A

PROJECT REPORT

ON

MACHINE LEARNING

FOR

BACHELOR OF TECHNOLOGY

(2022-2025)

IN

ARITIFICAL INTELLIGENCE AND DATA SCIENCE

AT

UNIIVERSITY SCHOOL OF AUTOMATION & ROBOTICS

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY

EAST DELHI, SURAJMAL VIHAR,DELHI-110092

SUBMITTED TO:

Dr. Sanjay

SUBMITTED BY

NAME : Prince Kumar

Enroll No : 00119011922

Batch : AI&DS_B2

LINKS-

DATASET : https://www.kaggle.com/datasets/mathchi/diabetes-data-set

GITHUB :
https://github.com/princeUsar/ML_PROJECT/blob/main/prince%20project%20-%20Jupyter%20Notebook.pdf

Title: Comparison of Classification and Clustering Algorithms for Diabetes Prediction

Abstract:

This report presents a study on the application of classification and clustering algorithms for diabetes prediction. The dataset used contains various features related to diabetes, such as pregnancies, glucose levels, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, and age. The goal is to evaluate the performance of different algorithms in predicting the outcome of diabetes and explore clustering techniques to discover patterns within the dataset.

Keywords: Classification, Clustering, Diabetes Prediction, Algorithms, Performance Evaluation

1. Introduction:

Diabetes is a prevalent chronic disease with significant health implications. Predicting the occurrence of diabetes can assist in early diagnosis and intervention, improving patient outcomes. In this study, we explore the application of classification and clustering algorithms to predict diabetes and discover inherent patterns within the dataset.

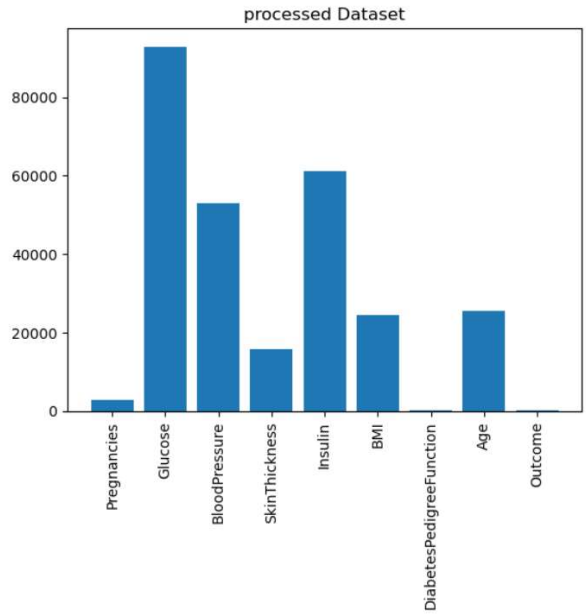
2. Proposed Methodology:

a. Dataset:

The diabetes dataset used in this study contains information about individuals, including various health-related attributes and the target variable indicating the presence or absence of diabetes. The dataset is loaded and preprocessed to prepare it for analysis.

b. Preprocessing:

Prior to model training, the dataset undergoes preprocessing steps such as handling missing values, encoding categorical variables, and scaling numerical features. This ensures that the data is suitable for the algorithms.



c. Classification Algorithms:

Several classification algorithms are employed, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gaussian Naive Bayes (GaussianNB), Perceptron, and Decision Trees. Each algorithm is trained on the preprocessed dataset and evaluated using accuracy scores.

```python
# importing all required libraries and modules
from sklearn.linear_model import Perceptron
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB


# Giving Parameters to the fumctions
clf1=KNeighborsClassifier(n_neighbors=5,metric="minkowski")
clf2=SVC(C=1.0,kernel="rbf")
clf3=GaussianNB(priors=None)
clf4=Perceptron(alpha=0,l1_ratio=0.15,max_iter=100)
clf5=DecisionTreeClassifier(criterion="gini",splitter="best", max_depth=5)


clf=[clf1,clf2,clf3,clf4,clf5]
clf_name=["kneighbors","svc","gaussionNB","perceptron","decisiontree"]

from sklearn.metrics import accuracy_score
accuracy={}
import time
acc={}
t={}
for model,model_name in zip(clf,clf_name):
    st=time.time()
    model.fit(X_train,y_train)
    pred=model.predict(X_test)
    et=time.time()
    acc[model_name]=accuracy_score(y_test,pred)
    t[model_name]=et-st

for i,j in acc.items():
    print(i,":-",j)
```

```
kneighbors :- 0.6623376623376623
svc :- 0.7662337662337663
gaussionNB :- 0.7662337662337663
perceptron :- 0.5714285714285714
decisiontree :- 0.7922077922077922
```

d. Clustering Algorithms:

