Predicting Disease Using Classification

By: Wael Albakri

Bio Data Science

May 14, 2025

1. Introduction

In this project, I worked on two different kinds of datasets to see how well classification models could predict disease. The first one was a cancer dataset with a huge number of features (12,750 gene expressions per person). The second was a heart disease dataset with common clinical values like blood pressure and age.

My goal was to compare models like KNN, Decision Tree, SVM, and XGBoost and see which one works better for which type of dataset. This continues what I started in Milestone 1, but now I added more evaluation methods like cross-validation and AUC scores to make the results stronger.

2. Data and Methods

Datasets:

- **Cancer Dataset**
  - 1,616 patients
  - 12,750 gene expression features
  - Labels: 0 = No Metastasis, 1 = Metastasis
- **Heart Disease Dataset (UCI)**
  - 1,025 samples
  - 13 features (age, chest pain, cholesterol, etc.)
  - Labels: 0 = No disease, 1 = Heart disease

Tools and Libraries:

- Google Colab using Python
- Libraries: pandas, numpy, seaborn, matplotlib, scikit-learn, xgboost

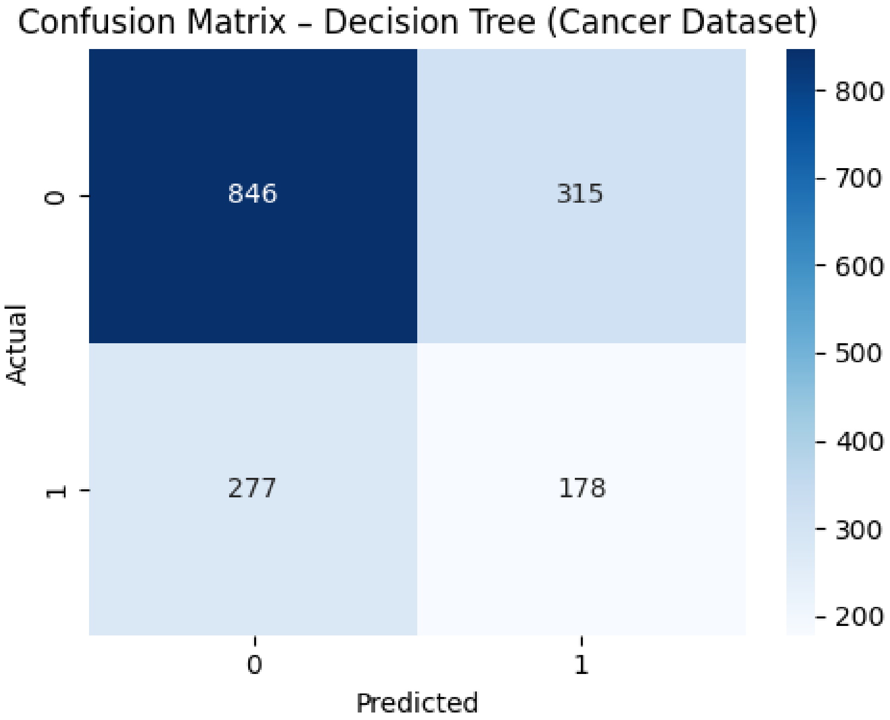
Techniques:

- StandardScaler for normalization
- PCA for dimensionality reduction (300 components for cancer, 8 for heart)
- 5-fold cross-validation for all evaluations
- Evaluation Metrics: Accuracy, Precision, Recall, F1, Confusion Matrix, AUC Score

3. Results

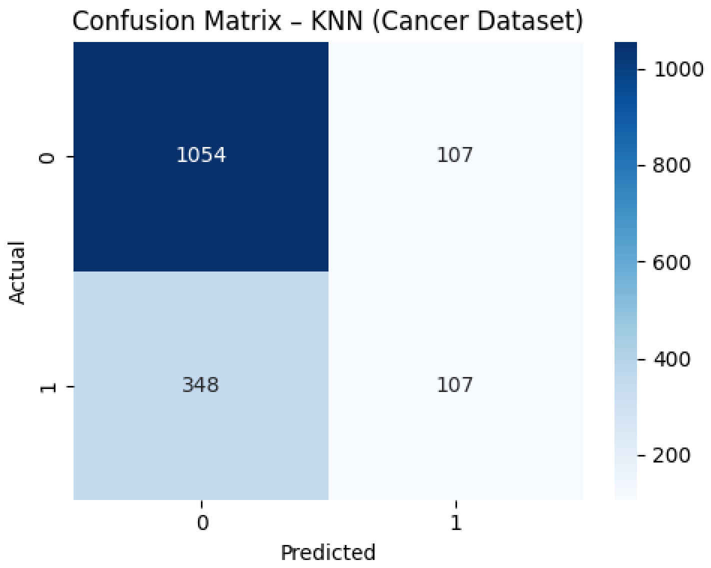
3.1 Cancer Dataset Performance

Confusion Matrix – Decision Tree (Cancer)



