Career Switch Prediction Using Machine Learning Models

Project Report

by

MD Ashraful Hoque
Student ID: 22101618

Constructed for the course Artificial Intelligence
BRAC University, Dhaka, Bangladesh

Course: **CSE422 – Artificial Intelligence**

January 2026

Abstract

Career switching has become increasingly common due to rapid technological advancement, evolving job market demands, economic uncertainty, and changes in personal interests and professional expectations. Predicting whether an individual is likely to change careers can be valuable for employers (to reduce turnover), for policymakers (to plan workforce development), and for individuals (to make better career decisions).

This project analyzes a real-world dataset containing demographic, education, and employment-related attributes of 5000 individuals and builds machine learning models to predict the binary target variable `will_change_career`. The dataset includes both numerical and categorical features, requiring preprocessing steps such as missing value handling, label encoding, and feature scaling [1]. Multiple supervised machine learning models are trained and tested, including K-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, and Gaussian Naive Bayes. In addition, a Multi-Layer Perceptron (MLP) neural network is implemented to capture non-linear feature interactions [2]. To satisfy unsupervised learning requirements, K-Means clustering is applied and visualized using Principal Component Analysis (PCA).

The experimental results suggest that Logistic Regression, Naive Bayes, KNN, and the Neural Network achieve competitive performance, while Decision Tree shows lower accuracy likely due to overfitting. The report also discusses how dataset imbalance affects evaluation, making precision, recall, confusion matrices, and ROC-AUC important for reliable comparison.

Keywords: Career switch prediction, classification, supervised learning, unsupervised learning, KNN, logistic regression, naive Bayes, decision tree, neural networks, K-Means.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Career switching refers to moving from one career domain to another—for example, transitioning from a non-technical profession to a technology-based role. Such decisions are influenced by several factors including skill mismatch, job satisfaction, salary expectations, work-life balance, personal interest, and the availability of better opportunities.

1.2 Motivation

The motivation behind this project is to explore how machine learning can identify patterns that indicate a potential career switch. Traditional statistical approaches may fail to capture complex relationships between multiple factors, whereas machine learning models can learn such patterns directly from data [1].

1.3 Objectives

The objectives of this project are:

- Understand the dataset and its feature types.
- Perform necessary preprocessing (missing values, encoding, scaling).
- Train multiple supervised learning classifiers and compare their results.
- Apply an unsupervised learning method (K-Means clustering) for pattern discovery.
- Evaluate models using suitable metrics beyond accuracy.

Chapter 2

Dataset Description

2.1 Dataset Overview

The dataset contains 5000 data points. There are 13 input features and 1 output feature. The target feature is `will_change_career`, where:

- 0 = No career change
- 1 = Career change

This makes the problem a **binary classification task**.

2.2 Feature Types

The dataset includes both numerical and categorical features.

2.2.1 Numerical Features

- `city_development_index`
- `training_hours`

`enrollee_id` is numeric but works as an identifier and is not predictive.

2.2.2 Categorical Features

- `city`
- `gender`
- `relevent_experience`
- `enrolled_university`

- `education_level`
- `major_discipline`
- `experience`
- `company_size`
- `company_type`
- `last_new_job`

2.3 Why Encoding is Required

Most machine learning algorithms cannot directly process text-based categories.Machine learning algorithms cannot directly process categorical (text-based) data. Therefore, categorical features must be converted into numerical form using encoding techniques such as: Label Encoding for ordinal features One-Hot Encoding for nominal features Therefore, In this project label encoding was applied to convert categorical values into numerical representations while preserving category distinctions.Encoding ensures that the dataset is compatible with machine learning models and prevents misinterpretation of categorical values.Categorical features must be transformed into numeric form [1]. In this project, **label encoding** was applied.

2.4 Correlation and Class Imbalance

A correlation heatmap was generated to analyze relationships between numerical features and the output [1]. The dataset is also imbalanced, meaning one class appears more frequently. In such cases accuracy is not sufficient; precision, recall, F1-score and ROC-AUC are more informative.

In this figure a bar chart was used to visualize the distribution of the target classes. The chart clearly shows that one class has a higher number of instances than the other. This imbalance must be considered during model evaluation, and performance metrics such as precision, recall, F1-score, and ROC–AUC are more appropriate than accuracy alone.

