AUTUMN INTERNSHIP PROJECT REPORT

Diabetes and Breast-Cancer Classification Models: Compared and Evaluated

Ananth Ram J,

Third Year in Integrated MSc in Economics, Dr B R Ambedkar School of Economics University, Bengaluru

Period of Internship: 25th August 2025 - 19th September 2025

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

This project explores the application of machine learning classification models for predicting diabetes based on diagnostic measurements from the Pima Indian Diabetes Dataset. The workflow involves standard steps including data loading, exploratory data analysis, preprocessing, and model training. Two common classification algorithms, K-Nearest Neighbours (KNN) and Support Vector Machine (SVM), are implemented and evaluated using key metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. The project then extends this workflow to the Breast Cancer dataset from scikit-learn to assess the generalizability of the approach and compare model performance across different medical classification problems. The evaluation metrics and visualizations (confusion matrices, ROC curves) are analysed to determine the effectiveness of each model on both datasets.

2. Introduction

The prediction of diseases like diabetes and breast cancer is a critical application of machine learning in healthcare. Early and accurate prediction can lead to timely intervention and improved patient outcomes. This project focuses on building and evaluating classification models for these tasks using a standard machine learning pipeline.

The project utilizes the Pima Indian Diabetes Dataset and the Breast Cancer dataset, both widely used benchmarks in machine learning. The technology involved includes Python and popular libraries such as pandas for data manipulation, matplotlib and seaborn for data visualization, and scikit-learn for machine learning algorithms and evaluation metrics.

The procedure used follows a typical supervised learning approach:

1. **Data Loading:** Loading the datasets into pandas DataFrames.
2. **Exploratory Data Analysis (EDA):** Understanding the data through descriptive statistics, visualizations (histograms, correlation heatmaps), and checking for missing values.
3. **Data Preprocessing:** Splitting the data into training and testing sets and scaling the features to ensure models are not biased by the magnitude of features.
4. **Model Training:** Training KNN and SVM classification models on the pre-processed training data.
5. **Model Evaluation:** Evaluating the trained models on the test data using various classification metrics and visualizations to assess their performance.

The purpose of this project is to demonstrate a practical machine learning workflow for binary classification, compare the performance of different models on medical datasets, and understand the importance of various evaluation metrics in the context of healthcare predictions.

During the initial phase of my internship, the training followed a carefully structured progression that built both technical and professional competencies. It began with **Python Basics – 1**, where I learned about data, variables, lists, and loops, establishing a strong programming foundation. This was followed by **Python Basics – 2**, which introduced me to core data structures, and **Python Basics – 3**, where I explored classes, functions, and object-

oriented programming to write modular and reusable code. After a brief pause in the schedule, the training resumed with **Python Basics – 4**, focusing on powerful libraries such as NumPy and Pandas for data handling and analysis. Once the Python groundwork was in place, the sessions transitioned toward applications, starting with a **Machine Learning Overview**, then moving into practical implementation through the **Regression Lab** and **Classification Lab**, where I applied concepts to real datasets. The training further expanded into modern AI with an introduction to **LLM Fundamentals**, and finally, it concluded with a dedicated session on **Communication Skills**, ensuring I could articulate technical insights effectively in collaborative and professional settings. This sequence created a smooth flow from programming basics to applied machine learning, while also emphasizing soft skills essential for real-world impact.

3. Project Objective

The primary objectives of this project are:

- Implement a standard machine learning workflow for binary classification tasks.
- Apply the workflow to the Pima Indian Diabetes Dataset and predict diabetes outcomes.
- Compare KNN and SVM classifiers on the diabetes dataset using multiple evaluation metrics (accuracy, precision, recall, F1-score, ROC-AUC).
- Extend the workflow to the Breast Cancer dataset and evaluate KNN and SVM performance in the same way.
- Analyse metric trade-offs and discuss the workflow's generalizability to other classification problems.

4. Methodology

The methodology followed in this project involves the following detailed steps:

1. **Data Loading:** The Pima Indian Diabetes Dataset was loaded directly from a URL into a pandas DataFrame. The Breast Cancer dataset was loaded using load_breast_cancer from sklearn.datasets and converted into pandas DataFrames for features (X_bc) and a pandas Series for the target variable (y_bc).
2. **Exploratory Data Analysis (EDA):**
    - The shape of each dataset was checked using .shape.
    - Basic information about the datasets, including data types and non-null counts, was obtained using .info().
    - Descriptive statistics (mean, std, min, max, quartiles) were generated using .describe().
    - Missing values were checked using .isnull().sum().
    - Correlation heatmaps were generated using seaborn.heatmap to visualize the relationships between features and the target variable.
    - Distribution plots (histograms) were generated using .hist() on the DataFrames to visualize the distribution of individual features.

3. **Data Preprocessing:**

    - For both datasets, the features (X) and the target variable (y) were separated.
    - The datasets were split into training and testing sets using train_test_split from sklearn.model_selection with a test_size of 0.2 (20% for testing), a random_state for reproducibility, and stratify=y to ensure the proportion of the target class is the same in both training and testing sets.
    - Feature scaling was performed using StandardScaler from sklearn.preprocessing. The scaler was fitted only on the training data (X_train_scaled = scaler.fit_transform(X_train)) and then used to transform both the training and testing data (X_test_scaled = scaler.transform(X_test)). The same process was repeated for the breast cancer dataset (scaler_bc, X_train_bc_scaled, X_test_bc_scaled).

