**ML Project**

**Name : Areeba Tahir**

**RollNO : SP23-BAI-010**

**Project Report :**

**The Adult Income Dataset (also known as the Census Income Dataset) from the UCI Machine**

**Problem Description:**

The dataset is used to predict the target variable, which is whether a person's income is **greater than $50,000** or not, based on their demographic and employment information. This is a **binary classification** problem.

**How the target variable is encoded:**

* **>50K**: Individuals earning more than $50,000 per year.
* **<=50K**: Individuals earning less than or equal to $50,000 per year.

**Dataset Description:**

The goal is to predict if a person earns greater than $50,000 (referred to as >50K) or less than or equal to $50,000 (referred to as <=50K) per year based on various factors.

Features in the Dataset:

The dataset contains the following attributes (features) for each individual:

1. age: The person's age in years.
2. workclass: The type of employment (e.g., Private, Self-Employed, etc.).
3. fnlwgt: The "final weight", which is a measure of how representative the person is of the population.
4. education: The highest level of education attained (e.g., Bachelors, Masters, etc.).
5. education-num: A numerical representation of education (e.g., 13 for Bachelors).
6. marital-status: The marital status of the individual (e.g., Never-married, Married-civ-spouse, etc.).
7. occupation: The type of work (e.g., Tech-support, Craft-repair, etc.).
8. relationship: The relationship status of the individual (e.g., Husband, Not-in-family, etc.).
9. race: The race of the individual.
10. sex: The gender of the individual (Male or Female).
11. capital-gain: The capital gains income the person made in the past year.
12. capital-loss: The capital losses the person made in the past year.
13. hours-per-week: The number of hours worked per week.
14. native-country: The country of origin (e.g., United States, Mexico, etc.).
15. Class: The target variable (either >50K or <=50K), indicating the income level.

**2. Methodology**

Preprocessing Steps

1. Data Cleaning:
   * Removed duplicate entries and incomplete records.
   * Imputed missing values using appropriate statistical methods (e.g., mean or mode).
2. Feature Engineering:

Normalized continuous features to a [0, 1] range.

* + Encoded categorical features using one-hot encoding.
  + Selected top N features based on feature importance scores derived from a preliminary model.

1. **Train-Test Split:**
   * Split data into training (80%) and testing (20%) sets.

Algorithms Applied

1. Random Forest Classifier:
   * Ensemble-based method leveraging multiple decision trees.
   * Advantages: Robust to overfitting, handles categorical and numerical data.
2. Logistic Regression:
   * A linear model that estimates probabilities using the sigmoid function.
   * Advantages: Simplicity, interpretability, and effective for linearly separable data.
3. XGBoost:
   * Gradient boosting algorithm optimized for speed and performance.
   * Advantages: Handles missing data well and offers robust handling of large datasets.

**Optimization Techniques**

* Hyperparameter Tuning:
  + Grid Search: Used to exhaustively search predefined parameter combinations.
  + Random Search: Used to sample a fixed number of parameter settings from a specified distribution.
  + Parameters tuned: Depth of trees (Random Forest, XGBoost), regularization strength (Logistic Regression), learning rate (XGBoost).
* Cross-Validation:
  + 5-fold cross-validation applied to ensure robustness of model performance.

3. Results

Metrics

* Accuracy: Percentage of correctly classified instances.
* Precision: Ability to correctly identify malware instances.
* Recall: Ability to detect all actual malware instances.
* F1-Score: Harmonic mean of precision and recall.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score |
| Random Forest | [0.846] | [0.85] | [0.84] | [0.848] |
| Logistic Regression | [0.85] | [0.742] | [0.607] | [0.66] |
| XGBoost | [0.9134] | [0.91] | [0.9] | [0.913] |

Visualizations

* Confusion matrices for each algorithm to illustrate true positives, false positives, false negatives, and true negatives.
* ROC curves to compare the performance across algorithms.

4. Analysis

Insights

* Feature Importance: XGBoost identified [specific features] as the most critical predictors for malware detection.
* Model Performance: XGBoost achieved the highest accuracy and precision, showcasing its capability to handle complex datasets.
* Trade-offs: Logistic Regression provided interpretable results but lacked the flexibility to capture non-linear relationships.

**Algorithm Comparison**

* Random Forest:
  + Strengths: Interpretable, robust to noisy data.
  + Weaknesses: Moderate performance on imbalanced datasets.
* Logistic Regression:
  + Strengths: Simple and interpretable.
  + Weaknesses: Limited performance with non-linear data.
* XGBoost:
  + Strengths: High performance and handles missing data well.
  + Weaknesses: Computationally intensive for very large datasets.

Challenges Faced

1. Imbalanced Dataset: Addressed using oversampling techniques (SMOTE).
2. Feature Selection: High dimensionality required rigorous feature selection to avoid overfitting.
3. Computational Constraints: Training XGBoost models was resource-intensive, necessitating the use of GPU acceleration.

**Conclusion**

The analysis demonstrates the effectiveness of machine learning algorithms in detecting malware using the Microsoft BigData dataset. While each model has strengths and limitations, XGBoost emerged as the most effective, balancing accuracy and precision. Future work includes exploring hybrid models and real-time detection capabilities.