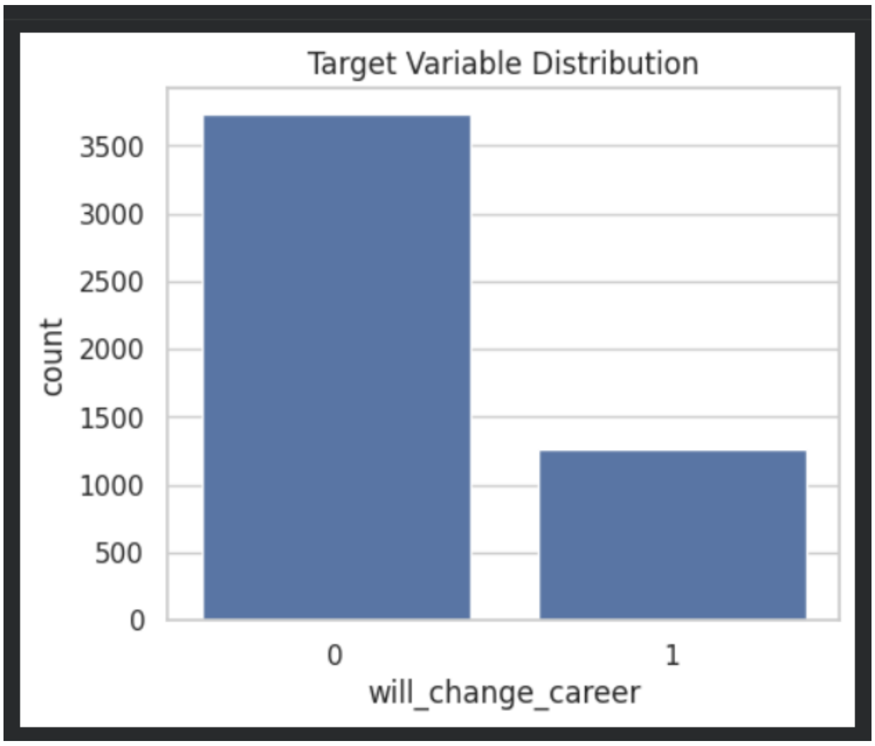

Figure 2.1: Target Variable Distribution

Chapter 3

Dataset Preprocessing

3.1 Handling Missing Values

Missing values reduce model performance. Therefore:

- Numerical missing values were handled using mean imputation [1].
- Categorical missing values were handled using the most frequent value [1].

3.2 Categorical Feature Encoding

All categorical features were converted into numerical form using label encoding so that machine learning models can process them [1].

3.3 Feature Scaling

Some algorithms are sensitive to feature scale. Standard Scaling was applied to normalize values and improve learning for distance-based and gradient-based models [1].

Chapter 4

Exploratory Data Analysis (EDA)

4.1 Definition of EDA

Exploratory Data Analysis (EDA) is the process of analyzing data using visual and statistical techniques before modeling [1]. It helps identify missing values, outliers, imbalance, and important patterns.

4.2 Importance of EDA in This Project

EDA was conducted to:

- Identify imbalance in the dataset.
- Study distributions of numerical features across target classes.
- Observe trends in categorical variables.

4.3 Key Observations

EDA indicates both numerical and categorical features show distinct behavior across the target classes, supporting the use of machine learning models. To understand the deeper relationships between our features and the target variable, we performed further exploratory analysis.. The EDA revealed meaningful patterns in both numerical and categorical features with respect to the target variable. Numerical features showed varying distributions across target classes, while categorical features demonstrated distinct trends that may influence career change decisions. These insights guided feature preprocessing and selection for subsequent model development.