4. **Train/Test Split:**
   Prior to training any models, each dataset was split into a training set and a testing set. This was done using the train_test_split function from sklearn.model_selection. A test_size of 0.2 (20% of the data) was allocated for the testing set, and the remaining 80% was used for the training set. A random_state was set to 42 to ensure the split was reproducible. Additionally, stratify=y was used to ensure that the proportion of the target variable (Outcome for Diabetes, target for Breast Cancer) was the same in both the training and testing sets. This is particularly important for classification problems to maintain representative class distributions in both sets.

5. **Model Selection:**

   The project focused on comparing two common and fundamental classification algorithms:

    - **K-Nearest Neighbors (KNN):** A simple instance-based learning algorithm that classifies a data point based on the majority class of its 'k' nearest neighbors in the feature space.
    - **Support Vector Machine (SVM):** A powerful algorithm that finds an optimal hyperplane to separate data points into different classes. A linear kernel was chosen for simplicity and interpretability in this project. These models were selected as they represent different approaches to classification and are commonly used as baseline models.

   These models were selected as they represent different approaches to classification and are commonly used as baseline models.

6. **Model Training:**

    - **K-Nearest Neighbors (KNN):** A KNeighborsClassifier was instantiated with n_neighbors=5 and trained on the scaled training data (knn.fit(X_train_scaled, y_train) and knn_bc.fit(X_train_bc_scaled, y_train_bc)).

- **Support Vector Machine (SVM):** An SVC classifier was instantiated with kernel="linear" and random_state=42 and trained on the scaled training data (svm.fit(X_train_scaled, y_train) and svm_bc.fit(X_train_bc_scaled, y_train_bc)).

7. **Model Evaluation:**

   - Predictions were made on the scaled test data for both models (y_pred_knn = knn.predict(X_test_scaled), y_pred_svm = svm.predict(X_test_scaled), y_pred_knn_bc = knn_bc.predict(X_test_bc_scaled), y_pred_svm_bc = svm_bc.predict(X_test_bc_scaled)).

   - **Confusion Matrices:** Confusion matrices were generated using confusion matrix from sklearn. metrics and visualized using seaborn.heatmap to show the counts of true positives, true negatives, false positives, and false negatives.

   - **Classification Reports:** Classification reports were printed using classification_report from sklearn. metrics to provide precision, recall, F1-score, and support for each class.

   - **ROC Curves and AUC:** ROC curves were generated by calculating the False Positive Rate (FPR) and True Positive Rate (TPR) at various threshold settings using roc_curve from sklearn.metrics. For SVM, decision_function was used to get scores for the ROC curve, while for KNN, predict_proba was used to get probabilities. The Area Under the ROC Curve (AUC) was calculated using roc_auc_score and displayed on the plot legend.

   - **Metric Comparison Table:** A pandas DataFrame was created to tabulate the key evaluation metrics (Accuracy, Precision, Recall, F1-Score, ROC-AUC) for each model on each dataset, providing a clear summary for comparison.

Code can be inspected at: https://github.com/ananthramj/ISI-Autumn-Internship-2025-ARJ/blob/main/Final_Code.py

5. Data Analysis and Results

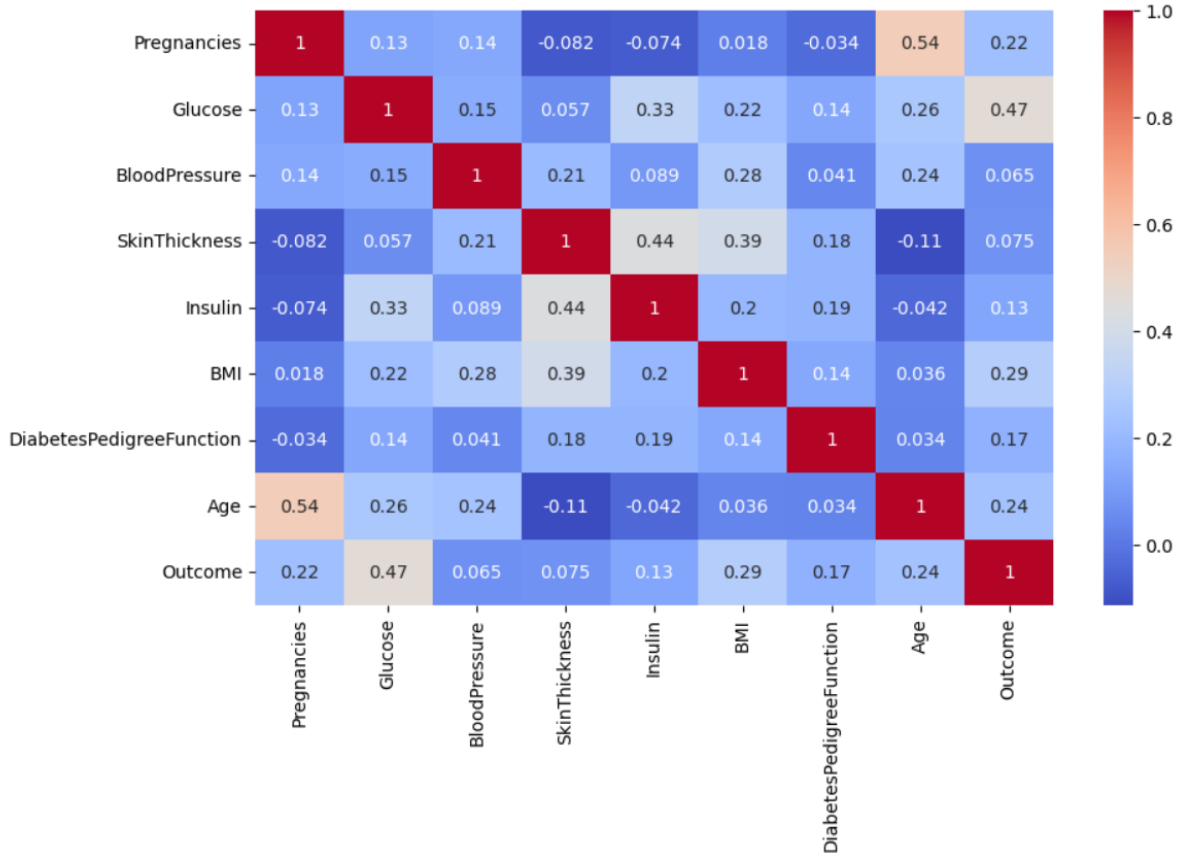
Basic Statistics – Diabetes Dataset

```python
# Basic EDA
print(df.shape)
print(df.info())
df.describe()
```

