SMS Spam Detection for Connect5G Networks

Revolution Consulting’s Machine Learning Solution

Presented by:
Michael Teixeira s4133975

02/10/2024

Introduction to Connect5G's Challenge

Overview of Connect5G Networks:

- Operates in Australia, Singapore, and the UK; known for premium customer experience.

Business Problem:

- Growing customer complaints about spam messages; leading to customer churn.

Need for a Solution:

- Connect5G needs an automatic, accurate spam detection service to retain customers.

Objective of the Project

Key Objective:

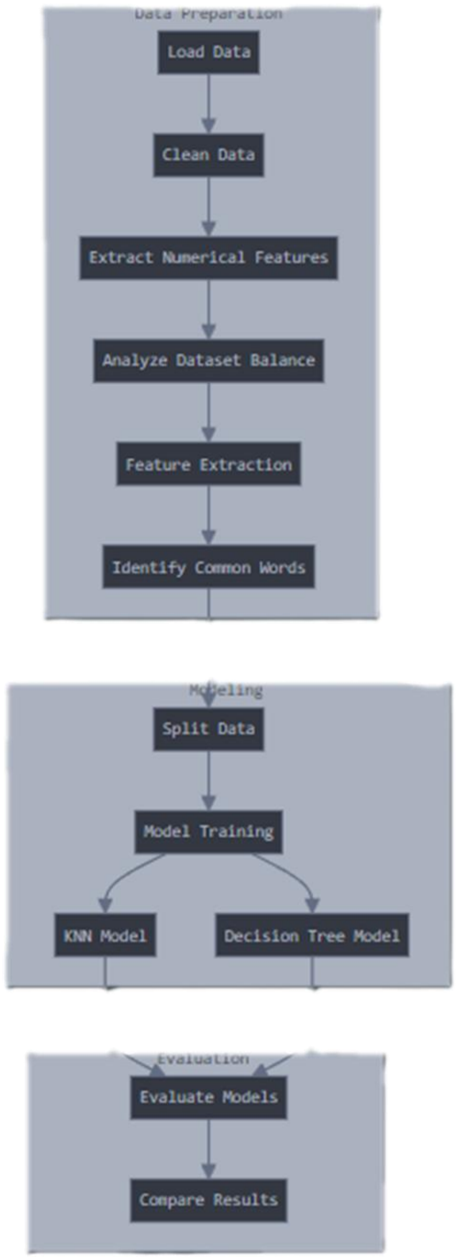
- Build and evaluate machine learning models to classify SMS as **spam or ham**.

Client's Requirements:

- Emphasize importance of the the average prediction time per sample, for real-time classification.

Deliverables:

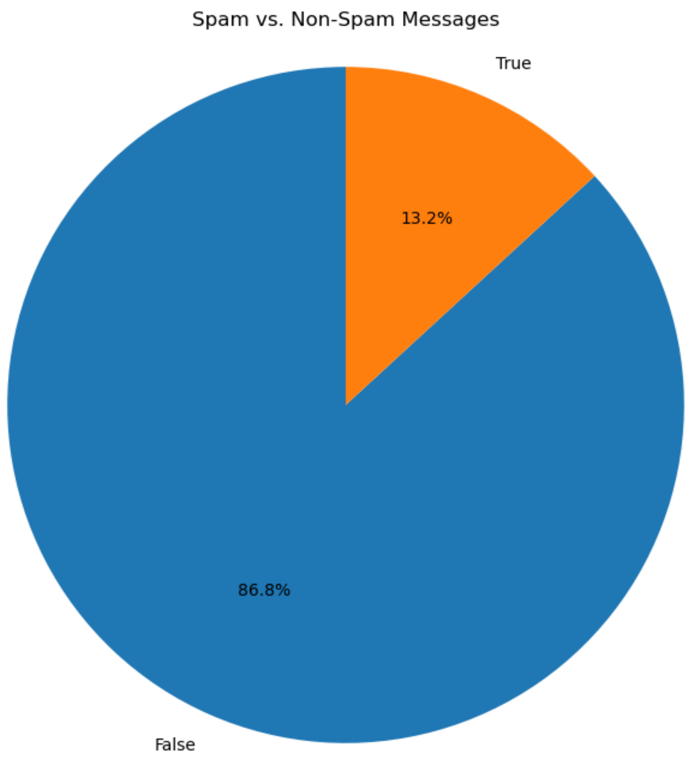
- Train and compare **two machine learning models**: K-Nearest Neighbors (KNN) and Decision Tree.



