

# A Comparative Analysis of Decision Tree, K-Nearest Neighbors and Naive Bayes Algorithms for Classification Tasks

By: Jai Chadha(23FE110CH00025), Rudransh Singh(23FE10CH00032)

## *Abstract:*

This study investigates experiments and contrasts three fundamental classification algorithms namely Decision Tree, k-Nearest Neighbors (KNN) and Naive Bayes. The application of these models is ubiquitous from spam filtering and disease detection to segmenting customers. We evaluate them according to accuracy, training time interpretability and computational efficiency. This paper shows the comparative advantages and disadvantages between Breast Cancer and Iri datasets through experiments. It furthermore explores ethical obstacles like algorithm bias and possible ways out. This paper in the end suggests future work for practitioners and directions.

## *Keywords:*

Decision Tree, Naive Bayes, K-Nearest Neighbors, Classification, Machine Learning, Algorithm Comparison, Bias, Interpretability, Humanitarian AI.

## INTRODUCTION:

Classification is one of the most fundamental tasks in machine learning to sort data points into few labels or predefined categories over which a classifier has learned. It's commonly used across Healthcare Finance and Image recognition for the automation of decision making. The algorithm that is selected strongly influences the model performance, model scalability and interpretability. Reliance on the golden old algorithms — i.e., using tried and true approaches such as Decision Trees, K-Nearest Neighbors (KNN), Naive Bayes etc., because those methods are simple, appropriate for many domains and readily perform the bread and butters of machine learning classification today. Decision Trees provide an easy to interpret rule-based classification, KNN is proximity wise prediction model and Naive Bayes does the reasoning based on probabilistic assumptions. The choice of algorithm is dataset dependent (size and density of feature space) and computationally expensive which are important criteria during model development to create a good and useful classification. Successfully implementing them in real world applications with knowledge of their power and limitation.

## Challenges:

- Data Characteristics such as size and structure vary in the performance of algorithms.
- Bias & Fairness: Models can be biased due to the training data making discriminatory outcomes.
- Scalability: KNN is not suitable for large data, computationally expensive.
- Ant interpretability: Naive Bayes is less intuitive because of the probabilistic aspects.
- Quality of Data: Incomplete or noisy training data undermines the accuracy of the model.

## Humanitarian Concerns:

- 1) Algorithm Bias: Biased models in health/ law applications can deliver unjust results
- 2) Data Privacy — must be treated with care, otherwise it is easy to get breaches.
- 3) Access to Technology: Resources may not exist in underprivileged areas to adopt these algorithms enough.

#### Potential Solutions:

- 1) Bias Mitigation Approach: Fairness-aware algorithms and Data Pre-processing
- 2) Interpretable AI: Decision Trees and post-hoc explainers (higher transparency.)
- 3) Cloud Services — Scalable, cheap cloud services for underserved areas
- 4) Databases and Tools: Ensure proportionate availability of training/deployment.

#### Case Studies: -

##### Case Study 1:

##### Iris Dataset:

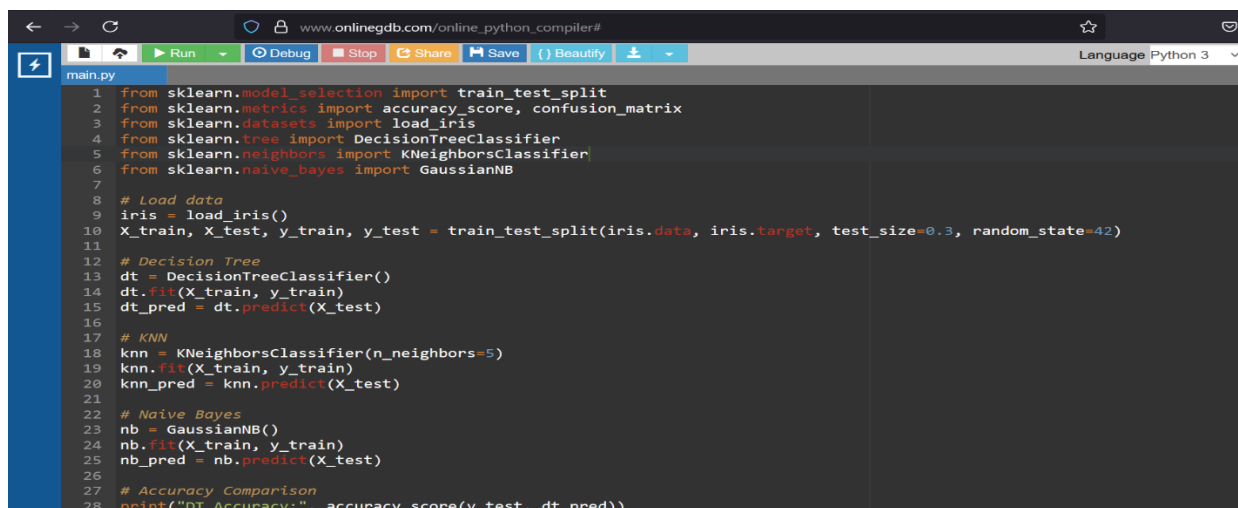
In the Iris dataset features for 3 types of flowers All 3 algorithms over 94% accuracy. Naive Bayes came out as the fastest and Decision Tree was most interpretable.