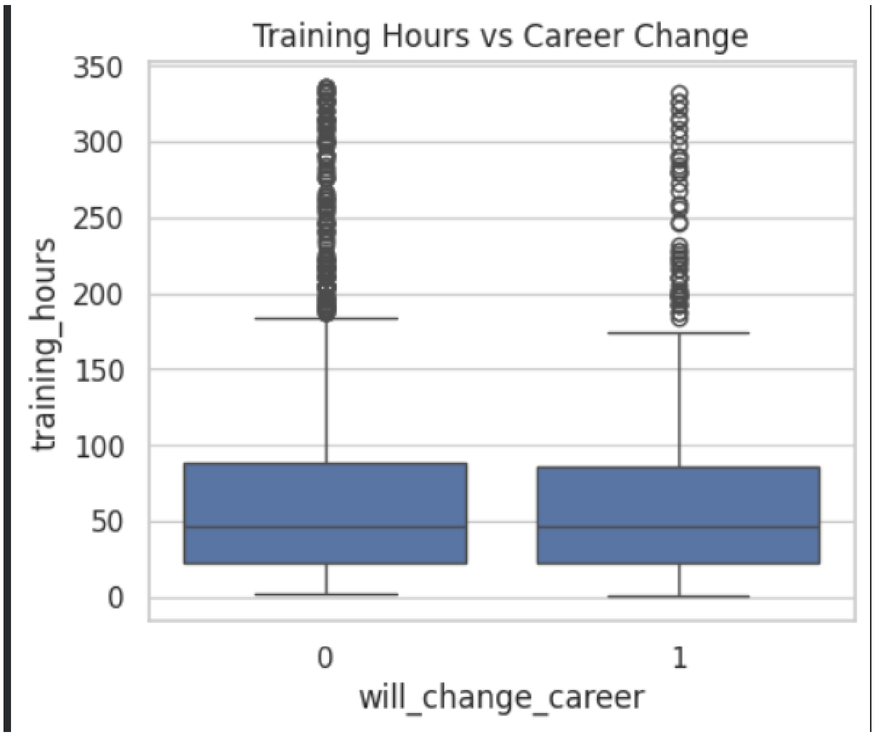

Figure 4.1: Training Hours vs Career Change

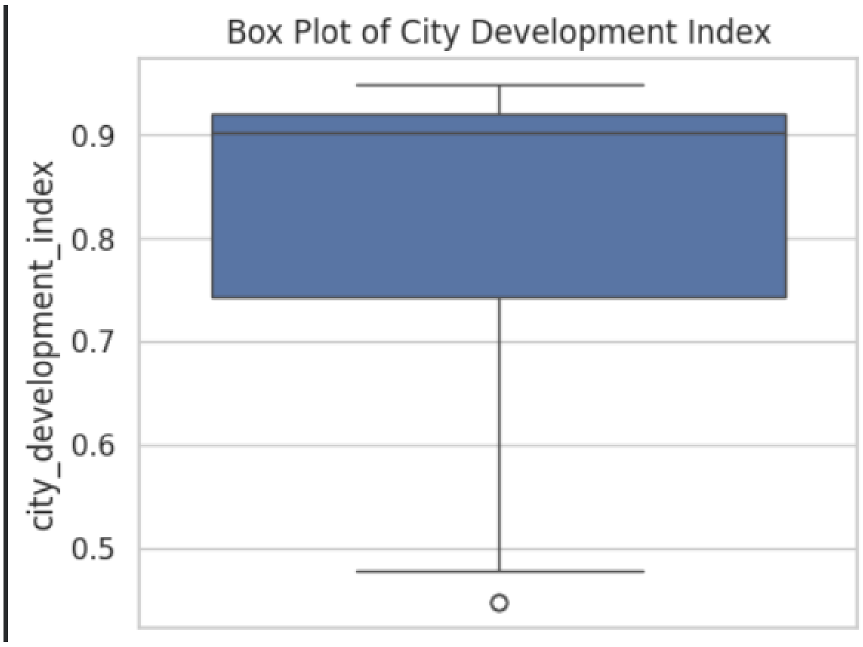

Figure 4.2: Box Plot of City Development Index

Chapter 5

Dataset Splitting

5.1 What is Dataset Splitting?

Dataset splitting divides data into training and testing subsets. The training set is used to learn patterns, while the test set evaluates performance on unseen data [1].

5.2 Splitting Strategy

Stratified sampling was used to maintain class distribution:

- Training set: 80%
- Testing set: 20%

Chapter 6

Supervised Learning Models

6.1 What is Supervised Learning?

Supervised learning trains models using labeled data where each example has input features and a known output label. The model learns a mapping from inputs to outputs.

6.2 Models Implemented

6.2.1 K-Nearest Neighbors (KNN)

KNN predicts class based on the majority class among the K nearest neighbors [2]. It is simple but sensitive to feature scaling.

6.2.2 Decision Tree Classifier

Decision trees classify using rule-based splits of features. Trees are interpretable but can overfit [3].

6.2.3 Logistic Regression

Logistic regression estimates class probability using a sigmoid function and works well for binary classification [4].

6.2.4 Gaussian Naive Bayes

Naive Bayes is based on Bayes' theorem. Gaussian NB assumes continuous features follow normal distribution [0].

