REPORT

Credora Internship – Data Science

WEEK 3 -Task 03

Decision Tree Classifier – Customer Purchase Prediction

Submitted by: AKANKSHA BHOSLE

DATE : 02-06-2025

Introduction:

This project focuses on predicting whether a customer will subscribe to a term deposit using the Bank Marketing dataset. A Decision Tree Classifier was used to analyze client features and predict outcomes based on behavior and demographics.

Dataset Overview:

- **Dataset:**
  deposit Bank Marketing Dataset from [UCI]
- **Goal:**
  Predict if a customer subscribes to a term (y)
- **Files used:**
  bank.csv – 10% sample
  bank-full.csv – full dataset
  bank-names.txt – column info
- No missing values
- Includes both numerical and categorical features
- Key features: age, job, balance, duration, contact, poutcome
- **Target:** y (yes/no – subscription)

Tools & Libraries:

- **Python**: Core programming language used for all data analysis and modeling tasks.

- **Pandas**:
    - For loading and exploring the dataset
    - Used to clean and manipulate data with DataFrames

- **NumPy**:
    - Used for numerical operations
    - Supports arrays and efficient math behind the scenes

- **Scikit-learn (sklearn)**:
    - Used to build and train the DecisionTreeClassifier
    - Provided tools for data splitting, encoding, and evaluation
    -

- **Matplotlib**:
    - Used to plot the decision tree structure
    - Helpful for basic visualizations

- **Seaborn**:
    - Used to create a heatmap of the confusion matrix
    - Makes statistical plots more visually appealing

- **Google Colab**:
    - Cloud platform used to run and share notebooks without local setup
    - Allows easy access to Python and libraries in the browser

Data Preprocessing:

- **Loaded**
  the dataset bank.csv using pandas. read_csv () with semicolon (;) as a separator.

- **Checked for missing values**
  Using isnull().sum()
  → No missing values were found in the dataset.

- **Identified categorical features**
  (like job, education, contact, etc.).

- **Encoded categorical columns**
  using Label Encoder for sklearn. preprocessing
  → Converted text labels into numeric format for modeling
  .

- **Separated features and target**
  - X → All columns except y
  - y → Target column (yes/no for term deposit)

- **Split the dataset**
  into training and testing sets using train_test_split()
  - 80% for training
  - 20% for testing

- **Final dataset was ready for building the Decision Tree model.**

Model Training :

- **Used**
  DecisionTreeClassifier from sklearn.tree to build the model.
- **Set**
  max_depth=**5** to prevent the tree from growing too deep and overfitting on the training data.

- **Trained the model**
  using .fit(X_train, y_train)
  → This allowed the model to learn patterns from the training dataset.

- The Decision Tree algorithm **automatically selected the most important features** (e.g., duration, poutcome, month) to make splits.

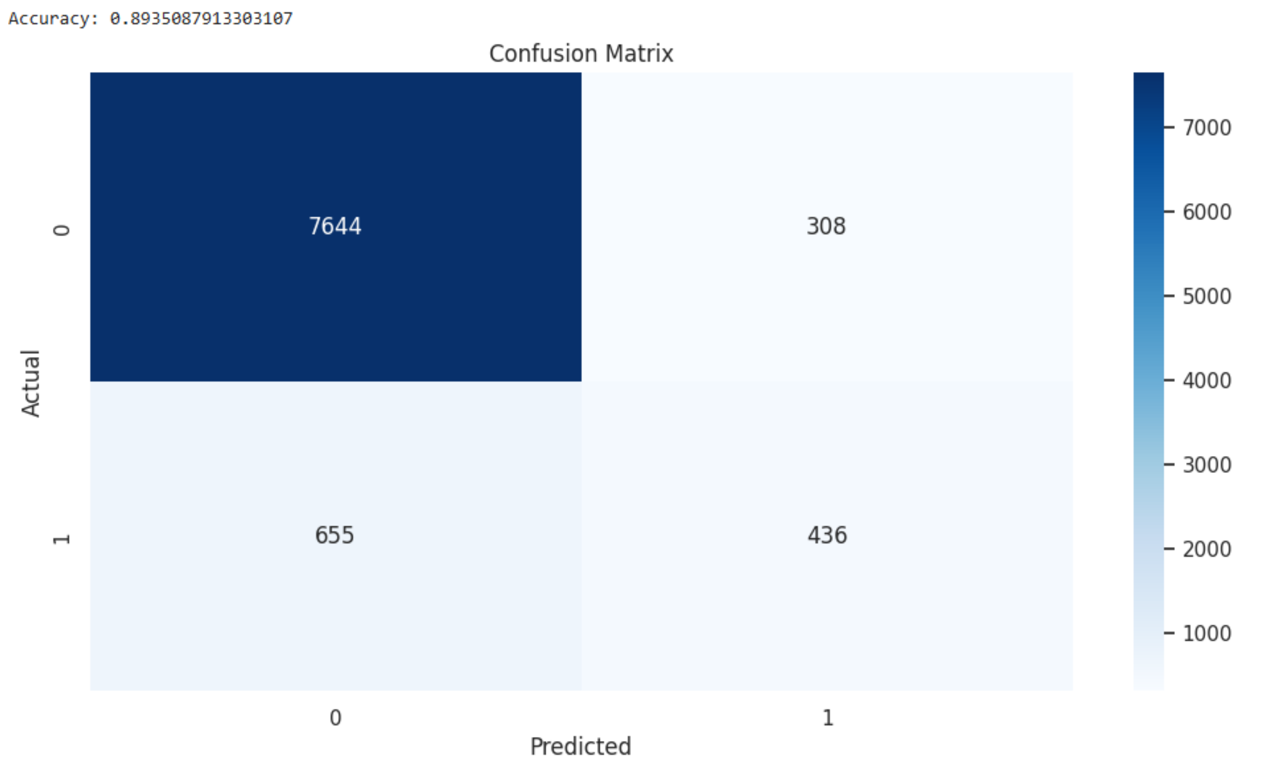
- The model uses a **tree structure** where internal nodes represent conditions and leaf nodes represent final predictions (yes/no for term deposit).

