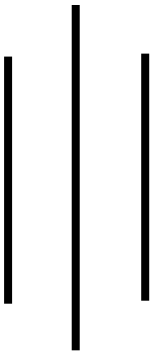
TRIBHUVAN UNIVERSTIY

INSTITUTE OF SCIENCE AND TECHNOLOGY



A SEMINAR REPORT ON

"COMPARATIVE ANALYSIS OF SVM AND KNN FOR HEART DISEASE PREDICTION"

Under the Supervision of Mr. Nawaraj Paudel CDCSIT, TU

Submitted to

Central Department of Computer Science and Information Technology
Institute of Science and Technology
Tribhuvan University

Submitted by

Binod Baral (Roll no. 26/079)

In the partial fulfillment for master's Degree in Computer science and Information Technology(M.Sc CSIT), 1st semester

Tribhuvan University

Institute of Science and Technology

Supervisor's Recommendation

I recommend that this seminar report, titled "**COMPARATIVE ANALYSIS OF SVM AND KNN FOR HEART DISEASE PREDICTION**" and prepared by **Mr. Binod Baral** under my supervision, be accepted as partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Information Technology. Mr. Baral has demonstrated a thorough understanding of the SVM and KNN and its application to heart disease prediction.

……..

Assoc. Prof. Nawaraj Poudel

(Supervisor)

Central Department of Computer Science and

Information Technology

Tribhuvan University

Institute of Science and Technology

LETTER OF APPROVAL

This is to approve the seminar report, titled "**COMPARATIVE ANALYSIS OF SVM AND KNN FOR HEART DISEASE PREDICTION**" prepared by **Mr. Binod Baral** in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Information Technology. The report has been well-studied and is satisfactory in scope and quality as a project for the required degree. Mr. Binod has demonstrated a thorough understanding of the K-Nearest Neighbor algorithm and its application to heart disease prediction. He has also shown a strong ability to conduct research and write clearly and concisely.

Evaluation Committee

……..……

Asst. Prof. Sarbin Sayami

(HOD)

Central Department of Computer Science and Information Technology

……

Assoc. Prof. Nawaraj Poudel

(Supervisor)

Central Department of Computer Science and Information Technology

……

(Internal)

ACKNOWLEDGEMENT

With great pleasure, I would like to express my deepest gratitude to the Central Department of Computer Science and Technology at **Tribhuvan University**.

First of all, I want to thank **Asst. Professor Sarbin Sayami**, the Head of the Central Department of Computer Science and Information Technology. His support and guidance have been invaluable throughout my academic journey.

I also want to acknowledge **Assoc. Professor Nawaraj Paudel** for his valuable guidance in carrying out in this project. His supervision and insights have been crucial to its success. I am grateful for his guidance, trust, and corrections in my seminar work.

Binod Baral(26/079)

ABSTRACT

Smart gadgets collect data from human bodies to analyze and predict future occurrences. During the study, SVM and KNN a machine learning algorithm, is implemented to predict heart attack. Thirteen features are considered, including personal details like chest pain type, blood pressure, cholesterol level, and heart rate. The implemented model is tested on the UCI healthcare heart disease dataset. Machine learning algorithms SVM and KNN are used for classification and regression tasks. SVM is a supervised learning algorithm that creates a hyperplane to separate two classes of data. KNN is a non-parametric algorithm that classifies a new data point based on the k nearest neighbors of the data point. SVM is more complex and accurate than KNN, but KNN is simpler and can handle complex or nonlinear boundaries. The choice of which algorithm to use depends on the specific problem being solved. In this study, SVM emerged as a more complex yet accurate approach, achieving an impressive accuracy of 88.78%, outperforming the KNN model's accuracy of 82.44%. Additionally, the SVM model exhibited higher precision (81.37%) and recall (95.40%) compared to the KNN model's precision (79.41%) and recall (84.37%). The F1 score, which considers both precision and recall, also indicated that the SVM model (87.83%) achieved a more balanced trade-off between these metrics compared to the KNN model (81.81%).

Keywords: *prediction, algorithm, SVM, supervised, features, chest pain, blood pressure, cholesterol, heart rate, UCI, healthcare, performance*

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Heart disease is the leading cause of death worldwide. An estimated 17.9 million people die each year from cardiovascular diseases (CVDs). Heart and blood vessel disorders account for CVDs, which include coronary heart disease, cerebrovascular disease, rheumatic heart disease, and others. Among people under 70, one third of CVD deaths are caused by heart attacks or strokes. [1]. To address the risk of developing heart disease, various methods and approaches are available. One such approach involves the use of computer tools, such as machine learning algorithms, to predict who is at risk. Support Vector Machines (SVM) and k-Nearest Neighbors (KNN) are two popular machine learning algorithms used for this purpose. SVM is a supervised learning algorithm that separates data into different classes using a hyperplane, maximizing the margin between the classes. It works by mapping data into a high-dimensional feature space and finding the optimal decision boundary. KNN, on the other hand, is a non-parametric and instance-based algorithm that classifies new data points based on their proximity to the existing data points. It assigns a class label to a new data point based on the majority class of its k nearest neighbors. By applying SVM and KNN algorithms to relevant medical data, researchers and healthcare professionals can gain insights into the factors and patterns that contribute to the development of heart disease and identify individuals who may be at a higher risk.