6.2.5 Neural Network (Multi-Layer Perceptron)

A Multi-Layer Perceptron (MLP) is a feedforward neural network capable of learning non-linear patterns [0]. The implemented architecture:

- Input: 14 neurons
- Hidden layers: 64 and 32 neurons (ReLU)
- Output: 1 neuron (Sigmoid)
- Optimizer: Adam

Chapter 7

Unsupervised Learning

7.1 What is Unsupervised Learning?

Unsupervised learning finds hidden patterns from data without labels. It is useful for clustering and grouping similar samples.

7.2 K-Means Clustering

K-Means clustering groups individuals into K clusters based on similarity. PCA was applied to reduce dimensions for better visualization.

Model comparison analysis: Best performance is given by Logistic regression, naive bayes, neural network and KNN.Decision Tree had the lowest accuracy likely due to overfitting on the training data.

In the bar chart the precision and recall comparison of each model is shown.

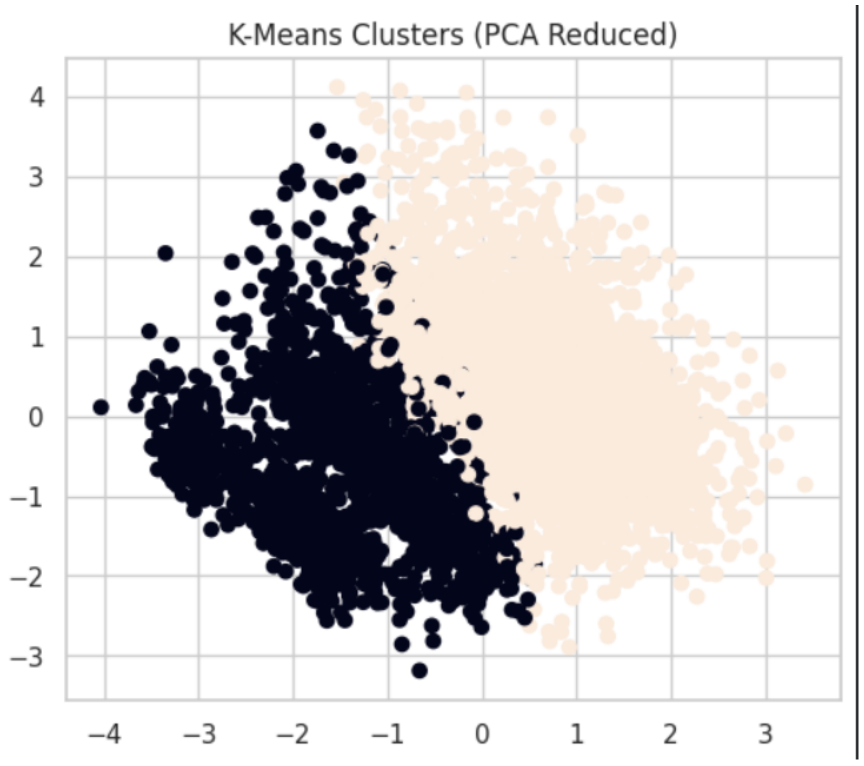

