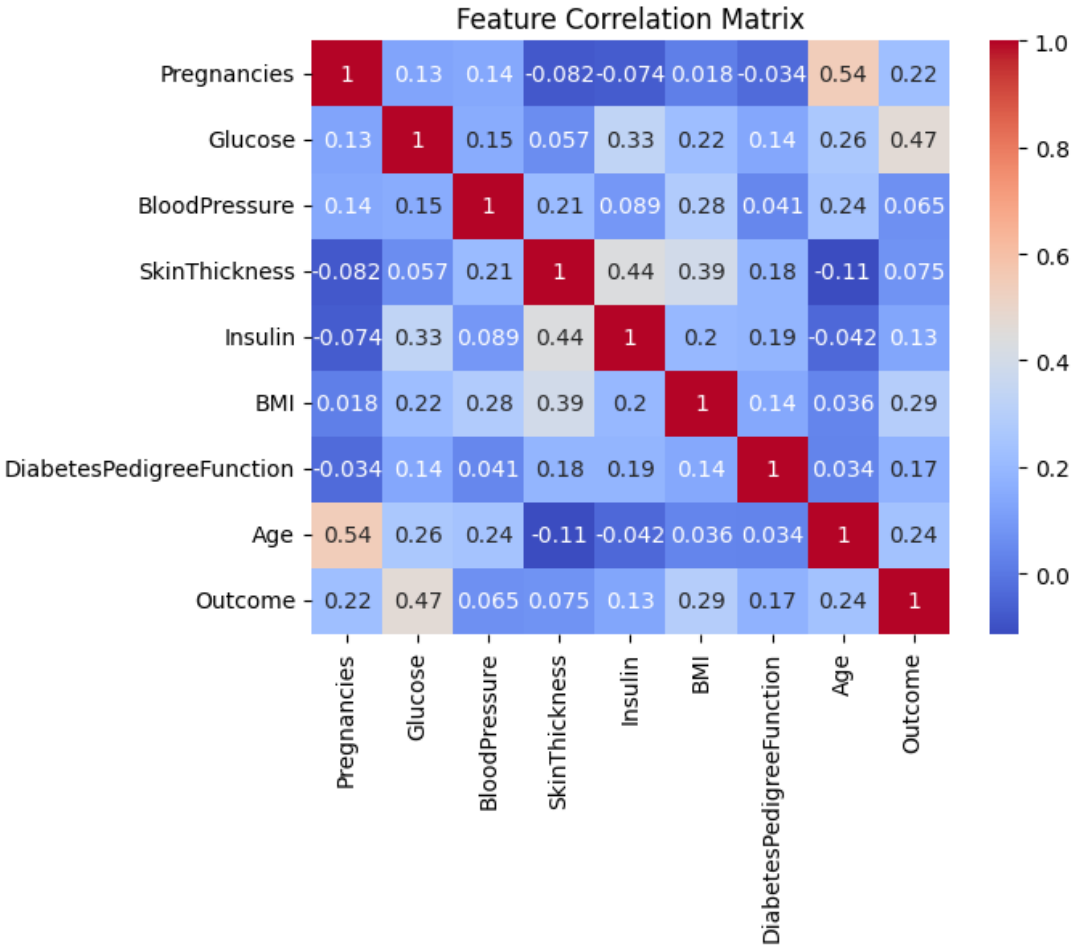

Figure 2. Correlation matrix

5. Select the target variable. Prepare training and test data for the models

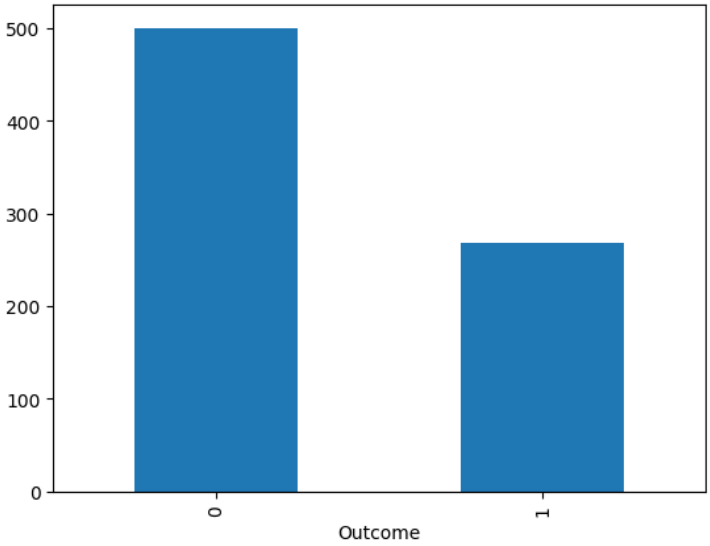

Figure 3. Distribution of data at the target variable

6. With the help of StandardScaler and RobustScaler transform the data

Training and testing the models

KNN and SVM classifiers were selected as models. In order to find out the best parameters for models, the GridSearchCV method was used.

KNN basically makes predictions based on the similarity of data points in the sample space. The performance of KNN is basically based on the choice of K. KNN works by memorizing the entire training dataset. When a new data point is given for prediction, KNN looks at the k-nearest data points in the training set based on a specified distance metric (commonly Euclidean distance). For classification, it assigns the majority class among the k-nearest neighbors to the new data point.

Best KNN Parameters: {'algorithm': 'auto', 'metric': 'euclidean', 'n_neighbors': 5, 'weights': 'uniform'}

The main objective of SVM is to find an optimal hyperplane that best separates the data into different classes in a high-dimensional space. The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points (support vectors) of each class. Kernel functions are used for transforming lower dimensional input space to higher dimensional output.

Best SVM Parameters: {'C': 1, 'class_weight': 'balanced', 'degree': 3, 'gamma': 'auto', 'kernel': 'poly'}

CHAPTER 3. RESULTS

The following metrics were used to compare the results of the models:

- Accuracy: The proportion of true results (both true positives and true negatives) among the total number of cases examined. It measures how often the model is correct overall.
- Precision: The proportion of true positives among the predicted positives. It indicates how many of the predicted positive cases are actually positive.
- Recall: The proportion of true positives among the actual positives. It shows how many of the actual positive cases were correctly identified by the model.
