# Machine Learning Report

**Akash Bahri**
College of Engineering
Northeastern University
Toronto, ON
Bahri.a@northeastern.edu

## Abstract

In this report, I explored the performance of various supervised learning algorithms on two interesting datasets. The learning algorithms that I have used for this assignment is Artificial Neural Networks, Decision Trees with pruning, K nearest neighbour, Boosting and Support Vector Machine. The first dataset is useful in the medical industry for breast cancer classification. The second dataset includes an interesting classification of song tracks into genres based on various features. The learning curve, model complexity and training time of each algorithm on both datasets have been explored and analyzed.

## 1 Datasets

I used the following two datasets as classification problems to be solved using Neural Networks. They are both very distinct in terms of number of quantity of data and complexity of feature dependency and we can observe how statistically varying datasets can be processed and used in neural networks.

Table 1: The basic feature of both datasets.

| | Data Set Characteristics | Attribute Characteristics | Associated Tasks | Number of Instances | Number of Attributes |
|---|---|---|---|---|---|
| **Dataset 1 – Song tracks** | Multivariate | Real | Classification | 17734 | 29 |
| **Dataset 2 – Cancer** | Multivariate | Real | Classification | 569 | 33 |

### 1.1 Data characteristics

The characteristics of both datasets are quite different from each other. I applied feature engineering and normalization techniques to better understand and analyze the data. Both datasets were somewhat complex, so I performed data preprocessing to clean and transform the data. This included handling missing values, outliers, feature engineering, and data splitting to divide the dataset into training and testing sets, using a specific percentage for testing. Additionally, I utilized data encoding methods such as label encoding for categorical features.

The song dataset is divided into two files. One contains publish related info on multiple songs like composer, date recorded, genre, language, licenses etc. While in the other file it contains numerical representation of various features of the audio like acousticness, energy, tempo, valence, instrumentalness, etc. With the help of these attributes linked thru common track id I have performed various data analysis techniques to preprocess the data and identify most relevant features.

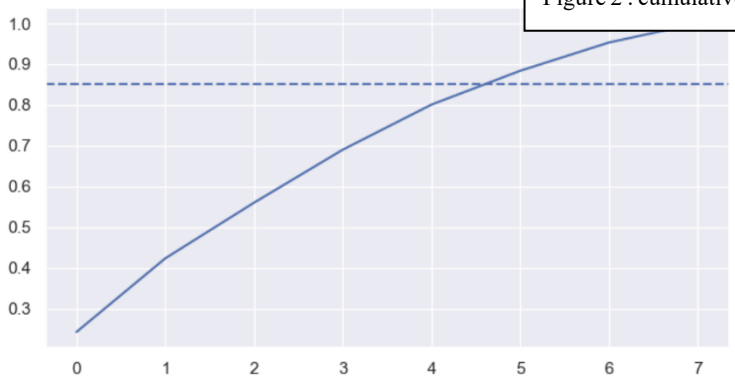
| | track_id | acousticness | danceability | energy | instrumentalness | liveness | speechiness | tempo | valence |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0.416675 | 0.675894 | 0.634476 | 0.010628 | 0.177647 | 0.159310 | 165.922 | 0.576661 |
| 1 | 3 | 0.374408 | 0.528643 | 0.817461 | 0.001851 | 0.105880 | 0.461818 | 126.957 | 0.269240 |
| 2 | 5 | 0.043567 | 0.745566 | 0.701470 | 0.000697 | 0.373143 | 0.124595 | 100.260 | 0.621661 |
| 3 | 10 | 0.951670 | 0.658179 | 0.924525 | 0.965427 | 0.115474 | 0.032985 | 111.562 | 0.963590 |
| 4 | 134 | 0.452217 | 0.513238 | 0.560410 | 0.019443 | 0.096567 | 0.525519 | 114.290 | 0.894072 |

| | track_id | bit_rate | comments | composer | date_created | date_recorded | duration | favorites | genre_top | genres | genres_all | inform |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 135 | 256000 | 1 | NaN | 2008-11-26 01:43:26 | 2008-11-26 00:00:00 | 837 | 0 | Rock | [45, 58] | [58, 12, 45] | |
| 1 | 136 | 256000 | 1 | NaN | 2008-11-26 01:43:35 | 2008-11-26 00:00:00 | 509 | 0 | Rock | [45, 58] | [58, 12, 45] | |
| 2 | 151 | 192000 | 0 | NaN | 2008-11-26 01:44:55 | NaN | 192 | 0 | Rock | [25] | [25, 12] | |

Figure 1: Feature information for song dataset.

