CSE 422 (SEC:13) LAB PROJECT REPORT

Topic : Predicting a Person's Likelihood to Switch Careers Using an AI Model

NABONITA SAHA	**22301645**
TASFIA ZAMAN	**22301779**

TABLE OF CONTENTS

1. Introduction

Understanding the factors that influence an individual's intent to switch careers can offer valuable insights for employers, job platforms, and educational institutions. This project aims to develop a predictive model that determines whether a person is likely to change careers based on a variety of demographic, educational, and professional attributes.

The dataset used for this task contains information on thousands of individuals, including features such as city development index, education level, relevant experience, company size, and training hours. The target variable is binary — whether or not the individual intends to change careers.

To address this binary classification problem, we applied a full machine learning pipeline: data preprocessing (handling missing values, encoding categorical variables, and feature scaling), exploratory data analysis, model training, and evaluation. We tested multiple algorithms including Logistic Regression, K-Nearest Neighbors, and a Neural Network, comparing their performance using standard classification metrics such as accuracy, F1-score, and ROC-AUC.

2. Dataset Description

A. Dataset Summary

- **Number of Features (Input Columns)**: 12 (excluding **enrollee_id** and target feature)
- **Number of Data Points (Rows)**: 5000
- **Target Feature**: **will_change_career** (0 = No, 1 = Yes)
- **Problem Type**: **Binary Classification**
    - Because the output variable is binary, this is a supervised classification problem.

B. Feature Types

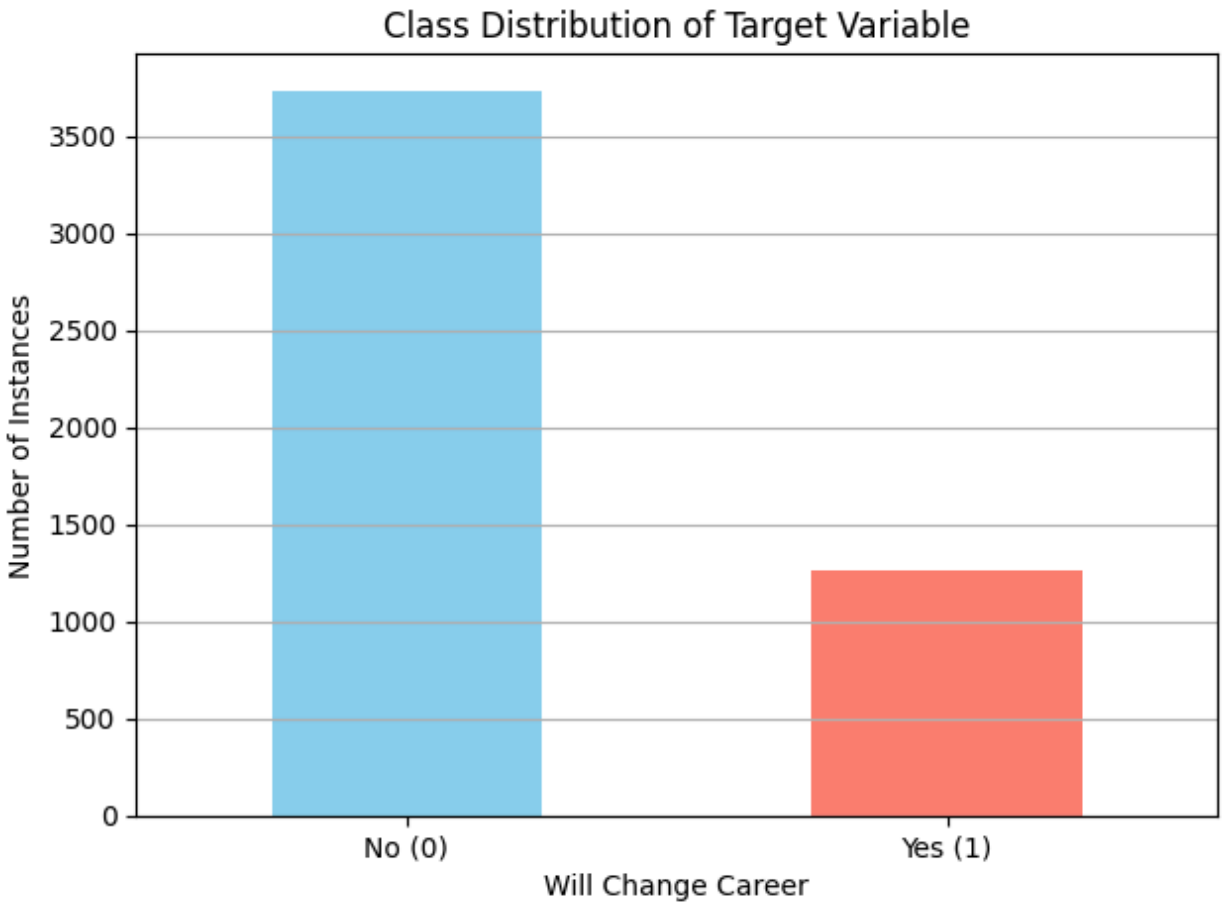
Type	Feature
Quantitative	**city_development_index, experience** (after cleaning), **training_hours, last_new_job** (after cleaning)