Healthcare data collected from various electronic devices of a patient reside in a format specified in electronic health record (EHR). The data available today are abundant, temporal, specific, contextual, and voluminous. To compare the effectiveness of different machine learning models, SVM, KNN, a comparative analysis is often conducted. This analysis involves evaluating the performance of these models in accurately classifying health disease cases as normal or abnormal. Performance metrics such as accuracy, precision, recall, and F1 score provide valuable insights into the model's predictive capabilities. Through feature importance analysis, the most influential factors contributing to health disease prediction can be identified. This

analysis provides a deeper understanding of disease patterns and risk factors, assisting in targeted interventions and personalized healthcare strategies.

1.2 Problem Statement

- A significant problem is identifying individuals at risk of heart disease, necessitating the use of machine learning algorithms such as SVM and KNN to enhance prediction accuracy.
- Discovering key factors for heart disease from complex healthcare data.

1.3 Objectives

- To compare SVM and KNN for health disease prediction and assess their effectiveness by evaluating performance metrics including accuracy, precision, recall, and F1 score.

CHAPTER 2: LITERATURE REVIEW

Healthcare data analytics poses challenges to research committees as it is voluminous and scanty in nature. Predictive models are implemented in smart devices like smart watches by Apple/Android/Fitbit and many more. The challenge that is still present in research committees is accuracy. The survey conducted here focuses on understanding the SVM model and also various predictive models implemented to achieve high accuracy and low computational complexity.

In a supervised machine learning techniques such as SVM, KNN, and Naive Bayes to forecast heart disorders. The R programming language was used to implement the machine learning algorithms. The algorithms' accuracy was used to assess their performance. KNN, Naive Bayes, and SVM were compared in terms of accuracy. The testing results show that the SVM algorithm correctly predicts heart illness 86.6 percent of the time [2].

In a proposed real-time prediction system for health issues based on cloud-based big medical data processing. The system uses machine learning algorithms to predict heart disease with an accuracy of 90.6%. The system is designed to be used in conjunction with electronic health records (EHRs), and it can be used to predict heart disease in real time. The system is still under development, but it has the potential to revolutionize the way heart disease is diagnosed and treated [3].

One such study conducted by Zhang compared the performance of KNN and SVM in classifying handwritten digits. The researchers found that KNN achieved higher accuracy rates compared to SVM, especially when dealing with datasets that have a large number of training instances. They concluded that KNN was more suitable for this particular classification task [4] .

Similarly, in a study by Wang, the researchers compared KNN and SVM for sentiment analysis in social media data. They found that KNN exhibited better performance in terms of precision and recall, particularly when dealing with imbalanced datasets. The study concluded that KNN was more effective in capturing minority classes and producing accurate sentiment classification results [5].

Furthermore, in 2019 conducted a comparative study between KNN and SVM for disease diagnosis using medical datasets. They observed that KNN outperformed SVM in terms of accuracy and achieved higher classification rates, particularly in scenarios where the dataset had noisy or overlapping features. The study recommended the use of KNN for disease diagnosis tasks in medical applications [6].

CHAPTER 3: METHODOLOGY

3.1 Methodology

Process diagram is a computational paradigm that focuses on the movement and transformation of data within a system. This flexible and modular approach enables efficient processing, parallelism, and scalability in complex data processing tasks.