Data Overview

Dataset Composition

- **Total messages:** 5,351
- **Spam messages:** 704 (13.2%)
- **Ham messages:** 4,647 (86.8%)



Is the Dataset Balanced?

- The dataset shows a **significant imbalance** between spam and ham messages, with ham messages being the majority class.

Solution:

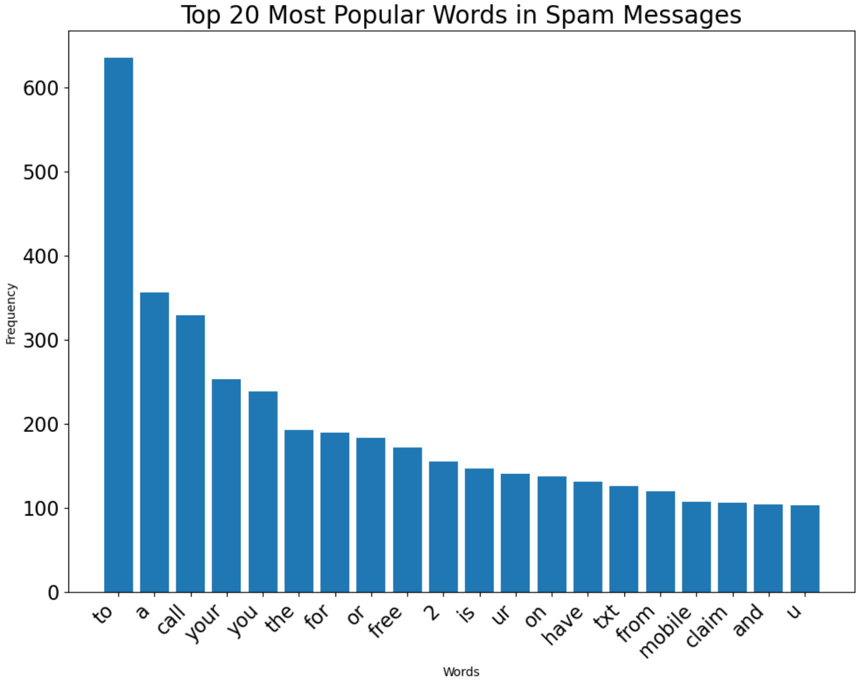
- Applied **SMOTE** (Synthetic Minority Over-sampling Technique).

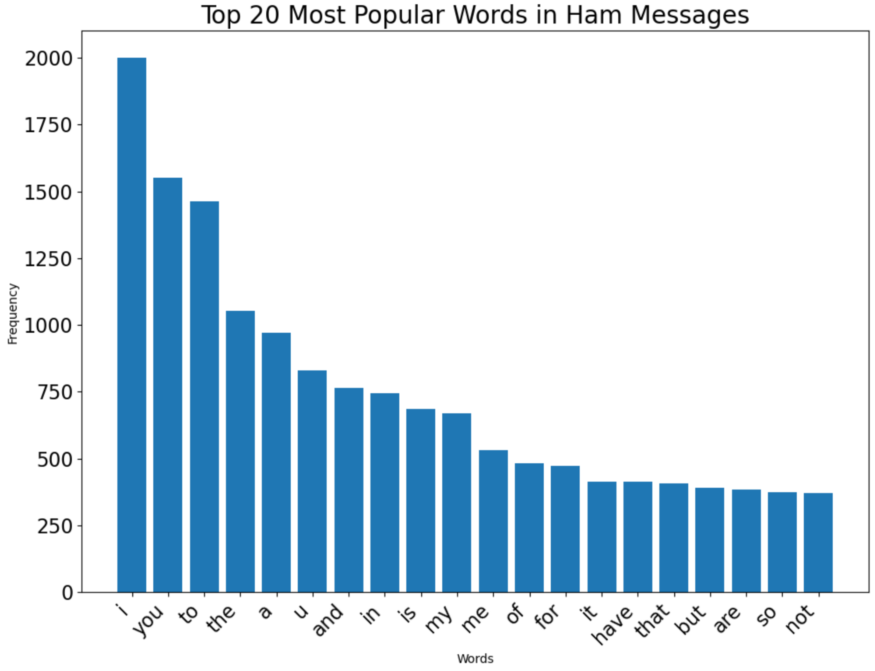
Preprocessing

Data Preprocessing:

Used **Count Vectorizer** to convert text messages into numerical features.