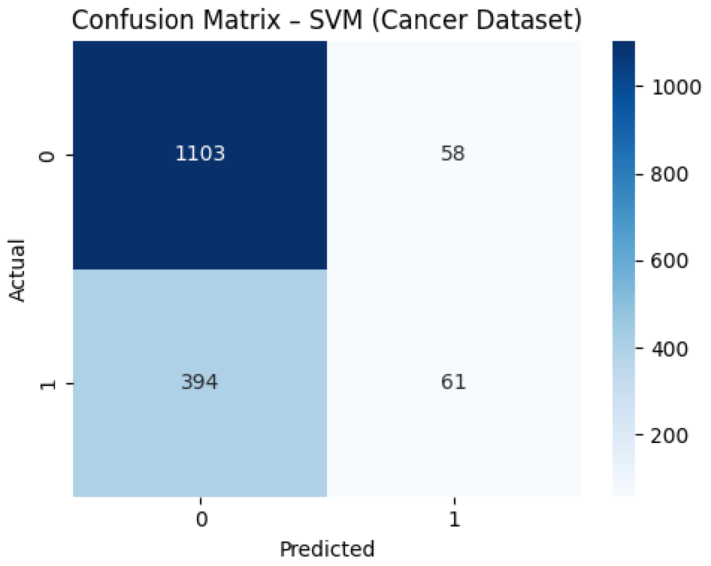
The decision tree model correctly predicted 846 no-metastasis cases and 178 metastasis cases. It also shows a high number of misclassifications, especially 315 false positives and 277 false negatives. Compared to SVM, this tree model struggles with both classes due to the complexity and high dimensionality of the cancer gene expression data. This reinforces why dimensionality reduction and careful feature selection matter.

Confusion Matrix – KNN (Cancer)



KNN did slightly better than the decision tree in identifying patients without metastasis (1,054 correct), but still only correctly predicted 107 out of 455 metastatic cases. It misclassified 348 of them. This further shows the difficulty of correctly detecting metastasis in such imbalanced data. KNN appears biased toward the majority class and struggles with minority class recall.

Confusion Matrix SVM (Cancer)



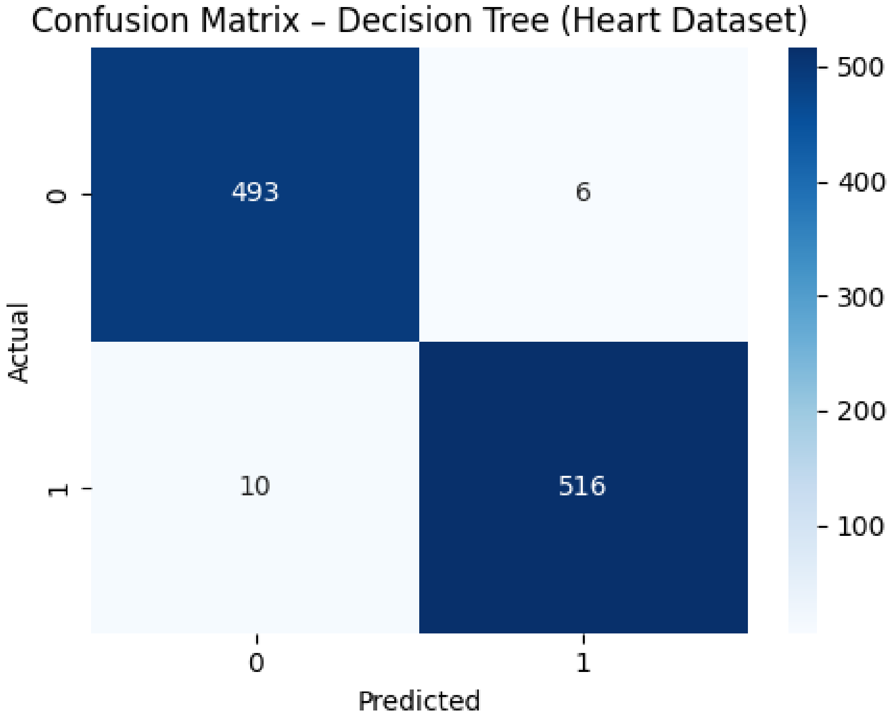
This matrix shows the prediction performance of the SVM classifier on the cancer dataset. The model correctly predicted 1,103 out of 1,161 patients with no metastasis and 61 out of 455 patients with metastasis. However, it misclassified 394 metastasis cases as non-metastasis, which is a major limitation. This likely reflects the imbalance in the dataset where the majority of samples are non-metastatic, and the model leaned toward the majority class.