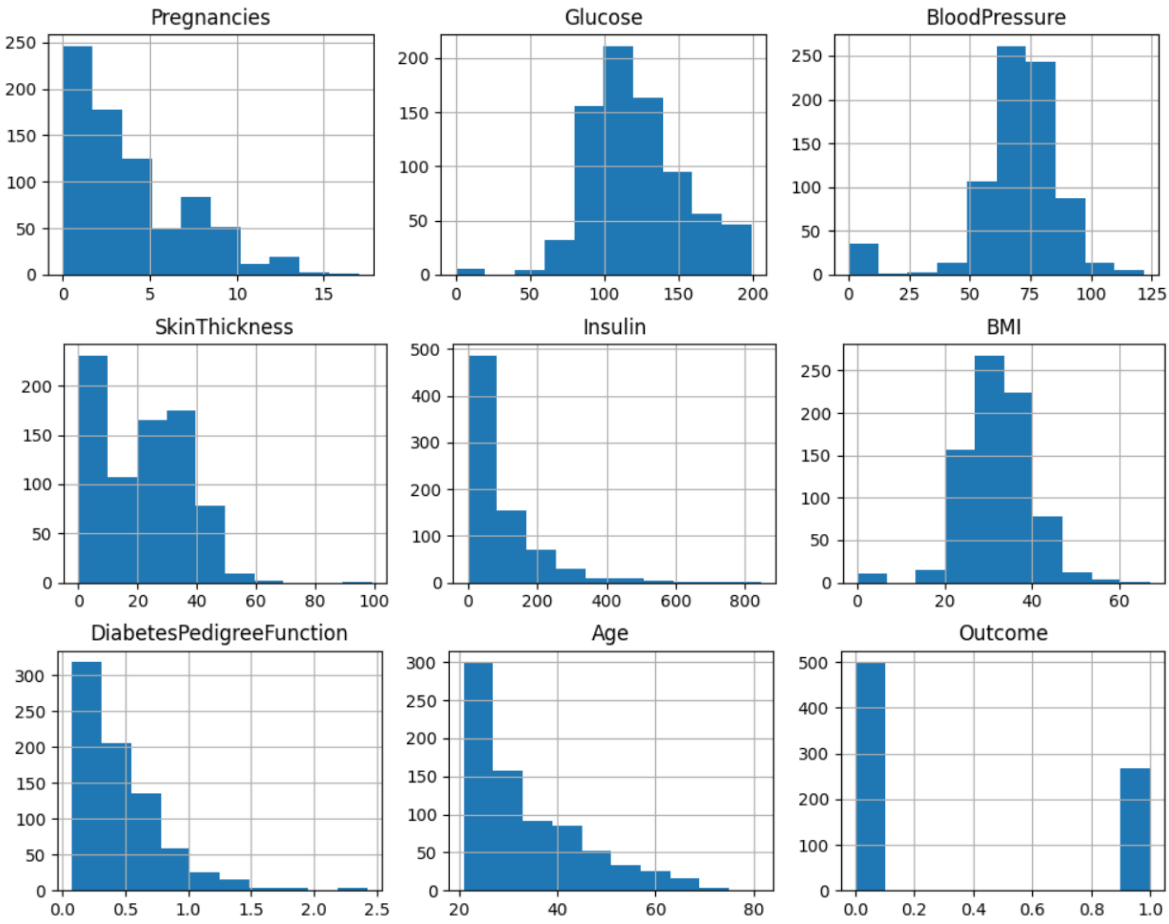
```
(768, 9)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Correlation heatmap – Diabetes Dataset



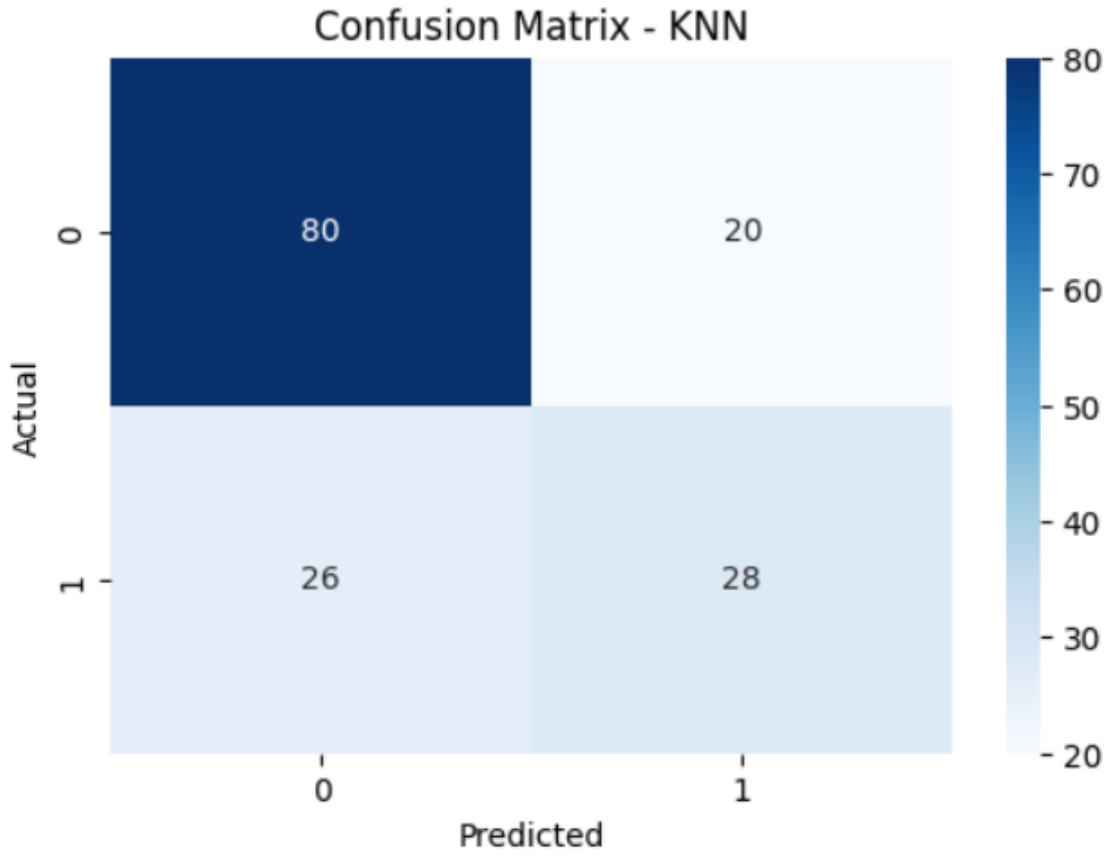
Distribution plots – Diabetes Dataset



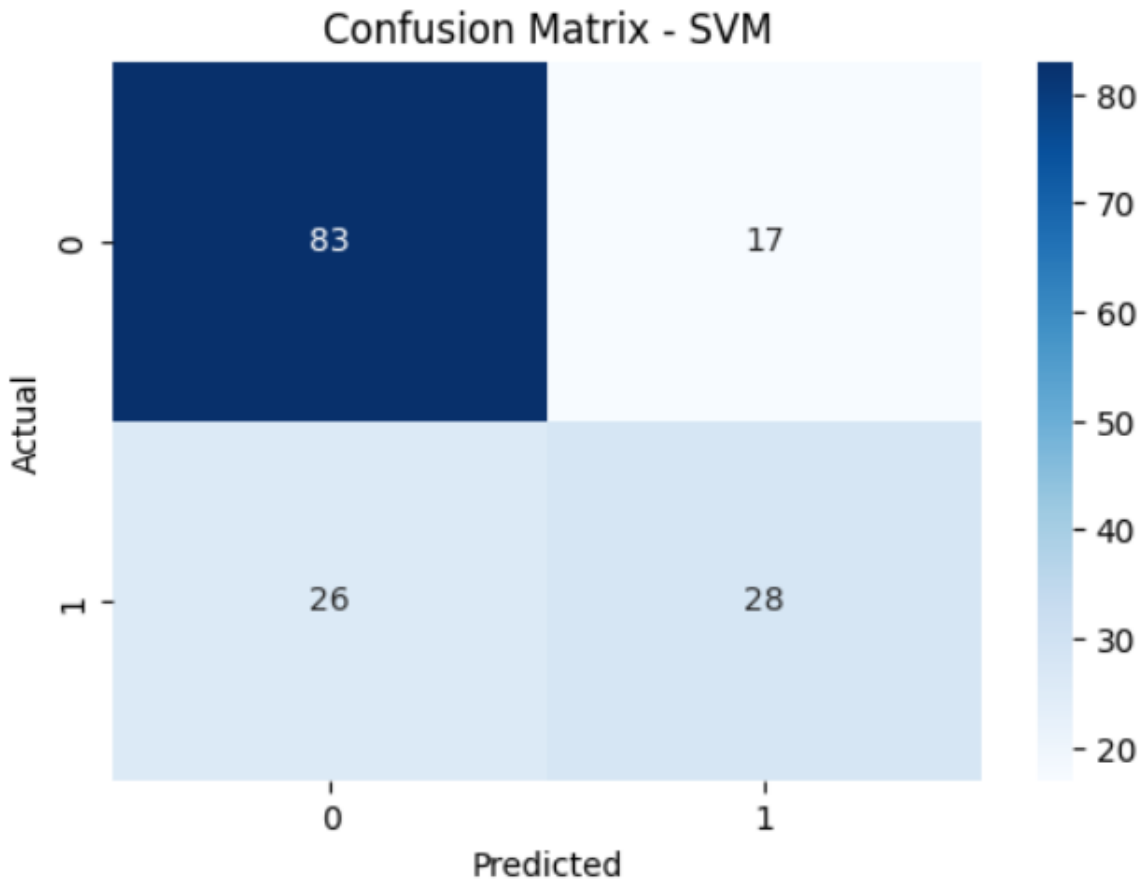
Diabetes Dataset Results:

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
KNN	0.70	0.58	0.52	0.55	0.74
SVM	0.72	0.62	0.52	0.57	0.83

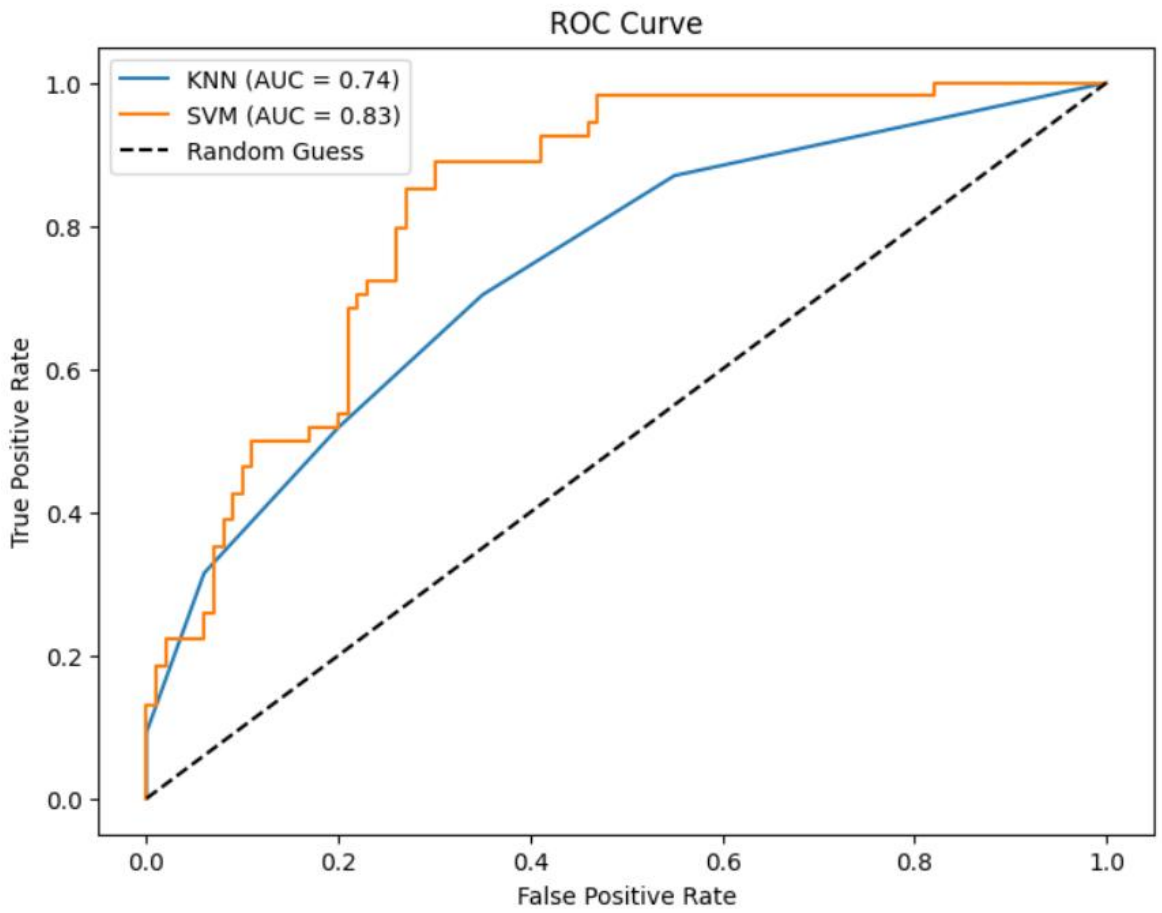
Confusion Matrix (KNN):
True Negatives: 80
False Positives: 20
False Negatives: 26
True Positives: 28



Confusion Matrix (SVM):
True Negatives: 83
False Positives: 17
False Negatives: 26
True Positives: 28



ROC Curve (Diabetes): The SVM model's ROC curve is higher and to the left of the KNN curve, indicating better discriminative power, as reflected in the higher AUC score (0.83 vs 0.74).