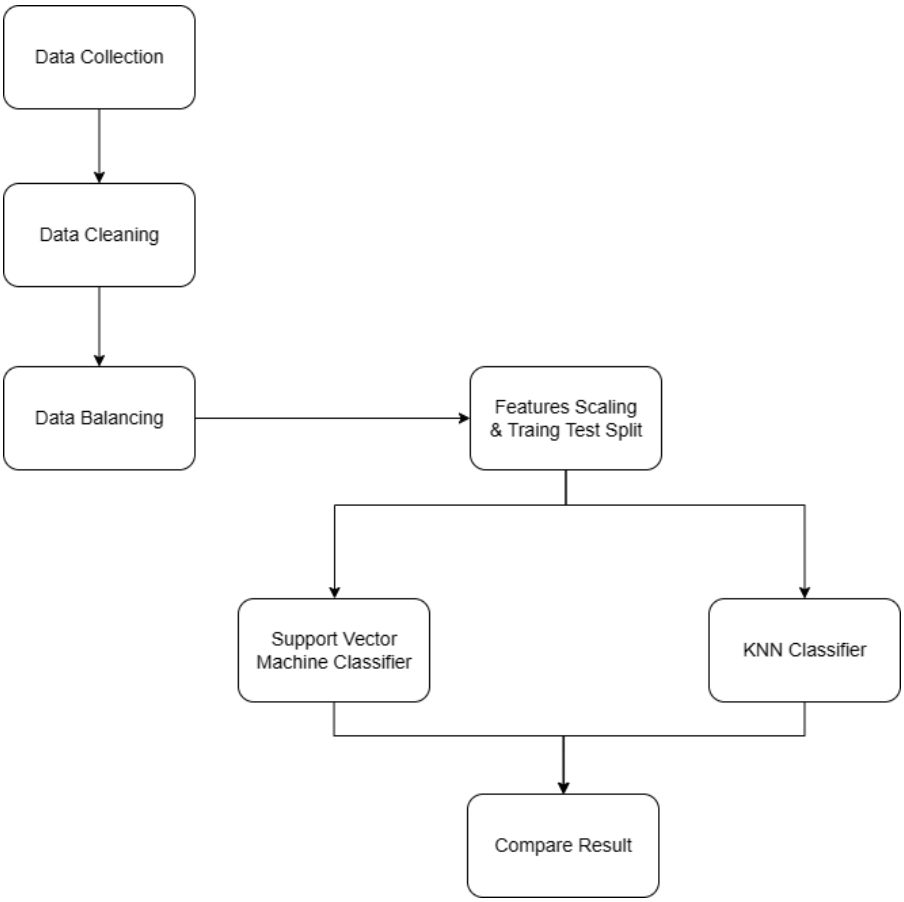


Fig 3.1: Process Diagram

3.2 Data Collection

This experiment is performed on data Heart Failure Prediction Dataset, file name “heart.csv” which is extracted from Kaggle website. It consists of 14 features and total of 1025 observations [7]. The "target" field refers to the presence of heart disease in the patient. It is integer valued 0 = no disease and 1 = disease.

3.2.1 Datasets(Sample)

Table 3.2.1: Dataset

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
52	1	0	125	212	0	1	168	0	1	2	2	3	0
53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
61	1	0	148	203	0	1	161	0	0	2	1	3	0
62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
58	0	0	100	248	0	0	122	0	1	1	0	2	1
58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
43	0	0	132	341	1	0	136	1	3	1	0	3	0
34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
51	1	0	140	298	0	1	122	1	4.2	1	3	3	0

Table 3.2.2: Attribute Information

1	age
2	Sex: 1:male , 0: female
3	chest pain type (4 values)
4	resting blood pressure
5	serum cholesterol in mg/dl
6	fasting blood sugar > 120 mg/dl
7	resting electrocardiographic results (values 0,1,2)
8	maximum heart rate achieved
9	exercise induced angina
10	oldpeak = ST depression induced by exercise relative to rest
11	the slope of the peak exercise ST segment
12	number of major vessels (0-3) colored by fluoroscopy
13	thal: 0 = normal; 1 = fixed defect; 2 = reversible defect
14	Target: 1 if the person has heart Disease, 0 if not

3.2.2 Visualizing the correlation among columns present in the dataset

In this study it is necessary to remove correlated variables to improve model. Correlations was found correlations using pandas “corr()” function and visualize the correlation matrix using plotly express.

Lighter shades represent positive correlation.

Darker shades represent negative correlation.

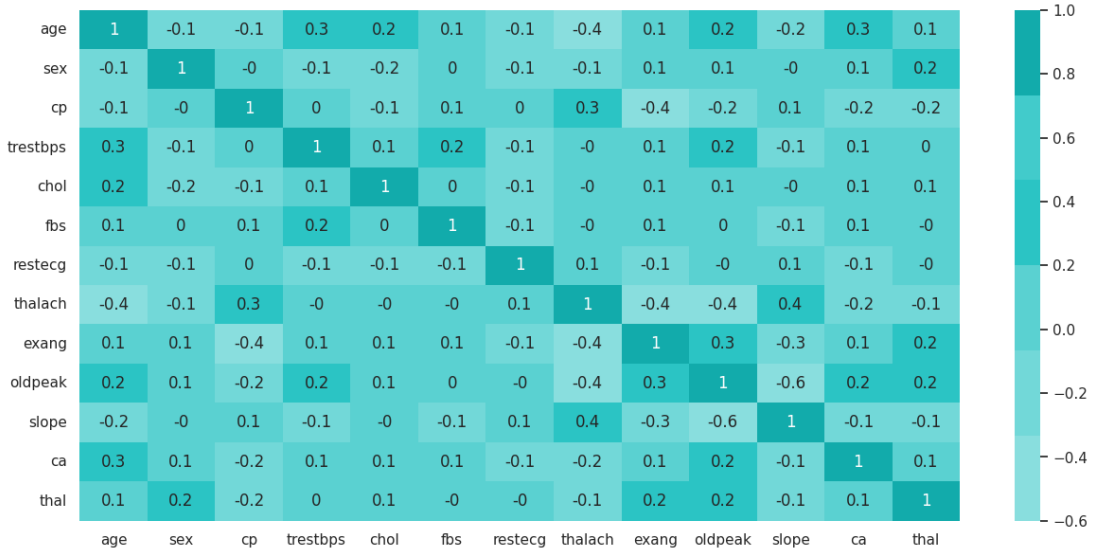


Fig 3.1.2: Correlation among data features.

3.3 Data Cleaning

- Identifying and removing outliers: In this case, outliers was identified and removed by checking each feature's data points. If a data point was more than six standard deviations away from the mean, it was considered an outlier and was removed from the dataset.
- Finding and filling missing values: Before further processing datasets missing values are identified and missing values are filled using mean of the other values in the column.
- Data balancing: The target values were analyzed to observe whether the datasets have a large class imbalance, meaning that one class has significantly more observations than the others. Because this can make difficult for model to learn to classify the minority class accurately.