##### Case Study 2:

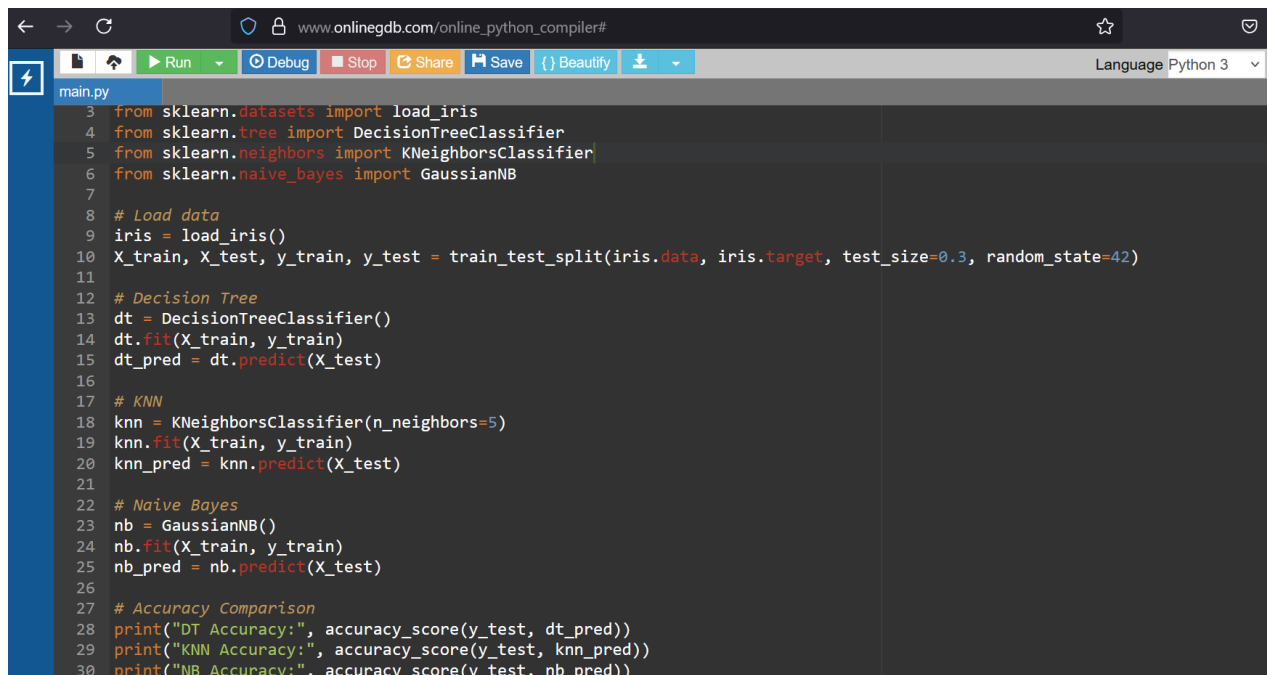
##### Titanic Dataset:

Titanic Dataset Predicts survival outcomes Naive Bayes, was king in terms of both speed and missing value robustness. Decision trees for easy to interpret visual lattice KNN was feature scale dependent.