Figure 7.1: K-Means Clusters (PCA Reduced

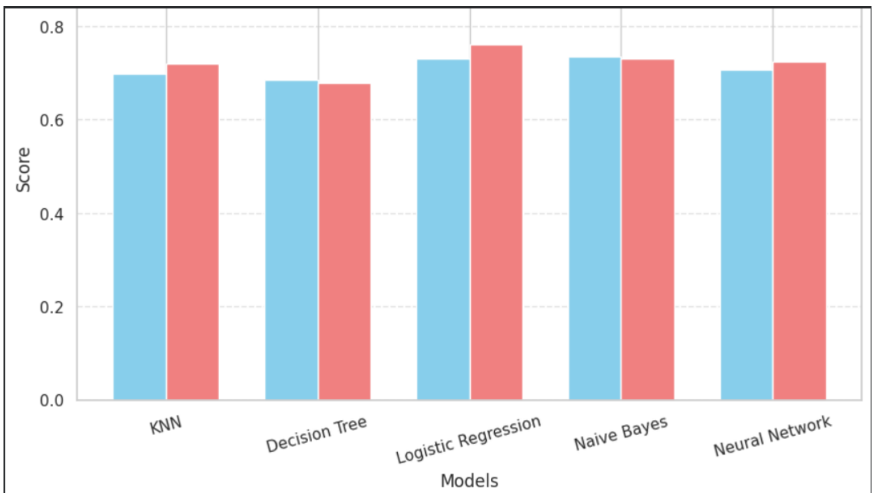

Figure 7.2: Precision and Recall Comparison of Classification Models

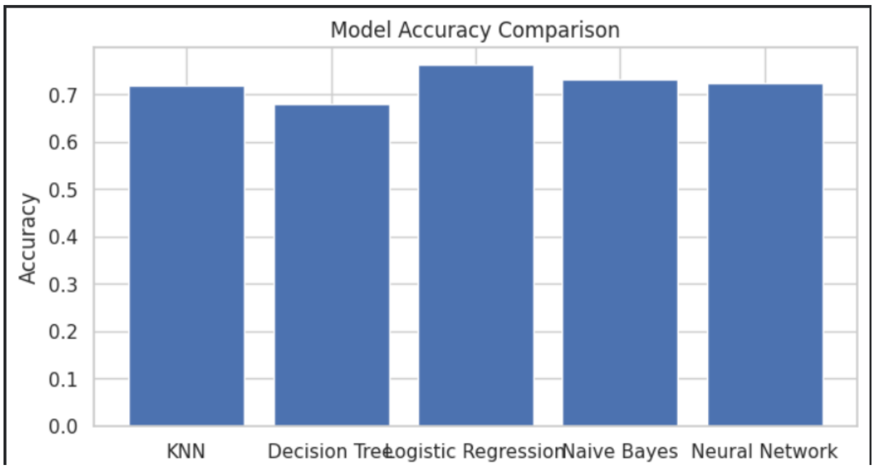

Figure 7.3: Model Accuracy Comparison

Chapter 8

Model Evaluation

8.1 What is Model Evaluation?

Model evaluation measures how well a trained model performs on unseen test data. Proper evaluation prevents misleading results due to overfitting.

8.2 Evaluation Metrics

Metrics used:

- Accuracy
- Precision
- Recall
- Confusion Matrix
- ROC curve and AUC score

8.3 Confusion Matrix

A confusion matrix summarizes prediction outcomes:

- TP: predicted 1, actual 1
- TN: predicted 0, actual 0
- FP: predicted 1, actual 0
- FN: predicted 0, actual 1

It is especially useful for imbalanced datasets.

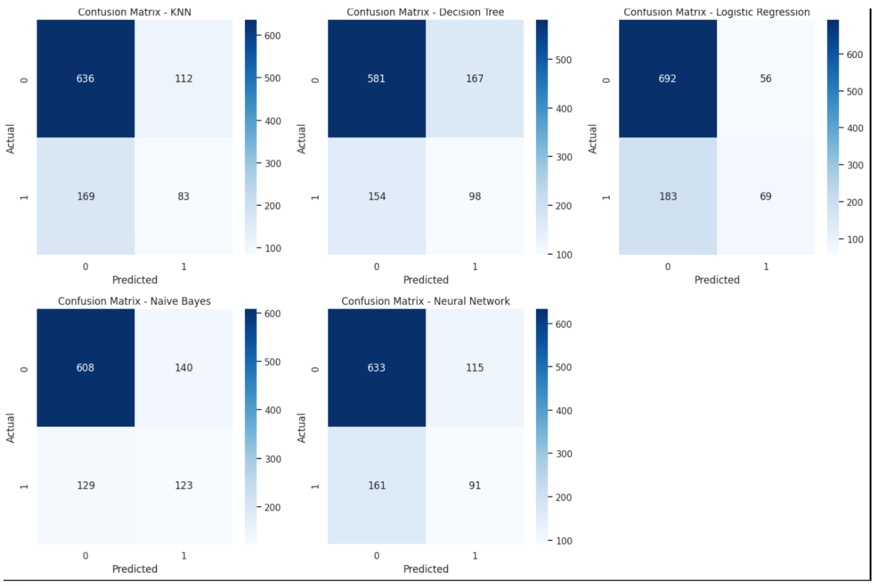

Figure 8.1: Confusion Matrix - Predicted,Actual

8.4 ROC Curve and AUC

ROC curves show trade-off between TPR and FPR. AUC measures class separation ability.

