Diabetes_prediction_classifier_comparison_and evaluation

PRAVIN KUMAR

Chemical Engineering, Birla Institue of Technology Mesra ,Ranchi

1. Abstract

In this project we mainly focus on the model classification of various dataset. We focus in KNN classification and SVM and compare the results between the models. KNN (**K-Nearest Neighbors**) classifies a data point based on the majority class among its closest neighbours. It is simple, intuitive, and works well with smaller datasets but can be slow for large datasets. SVM (**Support Vector Machine**) finds the best hyperplane that separates data into classes with maximum margin. The project typically involves preprocessing (handling missing values, scaling, encoding categorical data). The dataset is split into training and testing sets for evaluation. Both algorithms are trained and tested, and their performance is compared using accuracy, precision, recall, or F1-score. We perform this process for different datasets like Prima Indian Diabetes Dataset and Titanic Dataset.

2. Introduction:

The project is founded on applying and comparing K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) that are two of the popular machine learning algorithms. Those algorithms are quite applicable in today's advanced use cases including medical diagnostics, fraud identification, sentiment analysis for text classification, and face recognition. The project incorporates technologies including computer programming in Python, scikit-learn library, and data visualization software including Matplotlib and Seaborn.

KNN is a simple yet practical method for classification problems, especially for

non-linear boundary data. SVM can, nevertheless, handle large-dimensional data sets strongly and provide good generalization by making use of kernel transformations. The methodology employed is as follows:

Data preprocessing (managing missing values, encoding, normalization).

Split the data into test and train sets.

Training KNN and SVM models.

Measuring performance according to parameters such as accuracy, precision, recall, and F1-score.

Comparing results to choose the better algorithm for the dataset under consideration.

Comparing AUCs from ROC curves and finding the best model

3. Project Objective

❷ To implements and compare **K-Nearest Neighbors (KNN)** and **Support Vector Machine (SVM)** algorithms on a given dataset.

❷ To illustrate how these algorithms perform in terms of **accuracy,precision,recall, and F1-score**, thereby identifying the more suitable model.

❷ To demonstrate the impact of **data preprocessing techniques** (handling missing values, normalization, encoding) on model performance.

❷ To analyse the strengths and limitations of KNN and SVM in handling **different data characteristics** such as non-linear boundaries and high-dimensionality.

4. Methodology

Tools and Libraries Used : For loading, cleaning, and manipulating the dataset.

Pandas Scikit-learn(sklearn) : For splitting data, feature scaling, and implementing machine learning models.

 Matplotlib & Seaborn : For data visualization and exploratory data analysis.

Step 1: Data Loading and Initial Exploration

The dataset diabetes.csv was imported into a **Pandas DataFrame**. Initial exploration involved:

o Checking dimensions (rows and columns). o Inspecting column names, data types, and sample values using df.info(). o Summarizing basic statistics (mean, median, min, max, standard deviation) with df.describe().

 This step provided insight into the distribution of features and highlighted potential data quality issues.

Step 2: Data Pre-processing and Cleaning

- A key observation: Several medical features (**Glucose, BloodPressure, SkinThickness, Insulin, BMI**) contained **zero values**, which are biologically implausible.
- To handle this issue:
  - Zero values were treated as missing data.
  - Imputation was applied by replacing them with the **mean** of each respective feature.
- This method preserved all data points and avoided dropping records unnecessarily, ensuring maximum dataset utilization.

Step 3: Data Splitting and Feature Scaling

- The dataset was split into:
  - **Training set (80%)**
  - **Testing set (20%)**
  - Split performed using train_test_split() from Scikit-learn, ensuring unbiased model evaluation.
- **Feature Scaling**: StandardScaler was applied to normalize numerical attributes
  - **For SVM**: Scaling is essential since SVM is sensitive to differences in feature magnitude, which affects the positioning of the decision boundary.
  - **For KNN**: Scaling ensures all features contribute equally to distance calculations, preventing larger-range features from dominating.

Step 4: Model Development and Training

Two machine learning algorithms were developed:

1. **Support Vector Machine (SVM)**
   - Implemented using SVC from Scikit-learn.
   - Works by identifying the optimal hyperplane that separates different classes in the feature space.
2. **K-Nearest Neighbors (KNN)**
   - Implemented using KNeighborsClassifier.
   - Classifies k nearest test data points based on the majority vote from their neighbors.

- Both models were trained on the pre-processed **trainingdata(X_train,y_train)**

Step 5: Model Validation and Evaluation

The trainedmodels werevalidatedon testing data (X_test, y_test) .Evaluation was performed using thefollowing metrics:

Accuracy: Proportion of correct predictions.