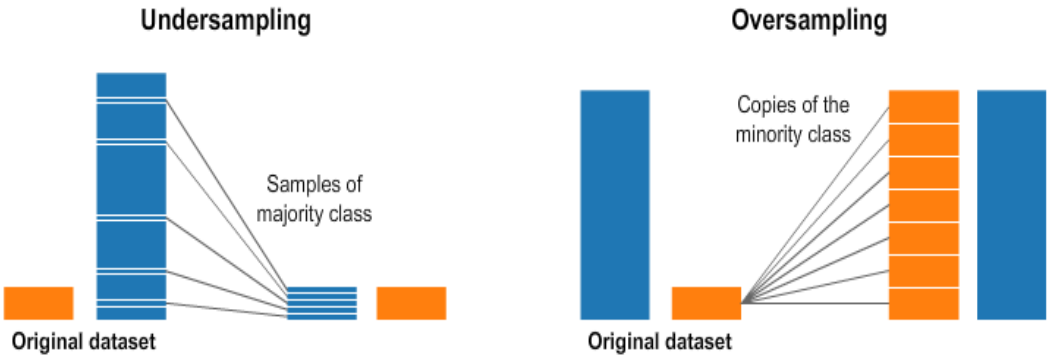


Fig 3.2: Oversampling and Under sampling

- Imblearn library was used to handle the imbalanced class distribution in the dataset. In this seminar, minority data are more than majority so, oversampled the minority class by creating more copies of the minority class observations. This helped to improve the performance of both SVM and KNN on the minority class.
- Normalizing the data. Normalization was done by scaling the data so that it has a mean of 0 and a standard deviation of 1.
- In this work, a SVM classification among different kernels (‘linear’,’poly’,’RBF’,’sigmoid’) RBF kernel was used and K-nearnest Neighbor value from (1 to 40) was implemented while taking the highest k-value according to accuracy using Python.

3.3. Support Vector Machines intuition

Hyperplane

To separate data points in the SVM classifier a hyperplane with maximum margin are obtained. And linear classifiers defined by maximum margin classes are called maximum margin classes.

Support Vectors

In the dataset data points closest to the hyperplane are called support vectors. Calculating margins enabled to define the separating line or hyperplane more accurately.

Margin

In data points with the closest separation, a margin is the gap between the two lines. The distance between the line and the nearest data point is calculated as a perpendicular

distance. By maximizing the separation gap it was able to get the maximum margin with SVMs.

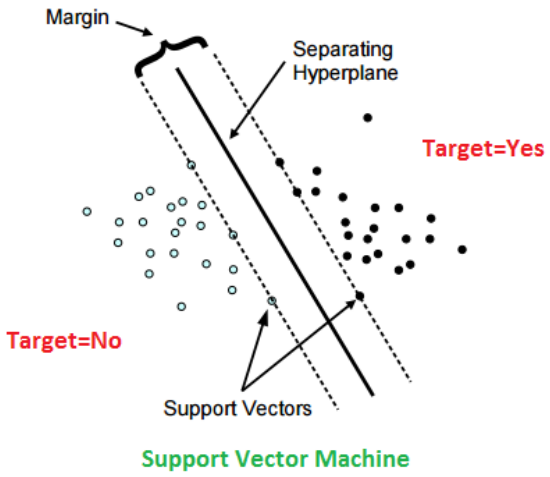


Fig 3.3: Support vector Machine hyperplane

Let n be the number of training examples { x_i, y_i }, i = 1,..., n, with each containing an input x_i and a class label y_i € {-1,1}. Each hyper plane has a weight vector (w) and a bias (b) that may be described using the equation below.

w^T. x + b = 0 ---------- (1)

Given the hyper plane, the following function can be used to classify training and testing data:

d(x) = σ(w.x + b) ---------- (2)

The previous function can be stated as follows when working with kernel functions.

$d(x) = \sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b$ ------(3)

Here, number of samples used for training is represented as N, $x_i$is the training samples input and y_i is the label for matching class, bias is represented by b, and the kernel function $k(x_i, x)$ that translates into an enlarged feature space from input vectors.

Radial basis function (RBF) kernel: The RBF kernel is a very powerful kernel function. It can learn non-linear relationships between the data points by computing the Gaussian distance between two data points. The Gaussian distance is a measure of how similar two data points are. The closer two data points are, the smaller the Gaussian distance will be.

$$k(x, y) = ((x.y) + c)^d$$

3.4 KNN

K-Nearest Neighbors (KNN) is a simple yet powerful machine learning algorithm for classification and regression. It works by finding the k most similar instances in the training set to a new instance, and then predicting the label of the new instance based on the labels of the k nearest neighbors.

k: The number of nearest neighbors to consider.

Distance metric: The distance metric was used to measure the similarity between distances.

Finding the value of k in K-Nearest Neighbors (KNN) was an essential step in utilizing the algorithm effectively. The value of k represents the number of nearest neighbors considered when classifying a new data point. The process of determining the optimal value of k involves a trade-off between model complexity and accuracy.