- Model training was **fast and interpretable**, making it suitable for business use cases.

Evaluation Metrics:

1.Accuracy score Measures the overall correctness of the model. It is the ratio of correctly predicted instances to the total instances.

2.Confusion matrix A table showing True Positives, True Negatives, False Positives, and False Negatives, helping visualize model errors.



3.Classification report (precision, recall, F1-score)
Includes:

- **Precision:** How many selected items are relevant.
- **Recall:** How many relevant items are selected.

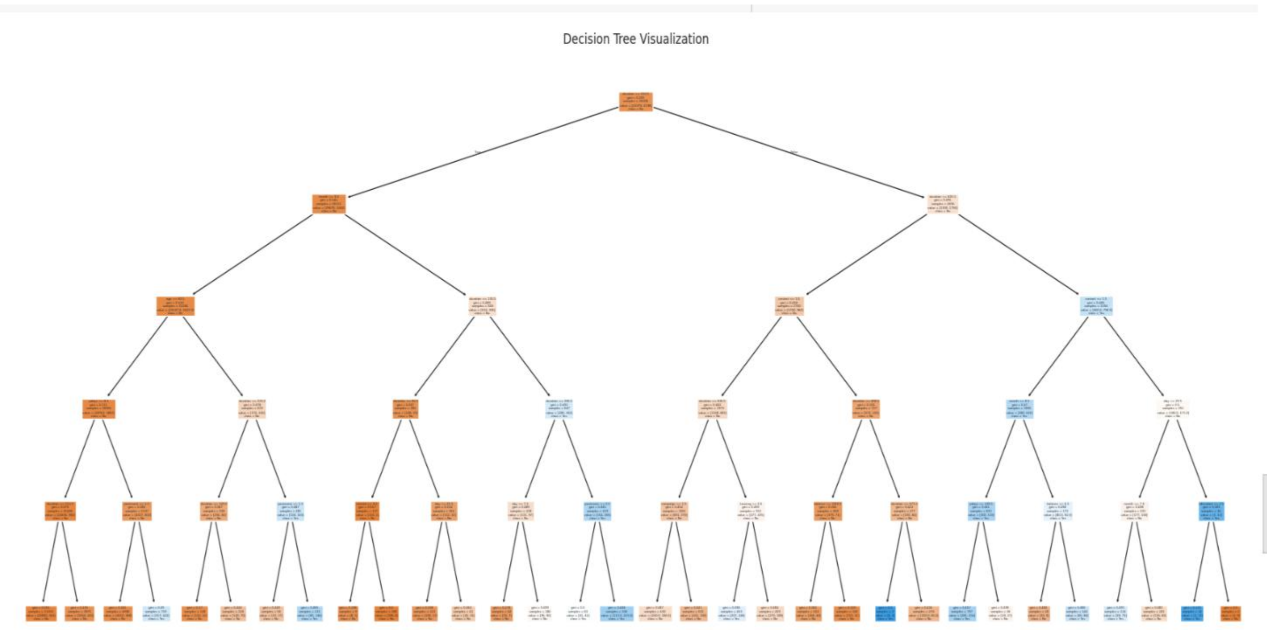
- **F1-score:** Harmonic mean of precision and recall.

Classification Report:

	precision	recall	f1-score	support
0	0.92	0.96	0.94	7952
1	0.59	0.40	0.48	1091
accuracy			0.89	9043
macro avg	0.75	0.68	0.71	9043
weighted avg	0.88	0.89	0.88	9043

Decision Tree Visualization:

The tree shows that features like call duration, month, and poutcome influence predictions the most.



Key Insights:

- Longer call durations lead to higher subscription rates
- Previous campaign outcomes matter
- Clients contacted in May or with short calls are less likely to subscribe

---

Challenges & Solutions:

Challenge	Solution
Many categorical columns	Used LabelEncoder
Overfitting risk	Limited tree depth
Large dataset	Used sample for testing, full for final training

---

Links:

- 📂 GitHub Repo: [REPO]
- 🔗 Google Colab Notebook: [colab]
- 🌐 Dataset: UCI Bank Marketing Repository

Contact

[AKANKSHA BHOSLE] Data Science Intern @Credora

- Email: [akanshabhosle31@gmail.com]
- LinkedIn: [LINKDIN_AKANKSHA]
- GitHub: [GITHUB_AKANKSHA]