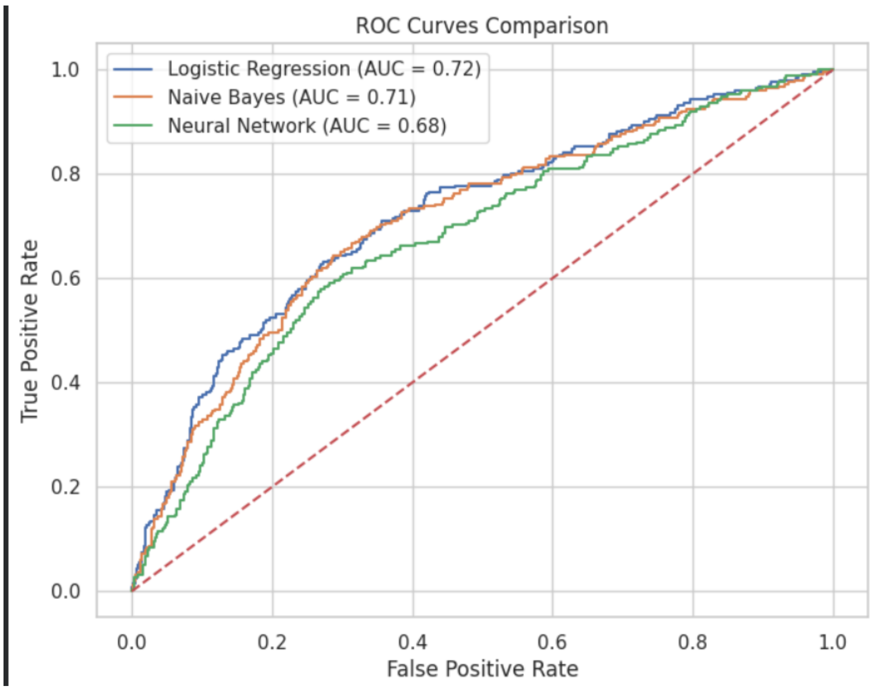

Figure 8.2: ROC Curves Compariso - False Positive Rate,True Positive Rate

Chapter 9

Results and Discussion

9.1 Performance Comparison

Model comparison analysis shows:

- Logistic Regression, Naive Bayes, KNN and Neural Network achieved best performance.
- Decision Tree showed the lowest accuracy likely due to overfitting.

9.2 Discussion

Performance differences occur because:

- Feature interactions and correlations
- Dataset imbalance
- Model assumptions
- Sensitivity to scaling and encoding

Chapter 10

Conclusion and Future Work

10.1 Conclusion

This project demonstrates that machine learning models can predict career switching behavior using demographic, educational, and professional features. The dataset contains missing values and categorical attributes, so preprocessing steps such as imputation, encoding, and scaling were required [1]. Due to class imbalance, accuracy alone is not enough to judge model performance; precision, recall, confusion matrices and ROC-AUC provide better understanding.

Overall, Logistic Regression and Gaussian Naive Bayes performed strongly and consistently. KNN also achieved competitive results after scaling. The Neural Network performed well due to its ability to capture non-linear feature relationships. Decision Tree showed weaker performance and likely overfitted the training data.

10.2 Limitations

- Label encoding may introduce artificial ordering for nominal variables.
- No advanced imbalance technique (SMOTE / class weights) was applied.
- Hyperparameter tuning was limited.
- Ensemble models like Random Forest or XGBoost were not tested.
- Neural networks have lower interpretability compared to linear models.

10.3 Future Work

- Apply SMOTE or class weighting to reduce imbalance.
- Use one-hot encoding for nominal categorical features.

- Tune hyperparameters using GridSearchCV.
- Try ensemble models for improved performance.
- Apply SHAP/feature importance analysis for interpretability.

IEEEexample:BSTcontrol

Bibliography

[1] “Eda cse422 lab (provided course material),” Course material, 2026.

[2] S. learn Developers, “sklearn.neighbors.kneighborsclassifier,” Scikit-learn documentation, 2026, retrieved January 4, 2026, from https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

[3] ——, “Decision trees,” Scikit-learn documentation, 2026, retrieved January 4, 2026, from https://scikit-learn.org/stable/modules/tree.html.

[4] ——, “sklearn.linear$_{m}odel.logisticregression,'' Scikit-learndocumentation$, 2026, $retrievedJanuar$

——, “Naive bayes,” Scikit-learn documentation, 2026, retrieved January 4, 2026, from `https://scikit-learn.org/stable/modules/naive_bayes.html`.

——, “sklearn.neural$_{n}etwork.mlpclassifier,'' Scikit-learndocumentation$, 2026, $retrievedJanuary$