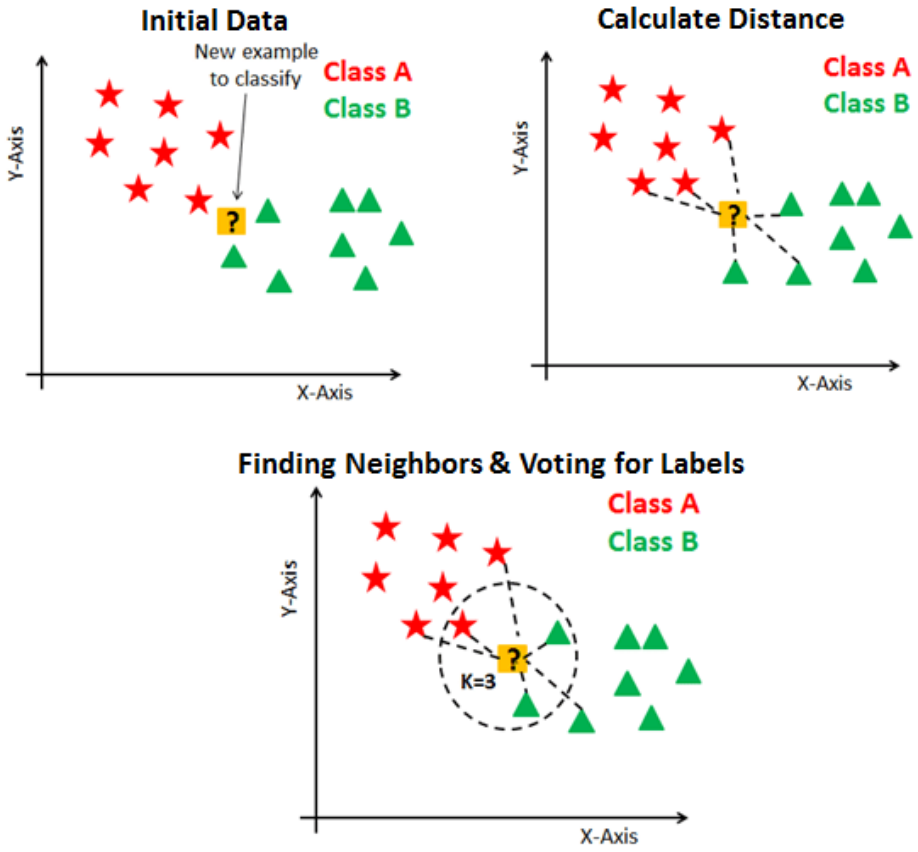


Fig 3.4: KNN classifier and finding value of K

Given a new instance x, the KNN algorithm finds the k most similar instances in the training set to x, where similarity is measured using a distance metric. The labels of the k nearest neighbors are then used to predict the label of x.The distance metric can be any metric that measures the similarity between two instances.

Input:

- Training set: X_train, y_train

- Test instance: x_test

- Number of neighbors: K

Procedure KNN:

1. Calculate the distance between x_test and each instance in X_train using a distance metric (e.g., Euclidean distance).

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2}$$

2. Sort the distances in ascending order and select the K nearest neighbors.

3. Retrieve the corresponding labels (y_train) for the K nearest neighbors.

4. Determine the majority class among the K neighbors.

5. Assign the majority class as the predicted class for the test instance.

3.5 Model evaluation

The initial step in the evaluation process involves predicting the health disease labels for the test dataset using the trained model. This prediction is made using the model's "predict" method applied to the test input data (X_test). The resulting predicted labels are then compared to the true labels in the test dataset to construct a confusion matrix. This matrix provides a visual representation of the true positives, true negatives, false positives, and false negatives, offering insights into the model's classification performance.

Derived from the confusion matrix are key evaluation metrics specific to heart disease prediction. Accuracy is utilized to gauge the overall correctness of predictions, while precision quantifies the proportion of predicted cases of heart disease that are truly correct, and recall measures the proportion of actual cases of heart disease that were accurately predicted.

Furthermore, the computation of loss, represented by the sum of false positives and false negatives, sheds light on instances of misclassification within the model. Collectively, these metrics offer a comprehensive assessment of the model's performance in identifying individuals at risk of heart disease. Additionally, visualizations such as plots illustrating training and validation loss, accuracy, and other relevant metrics across different stages of model training help monitor the model's learning progression and potential overfitting.

CHAPTER 4: IMPLEMENTATION

4.1 Implementation Tools and Packages Used

For this study, Python was used along with Google Collab as primary tools. Python, with its extensive libraries and frameworks for data analysis and machine learning. Google Collab, being a cloud-based Jupyter notebook environment, offers the convenience of working environment allocating 12 GB of RAM and 126 GB of HDD space by default.

The different libraries that are used in this project are as follows:

• Pandas, a Python library used for data analysis. It provides a variety of tools for loading, cleaning, manipulating, and analyzing data. Also used Pandas to load data into a DataFrame, which is a powerful data structure that makes it easy to work with data.

• NumPy used to provide a variety of mathematical functions and data structures that can be used for data analysis. Also to calculate statistical measures, perform linear algebra operations, and manipulate arrays.

• Scikit-Learn provides a variety of machine learning algorithms that can be used for classification, regression, clustering, and dimensionality reduction. In this study, Scikit-Learn is used to train a KNN model on data and to classify new data points.

• Seaborn library used for statistical data visualization which provides a variety of functions for creating informative visualizations. It was used Seaborn to create visualizations of data, such as scatter plots, line plots, and histograms [8].

