**Income Prediction**

Income dataset splits into train and test datasets and contain 14 columns: age, workclass , final-weight , education , education-number , marital-status , occupation , relationship , race , sex , capital-gain , capital-loss , hours-per-week and native-country and a target column in the last called Income.

Preprocessing:

Data Cleansing:

* Remove rows that contain nulls and duplicates.
* Replace character ‘?’ with the mode of the column.
* Convert target column ‘Income’ to binary.

Outliers:

A picture containing diagram, plan, line, technical drawing

Description automatically generated

There are two choices for getting rid of it: Replacing outlier with mode of the column or Removing row that contain outlier.

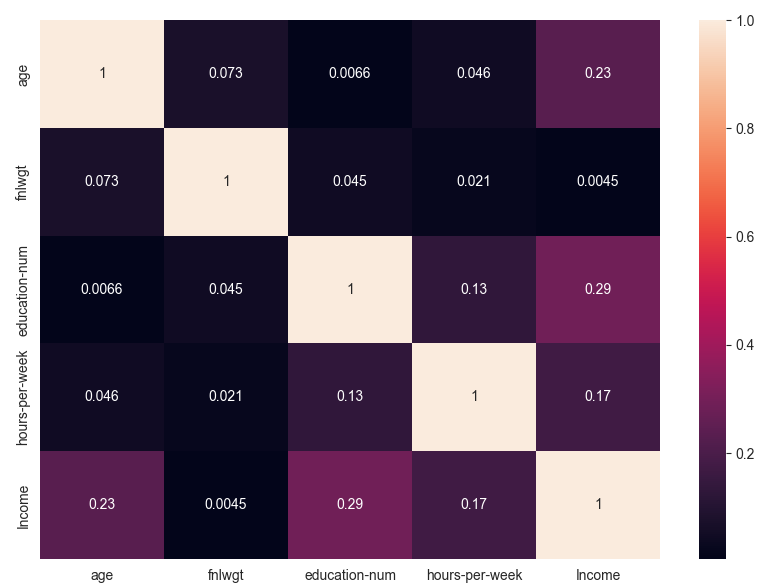
Feature Selection:

The columns will be removed from the train and test dataframes.

There are 4 techniques used for selecting features:

1. Univariate Selection using SelectKBest and the score\_func is f\_classif.
2. Feature Importance using ExtraTreesClassifier.
3. Recursive Feature Elimination using RFE and use LogisticRegression as a model.
4. Ignore Weak Correlation by ignoring features has weak correlation with target column.

After replacing outliers After removing outliers

There are 3 ways for selecting features:

1. Remove the following columns: ‘fnlwgt’, ‘education’, ‘relationship’, ‘race’.
2. Select common features from the previous techniques.
3. Select a technique from the previous techniques.

Convert strings:

To convert categorical columns to numeric columns, there are two ways:

1. Use Normalizer after using LabelEncoder.
2. Use StandardScaler after using LabelEncoder.

For good accuracy, you should choose the second way.

Data Balancing:

For balancing data, we use to ways:

1. RandomUnderSampler for under sampling.
2. RandomOverSampler for over sampling.

Training:

We use 4 models for training:

1. Logistic Regression.
2. Support Vector Classifier.
3. Decision Tree Classifier.
4. Random Forest Classifier.

By using GridSearchCV to get the best hyper parameters, the hyper parameters we give it to GridSearchCV for:

1. Logistic Regression, we use

* solver and its values are ‘lbfgs’ , ‘liblinear’
* Penalty and its values are ‘l1’ , ‘l2’
* C and its values are 100 , 10 , 1 , 0.1

1. Support Vector Classifier, we use

* kernel and its values are ‘poly’ , ‘rbf’
* max\_iter and its values are 5000 , 8000 , -1

1. Decision Tree Classifier, we use
   * max\_leaf\_nodes and its values from 10 to 30
   * criterion and its values are ‘gini’ , ‘entropy’ , ‘log\_loss’
2. Random Forest Classifier, we use
   * n\_estimators and its values are a range from 10 to 100 with step = 8
   * max\_depth and its values are a range from 5 to 20 with step = 3
   * criterion and its values are ‘gini’ , ‘entropy’ , ‘log\_loss’

Finally, by using some evaluation like classification\_report and confusion\_matrix, we get the best model.

A screenshot of a computer

Description automatically generated

|  |  |  |
| --- | --- | --- |
| marks | id | name |
|  | 2021170647 | يوسف محمود محمد طه |
|  | 2021170496 | محمود حسن سيد محمد |
|  | 2021170629 | يوسف إسماعيل ذكي |
|  | 2021170501 | مروان احمد محمد حسين |
|  | 2021170213 | زياد علي محمود علي |
|  | 2021170400 | كريم أحمد عبد المجيد السيد |
|  | 2021170527 | مصطفى مجدى احمد ابوالمجد |