Clustering techniques, namely K-Means, Agglomerative Clustering, DBSCAN, Gaussian Mixture Models (GMM), and BIRCH, are applied to identify clusters or groups within the dataset. The algorithms assign labels to each data point based on their similarities.

```python
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN
from sklearn.mixture import GaussianMixture
from sklearn.cluster import Birch

clustering_algorithms = {
    'K-Means': KMeans(n_clusters=5),
    'Agglomerative': AgglomerativeClustering(n_clusters=5),
    'DBSCAN': DBSCAN(eps=0.5, min_samples=5),
    'GMM': GaussianMixture(n_components=5),
    'BIRCH': Birch(n_clusters=5)
}

data = df

for algorithm_name, algorithm in clustering_algorithms.items():
    labels = algorithm.fit_predict(data)
    print(f"Algorithm: {algorithm_name}")
    print("Cluster Labels:")
    print(labels)
    print("------")
```

```
Algorithm: K-Means
Cluster Labels:
[3 3 3 0 1 3 0 3 2 3 3 3 3 2 1 3 1 3 0 0 1 3 3 3 1 0 3 1 0 3 3 4 0 3 3 1 3
 3 3 1 0 3 3 4 3 3 3 3 3 3 0 3 3 4 4 3 4 0 3 1 3 3 3 1 3 3 3 3 0 0 0 1 3 4
 3 3 3 3 3 3 3 3 0 3 3 0 3 0 0 3 3 1 0 3 0 1 3 0 0 1 3 3 3 0 3 1 3 1 3 0 1
 2 0 3 1 3 3 3 3 0 0 3 0 3 3 0 1 0 1 3 1 3 1 3 0 1 0 0 3 4 3 3 0 3 4 3 3 0
 3 3 1 3 1 2 3 3 0 1 0 1 3 0 4 3 3 1 3 3 3 0 3 1 3 0 0 1 3 1 3 3 3 0 0 3 3
 3 2 0 0 1 3 0 3 3 3 1 3 0 0 4 3 3 3 0 1 3 4 3 0 3 3 3 3 1 1 4 1 0 3 3 2 3
 3 1 0 3 3 3 2 0 3 4 0 3 0 3 1 3 3 3 3 0 3 1 1 3 3 2 4 3 3 3 0 3 4 3 3 3 4
 1 1 3 3 3 3 0 3 3 3 3 3 0 3 0 3 0 3 0 3 4 3 1 1 3 3 1 2 1 0 0 0 0 1 1 3 1
 4 1 1 3 3 1 0 3 3 0 1 1 1 1 3 1 0 0 3 0 0 3 1 3 1 3 3 3 3 1 1 3 0 0 3 0 3
 3 0 4 3 3 1 3 0 0 3 3 3 0 0 3 0 3 3 3 3 0 3 3 1 3 0 4 4 3 3 3 4 0 3 3 0 1
 2 0 0 0 1 4 0 0 3 0 0 3 1 0 0 0 3 3 4 0 1 3 4 0 3 4 0 3 3 3 3 3 0 3 3 1 3
 3 3 2 3 1 4 0 1 2 3 3 3 0 1 0 0 3 1 4 3 1 1 1 3 0 0 3 3 3 3 3 3 3 3 0 0 3
 3 3 0 0 0 0 0 3 1 3 0 3 3 0 4 0 0 3 0 3 3 0 0 0 3 1 3 3 3 3 3 3 1 0 0 3 4
 3 0 0 3 4 2 4 3 3 0 3 3 1 3 3 3 0 1 1 0 3 3 0 3 3 0 1 0 3 3 1 3 3 0 0 1 3
 3 4 0 1 3 3 3 3 0 0 1 3 0 3 0 3 0 3 3 3 1 1 1 1 3 0 0 1 1 1 0 3 3 0 3 0 0
 1 3 3 3 3 3 4 0 0 3 0 3 0 1 1 3 3 0 0 4 0 1 3 3 3 3 3 3 3 2 3 3 3 1 3 3 1
 3 0 1 1 3 3 3 0 3 3 3 1 3 3 4 0 4 1 1 1 4 3 1 3 3 3 3 3 1 3 3 0 3 0 3 3 3
```

```
Algorithm: Agglomerative
Cluster Labels:
[0 0 0 1 3 0 1 0 2 0 0 0 0 2 3 0 3 0 1 1 3 0 0 0 3 1 0 1 1 0 0 3 1 0 0 3 0
 0 0 3 1 0 0 3 0 0 0 0 0 0 1 1 0 4 4 0 4 1 0 1 0 0 0 3 0 0 0 0 1 1 1 3 0 4
 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 3 1 0 1 3 0 1 1 3 0 0 0 1 0 3 0 3 0 1 3
 2 1 0 3 0 0 0 0 1 1 0 1 0 0 1 1 1 3 0 3 0 3 0 1 3 1 1 0 4 0 0 1 0 4 0 0 1
 0 0 3 0 3 2 0 0 1 1 1 3 0 1 4 0 0 3 0 0 0 1 0 3 0 1 1 3 0 3 0 0 0 1 0 0 0
 0 2 1 1 3 0 1 0 0 0 3 0 1 1 4 0 0 0 1 3 0 4 0 1 0 0 0 0 3 3 4 1 1 0 0 2 0
 0 3 1 1 0 0 2 1 0 4 1 0 1 0 3 0 0 0 0 1 0 3 3 0 0 2 4 0 0 0 1 0 4 0 0 0 4
 3 3 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 4 0 1 3 0 0 3 2 3 1 1 1 1 3 3 0 3
 4 3 3 0 0 3 1 0 0 1 3 3 3 3 0 1 1 1 0 1 1 0 1 0 3 0 0 0 0 3 3 0 1 1 0 1 0
 0 1 3 0 0 3 0 1 1 0 0 0 1 1 0 1 0 0 0 0 1 0 0 3 0 1 4 4 0 0 0 4 1 0 0 1 3
 2 1 1 1 3 4 1 1 0 1 1 0 3 1 1 1 0 0 4 1 3 0 4 1 0 4 1 0 0 0 0 0 1 0 0 3 0
 0 0 2 0 3 4 1 3 2 0 0 0 1 3 1 1 0 3 4 0 3 3 3 0 1 1 0 0 0 0 0 0 0 0 1 1 0
 0 0 1 1 1 1 1 0 3 0 1 0 0 1 3 1 1 0 1 0 0 1 0 1 0 3 0 0 0 0 0 0 3 1 1 0 4
 0 1 1 0 3 2 4 0 0 1 0 0 1 0 0 0 1 3 3 1 0 0 1 0 0 1 3 1 0 0 3 0 0 1 1 3 0
 0 4 1 1 0 0 0 0 1 1 3 0 1 0 1 0 1 0 0 0 3 3 3 3 0 1 1 3 3 3 1 0 0 1 0 1 1
 3 0 0 0 0 0 4 1 1 0 1 0 1 3 3 0 0 1 1 4 1 1 0 0 0 0 0 0 0 2 0 0 0 3 0 0 1
 0 1 3 3 0 0 0 1 0 0 0 3 0 0 4 1 4 3 1 3 4 0 3 0 0 0 0 0 3 0 0 1 0 1 0 0 0
```

```
Algorithm: DBSCAN
Cluster Labels:
[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