Using Cumulative explained variance we could see that about 5 features are required to explain approx. 90% of the variance of data. So, I decided to use Principal Component Analysis (**PCA**). Using this we can achieve dimensionality reduction, noise reduction, and improved visualization. By transforming the original features into a smaller set of uncorrelated components that capture the most variance, PCA helps simplify the dataset, making it easier to analyze and visualize.

Figure 2 : cumulative explained variance for song dataset




The breast cancer dataset is contained in a single csv file. This dataset includes detailed information about various patients, such as their diagnosis (benign or malignant), and a range of medical features like radius, texture, perimeter, area, smoothness, compactness, concavity, and more. Each row in the dataset represents a unique patient, identified by an ID. I performed several steps including dropping irrelevant and corrupted data, normalizing data and others. Using a correlation matrix I identified highly dependent features.

Highly correlated features can introduce multicollinearity into the dataset, which can negatively impact the performance of certain machine learning algorithms. So, I decided to drop features with a correlation coefficient greater than 0.97 to reduce multicollinearity, thereby improving the model's performance and interpretability. The pair plot helps visualize the relationships between the highly correlated features and the target variable.

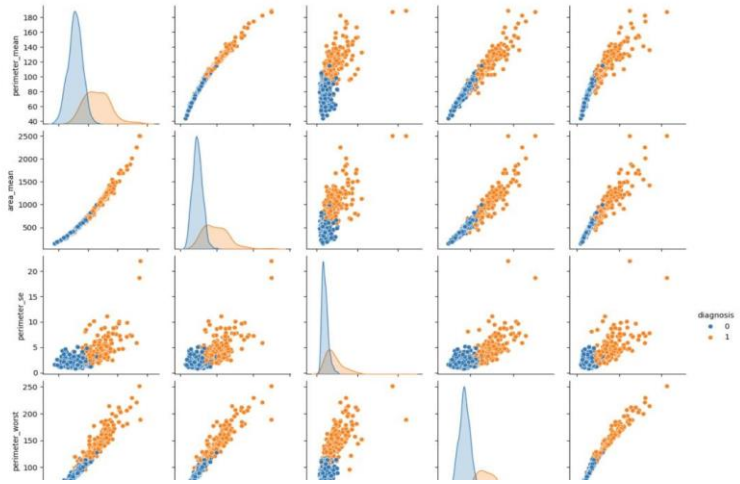



Figure 3: Pear plot for cancer dataset (Only the Features to be dropped > 97% collinearity)

## 1.2 Why are these interesting datasets?

Both datasets are interesting for their practical applications and more. Coming to the song genre classification problem I am a music enthusiast and often wondered on what basis are various songs are categorized into a genre especially in this modern era where there is so much overlap of music styles and ethnic musical instruments being incorporated into electronically created music. With the varying beats per minute (bpm) or tempo we can form the most fundamental block for classification and then getting into further layers of musical chemistry we can identify songs into various classifications.

The breast cancer classification problem is something I believe works towards the progress of humans on a species level. Using statistical data of millions of patients and modern computing power and intelligent algorithms and deep learning techniques like neural network (a concept taken from study of biology) we are giving back to the field of biology and helping fight diseases. There is truly no limit to what constructive struggle of learning can achieve. Be it machine or man.

