Income predection

(ML Report)

By:

- Moamen Sherif
- Mohab Ashraf
- Mostafa Bahnasy
- Omar Alaa
- Ali Samy Gadallah
- Kerollos Rady
- Abanoub Ashraf

Introduction:

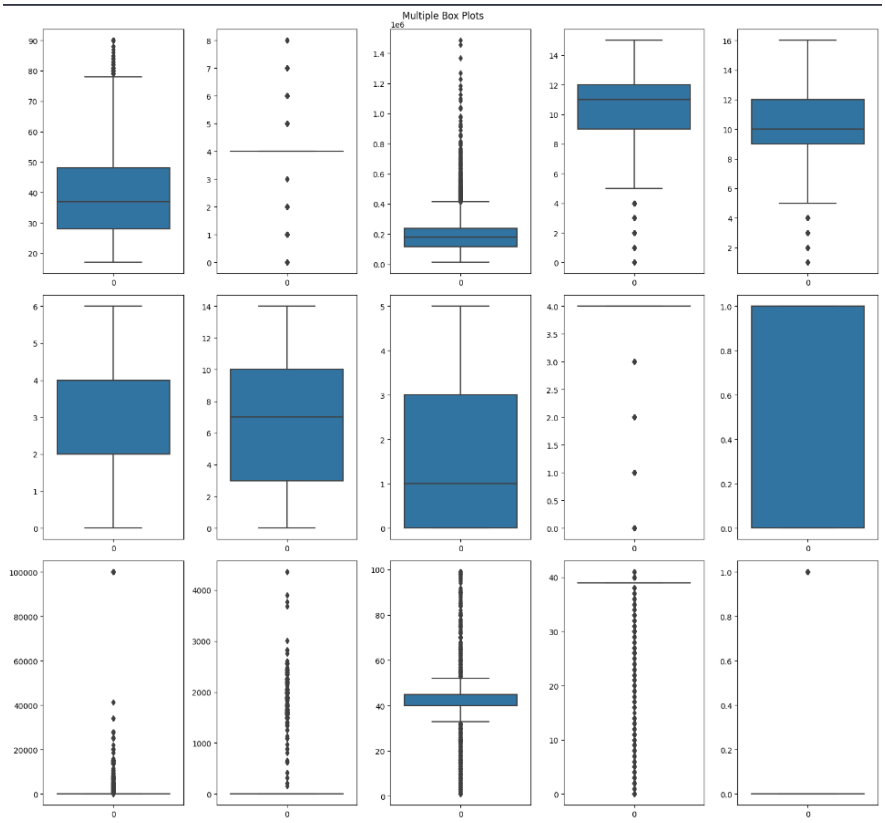
This report describes a machine learning project that involves the classification of income based on various features. The project utilizes several machine learning algorithms, data pre-processing techniques, and evaluation metrics.

Data Exploring:

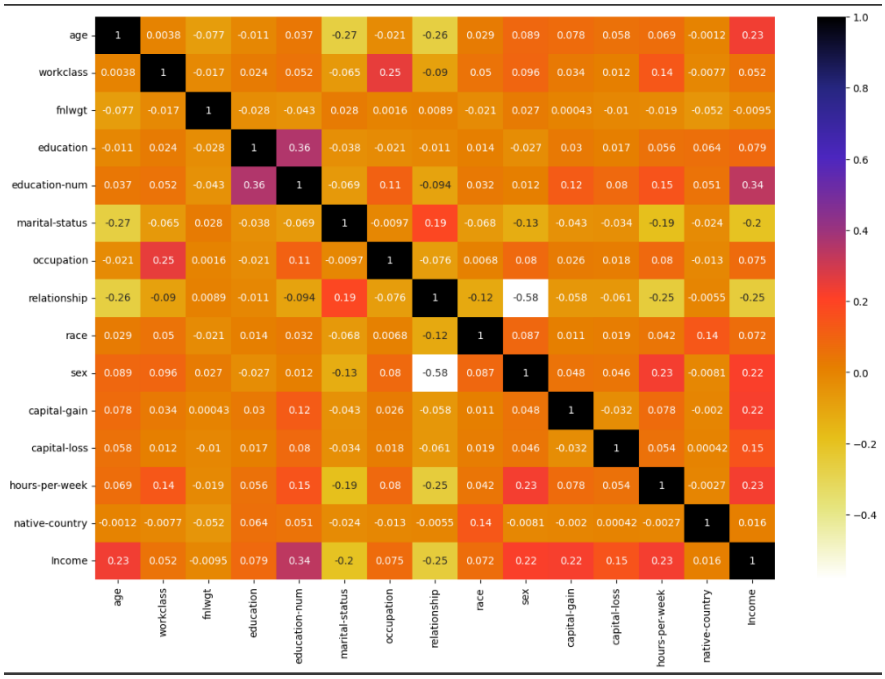
The project starts with data exploration to gain an understanding of the dataset. The dataset is loaded from two CSV files: "train_data.csv" and "test_data.csv". The data is then examined using various methods such as info(), describe(), and head(). The goal is to understand the structure of the data, identify missing values, and detect outliers.

