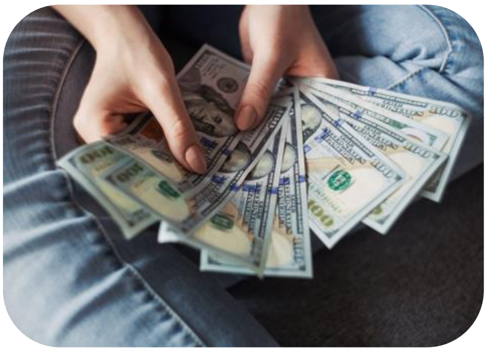
DS 861 - Data Mining and Advanced Statistical Methods for Business Analysts - Spring 2023

Project Report - Income Classification using Adult Census Data



Submitted by -

Saksham Motwani - 922988440

Under the supervision of Professor Minh Pham

Contents

Project Dataset and Objective

For this project, we examined the Census Income dataset [1] available at the UCI ML repository to predict whether an individual's annual income exceeds $50,000 based on various demographic and socio-economic attributes i.e. a binary classification problem. The dataset, extracted from the 1994 US census database, contains both numerical and categorical variables and 48,842 instances making it a moderately sized dataset suitable for training and testing machine learning models. The response variable is imbalanced, with about 75% of the instances being a part of the majority class (income <=$50K). First, we preprocess the data, followed by developing an understanding of the data by doing some exploratory analysis, and finally move on to classification tasks by implementing various machine learning models and their comparisons.

Dataset Description

Each entry in the dataset contains the following information about an individual

Feature	Type	Values	Feature description
age	Numerical	Positive integer between 17 and 90	Age of the individual
workclass	Categorical	State-gov, Self-emp-not-inc, Private, Federal-gov, Local-gov, Self-emp-inc, Without-pay, Never-worked	Represents the type of employment of the individual
fnlwgt	Numerical	Positive integer value	Final weight. A weight assigned by the census bureau. Allocate similar weights to people with similar demographic characteristics
education	Categorical	Bachelors, HS-grad, 11th, Masters, 9th, Some-college, Assoc-acdm, Assoc-voc, 7th-8th, Doctorate, Prof-school, 5th-6th, 10th, 1st-4th, Preschool, 12th	Highest level of education of an individual
education-num	Numerical	Positive integer between 1 and 16.	A label encoded version of education. 1 represents Preschool and 16 represents Doctorate
marital-status	Categorical	Never-married, Married-civ-spouse, Divorced, Married-spouse-absent, Separated, Married-AF-spouse, Widowed	Marital status of an individual
occupation	Categorical	Adm-clerical, Exec-managerial, Handlers-cleaners, Prof-specialty, Other-service, Sales, Craft-repair, Transport-moving, Farming-fishing, Machine-op-inspct, Tech-support, Protective-serv, Armed-Forces, Priv-house-serv	A general type of occupation of an individual
relationship	Categorical	Not-in-family, Husband, Wife, Own-child, Unmarried, Other-relative	Represents what this individual is relative to others.
race	Categorical	White, Black, Asian-Pac-Islander, Amer-Indian-Eskimo, Other	Race of an individual
sex	Categorical	Male, Female	Gender of an individual

capital-gain	Numerical	Non-negative integer value	Capital gains for an individual
capital-loss	Numerical	Non-negative integer value	Capital loss for an individual
hours-per-week	Numerical	Positive integer value from 1 to 99	Represents the hours an individual reported to work
native-country	Categorical	A list of 41 countries	Country of origin for an individual
income	Categorical	<=50k, >50k	Represents whether or not an individual makes more than $50K annually

Literature Review

Researchers have previously employed several machine learning models to make predictions about income levels, demonstrating a notable dedication exploring this dataset:

- In their study, Chockalingam et al. [2] extensively examined the Adult Dataset and applied various Machine Learning Models, including Logistic Regression, Stepwise Logistic Regression, Naive Bayes, Decision Trees, Extra Trees, k-Nearest Neighbor, SVM, Gradient Boosting, and six different configurations of Activated Neural Network. Additionally, they conducted a comprehensive comparative analysis to evaluate the predictive performance of these models.
- Lemon et al. [3] aimed to identify key features in the data to optimize the complexity of different machine learning models used in classification tasks.

We also found a lot of helpful references in Kaggle where people had used this dataset for classification purposes

- IPByrne's solution [4] where he performs feature engineering on the marital-status feature gave us ideas to recategorize some features where the same technique could be used to decrease complexity.