Categorical	**gender,** **relevent_experience** (binary)**,** **enrolled_university,** **education_level** (ordinal)**,** **major_discipline,** **company_size** (ordinal)**, company_type, city**

C. Imbalance Analysis

☐ The target variable **will_change_career** is binary (0 or 1). We visualized the class distribution using a bar chart, which shows that the dataset is imbalanced, with significantly more instances of **class 0** (no career change) compared to **class 1** (career change).

This imbalance could bias classification models toward predicting the majority class, so we will apply stratified splitting and evaluate models using appropriate metrics such as precision, recall, and ROC-AUC, instead of relying solely on accuracy.



D. Correlation Heatmap (After Preprocessing)

The correlation heatmap below was generated using the *seaborn* library after completing all preprocessing steps, including missing value imputation and encoding. At this stage, selected features were either continuous or ordinal, and all were converted to numeric format, allowing for valid Pearson correlation analysis.

The heatmap focuses on key features that were either continuous or meaningfully ordered. Features with high cardinality (like city) or one-hot encoded dimensions (like gender or company_type) were excluded to avoid clutter and misleading interpretations.

Key Observations:

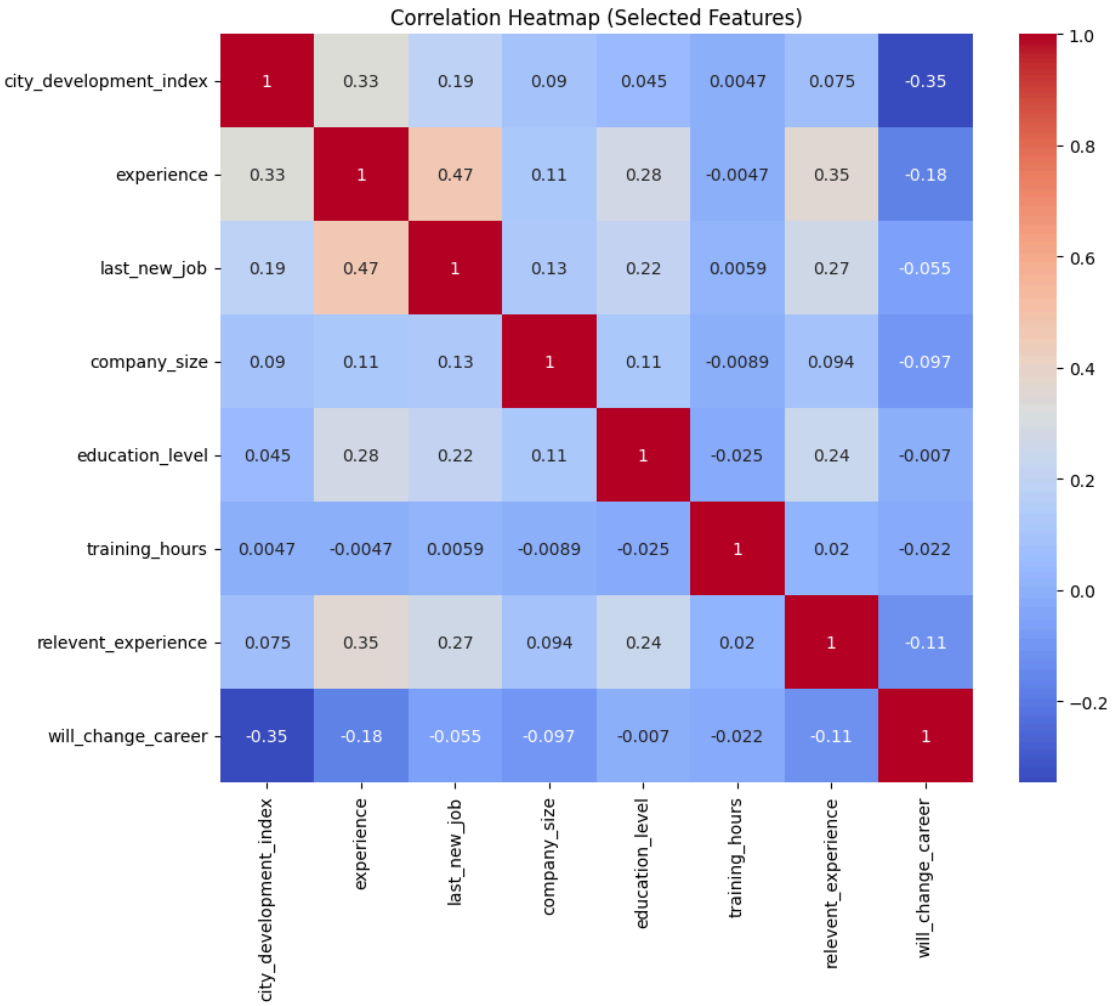
- **city_development_index** shows a weak **negative** correlation (~**-0.35**) with the target variable. This suggests individuals from less developed cities may be more inclined to switch careers.