- F1 Score: The harmonic mean of precision and recall. It provides a single metric that balances both precision and recall.

KNN Classifier:

KNN metrics:

	precision	recall	f1-score	support
0	0.79	0.87	0.83	150
1	0.71	0.57	0.63	81
accuracy			0.77	231
macro avg	0.75	0.72	0.73	231
weighted avg	0.76	0.77	0.76	231

Figure 4. Metrics of the KNN model

Time of training KNN: 0.153 seconds

SVM Classifier:

SVM metrics:

	precision	recall	f1-score	support
0	0.77	0.89	0.83	150
1	0.71	0.52	0.60	81
accuracy			0.76	231
macro avg	0.74	0.70	0.71	231
weighted avg	0.75	0.76	0.75	231

Figure 5. Metrics of the SVM model

Time of training SVM: 0.311 seconds

The KNN and SVM models showed similar very similar results, but KNN is slightly better. (Accuracy: 77% over 76%)

It took two times less time to train KNN than SVM. (0.15 over 0.31 seconds)

After analyzing the metrics of the models, we can conclude that she has no problems determining the absence of diabetes in a patient. However, the model has some difficulties in determining whether a patient has diabetes (recall metric). For more accurate predictions, a more extensive dataset is required for training models.

CHAPTER 4. CONCLUSION

In the initial part of the coursework, we focused on developing and testing machine learning algorithms designed for the detection of diabetes using medical data. This phase of the project encompassed a comprehensive data processing cycle, which included stages of preparation, analysis, and model training and testing.

As a result of the comparison, the KNN model was selected for further use in the project. It has greater accuracy and predicts the presence of diabetes in humans more often.

The next part of the course work will focus on system development. A mobile application will be written using the Jetpack Compose UI framework. To integrate the KNN model and connect the PostgreSQL database, a backend will be written in the Go programming language. Communication will take place through the architectural style of the REST API.