## 2 Neural Networks

I began by importing the necessary libraries to implement the neural network algorithm. Then, I split the data and assigned it to my training and testing variables (X, y). As we were asked to choose a network with multiple layers and activation functions that can fit the model, I opted for a simple feedforward neural network model with 4 layers. I used activation functions such as 'relu,' 'softmax,' and 'sigmoid.' Next, I compiled the model using the 'adam' optimizer and chose the 'sparse_categorical_crossentropy' loss function because we needed the output labels in a sparse matrix format. During training, the model will process 32 samples at a time before updating the model's weights for a total of 50 epochs. After evaluating the model, I obtained the following results:

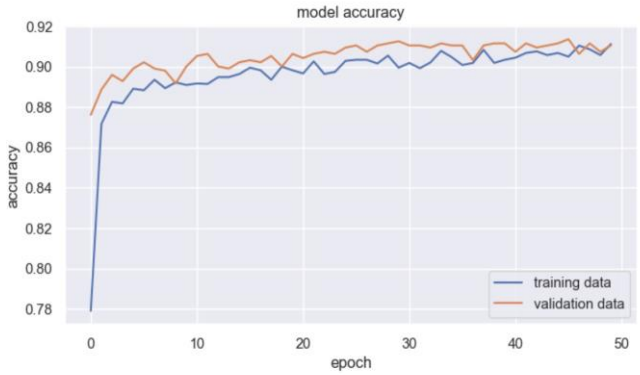



Figure 4: 91% Accuracy for Songs

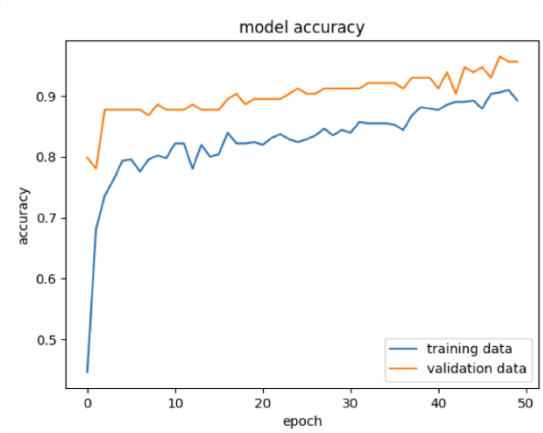



Figure 5: 95% accuracy for cancer as shown in figure below

## 3 Decision Trees

First I implemented the decision tree algorithm from Scikit learn by default with no additional parameters or pruning. After checking the training and testing accuracy I found, the model was overfitting as expected. So I decided to do pruning of decision tree. I decided to go with pre-pruning. To find out the most optimized hyperparameters I used GridsearchCV and in results got the best values for various hyperparameters like max_depth, min_samples_leaf, etc. Then I trained the new model and as a result the training accuracy decreased but the testing accuracy increased.