- **experience** and **last_new_job** are **positively correlated** with each other (~0.47), indicating that individuals with more experience may also have stayed longer at their previous job.

- **experience** also shows a weak **negative** correlation (~**-0.18**) with the target, implying that less experienced individuals are slightly more likely to consider a career change.

- **relevent_experience** and **education_level** show **very weak** correlations (~±0.1 or less), which suggests limited standalone predictive power but potential usefulness when combined with other features.

- **training_hours** has **almost zero correlation** with the target, indicating that it may not influence career switching behavior in a linear way.



Most features show only weak linear relationships with the target variable, which is expected in real-world classification problems. This supports the use of machine learning models capable of capturing **nonlinear interactions** among variables, such as neural networks.

E. Exploratory Data Analysis (EDA)

We conducted exploratory data analysis (EDA) to identify key trends and inform preprocessing and modeling choices. Only features with low proportions of missing data were included in this step to ensure reliability.

- **relevant_experience**
  Individuals without relevant experience showed a higher likelihood of switching careers. This justified preserving the feature and encoding it as binary during preprocessing.

- **training_hours**
  The average training hours were slightly higher among those who did not plan to change careers. Combined with its near-zero correlation with the target, this suggested that the

feature might have non-linear effects, supporting the use of flexible models like neural networks.

- **Dominant category check**
  Visualizations of categorical features like **gender, education_level, company_size**, and **enrolled_university** helped confirm the feasibility of using mode imputation where appropriate. For example, Graduate was the most common education level, and missing values were relatively few (118 out of 5000), making mode imputation a safe choice.

- **Missing data patterns**
  Most rows had missing values, so we chose imputation over row deletion to retain data. This decision was confirmed by visualizing missing feature counts per row.

EDA results showed that career change intent is influenced by multiple subtle factors rather than any single dominant variable. This reinforced our choice of models that can capture complex interactions among features.

3. Dataset Preprocessing

A. Missing Values

- Instead of dropping rows with missing values, we first handled all missing data using imputation (mode for categorical, median for numeric).

Feature	Type	Missing	Action Taken	Justification
gender	Categorical	1113	Imputed with mode	Key demographic feature; mode retains majority pattern
enrolled_university	Categorical	107	Imputed with mode	Important for educational context; low missing count
education_level	Categorical	118	Imputed with mode	Reflects qualification; low missing count
major_discipline	Categorical	724	Imputed with mode	Related to skill background; imputation avoids data loss
experience	Quantitative	11	>20 → 21, <1 → 0.5, converted to numeric; imputed with median	Numeric conversion retains meaning; median handles minor missingness
company_size	Categorical	1571	corrupted entry "Oct-49" was corrected to "10-49"; imputed with mode	Fixes structural error and preserves ordinal mapping despite high missing rate
company_type	Categorical	1621	Imputed with mode	High missing rate; imputed to retain useful company info
last_new_job	Quantitative	104	>4 → 5; converted to numeric; imputed with median	Reflects job change recency; numeric form enables ordinal treatment

B. Categorical Encoding

- Categorical features were encoded using multiple strategies, selected based on the nature and semantics of each feature:
    - **Binary Encoding: relevent_experience** was manually mapped to 0 and 1.
    - **Integer Mapping:** Features with an inherent order were mapped to integers:
      education_level (e.g., Primary School → 0, PhD → 4)
      company_size (e.g., <10 → 0, 10000+ → 7)
      These features are treated as discrete numeric variables during modeling and correlation analysis.
    - **Label Encoding: city** was label-encoded due to its high cardinality (113 unique values).
    - **One-Hot Encoding:** Categorical features without meaningful order and with low cardinality were one-hot encoded: **gender**, **enrolled_university**, **major_discipline**, and **company_type.** The 'drop_first=True' parameter was used to avoid multicollinearity.
- This encoding approach preserves important ordinal relationships while minimizing unnecessary dimensionality.