Precision: Ability to avoid false positives.

Recall: Ability to capture all true positives.

F1-Score: Balance between precision and recall.

 A **Confusion Matrix** was generated for both models to visualize performance (True Positives, True Negatives, False Positives, False Negatives).

5. Data Analysis and Results

For Diabetes Data:

```
KNN Results:
Accuracy: 0.7012987012987013
[[80 20]
 [26 28]]
              precision    recall  f1-score   support

           0       0.75      0.80      0.78       100
           1       0.58      0.52      0.55        54

    accuracy                           0.70       154
   macro avg       0.67      0.66      0.66       154
weighted avg       0.69      0.70      0.70       154
```

80 non-diabetic patient are correctly predicted as non_diabetic(tn). (true negative) 20 non-diabetic patient are incorrectly predicted as diabetic(fp). (false positive) 26 diabetic patients are incorrectly predicted as non-diabetic(fn). (false negative) 28 diabetic patients are correctly predicted as diabetic (tp). (true positive) 70.13% percent accuracy of the model

```
SVM Results:
Accuracy: 0.7207792207792207
[[83 17]
 [26 28]]
              precision    recall  f1-score   support

           0       0.76      0.83      0.79       100
           1       0.62      0.52      0.57        54

    accuracy                           0.72       154
   macro avg       0.69      0.67      0.68       154
weighted avg       0.71      0.72      0.71       154
```

83 non-diabetic patient are correctly predicted as non_diabetic(tn)

17 non-diabetic patient are incorrectly predicted as diabetic(fp) 26 diabetic patients are incorrectly predicted as non-diabetic(fn) 28 diabetic patients are correctly predicted as diabetic(tp) 72.08% percent accuracy of the model

For diabetes patient (class1) in knn model precision for class 1 =0.58 recall=0.52,f1 score: 0.55, and for non_diabete patient (class 0) precision=0.75,recall=0.80,f1-score=0.78.Whereas in knn model for class 1 precision=0.62,recall=0.52,f1-score=0.57 for class0 (non_diabetes) precision=0.76,recall=0.83,f1-score=0.79.SVM model has more accuracy than KNN model

For Titanic Data:

```
KNN Results (Titanic):
Accuracy: 0.6536312849162011
[[88 22]
 [40 29]]
              precision    recall  f1-score   support

           0       0.69      0.80      0.74       110
           1       0.57      0.42      0.48        69

    accuracy                           0.65       179
   macro avg       0.63      0.61      0.61       179
weighted avg       0.64      0.65      0.64       179
```

88 people who were predicted dead and actually dead(tn).(true negative) 22 people who were predicted to be alive but actually dead(fp).(false positive) 40 people who were predicted dead but actually alive(fn).(false negative) 29 people who were predicted alive and actually alive (tp).(true positive)

65.36% percent accuracy of the model

```
SVM Results:
Accuracy: 0.6815642458100558
[[92 18]
 [39 30]]
              precision    recall  f1-score   support

           0       0.70      0.84      0.76       110
           1       0.62      0.43      0.51        69

    accuracy                           0.68       179
   macro avg       0.66      0.64      0.64       179
weighted avg       0.67      0.68      0.67       179
```

92 people who were predicted dead and actually dead(tn) 18 people who were predicted to be alive but actually dead(fp) 39 people who were predicted dead but actually alive(fn) 30 people who were predicted alive and actually alive(tp) 68.15% percent accuracy of the model For survived passengers (class1) in knn model precision for class 1 =0.57 recall=0.42,f1
score: 0.48, and for dead passengers (class 0) precision=0.69,recall=0.80,f1-score=0.74.
Whereas in svm model for class 1 precision=0.62,recall=0.43,f1-score=0.51 for class0

(dead
passengers) precision=0.70,recall=0.84,f1-score=0.76
SVM model has more accuracy than KNN model

6.Conclusion: For Diabetes Data:

SVM model performs slightly better than KNN model .SVM model has higher accuray (72%) than KNN model (70%). Recalls and presicions of svm model w.r.t class1 and class0 is slightly higher that KNN model. Also SVM has higher macro and weighted average indicates that SVM is the better choice

Recall for class 1 in both the models are low which shows that it struggles to distinguish between class1 (diabetes) and class0(non-diabetic) patients

For Titanic Data:

SVM model performs slightly better than KNN model .SVM model has higher accuray (68.15%) than KNN model (65.36%).Recalls and presicions of svm model w.r.t class1 and class0 is slightly higher that KNN model.Also SVM has higher macro and weighted average indicates that SVM is the better choice

In Both the Datasets SVM models have more accuracy.