4.2 Implementation Details

4.2.1 Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Fig 4.2.1 : Code for Importing libraries

4.2.2 Data Preprocessing

a) Finding Null values

Since there were no null values.

```
data.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

Fig 4.2.2 'a': code for finding null values

b) Data imbalance check

Datasets are also well balanced i.e. almost equal number of Healthy and Heart Disease patients.

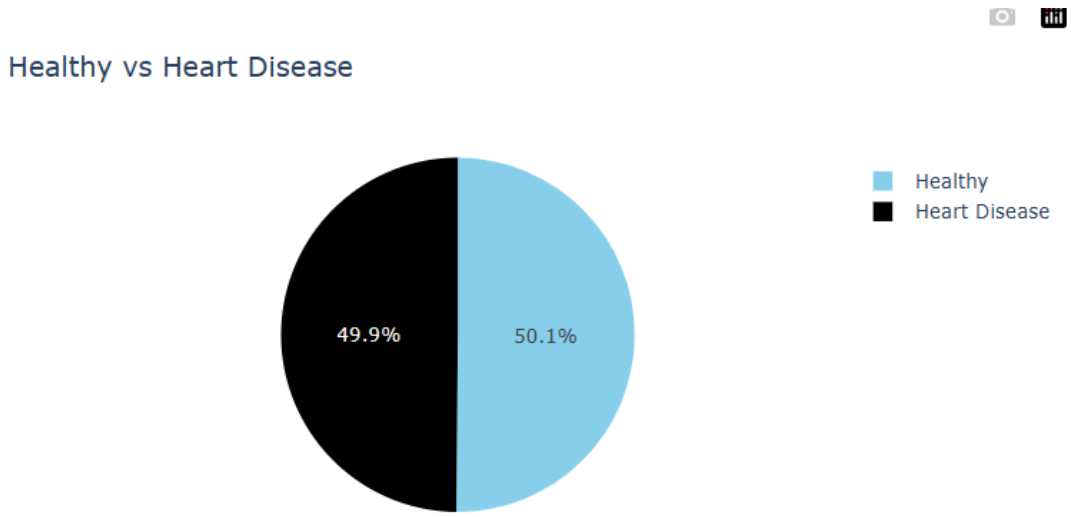
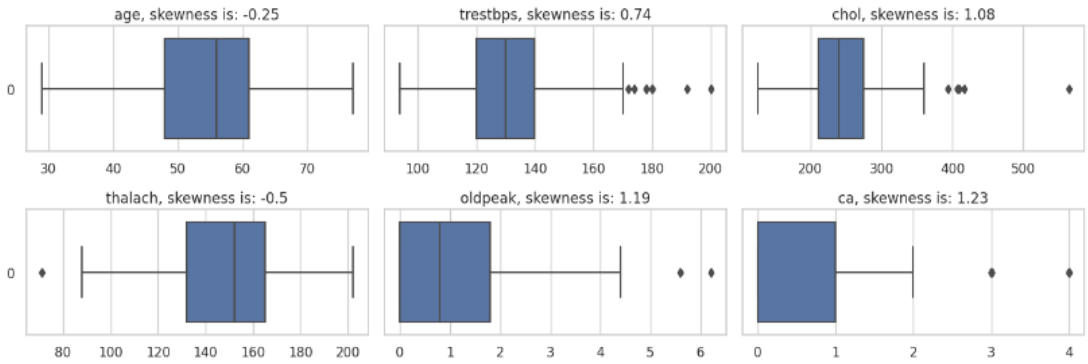


Fig 4.2.2 'b': Data balance identification

c) Checking and removing Outliers




```
for feature in df.columns[:-1]:

    mean=df[feature].mean()
    std=df[feature].std()
    #I have chosen to consider outliers all elements outside 6*std
    remove=(df[feature]<mean-6*std) | (df[feature]>mean+6*std)
    if (remove).any():
        #print("Removed element(s) with:",feature,round(df[feature].loc[remove].values[0],2))
        df=df.loc[~remove]
        df.reset_index()
print("# of outliers removed:",df.shape[0]-1025)
```

```
# of outliers removed: 24
```

Fig 4.2.2 'c': Checking and removing outliers

4.2.3 Transforming Data using Standard Scalar

```
st = StandardScaler()
data[cont_val] = st.fit_transform(data[cont_val])
```

Fig 4.2.3: Transforming Data using Standard Scalar

4.2.4 Splitting data into training and test dataset

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

Fig 4.2.4: Splitting data into training and test dataset

4.2.5 Training of SVM and KNN

a) SVM

In Python, I use scikit-learn to implement SVM with an RBF kernel. Support Vector Machine (SVM) with the Radial Basis Function (RBF) kernel which is a powerful classification algorithm.

```
from sklearn import svm
# Create an SVM classifier
clf = svm.SVC(kernel='rbf')

# Training the SVM model
clf.fit(X_train, y_train)  # X_train contains input features, y_train contains target labels

# Makingpredictions on new data
y_pred = clf.predict(X_test)  # X_test contains new data for prediction
```

Fig 4.2.5 ‘a’: Code implementation of SVM

b) k-Nearest Neighbors (KNN) with k=7:

KNN is also classification algorithm that assigns a data point's class label based on the majority class among its 7 nearest neighbors in the feature space.

The choice of k depends on the dataset and problem. Picking k=7 means considering the seven closest neighbors for classification decisions. It's a moderate value that provides a balance between overfitting (small k) and smoothing the decision boundary (large k).