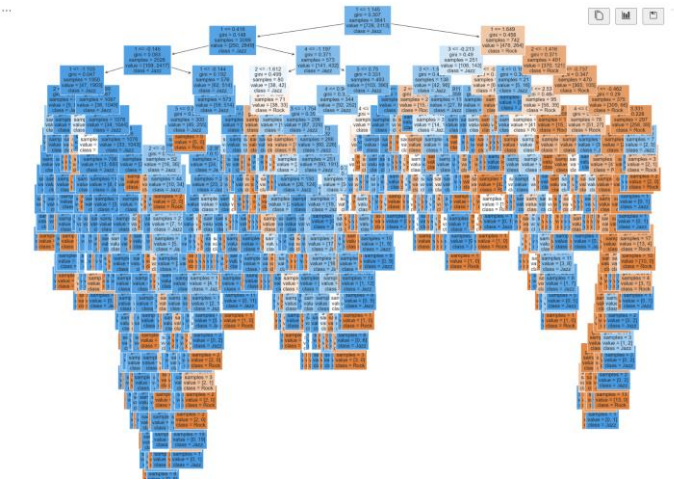


Figure 6: Decision Tree Before pruning

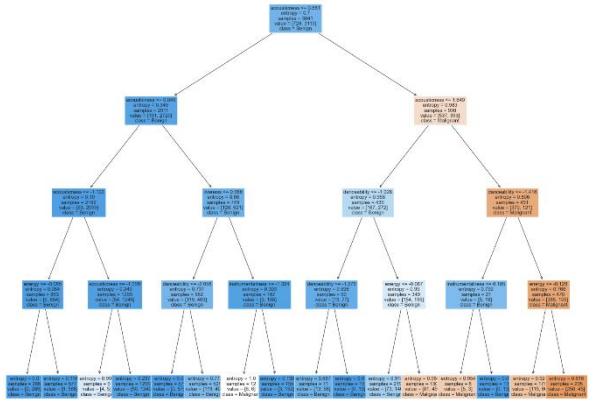


Figure 7: Decision Tree after Pruning

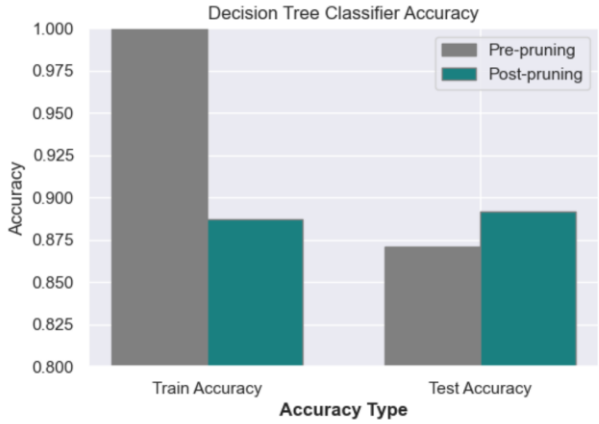



Figure 8: Accuracy comparison of pre pruning vs post pruning

## 4 K-Nearest Neighbour

K-Nearest Neighbour (KNN) is a non-parametric, lazy learning algorithm used in classification and regression by classifying data points based on the classification of their neighbors. For the Cancer dataset, features (X) and target (Y) were separated, and data was standardized using StandardScaler to ensure equal feature contribution in distance calculations. The data was divided into training and testing sets with train_test_split, and a KNN model with n_neighbors=5 was trained. Hyperparameter tuning was performed using GridSearchCV on parameters such as n_neighbors, weights, algorithm, leaf_size, and p. The best model achieved training and testing accuracies, visualized for comparison before and after tuning. Similarly, for the Song dataset, KNN was applied with the same steps, leading to accuracy improvements after hyperparameter tuning.