C. <u>Feature Scaling</u>

- After splitting the dataset, we applied StandardScaler to normalize the entire feature set. Scaling was performed on the training data using *fit_transform*, and the same transformation was applied to the test set using *transform*.
- This approach avoids data leakage and ensures models that are sensitive to feature magnitude (such as KNN, Neural Networks and Logistic Regression) perform effectively

D. <u>Dropped Rows</u>

- We dropped the column **enrollee_id**, as it is a unique identifier and does not contribute to prediction. Retaining such columns would introduce noise and provide no signal to the model.

4. Dataset splitting

We used **stratified sampling** during the train-test split to ensure that the **class distribution** of the target variable (**will_change_career**) was preserved in both the training and testing sets. This was done using the **train_test_split()** function from the **scikit-learn** library with the parameter **stratify=y**.

This approach was essential due to the class imbalance in the dataset, where the **"No Change" (class 0)** category was significantly more frequent than the **"Change" (class 1)** category. Without stratification, the test set could end up with too few minority class samples, resulting in misleading evaluation metrics.

By maintaining the original proportion of classes in both sets, we ensured that the models were trained and evaluated on data that reflects the true class distribution, leading to fairer and more reliable performance measurements.

5. Model Training & Testing

We trained three machine learning models on the SMOTE-balanced and scaled training set:

- K-Nearest Neighbors (KNN)
- Neural Network
- Logistic Regression

To address the class imbalance problem in the dataset, SMOTE (Synthetic Minority Oversampling Technique) was applied after the train-test split to oversample the minority class in the training data. This ensured a more balanced class distribution during training while preserving the integrity of the test set.

All models were trained using the fit() method from the scikit-learn library. Features were standardized using StandardScaler prior to training, which is particularly important for distance-based models (like KNN) and gradient-based models (like Neural Networks and Logistic Regression).

The models were tested on the original (unbalanced) test set to evaluate their ability to generalize. Performance was assessed using the following metrics:

- Accuracy – Proportion of total correct predictions.
- Precision, Recall, and F1-Score – Particularly important for imbalanced classification.
- Confusion Matrix – To visualize true positives, false positives, and false negatives.

These metrics gave a detailed picture of each model's strengths and weaknesses, especially in detecting the minority class ("Change" = 1). Detailed comparisons of these results are discussed in Section 6.

6. Model Evaluation and Comparison

We evaluated and compared the performance of three classification models — K-Nearest Neighbors (KNN), Logistic Regression, and a Neural Network (MLPClassifier) — using multiple performance metrics suited for imbalanced binary classification tasks.:

Model	Accuracy	Precision	Recall	F1-Score	AUC Score
Logistic Regression	71.2%	0.44	0.50	0.46	0.68
KNN (k=5)	68.9%	0.41	0.52	0.46	0.67
Neural Network (MLP)	69.6%	0.41	0.46	0.43	0.68

Confusion Matrices:

Each model's confusion matrix was visualized to provide a clearer understanding of the classification balance. All models performed better at predicting the majority class (**"No Change"**), but varied in how well they captured the minority class (**"Change"**):

- **KNN** predicted 130 true positives (Change), misclassifying 122.
- **Logistic Regression** performed slightly better, identifying 125 true positives.
- **Neural Network** identified 116, with a slightly higher false negative rate.

ROC Curve Comparison:

A combined ROC curve was plotted for all three models. Logistic Regression and Neural Network tied with the highest AUC score of **0.68**, slightly outperforming KNN (**0.67**). While all models are better than random guessing, there is still room for improvement in discriminative ability.

Overall, Logistic Regression achieved the best balance of precision, recall, and interpretability. Neural Network was competitive in AUC but slightly weaker in

F1-score. KNN lagged behind, possibly due to sensitivity to high-dimensional data and class imbalance.

7. Conclusion

This project aimed to predict whether an individual is likely to switch careers using machine learning. After preprocessing a real-world imbalanced dataset, we trained and evaluated multiple classification models using metrics suited to the challenge.

Key Findings:

- The **Logistic Regression model** achieved the best overall performance, offering strong interpretability and balanced metrics.
- The **Neural Network** showed strong AUC performance, suggesting it can model non-linear feature interactions, but its recall and F1-score were slightly lower.
- **KNN** was the weakest model, likely due to high-dimensionality and class imbalance affecting distance-based learning.

Observations on Class Imbalance:

- The dataset showed a strong imbalance (about 3:1) favoring the "No Change" class.
- As a result, all models had significantly better performance on that class.
- We addressed this using **SMOTE (Synthetic Minority Oversampling Technique)** to balance the training data — improving recall and F1-score on the minority class compared to earlier baselines.

Challenges Faced:

- **Dealing with class imbalance** without overfitting or underfitting.
- Choosing encoding techniques that respected feature semantics (ordinal vs nominal).
- Interpreting weak feature correlations, which reinforced the need for models that capture non-linear interactions.