Breast Cancer Dataset Results:

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
KNN	0.96	0.96	0.97	0.97	0.98
SVM	0.97	0.99	0.97	0.98	0.996

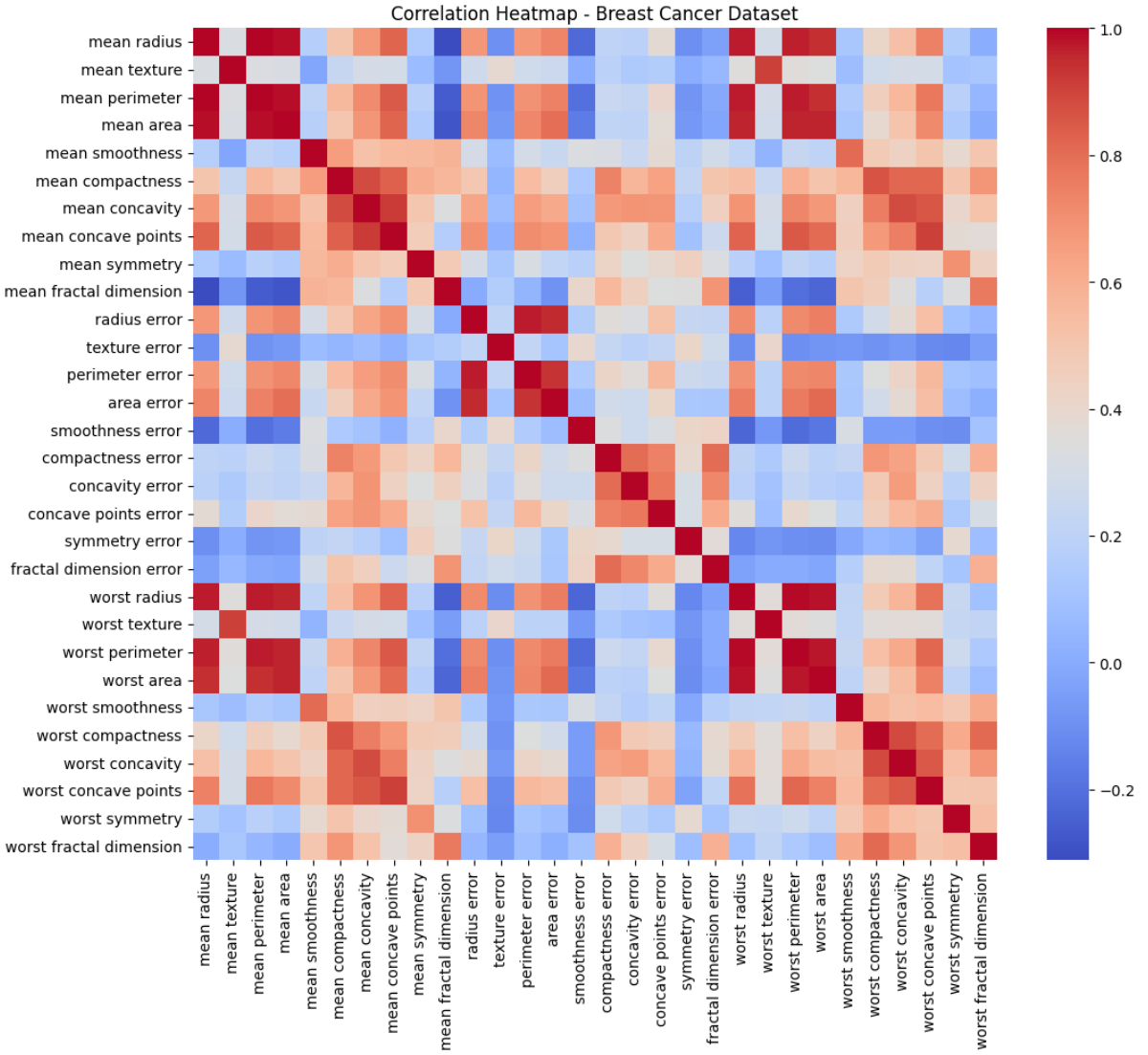
Basic Statistics:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883

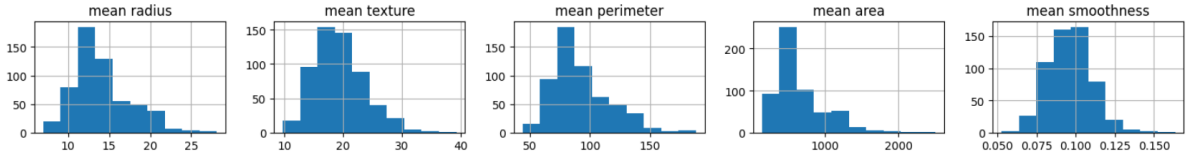
5 rows × 30 columns

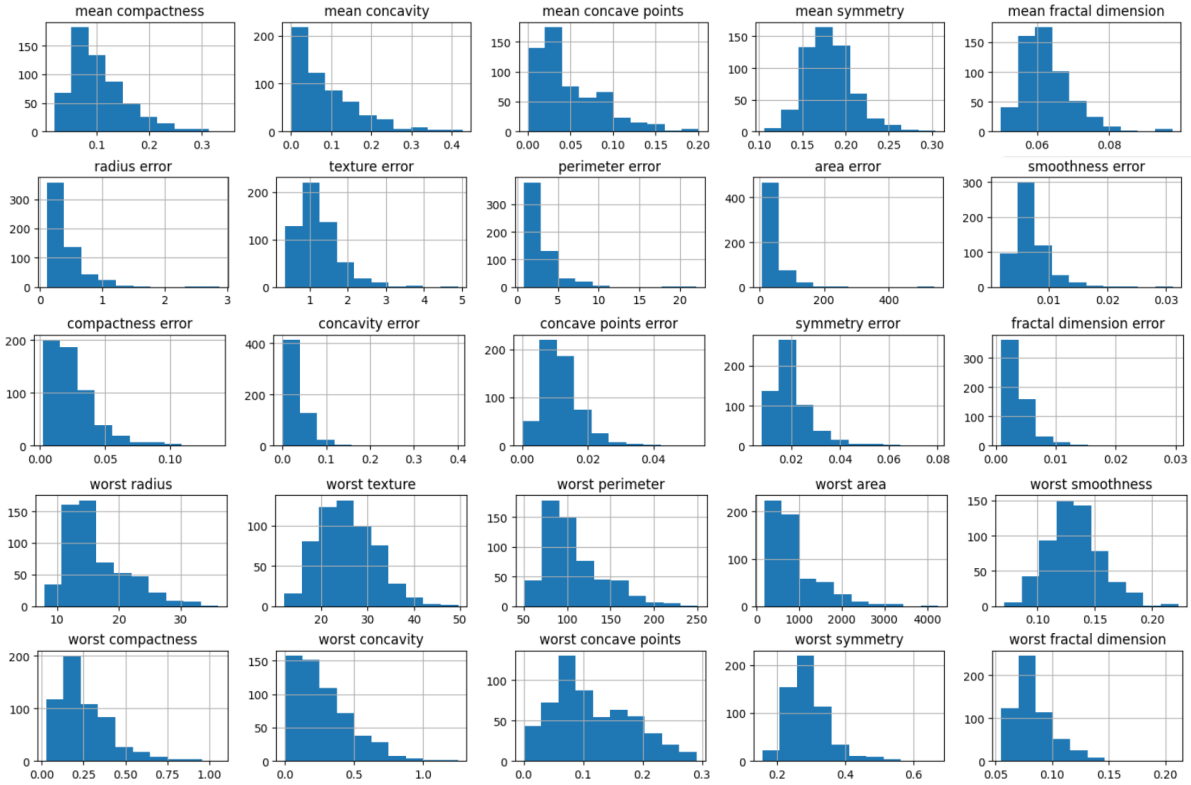
...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
...	25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
...	24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
...	23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
...	14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
...	22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

Correlation heatmap:

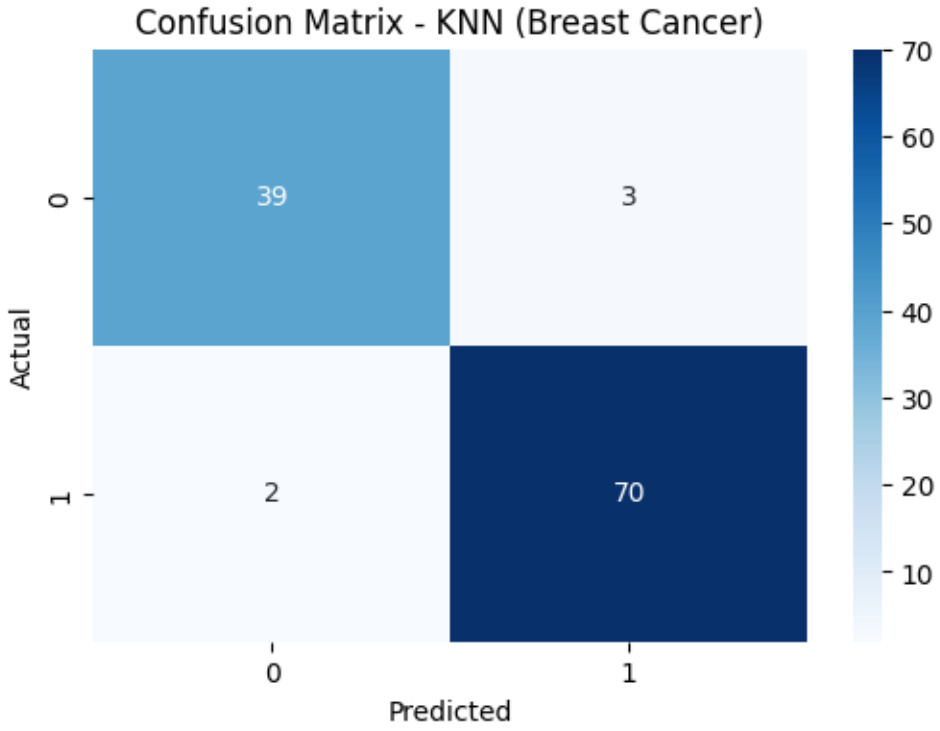


Distribution plots:



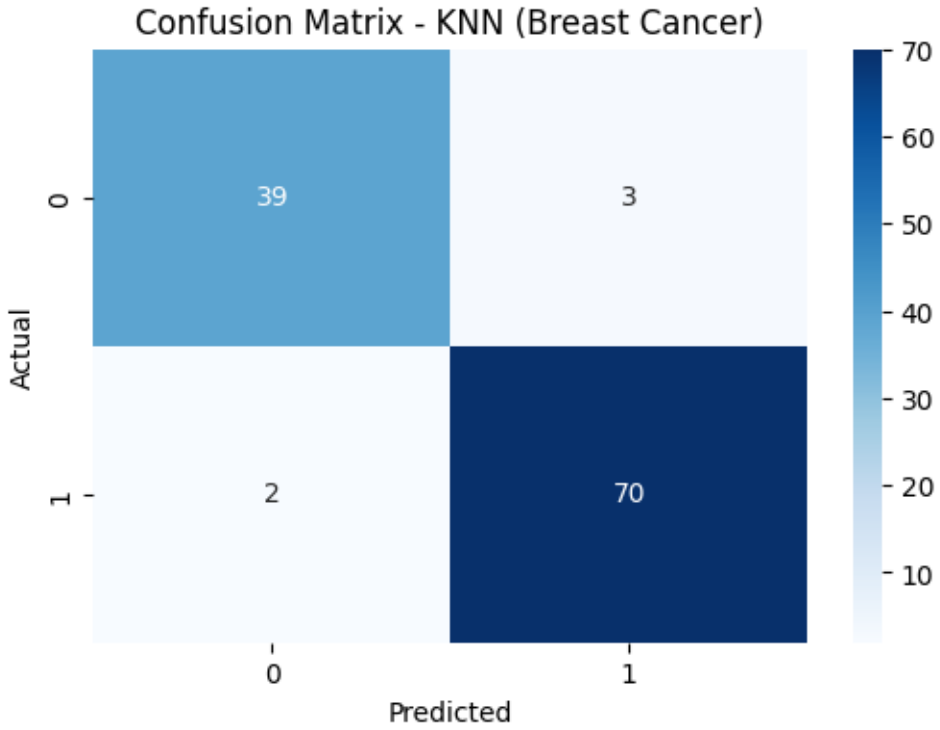


Confusion Matrix (KNN - Breast Cancer):
True Negatives: 39
False Positives: 3
False Negatives: 2
True Positives: 70

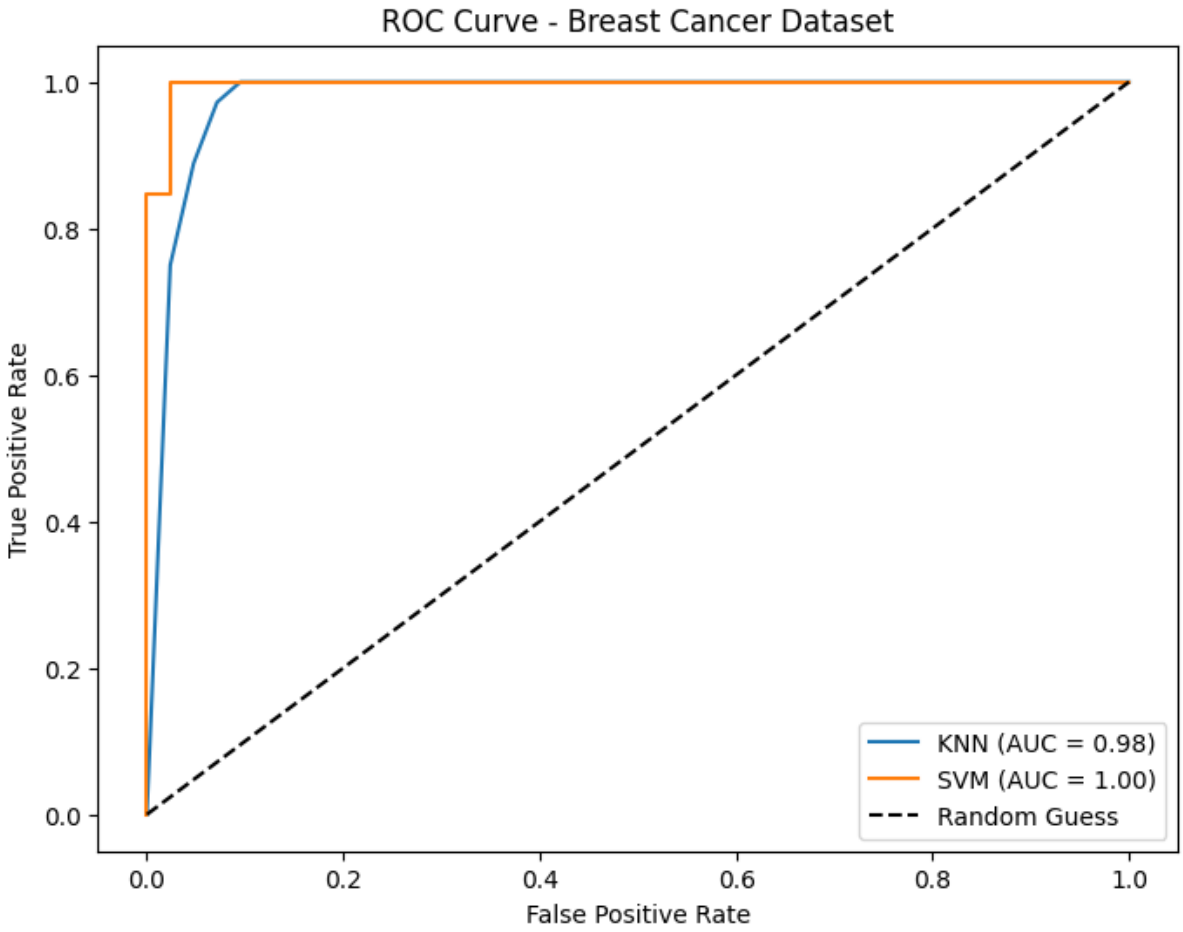


Confusion Matrix (SVM - Breast Cancer):
True Negatives: 41

False Positives: 1
False Negatives: 2
True Positives: 70



ROC Curve (Breast Cancer): Both models show excellent performance with curves close to the top-left corner. The SVM model has a slightly higher AUC (1.00 vs 0.98), suggesting near-perfect discrimination on this dataset.



6. Conclusion

This project successfully implemented and evaluated a machine learning workflow for binary classification on the Pima Indian Diabetes and Breast Cancer datasets.