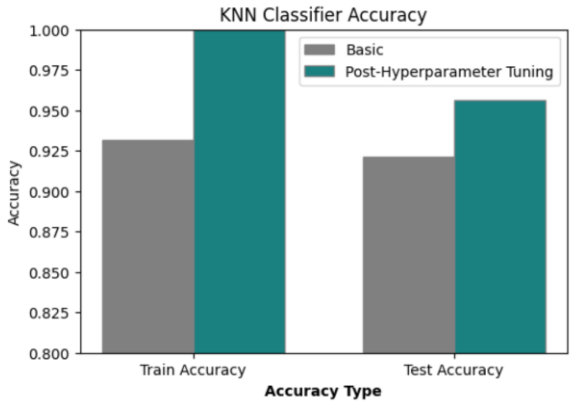



Figure 9: Accuracy of Knn for cancer dataset

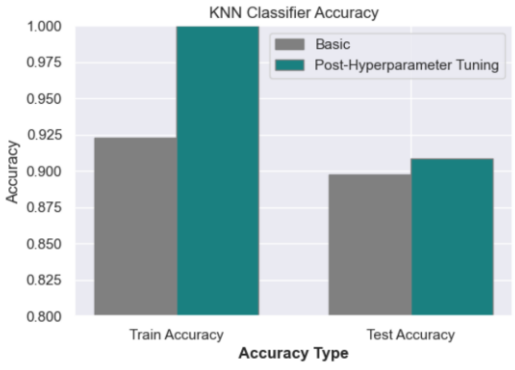



Figure 10: Accuracy of Knn for song dataset

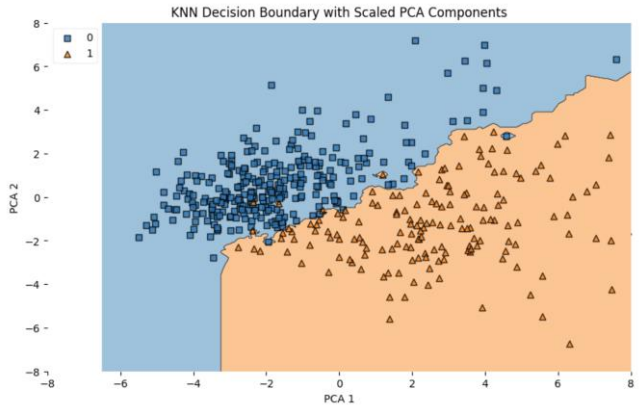



Figure 11: Decision Boundry of Knn for cancer dataset

## 5 Boosting

I used Gradient Boosting, which combines several base estimators to enhance model accuracy. For the Cancer dataset, data preparation involved standardizing features, splitting into training and testing sets, and training a Gradient Boosting model with a base Logistic Regression estimator using parameters n_estimators=1000, learning_rate=0.001, and max_depth=3. Model accuracy was evaluated and visualized with a learning curve, and feature importances were plotted to assess contribution levels.

For the Song dataset, similar steps were followed, using n_estimators=500, learning_rate=0.1, and max_depth=5, achieving high accuracy, with feature importances displayed for better insight.

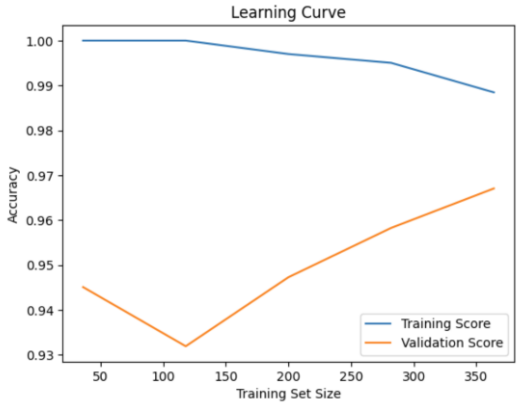



Figure 12: Accuracy of boosting for cancer dataset

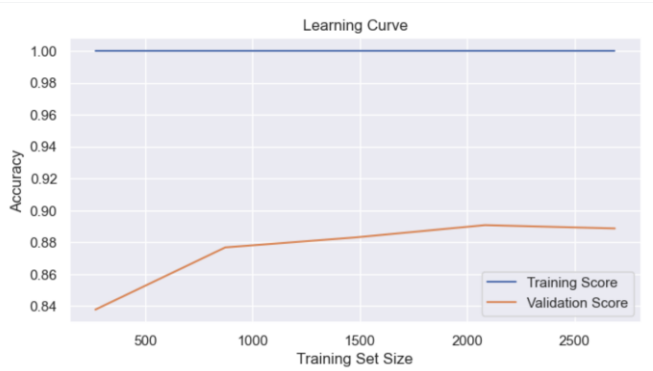



Figure 13: Accuracy of boosting for song dataset

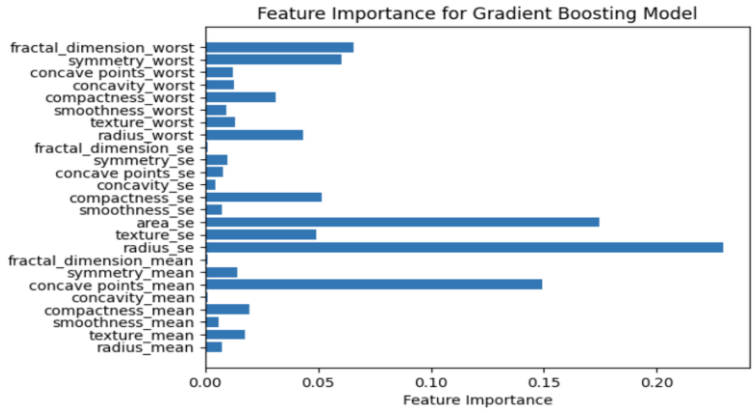



Figure 14: Feature importance of cancer dataset

## 6 Support Vector Machine (SVM)

The **SVM model** was initially implemented using **Scikit-learn** with a **linear kernel** and default parameters. The model achieved a high **training accuracy** of 96.48% and a **test accuracy** of 95.61%, indicating strong generalization. **Feature importance** was analyzed based on the coefficients of the linear SVM, and the **top two features** were selected for further exploration.