The data exploration phase includes the following steps:

- Removing white spaces in column names and values.
- Checking for null values in the dataset.
- Encoding categorical variables to numerical values using label encoding.
- Analysing the presence of outliers using box plots.



- Computing correlations between variables using point-biserial correlation and Spearman correlation.
- Visualizing the correlations using a heatmap.



Data Pre-processing:

Data pre-processing is performed to prepare the dataset for training the machine learning models. The following pre-processing steps are applied:

- Replacing common marital status values with equivalent values in both the train and test datasets.
- Dropping duplicated rows in the dataset.

Data Sampling:

In this project, data sampling techniques such as oversampling and SMOTE (Synthetic Minority Over-sampling Technique) are used to handle imbalanced classes. However, these techniques are commented out in the code, and the original dataset is used for training the models.

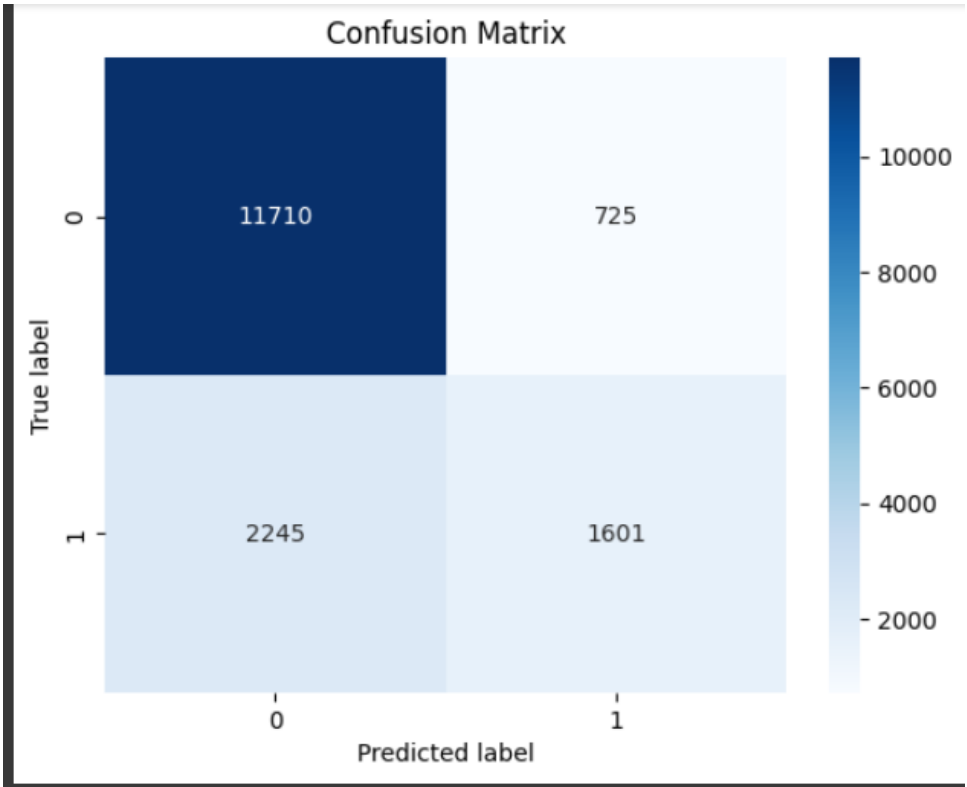
Test Algorithms:

Several machine learning algorithms are tested on the dataset. The algorithms used in this project are Logistic Regression, XGBoost, and Random Forest Classifier. For each algorithm, the following steps are performed:

- Creating an instance of the algorithm.
- Fitting the algorithm to the training data.
- Making predictions on the test data.
- Evaluating the performance using accuracy score and confusion matrix.
- Displaying the confusion matrix using a heatmap.