Data Preprocessing

1. Handling missing values

   Although the dataset initially did not seem to have any missing values, upon observing the unique values in every feature, it was noticed that 3 features, ‘workclass’, ‘occupation’, and ‘native-country’ had values (2799 instances, 2809 instances, and 857 instances respectively) marked by ‘?’. We naturally assumed these to be missing data and effectively dealt with it by first converting them into missing values and then dropping these instances. As a result, our dataset had 45222 observations and 15 features.

2. Handling duplicate entries

   The dataset had 47 duplicate observations and these were dropped too. After this, the dataset had 45175 observations and 15 features.

3. Feature engineering

   Since some features had a lot of categories, we decided to re-categorize some of these features into lesser categories in a way which ensured minimal information loss.

- Features - ‘education’ & ‘education-num’

There were 2 features that conveyed the education level of an individual. The feature ‘education’ had categories pertaining to the education of a person (ex: ‘Preschool’, ‘1st - 4th’, ‘Doctorate’) and the feature ‘education-num’ which was a label encoded form of the first feature. For instance, if an individual only studied until preschool, the corresponding value in the ‘education-num’ column was 1, all the way until the value 16 which meant an individual has completed their doctorate. ‘Education-num’ was **not** the number of years an individual has spent in their education.

‘Education-num’ was dropped, and ‘education’ was re-categorized as below:

Old Category	New Category	Logic
Preschool 1st-4th 5th-6th 7th-8th 9th 10th 11th 12th	School-dropout	Since individuals in these categories have not completed their high-school, clubbing them into School-dropouts makes sense.
HS-grad	High-school-grad	Just for better naming
Assoc-voc Assoc-acdm Some-college	Associate-degree	These are individuals who went to a community college but do not have bachelors degrees
Bachelors	Bachelors (No Change)	
Masters	Masters (No Change)	
Prof-school	Prof-school (No Change)	
Doctorate	Doctorate (No Change)	

- Feature - ‘marital-status’

Old Category	New Category	Logic
Married-civ-spouse Married-AF-spouse Married-spouse-absent Separated	Married	These individuals are legally married.
Never-married Divorced Widowed	Single	These individuals are legally single.

- Feature - ‘native-country’

41256 out of 45175 observations (91%) had native-country as US.

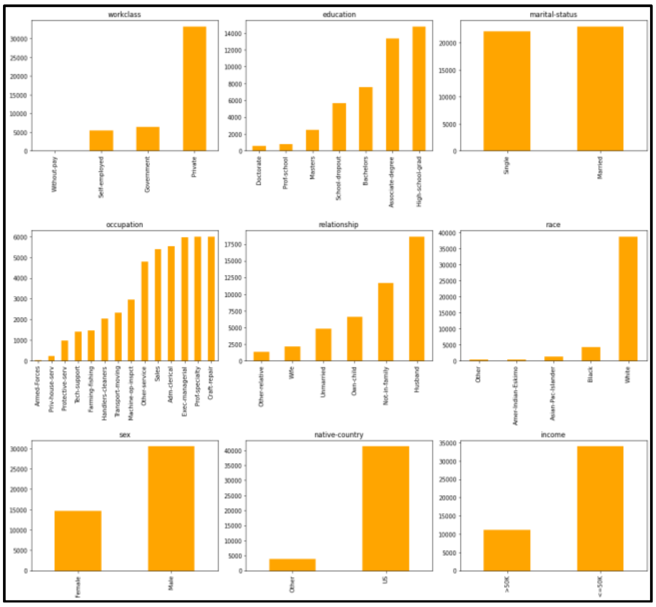
Old Category	New Category	Logic
United-States	US	Just for better naming
Philippines, Germany, Puerto-Rico, Canada, El-Salvador, India, Cuba, England, China, South, Jamaica, Italy, Dominican-Republic, Japan, Guatemala, Poland, Vietnam, Columbia, Haiti, Portugal, Taiwan, Iran, Greece, Nicaragua, Peru, Ecuador, France, Ireland, Hong, Thailand, Cambodia, Trinadad&Tobago, Laos, Yugoslavia, Outlying-US(Guam-USVI-etc), Scotland, Honduras, Hungary, Holand-Netherlands,Mexico	Other	

- Feature 'workclass'