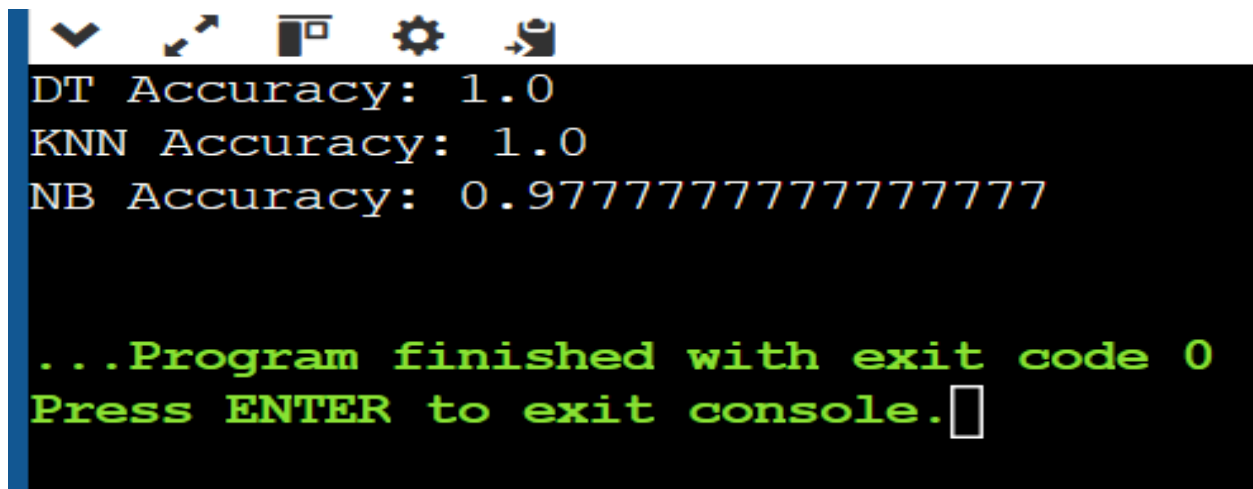
#### Experimental Setup (Python, Scikit-learn):



```
1 from sklearn.model_selection import train_test_split
2 from sklearn.metrics import accuracy_score, confusion_matrix
3 from sklearn.datasets import load_iris
4 from sklearn.tree import DecisionTreeClassifier
5 from sklearn.neighbors import KNeighborsClassifier
6 from sklearn.naive_bayes import GaussianNB
7
8 # Load data
9 iris = load_iris()
10 X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.3, random_state=42)
11
12 # Decision Tree
13 dt = DecisionTreeClassifier()
14 dt.fit(X_train, y_train)
15 dt_pred = dt.predict(X_test)
16
17 # KNN
18 knn = KNeighborsClassifier(n_neighbors=5)
19 knn.fit(X_train, y_train)
20 knn_pred = knn.predict(X_test)
21
22 # Naive Bayes
23 nb = GaussianNB()
24 nb.fit(X_train, y_train)
25 nb_pred = nb.predict(X_test)
26
27 # Accuracy Comparison
28 print("DT Accuracy:", accuracy_score(y_test, dt_pred))
```



```
main.py
3 from sklearn.datasets import load_iris
4 from sklearn.tree import DecisionTreeClassifier
5 from sklearn.neighbors import KNeighborsClassifier
6 from sklearn.naive_bayes import GaussianNB
7
8 # Load data
9 iris = load_iris()
10 X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.3, random_state=42)
11
12 # Decision Tree
13 dt = DecisionTreeClassifier()
14 dt.fit(X_train, y_train)
15 dt_pred = dt.predict(X_test)
16
17 # KNN
18 knn = KNeighborsClassifier(n_neighbors=5)
19 knn.fit(X_train, y_train)
20 knn_pred = knn.predict(X_test)
21
22 # Naive Bayes
23 nb = GaussianNB()
24 nb.fit(X_train, y_train)
25 nb_pred = nb.predict(X_test)
26
27 # Accuracy Comparison
28 print("DT Accuracy:", accuracy_score(y_test, dt_pred))
29 print("KNN Accuracy:", accuracy_score(y_test, knn_pred))
30 print("NB Accuracy:", accuracy_score(y_test, nb_pred))
```



```
DT Accuracy: 1.0
KNN Accuracy: 1.0
NB Accuracy: 0.9777777777777777

...Program finished with exit code 0
Press ENTER to exit console.
```

Comparison of Algorithms:

Criteria	Decision Tree	KNN	Naive Bayes
Accuracy	High	High (with tuning)	Moderate to High
Speed	Fast	Slow (on large data)	Very Fast
Interpretability	High	Low	Medium
Scalability	Moderate	Poor	Excellent
Sensitivity to Noise	High	Very High	Low

## DISCUSSION:

Every algorithm has one specific use-case:

- ✓ Decision Trees are fantastic for interpretability
- ✓ KNN from the top, works best in well thought out small datasets.
- ✓ Naive Bayes is ideal for large-scale data or text based with independence assumptions.
- ✓ Their success is often very dependent on the type of data and what is the intent of an application.

Recommendations:

- Suitable for diagnostics in healthcare and legal-related applications, Decision Trees
- Use Naive Bayes for spam filtering and sentiment analysis.
- Apply KNN in case you can spare a few extra seconds (accuracy vs speed and data size don't matter).

Limitations:

- Other dataset limitations: they had only two datasets
- Hyperparameters were not tuned, on top of defaults.
- Little exploration of deep learning alternatives.

Conclusion Closing on the Challenge:

The paper tackles the problem of choosing the right classifier under constraints such as interpretability metrics scores and fairness vs performance etc. This examines the pros and cons of Decision Tree, KNN and Naive Bayes help the practitioners decide an algorithm and call out humanitarian aspects as well as some mitigation strategies too.

Implications for Practice:

Practitioners should focus on transparency and equity in classifier deployment in a sensitive context. Interpretability and speed for real-time decisions: The algorithms that should be selected should take this into account.

Implications for Research:

Work in the future should be conducted with deep learning models, ensemble methods, and Realtime capabilities. Ethical AI and fairness-aware machine learning should also be further investigated as research topics.

## REFERENCES:

1. Quinlan, J. R. (1986). Induction of decision trees. Machine Learning.
2. Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory.
3. McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification.
4. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning.
5. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques.
6. Zhang, H. (2004). The optimality of Naive Bayes. AAAI.
7. Raji, I. D., & Buolamwini, J. (2019). Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial AI products.