```
from sklearn.neighbors import KNeighborsClassifier

# Creating a k-NN classifier with k=7
knn = KNeighborsClassifier(n_neighbors=7)

# Training the k-NN classifier
knn.fit(X_train, y_train)

# Making predictions on new data
y_pred = knn.predict(X_test)
```

Fig 4.2.3 ‘b’: Code Implementation of KNN

4.2.6 Model Evalutaion

To evaluate the performance of both Support Vector Machine (SVM) and k-Nearest Neighbors (k-NN) classifiers using classification reports and confusion matrices,was done by these steps:

The confusion_matrix function computes the confusion matrix, which is a table that summarizes the performance of a classification algorithm. It shows the number of true positives, true negatives, false positives, and false negatives.

The classification_report function generates a text report that includes precision, recall, F1-score, and support for each class in the classification. It provides a detailed overview of the model's performance.

```
#computing confusion matrix
confusion_matrix(y_test, predicted)

#Computing classification report
classification_report(y_test,preds))
```

Fig 4.2.6 : Code for Model evaluation

CHAPTER 5: RESULT AND ANALYSIS

In this experiment, the aim was to predict heart disease using two popular machine learning algorithms: Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). The dataset used for this analysis contained approximately 1025 data points with 14 features. The goal was to develop models that could effectively classify individuals as either having or not having heart disease based on the provided features.

The experiment aimed to assess the accuracy, precision, recall, and F1-score of both algorithms in predicting heart disease. The results obtained from this analysis would provide insights into the effectiveness of SVM and KNN in identifying individuals at risk of heart disease based on the given dataset.

5.1 Performance Measure

Confusion matrix was used to measure the performance of classification models. A confusion matrix is a table that shows the number of correct and incorrect predictions made by my model for each class. The table is divided into four quadrants:

True Positives (TP): These are the instances where the model correctly predicted the positive class.

True Negatives (TN): These are the instances where the model correctly predicted the negative class.

False Positives (FP): These are the instances where the model incorrectly predicted the positive class.

False Negatives (FN): These are the instances where the model incorrectly predicted the negative class.

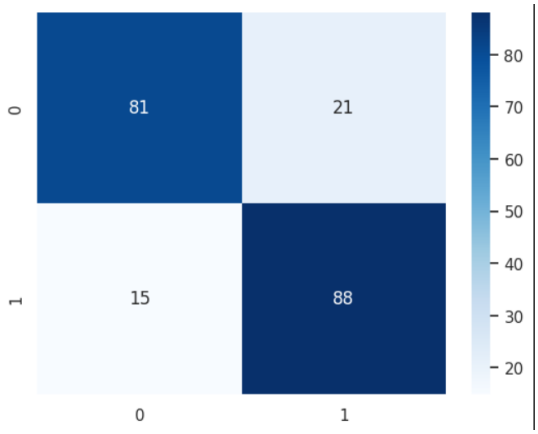


Fig 5.1: Confusion Matrix of KNN at k = 7

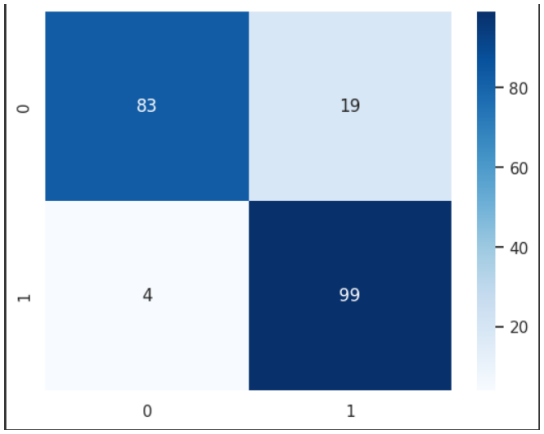


Fig 5.2: Confusion Matrix of SVM using 'rbf' kernel

5.2 Training Time

The training time indicate that k-Nearest Neighbors (k-NN) is faster to train compared to Support Vector Machine (SVM) with the given dataset and configurations:

SVM Training Time: 0.092 seconds

k-NN Training Time: 0.027 seconds

It's not uncommon for k-NN to have shorter training times than SVM, especially for small to moderately sized datasets, as k-NN's training time is generally less computationally intensive compared to SVM, which involves optimization during training. However, the specific training times can vary based on the dataset size, dimensionality, and the computational resources available.

5.4 Comparison between KNN and SVM

The values in the confusion matrix to calculate a number of performance metrics, such as accuracy, precision, recall, and F1 score. Accuracy is the overall percentage of predictions that my model made correctly. Precision is the percentage of positive predictions that were actually positive. Recall is the percentage of positive instances that my model correctly predicted. F1 score is a harmonic mean of precision and recall.