On the **Diabetes dataset**, the SVM model generally outperformed the KNN model, particularly in its ability to distinguish between classes as indicated by the higher ROC-AUC.

On the **Breast Cancer dataset**, both models performed exceptionally well, with SVM showing a slight edge across most metrics. The high performance on this dataset suggests strong predictive features.

The **trade-offs between metrics** highlight the importance of choosing appropriate evaluation measures based on the problem context, especially in medical diagnosis where minimizing false negatives is crucial.

The implemented workflow is **generalizable** to other binary classification tasks, providing a solid foundation. However, adapting EDA for high-dimensional data, considering feature engineering, exploring diverse models, and hyperparameter tuning are important considerations for optimizing performance on different datasets.

Recommendation for Future Work:

- Explore additional classification algorithms (e.g., Random Forest, Gradient Boosting) and compare their performance.

- Perform hyperparameter tuning for the models using techniques like cross-validation to potentially improve results.
- Investigate feature selection or dimensionality reduction techniques for datasets with a large number of features.
- Address potential data imbalance issues if encountered in other datasets.

7. APPENDICES

1. GitHub Repository relevant to this project (Public): https://github.com/ananthramj/ISI-Autumn-Internship-2025-ARJ/
2. Specific link to final code hosted on GitHub: https://github.com/ananthramj/ISI-Autumn-Internship-2025-ARJ/blob/main/Final_Code.py
3. Code Demonstration Video: https://github.com/ananthramj/ISI-Autumn-Internship-2025-ARJ/blob/main/Video_Code_Demonstration.mp4
4. Google Colab Drafting Notebook: https://colab.research.google.com/drive/1OTD-1MuOMs48LdUxLXbu8uLQPJrWStQy?usp=sharing
5. Pima Indian Diabetes Dataset: https://raw.githubusercontent.com/plotly/datasets/master/diabetes.csv
6. Sci-kit learn dataset: The Breast Cancer Wisconsin (Diagnostic) Dataset is one of the datasets built into scikit-learn. While the dataset is built into scikit-learn, it shouldn't be dismissed as a fake toy dataset, but rather an old, filtered dataset. All entries in the scikit-learn dataset come from the original Breast Cancer Wisconsin dataset. Review more on their page: https://sklearner.com/scikit-learn-load_breast_cancer/