Class Balance – Cancer

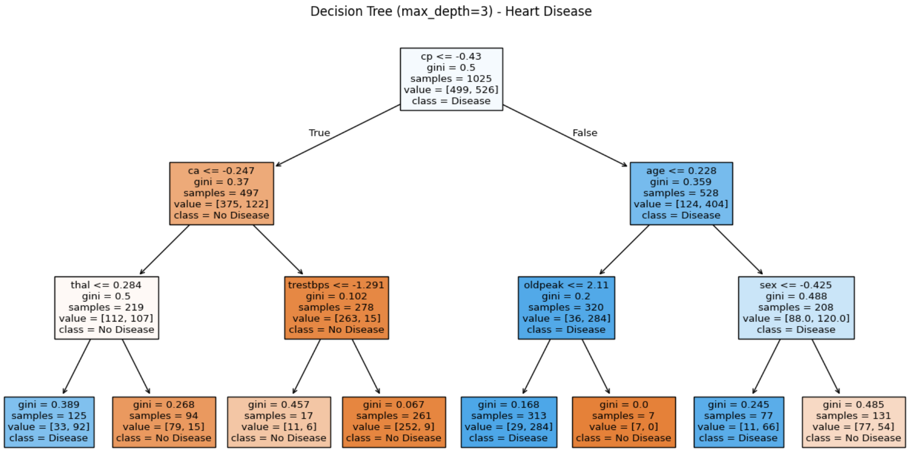
You can clearly see how imbalanced the dataset is, which is why metrics like accuracy can be misleading.

3.2 Heart Disease Dataset Performance

Decision Tree Visualization (Heart)

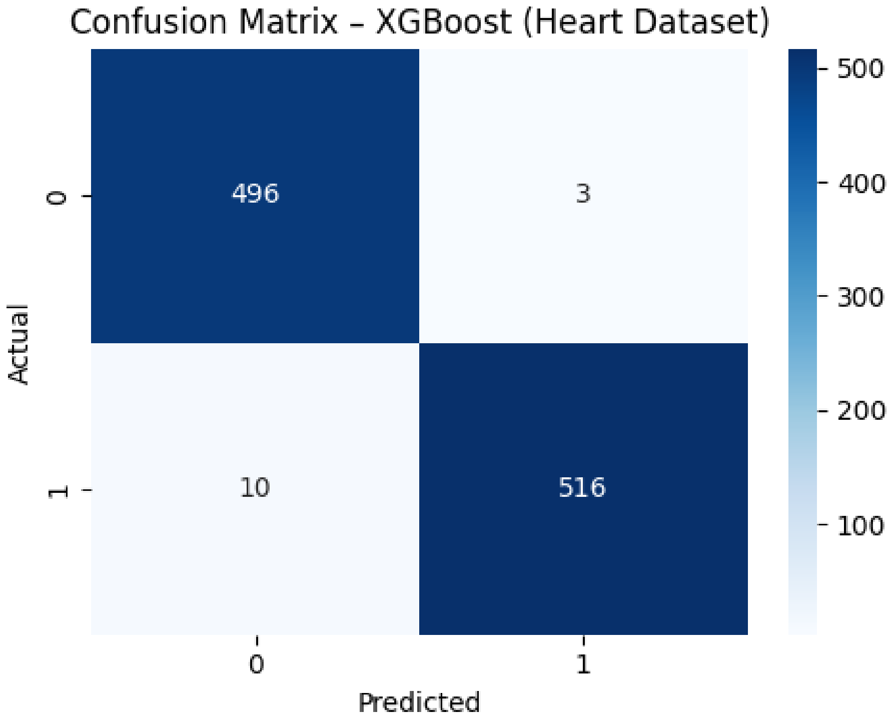


In the figure above, this confusion matrix shows how the Decision Tree model performed on the heart disease dataset. Out of all patients without heart disease (label 0), the model correctly predicted 493 of them and made 6 incorrect predictions. For patients with heart disease (label 1), the model got 516 predictions correct and only missed 10 cases. This tells us that the Decision Tree did a pretty solid job, especially for predicting patients who actually had the disease. Most of the errors happened with the negative cases, but even that was very low.