- Lowercasing.
- Tokenization.
- Removal of stopwords.





Model Training and Hyperparameter Tuning

Model Selection:

- **K-Nearest Neighbors (KNN)** and **Decision Tree.**

Hyperparameter Tuning:

- Used **GridSearchCV** for both models to optimize parameters:

Model	Parameter	Values
KNN	n_neighbors	1, 3, 5, 9, 11
	p	1, 2
Decision Tree	min_samples_split	2, 3, 5
	min_samples_leaf	5, 10, 20, 50, 100
	max_depth	2, 3, 5, 10, 20

Handling Imbalanced Data:

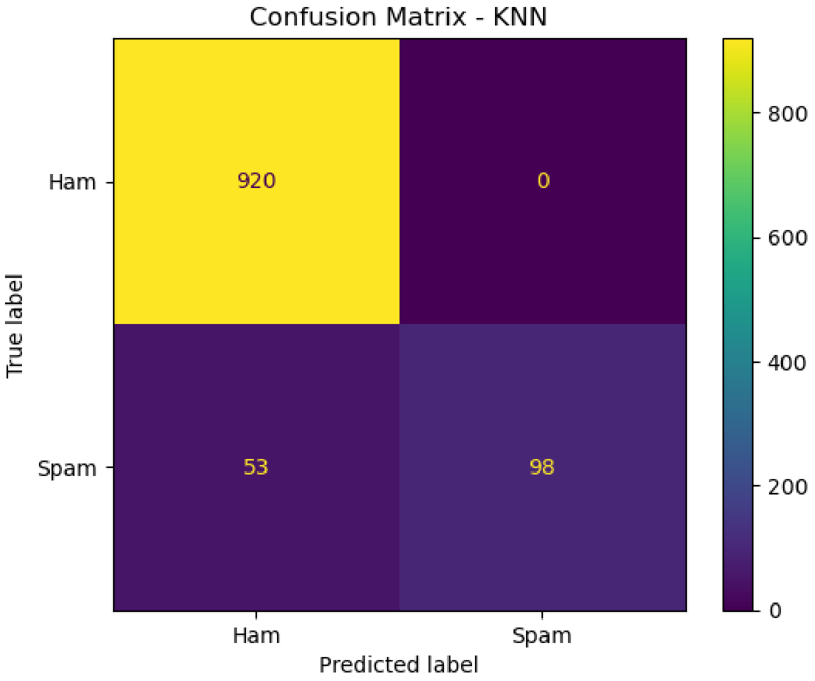
- Trained models with and without SMOTE to compare the effects on performance.

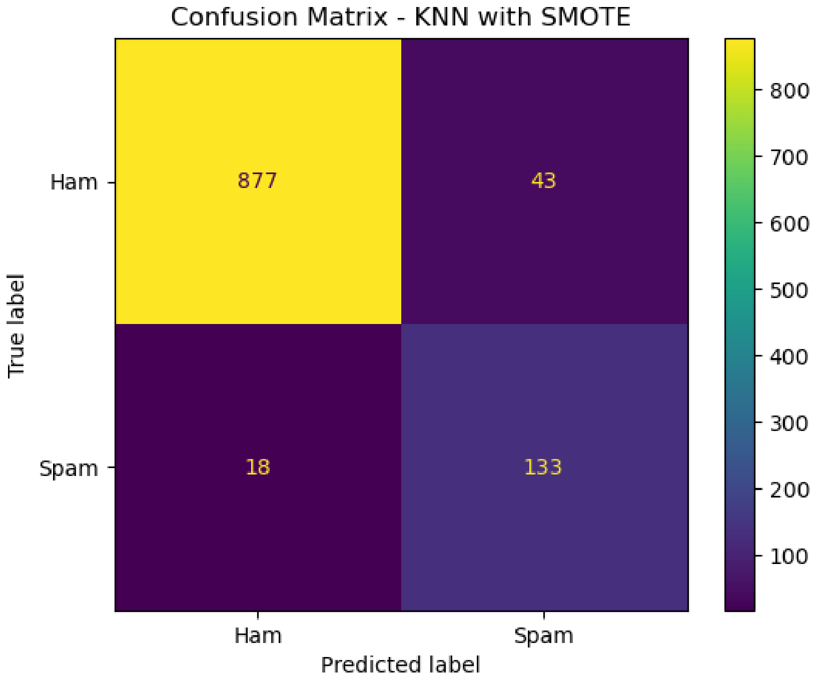
Confusion Matrix

Metrics for Evaluation:

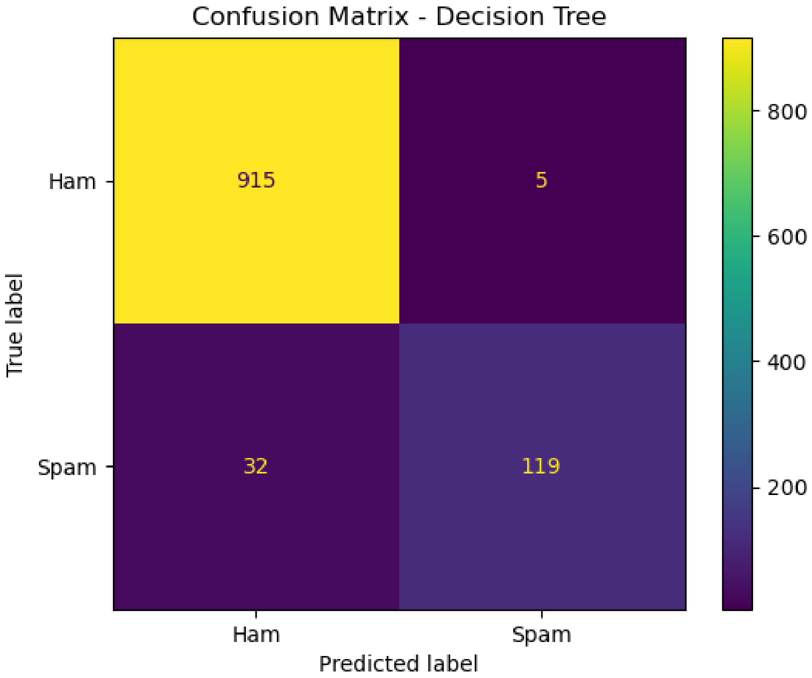
- **Accuracy:** Percentage of correctly predicted instances.
- **Balanced Accuracy:** Accounts for class imbalance in evaluation.
- **Training Time:** Time taken to fit the model.
- **Prediction Time:** Average time taken for model predictions.

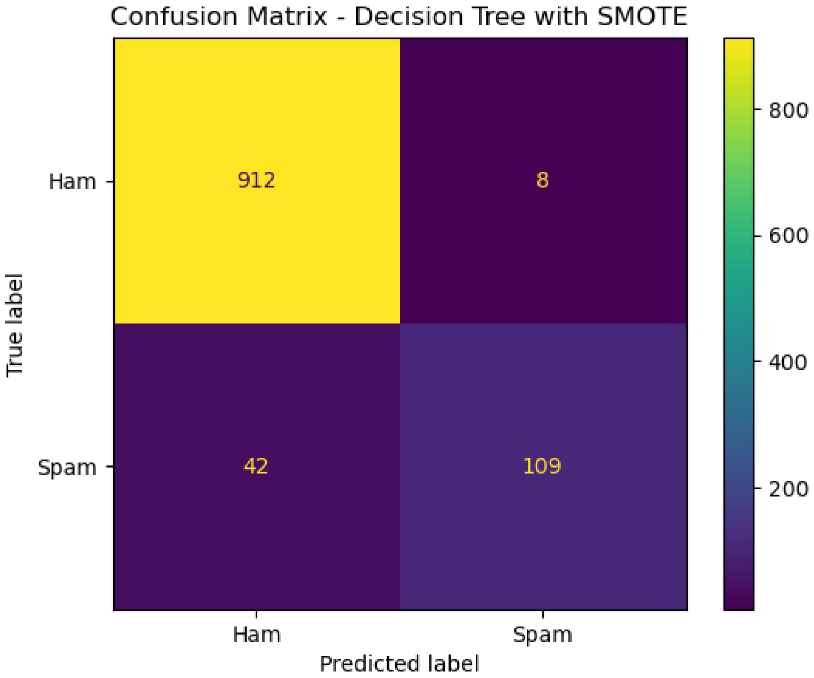
KNN:





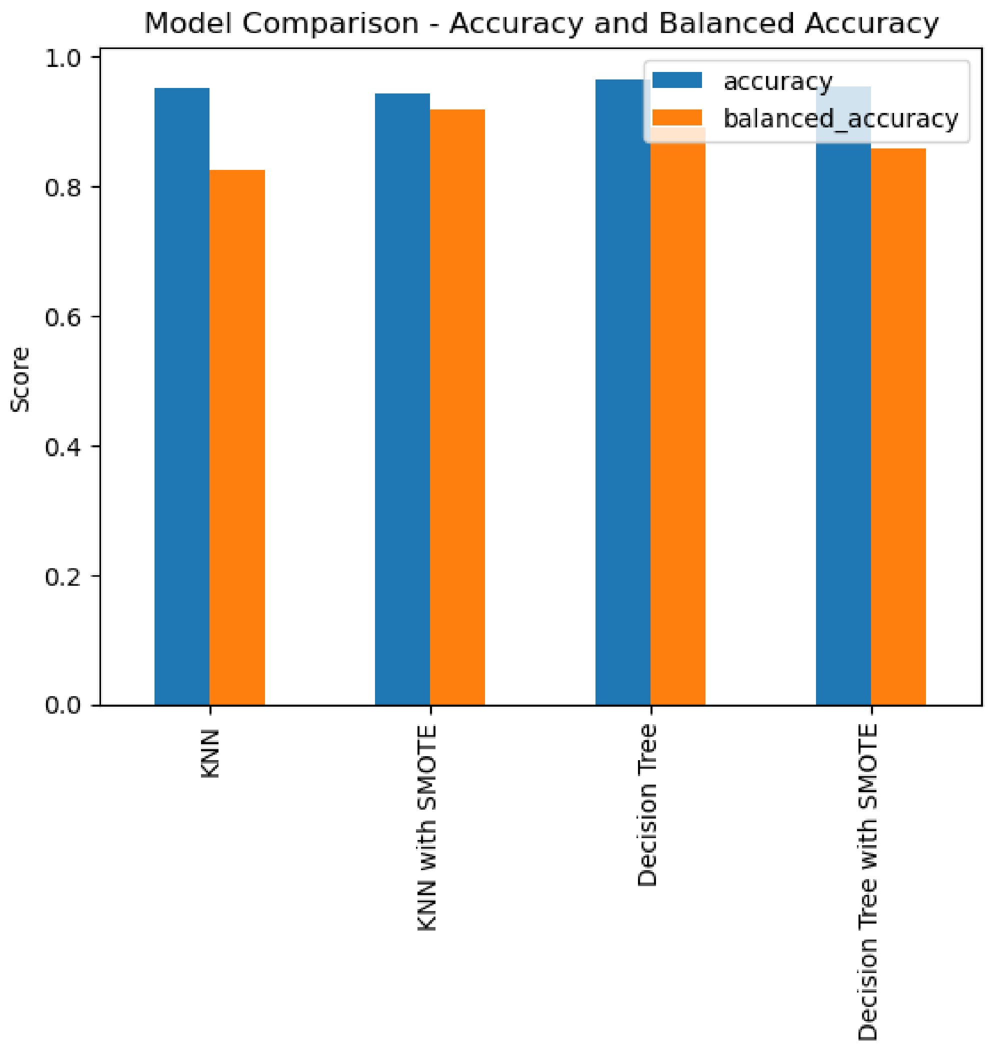
Decision Tree

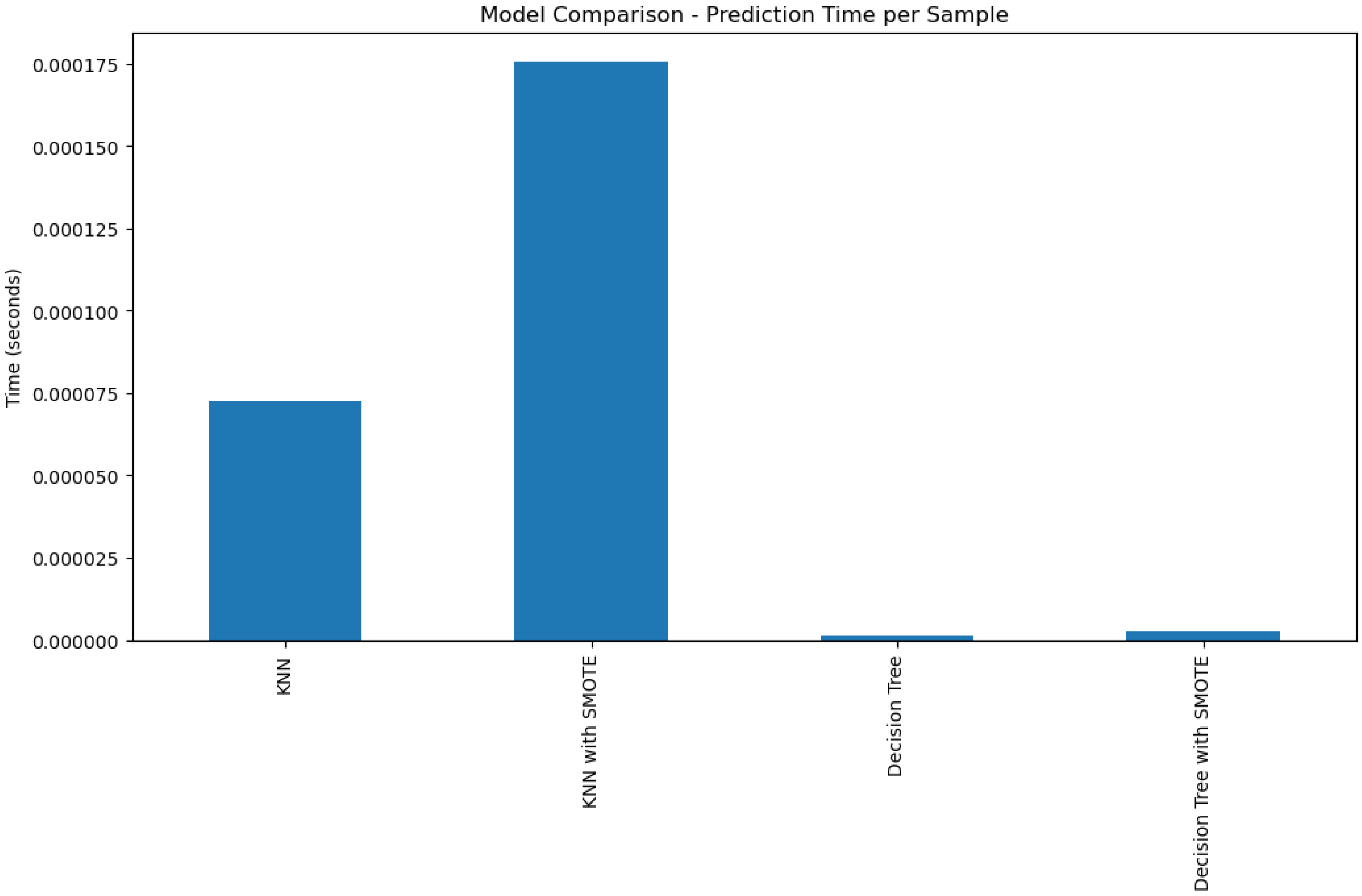




Model	Accuracy	Balanced Accuracy	Training Time	Prediction Time
KNN	0.95	0.824	0.001	6.22e-05
KNN with SMOTE	0.943	0.917	0.001	1.76e-04
Decision Tree	0.966	0.894	0.045	9.22e-07
Decision Tree with SMOTE	0.95	0.852	0.049	1.05e-06

- **Decision Tree** showed higher accuracy, and excelled in faster prediction times.
- Training time is slightly longer, but this doesn't significantly impact model performance.


Model Comparison - Accuracy and Balanced Accuracy
accuracy
balanced_accuracy
Score
1.0
0.8
0.6
0.4
0.2
0.0
KNN
KNN with SMOTE
Decision Tree
Decision Tree with SMOTE


Model Comparison - Prediction Time per Sample
Time (seconds)
0.000000
0.000025
0.000050
0.000075
0.000100
0.000125
0.000150
0.000175
KNN
KNN with SMOTE
Decision Tree
Decision Tree with SMOTE

Conclusion & Recommendation

- **Decision Tree with SMOTE**

After applying SMOTE, the model introduces more errors compared to the basic Decision Tree.

However, considering real-time performance and the imbalanced nature of spam detection, **Decision Tree with SMOTE** is better suited for production because it provides a more balanced approach to detecting both spam and ham in dynamic environments.

Thank You!