```
Algorithm: GMM
Cluster Labels:
[1 1 1 0 4 1 4 1 2 1 1 1 1 3 4 1 4 1 0 4 0 1 1 1 4 4 1 0 0 1 1 4 0 1 1 0 1
 1 1 4 0 1 1 4 1 4 1 1 1 1 0 0 0 4 0 1 4 0 1 0 1 1 1 0 1 1 1 1 0 0 4 0 1 0
 1 1 1 1 1 1 1 1 0 1 1 0 1 0 4 1 1 0 0 1 0 0 1 0 0 4 1 1 1 0 1 0 1 0 0 4 4
 2 0 1 4 1 1 1 1 0 4 1 0 1 1 4 0 0 4 1 4 1 4 1 0 0 0 0 1 0 1 1 0 1 0 1 1 0
 1 1 0 1 4 2 1 1 0 0 0 4 1 0 0 1 1 4 1 1 1 0 1 4 1 0 0 4 1 4 1 1 1 0 0 1 1
 1 2 4 4 4 1 0 1 1 1 4 1 4 4 4 1 1 1 0 0 1 4 1 0 1 1 1 1 4 4 4 4 0 1 1 2 1
 1 0 0 0 1 1 3 0 1 4 0 1 0 1 4 1 1 1 1 0 1 4 0 1 1 3 2 1 1 1 0 1 4 1 1 1 0
 4 0 1 1 1 1 0 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 0 2 4 0 0 0 4 4 4 1 0
 4 0 4 1 1 4 0 1 1 0 4 0 4 4 1 0 4 0 1 0 0 1 0 1 0 1 1 4 1 0 4 1 4 0 1 0 1
 1 0 0 1 1 4 1 0 0 1 1 1 0 0 1 0 1 1 1 1 0 1 1 4 1 0 4 4 1 1 1 0 0 1 1 0 4
 2 0 0 0 0 4 0 0 1 0 0 1 0 0 0 0 1 1 4 0 0 1 0 0 1 0 0 1 1 1 1 1 4 1 1 0 1
 1 1 2 1 0 0 0 4 2 1 1 1 4 0 0 0 1 4 4 1 4 0 4 1 0 0 1 1 1 1 1 1 1 1 0 0 1
 1 4 0 0 4 0 0 1 0 1 0 1 1 0 4 0 0 1 0 1 1 0 0 0 1 0 1 1 1 1 1 1 4 0 0 1 4
 1 0 0 1 4 2 0 1 1 0 1 1 4 1 1 1 0 4 0 0 1 1 0 1 1 4 0 0 1 1 0 1 1 0 4 4 1
 1 0 0 0 1 1 1 1 0 0 0 1 0 1 0 1 0 1 1 1 0 4 4 4 1 0 0 4 4 0 0 1 1 0 1 0 0
 0 1 1 1 1 1 4 0 0 1 0 0 0 0 4 1 1 0 0 0 0 0 1 1 1 1 1 1 1 2 1 1 1 4 1 1 0
 1 0 0 4 1 0 1 0 1 1 1 4 1 1 4 0 0 0 0 4 4 1 4 1 1 0 1 1 0 1 1 0 1 0 1 1 1
```