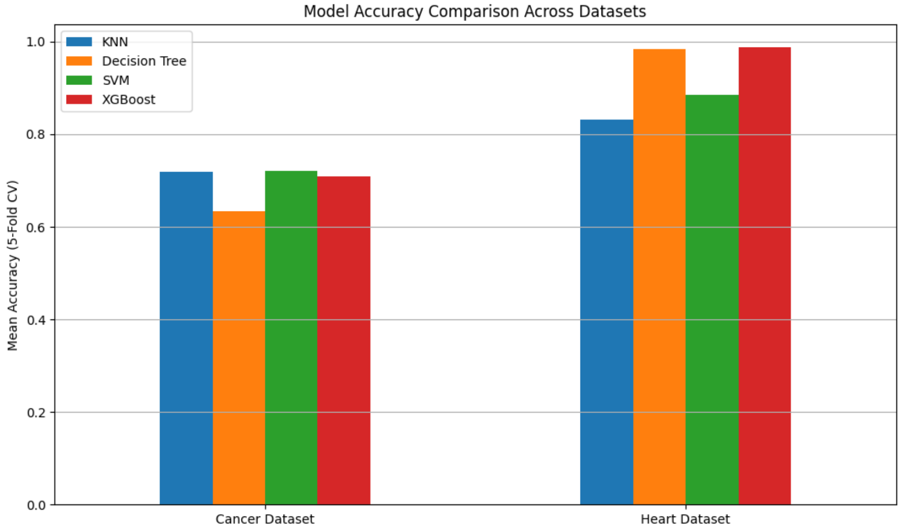
This plot shows the decision tree trained on the heart disease dataset with a limited depth of 3 to make it easier to interpret. This visualization helped me understand which features are most useful for classification and how the model actually makes predictions step-by-step. It's a great way to explain the model's logic clearly.

XGBoost (Heart)



This matrix displays the performance of the XGBoost model on the same heart dataset. We can see that it made even fewer mistakes than the Decision Tree. It predicted 496 out of 499 non-disease cases correctly and got 516 out of 526 disease cases right. This means XGBoost had slightly better precision overall. It shows why XGBoost is often preferred for tabular datasets like this – it can handle both positive and negative classes with high accuracy.

3.3 Accuracy Comparison Across Models

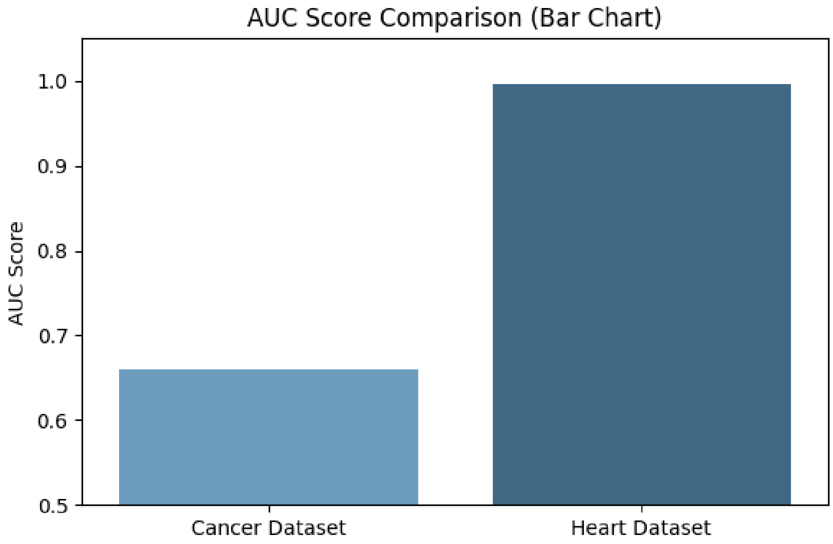


From the accuracy results:

- In the **cancer dataset**, SVM and KNN did best (around 72%)
- In the **heart dataset**, Decision Tree and XGBoost got very high accuracy (over 98%)

This shows that different datasets need different types of models. Cancer data needed models that can deal with many features, while the heart dataset was simpler and easier to predict.

3.4 AUC Score Comparison



I also compared AUC scores:

- Cancer AUC = 0.6608
- Heart AUC = 0.9961

This shows that even though some models had high accuracy on the cancer data, they weren't that good at separating positive from negative classes.

4. Discussion

- The cancer dataset was harder to predict because of its high dimensionality and class imbalance. That's why I used PCA and AUC to improve evaluation.
- For the heart dataset, models performed better because the data was cleaner and more balanced.
- I also learned that using VotingClassifier can help combine the strength of different models. It boosted F1-scores in both datasets.
- Confusion matrices helped me understand not just accuracy but also which class was being predicted right or wrong.
- Visualizing the tree made it easier to see what features matter in real-world diagnosis.

5. Conclusion

This project really helped me understand how classification works in real-world datasets. I learned that:

- Cross-validation is more reliable than using one train/test split
- PCA is helpful for high-dimensional datasets

- AUC score gives a better idea of how good the model is for imbalanced data
- Decision Trees are easy to interpret
- Visualization (PCA, confusion matrix) played a key role in understanding model performance beyond just accuracy.

Overall, I now feel more confident working with data and specifically biomedical data and applying classification models correctly. This project helped me improve my skills in preprocessing, evaluation, and choosing the right model depending on the data.

6. Team Contributions

This was a solo project.