Models	TP	FP	FN	TN	Accuracy	Precision	Recall	F1
KNN	81	21	15	88	82.44%	79.41%	84.37%	81.81
SVM	83	19	4	99	88.78%	81.37%	95.40%	87.83

Table 5.4: Comparison between KNN and SVM

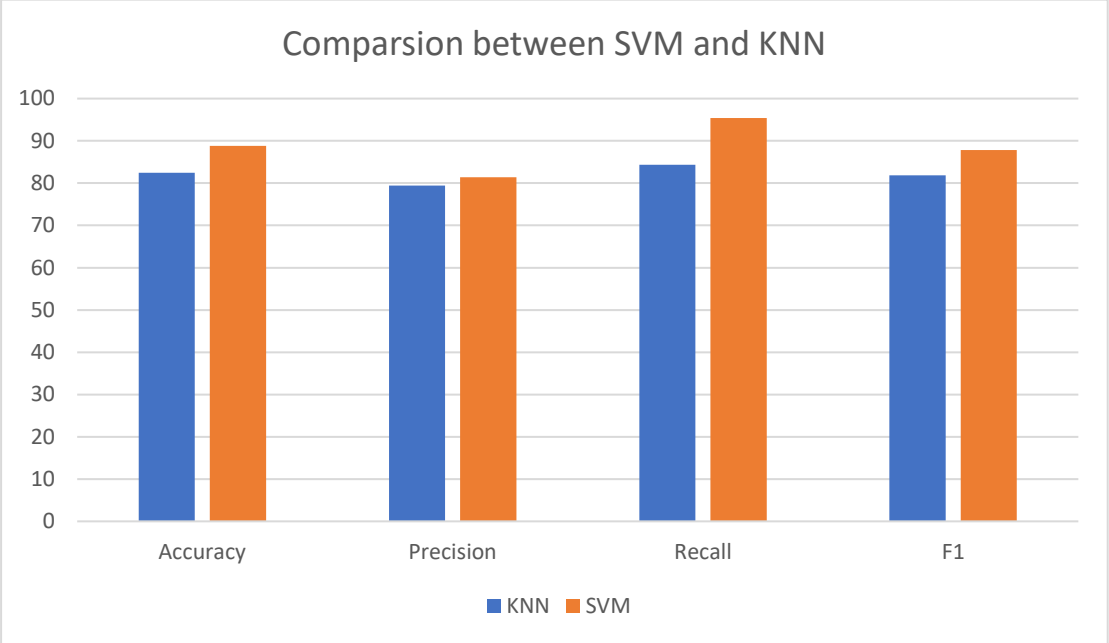


Fig 5.4: Bar Graph representing Comparison of KNN and SVM

CHAPTER 6: CONCLUSION

6.1 Conclusion

In conclusion, during experimentation, it is observed that SVM exhibited a slightly longer training time of approximately 0.092 seconds compared to k-NN's training time of approximately 0.027 seconds.

The performance evaluation of the K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) models reveals valuable insights into their predictive capabilities for the specific task. The SVM model demonstrates an impressive accuracy of 88.78%, outperforming the KNN model's accuracy of 82.44%. Additionally, the SVM model exhibits higher precision (81.37%) and recall (95.40%) compared to the KNN model's precision (79.41%) and recall (84.37%). The F1 score, which considers both precision and recall, also indicates that the SVM model (87.83%) achieves a more balanced trade-off between these metrics compared to the KNN model (81.81%). These results indicate that the SVM model is a more promising candidate for accurately predicting positive cases and effectively identifying true negatives, making it a suitable choice for the given task. However, further analysis and fine-tuning may be necessary to optimize the models' performance and adapt them to specific application requirements.

6.2 Future Recommendation

Based on the study when we compared Support Vector Machine (SVM) and k-Nearest Neighbors (k-NN) for classification, some suggestions for future research are Firstly, we should try adjusting the settings of both algorithms to see if we can make them work even better. We can also think about improving the data we use to train these algorithms. It might help to combine the results of multiple classifiers to get better results. We should also make sure we're preparing the data properly before using these algorithms. Testing them on bigger sets of data can help us see how they perform on larger problems. Dealing with data that isn't perfect, with some errors, is another area to look into, especially for KNN. Also we can make a software that takes user input and directly predict the heart disease based on the previous data available. Lastly, we need to think about how easy it is to understand the results of these algorithms versus how well they work.

REFERENCES

[1] W. teams, "Cardiovascular disease," World Health Organization, 2023. [Online]. Available: https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1.

[2] D. S. Anitha and S. Dr. N, "HEART DISEASE PREDICTION USING DATA MINING TECHNIQUES," *Journal of Analysis and Computation*, vol. XIII, no. II, pp. 0973-2861, February 2019.

[3] "BIG DATA ANALYTICS IN HEART DISEASE PREDECTION," *Journal of Theoretical and Applied Information Technology*, vol. Vol.98 No 11, no. ISSN: 1992-8645, pp. 1817-3195, 15 June 2020.

[4] J. L. Z. &. D. Q. Zhang, "A Comparative Study of KNN and SVM for Handwritten Digits Classification," in *International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2017.

[5] S. J. J. D. Z. &. G. H. Wang, "A Comparative Study of K-Nearest Neighbor and Support Vector Machine for Sentiment Analysis," in *2nd International Conference on Computer Science and Artificial Intelligence*, 2018.

[6] M. &. C. Y. Chen, "Comparative Study of KNN and SVM for Disease Diagnosis in Medical Applications," in *8th International Conference on Software and Computer Applications (ICSCA)*, 2019.

[7] D. W. Aha, "Kaggle," 2021. [Online]. Available: https://www.kaggle.com/fedesoriano/heart-failure-prediction. [Accessed sep 2023].

[8] S. 2.0, "Machine Learning Library (MLlib) Guide," 2023. [Online]. Available: https://spark.apache.org/docs/2.0.0-preview/mllib-guide.html.