The **decision boundary** and **support vectors** were visualized using these features, providing insights into the model's **classification process**. Additionally, the SVM was evaluated with different **kernels** (**linear**, **polynomial**, **RBF**, and **sigmoid**).

In the **Cancer Dataset**, the **linear kernel** performed the best. The model's performance with each kernel was compared, demonstrating the superiority of the linear kernel for this dataset.

In contrast, for the **Song dataset**, the **RBF kernel** outperformed the others, demonstrating its ability to handle more **complex patterns** within the data.

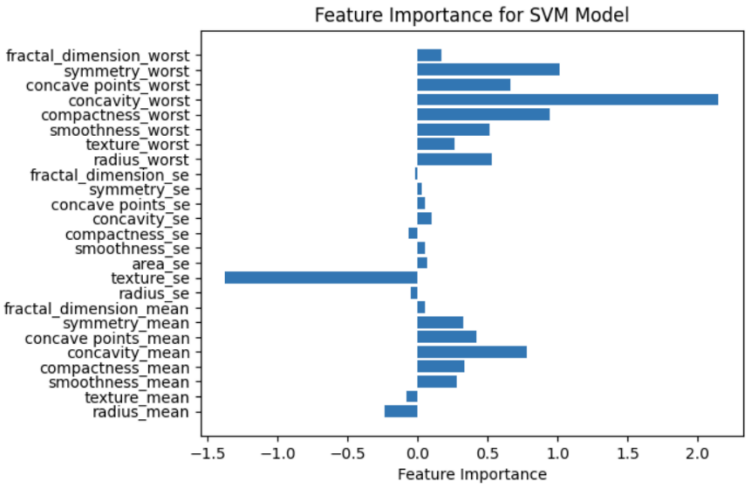



Figure 15: Feature importance of cancer dataset

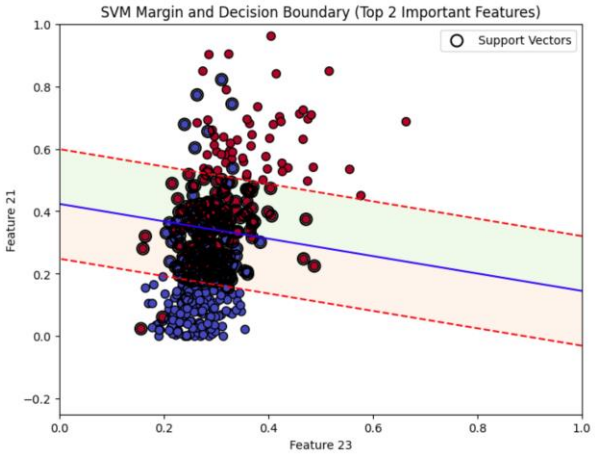



Figure 16: SVM Decision Boundary Cancer Dataset

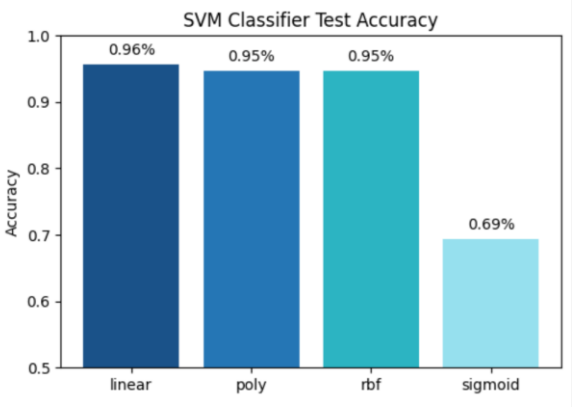



Figure 17: SVM Classifier Accuracy with Different Kernels (Cancer Dataset)

## 7 Conclusions

In this project, I explored the performance of supervised learning algorithms on two distinct datasets: one focused on **breast cancer classification** and the other on **song genre classification.** I used five different machine learning algorithms **Artificial Neural Networks (ANNs)**, **Decision Trees, K-NN, GradientBoosting** and **SVM**

The Decision Tree initially resulted in overfitting. To address this, I performed **pre-pruning** with hyperparameter optimization using **GridSearchCV**. This improved the model's generalization by reducing the training accuracy but increasing the test accuracy.

