Project Overview: Fake Job Postings Detection

# Objective:

The primary goal of this project is to develop a machine learning model capable of identifying fake job postings. By leveraging various features such as job titles, locations, company profiles, and more, the model aims to distinguish between fraudulent and legitimate job listings.

# Dataset:

The project utilizes a dataset stored in a CSV file ('fake\_job\_postings.csv'). This dataset contains information such as job titles, locations, company profiles, and a binary indicator ('fraudulent') denoting whether a job posting is fraudulent (1) or not fraudulent (0).

# Steps:

## 1. Data Loading and Exploration:

* Loaded the dataset and explored its structure.
* Gained insights into the distribution of fraudulent and non-fraudulent instances.
* Explored missing values in the dataset and visualized their distribution.

## 2. Data Pre-processing:

* Handled missing values in relevant columns.
* Consolidated relevant text features ('title' and 'description') into a single 'text' feature.
* Prepared the data for model training by addressing missing values and encoding categorical features.

## 3. Model Development:

* Selected a variety of machine learning models, including Gradient Boosting, Random Forest, and XGBoost.
* Utilized a pipeline for efficient data pre-processing and model training.

## 4. Model Evaluation:

* Conducted cross-validation to assess model performance.
* Selected XGBoost as the best-performing model based on accuracy and F1 score.
* Evaluated the model on a test set to validate its effectiveness.

## 5. Visualization:

* Visualized confusion matrices and classification reports to provide insights into model behaviour.
* Plotted ROC curves and precision-recall curves to assess model performance across different thresholds.

## 6. Additional Analysis:

* Explored the imbalance in the dataset and acknowledged the class imbalance issue.
* Conducted a threshold analysis to understand the trade-offs between precision and recall.

## Results:

* XGBoost Classifier emerged as the best-performing model with high accuracy and F1 score.
* Confusion matrices and classification reports were provided to detail the model's performance on the test set.

## Recommendations:

* Consider further exploration of text pre-processing techniques.
* Experiment with advanced modelling techniques like ensemble methods.
* Address class imbalance using oversampling, under sampling, or other strategies.
* Explore the potential of neural networks with access to more processing power.

## Future Work:

* Regularly update the model with new data to maintain accuracy.
* Consider deploying the model in a production environment for real-time detection.