Report on classification models (Decision Tree & Naive Bayes)

- Riya Huddar (MDS202431)
- Shruti Sharma (MDS202435)

In this report, we compare the performance of two popular classification models: **Decision Tree** and **Naive Bayes**. We evaluate their performance on two distinct datasets: the **Bank Marketing** dataset and the **Movies** dataset. Both models are applied to predict binary outcomes (whether a person subscribes to a bank product or whether a movie is a hit or flop) based on multiple features.

The goal of this analysis is to determine which model performs better in terms of key metrics like **accuracy**, **precision**, **recall**, and **F1-score**, and to understand the strengths and weaknesses of each model in different scenarios.

Libraries Used:

1. **pandas**: A data manipulation library used for reading, cleaning, and processing datasets in tabular form (e.g., CSV files).
2. **Sklearn.model_selection**:
    - **train_test_split**: Splits the data into training and testing subsets to evaluate model performance.
    - **GridSearchCV**: Performs hyperparameter tuning by exhaustively searching over a specified parameter grid using cross-validation.
3. **sklearn.naive_bayes** : Implements the Naive Bayes classification algorithm, specifically for Gaussian-distributed data
4. **Sklearn.metrics**:
    - **accuracy_score**: Computes the proportion of correct predictions.
    - **precision_score**: Measures the proportion of true positives among predicted positives.
    - **recall_score**: Measures the proportion of true positives among actual positives.
    - **f1_score**: Harmonic mean of precision and recall, balancing both metrics.
5. **roc_auc_score**: Computes the area under the ROC curve, indicating the model's ability to distinguish between classes.
6. **classification_report**: Provides a summary of precision, recall, F1-score, and support for each class.
7. **confusion_matrix**: Generates a matrix showing true vs. predicted class values.
8. **matplotlib.pyplot** : A plotting library used to create static, interactive, and animated visualizations.

9. **seaborn** : A statistical data visualization library built on matplotlib, used for making more attractive and informative plots, like heatmaps and pairplots.
10. **Sklearn.tree**(DecisionTreeClassifier): DecisionTreeClassifier from sklearn.tree is a classification algorithm that builds a tree-like model by splitting data based on feature values. It handles both numerical and categorical data and is interpretable.

The "Bank Marketing Data Set" from the UCI Machine Learning Repository is related with direct marketing campaigns (phone calls) of a Portuguese banking institution.The classification goal is to predict if the client will subscribe a term deposit (variable y).

Task 1: Bank marketing dataset

Model Comparison Table

Metric	Decision Tree	Naïve Bayes
Accuracy	0.84	0.87
Precision	0.37	0.43
Recall	0.58	0.44
F1-score	0.45	0.43

Interpretation

- **Accuracy**: Naïve Bayes performed slightly better (**0.87 vs. 0.84**).
- **Precision**: Naïve Bayes had higher precision (**0.43 vs. 0.37**), meaning it made fewer false positive predictions.
- **Recall**: Decision Tree had a better recall (**0.58 vs. 0.44**), meaning it made fewer false negative predictions.
- **F1-score**: Decision Tree performed slightly better (**0.45 vs. 0.43**), indicating a better balance between precision and recall.
- **Key Takeaway**: If the goal is to **reduce false negatives** (i.e., capture more potential subscribers), Decision Tree is preferable. If the goal is to **reduce false positives**, Naïve Bayes is a better choice.

The movie database consists of 1698 Hindi Movies from 2005-2017.
Source: https://www.kaggle.com/datasets/rishidamarla/bollywood-movies-dataset.A movie is a hit if ,revenue > budget, and it is a flop otherwise. The goal is to predict whether a movie will be a hit or flop, given all the other attributes.

Task 2: Movies dataset

Model Comparison Table

Metric	Decision Tree	Naïve Bayes
Accuracy	0.90	0.85
Precision	0.97	0.86
Recall	0.89	0.96
F1-score	0.93	0.91

Interpretation

- **Accuracy**: Decision Tree performed better (**0.90 vs. 0.85**).
- **Precision**: Decision Tree had a much higher precision (**0.97 vs. 0.86**), meaning it was more confident in predicting hits.
- **Recall**: Naïve Bayes had higher recall (**0.96 vs. 0.89**), meaning it was better at identifying all hit movies but at the cost of more false positives.
- **F1-score**: Decision Tree had a slightly better F1-score (**0.93 vs. 0.91**), indicating a better balance.
- **Key Takeaway**: If we want a **high-confidence prediction** of hit movies (minimizing false hits), Decision Tree is better. If we **don't want to miss any hits**, Naïve Bayes is preferable.

Conclusion: We have compared the performance of the **Decision Tree** and **Naïve Bayes** classifiers on two different datasets: the **Bank Marketing dataset** and the **Movies dataset**. While our initial results show differences in metrics like **accuracy**, **precision**, and **recall**, it's important to note that these outcomes can change significantly with careful **hyperparameter tuning**. Additionally, model performance is heavily influenced by the **features selected** and the **type of data**.

For example, **Naive Bayes** tends to perform well when the features are independent and not highly correlated. In cases where features are strongly correlated, **Naive Bayes** may not perform as well, as it assumes independence among the features. On the other hand, **Decision**

Trees can handle correlated features better and often perform well with more complex relationships between features.

Thus, the choice of model and its performance will also depend on the nature of the data and the feature engineering process. Therefore, tuning the models, selecting the right features, and considering the type of data are critical steps in optimizing classifier performance.