Logistic Regression:

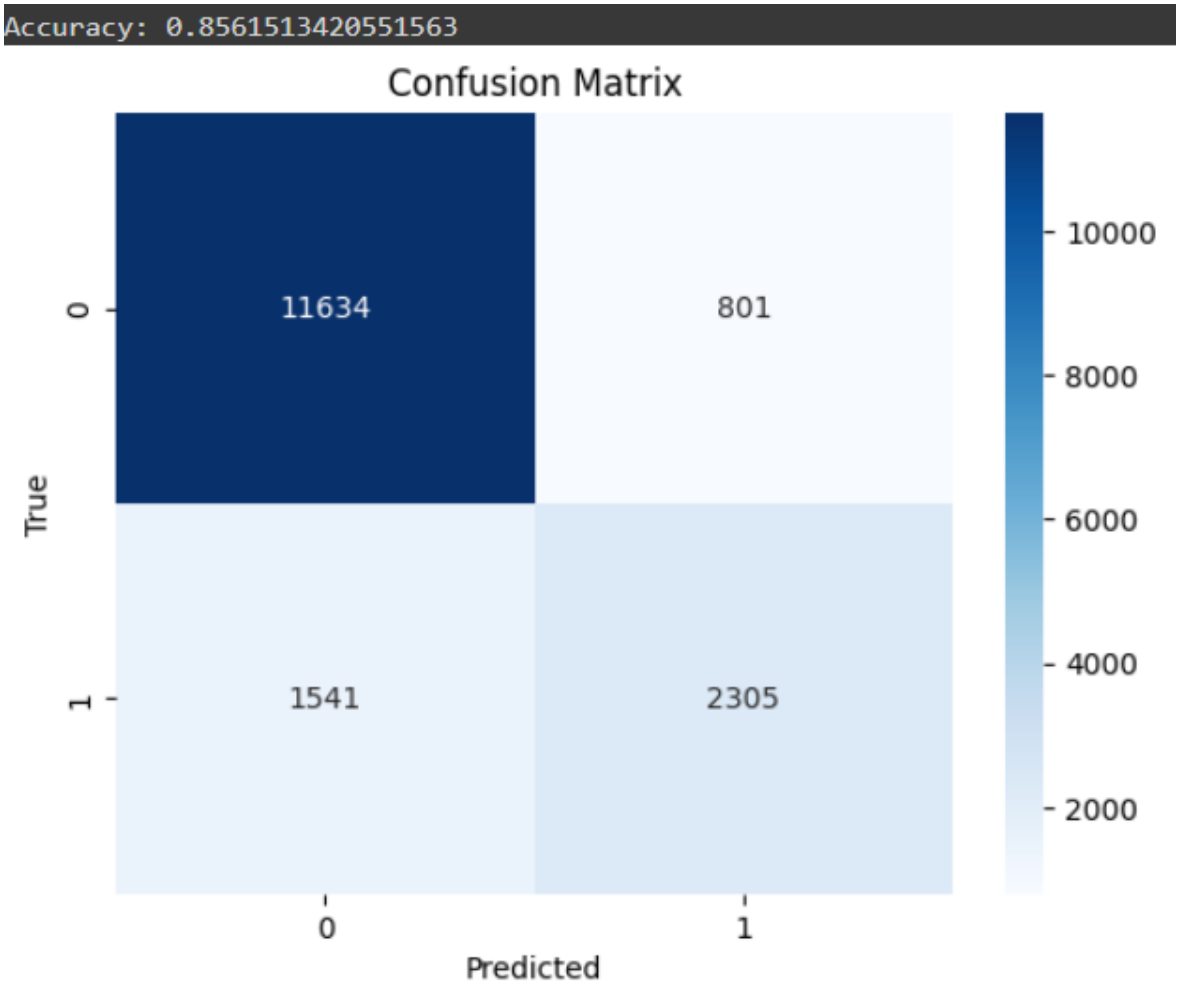
Logistic Regression is applied using the ‘Logistic Regression’ class from scikit-learn. The algorithm is initially tested without hyperparameter tuning. The accuracy score and confusion matrix are calculated and displayed. Then, a grid search is performed to find the best hyperparameters using ‘GridSearchCV’. The algorithm is retrained with the best hyperparameters, and the accuracy score and confusion matrix are displayed again.



	precision	recall	f1-score	support
0	0.84	0.94	0.89	12435
1	0.69	0.42	0.52	3846
accuracy			0.82	16281
macro avg	0.76	0.68	0.70	16281
weighted avg	0.80	0.82	0.80	16281