The comparison between **pruned decision trees, knn, SVM, boosting** and **ANNs** showed that while all algorithms performed quite closely, **ANNs provided higher accuracy** overall, particularly for the song dataset.

For the more complex Cancer Dataset which also has more collinearity we can see that **Boosting** algorithm provided much better results and peaked accuracy of **97.37%.**

Also, an interesting observation that for the cancer dataset KNN and SVM have exactly same accuracy. Furthermore, neural networks and boosting have exactly same accuracies. The similar accuracies could result from optimal hyperparameter tuning, dataset characteristics, and the models **converging to similar decision boundaries** due to capturing the same underlying **data patterns**.

This analysis highlights the importance of **feature engineering**, **model selection**, and **hyperparameter tuning** in building effective machine learning models. The project also demonstrates how **neural networks** excel in capturing non-linear relationships in the data, while **decision trees**, with appropriate pruning, offer interpretable models suitable for scenarios where simplicity and interpretability are prioritized.

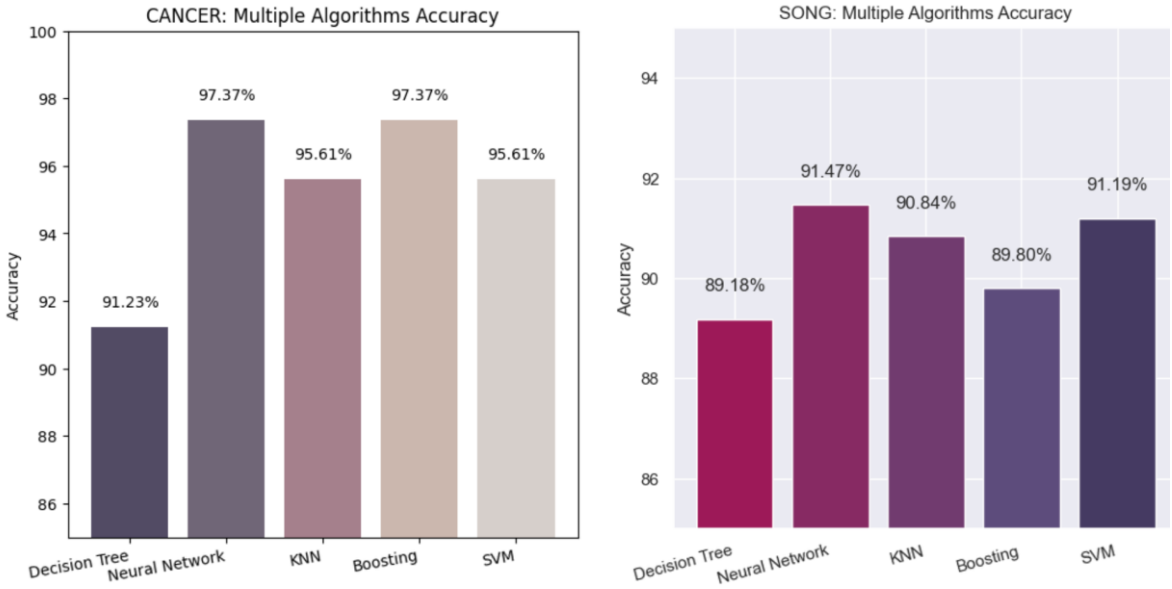



Figure 15 & 16 : Comparing the accuracy of various algorithms for **Cancer** and **Song** datasets.

## 8 References

[1] Kaggle Dataset: https://www.kaggle.com/datasets

[2] Sklearn Libraries

[3] YouTube videos, Stack Overflow, and Google