```
Algorithm: BIRCH
Cluster Labels:
[0 0 0 1 3 0 1 0 2 0 0 0 0 2 3 0 3 0 1 1 3 0 0 0 3 1 0 1 1 0 0 3 1 0 0 3 0
 0 0 3 1 0 0 3 0 0 0 0 0 0 1 1 0 4 4 0 4 1 0 1 0 0 0 3 0 0 0 0 1 1 1 3 0 4
 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 3 1 0 1 3 0 1 1 3 0 0 0 1 0 3 0 3 0 1 3
 2 1 0 3 0 0 0 0 1 1 0 1 0 0 1 1 1 3 0 3 0 3 0 1 3 1 1 0 4 0 0 1 0 4 0 0 1
 0 0 3 0 3 2 0 0 1 1 1 3 0 1 4 0 0 3 0 0 0 1 0 3 0 1 1 3 0 3 0 0 0 1 0 0 0
 0 2 1 1 3 0 1 0 0 0 3 0 1 1 4 0 0 0 1 3 0 4 0 1 0 0 0 0 3 3 4 1 1 0 0 2 0
 0 3 1 1 0 0 2 1 0 4 1 0 1 0 3 0 0 0 0 1 0 3 3 0 0 2 4 0 0 0 1 0 4 0 0 0 4
 3 3 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 1 0 4 0 1 3 0 0 3 2 3 1 1 1 1 3 3 0 3
 4 3 3 0 0 3 1 0 0 1 3 3 3 3 0 1 1 1 0 1 1 0 1 0 3 0 0 0 0 3 3 0 1 1 0 1 0
 0 1 3 0 0 3 0 1 1 0 0 0 1 1 0 1 0 0 0 0 1 0 0 3 0 1 4 4 0 0 0 4 1 0 0 1 3
 2 1 1 1 3 4 1 1 0 1 1 0 3 1 1 1 0 0 4 1 3 0 4 1 0 4 1 0 0 0 0 0 1 0 0 3 0
 0 0 2 0 3 4 1 3 2 0 0 0 1 3 1 1 0 3 4 0 3 3 3 0 1 1 0 0 0 0 0 0 0 0 1 1 0
 0 0 1 1 1 1 1 0 3 0 1 0 0 1 3 1 1 0 1 0 0 1 0 1 0 3 0 0 0 0 0 0 3 1 1 0 4
 0 1 1 0 3 2 4 0 0 1 0 0 1 0 0 0 1 3 3 1 0 0 1 0 0 1 3 1 0 0 3 0 0 1 1 3 0
 0 4 1 1 0 0 0 0 1 1 3 0 1 0 1 0 1 0 0 0 3 3 3 3 0 1 1 3 3 3 1 0 0 1 0 1 1
 3 0 0 0 0 0 4 1 1 0 1 0 1 3 3 0 0 1 1 4 1 1 0 0 0 0 0 0 0 2 0 0 0 3 0 0 1
 0 1 3 3 0 0 0 1 0 0 0 3 0 0 4 1 4 3 1 3 4 0 3 0 0 0 0 0 3 0 0 1 0 1 0 0 0
```

3. Results & Discussion:

The performance of each classification algorithm is evaluated using accuracy scores. The results show that KNN and Decision Trees achieve the highest accuracy in predicting diabetes. The clustering algorithms reveal different patterns within the dataset, allowing for potential insights into subgroups of individuals with similar characteristics.

4. Conclusion & Future Work:

In conclusion, the study demonstrates the effectiveness of classification algorithms, such as KNN and Decision Trees, in predicting diabetes. Additionally, clustering algorithms provide insights into inherent patterns within the dataset. Future work could involve further optimizing the classification models and exploring additional clustering techniques to gain more comprehensive insights into the data.

By conducting this study, aim to contribute to the field of diabetes prediction and provide a basis for further research in the development of accurate and efficient algorithms for diabetes diagnosis and patient management.