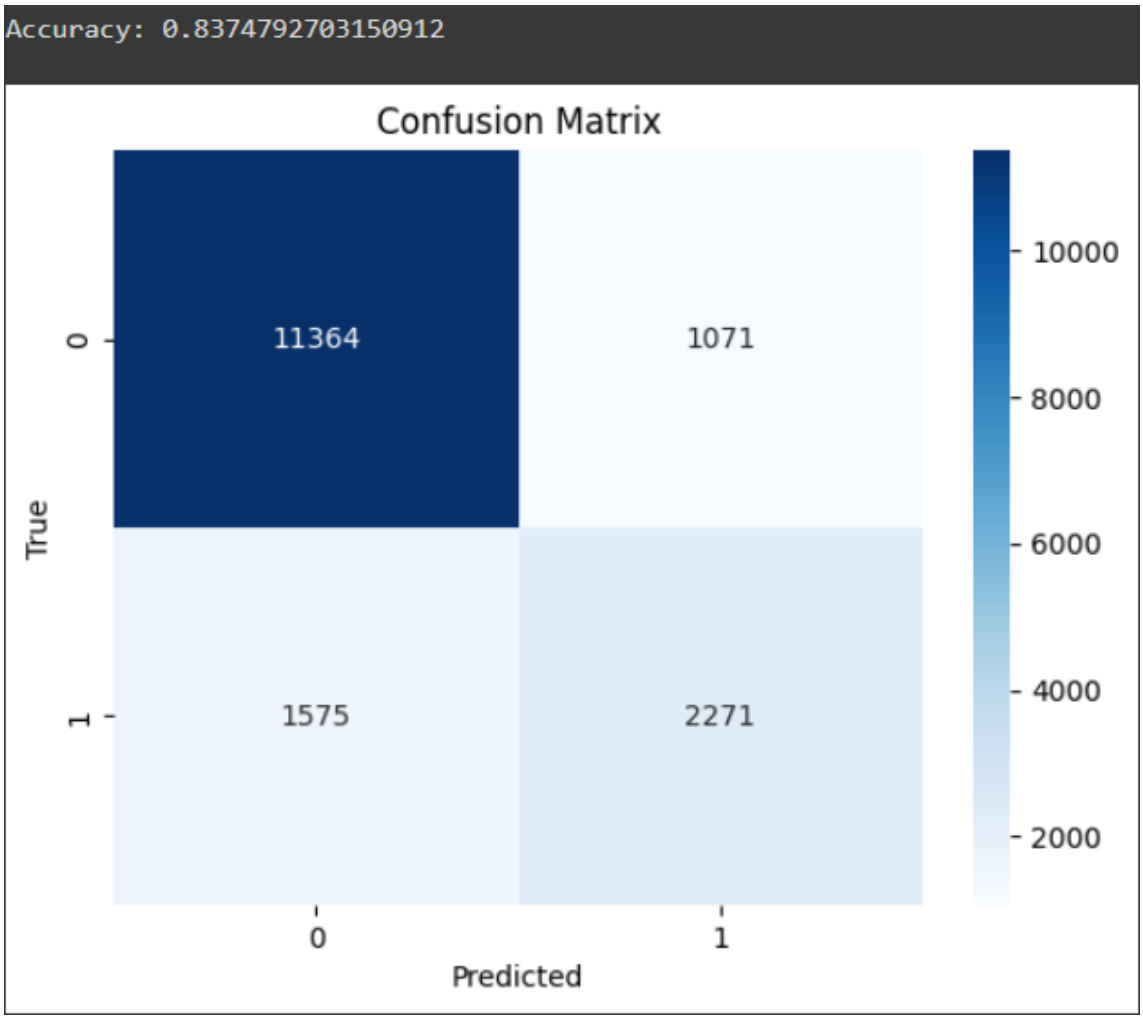
XGBoost:

XGBoost (Extreme Gradient Boosting) is implemented using the 'XGBClassifier' class from the xgboost library. A grid search is conducted to find the best hyperparameters by searching over a predefined parameter grid. The algorithm is trained with the best hyperparameters, and the accuracy score and confusion matrix are computed and displayed.



Random Forest Classifier:

Random Forest Classifier is used with the ‘RandomForestClassifier’ class from scikit-learn. The classifier is trained on the training data, and predictions are made on the test data. The accuracy score and confusion matrix are calculated and displayed.



Conclusion:

This machine learning project involved the classification of income based on various features. The data was explored, pre-processed, and sampled using oversampling and SMOTE techniques. Three algorithms were tested on the dataset: Logistic Regression, XGBoost, and Random Forest Classifier. The performance of each algorithm was evaluated using accuracy scores and confusion matrices. The best hyperparameters for Logistic Regression and XGBoost were found using grid search. The results provide insights into the performance of different algorithms and their suitability for the income classification task. Notably, the XGBoost algorithm get the best performance, achieving the highest accuracy and delivering the most accurate results among the tested algorithms.