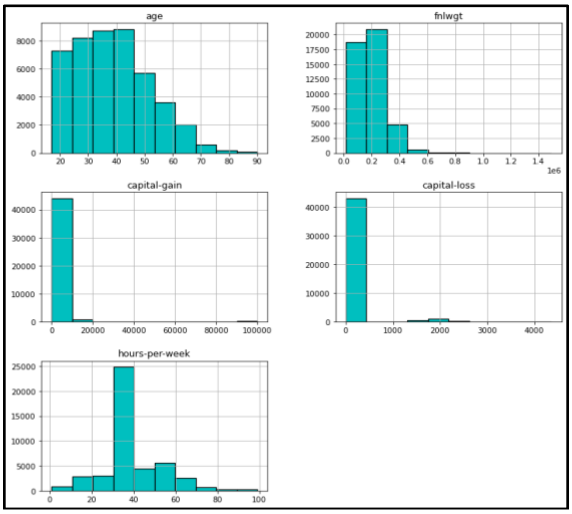
Old Category	New Category	Logic
Private	Private (No Change)	
Without-pay	Without-pay (No Change)	
Local-gov State-gov Federal-gov	Government	All the 3 older categories are government jobs.
Self-emp-not-inc Self-emp-inc	Self-employed	Both older categories are individuals who are self employed (some taking a salary out of their business and some not)

Exploratory Data Analysis

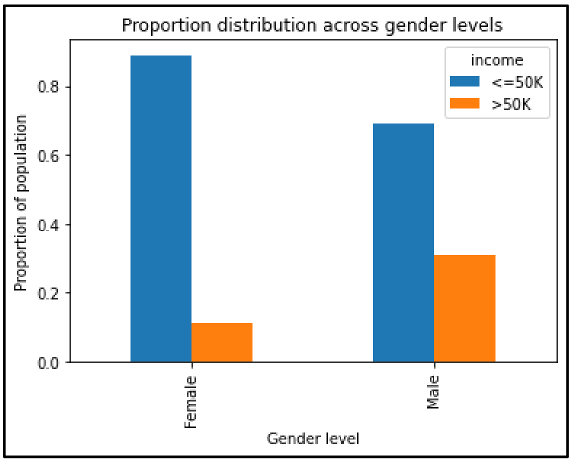
1. Distribution of categorical features



2. Distribution of numerical features

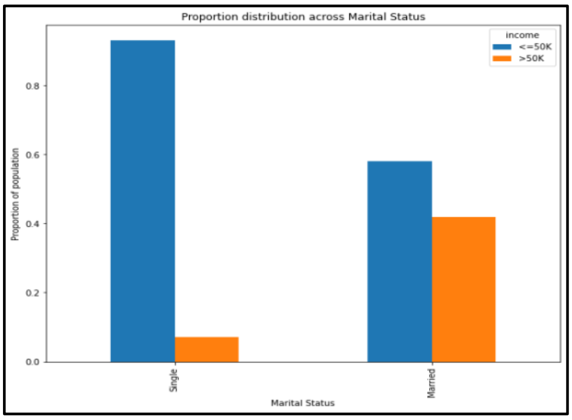


3. Gender v/s Income



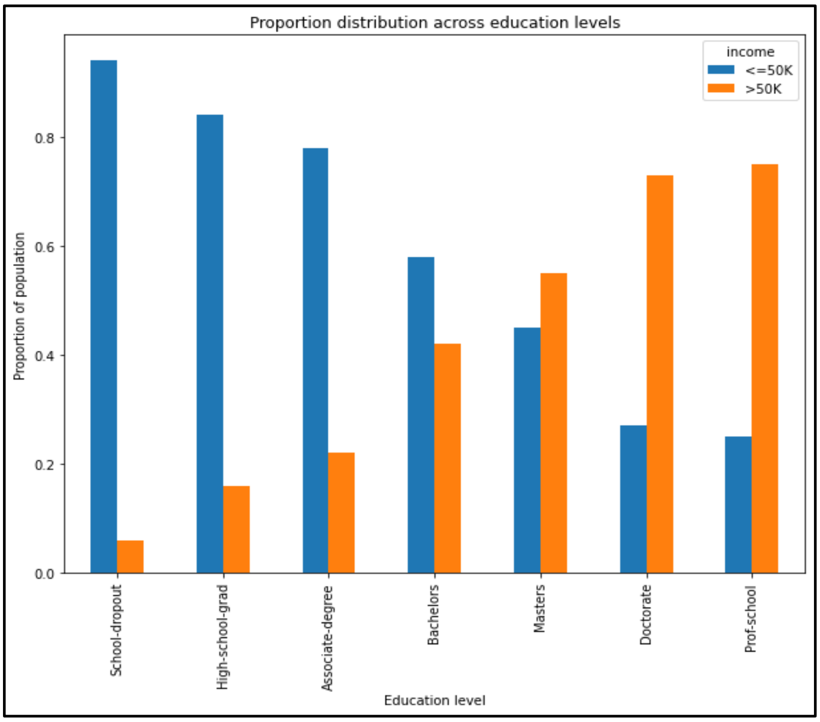
It can be observed that there exists an income gap between females and males. The proportion of males earning more than $50K per annum is more than double of their female counterparts.

4. Marital Status v/s Income



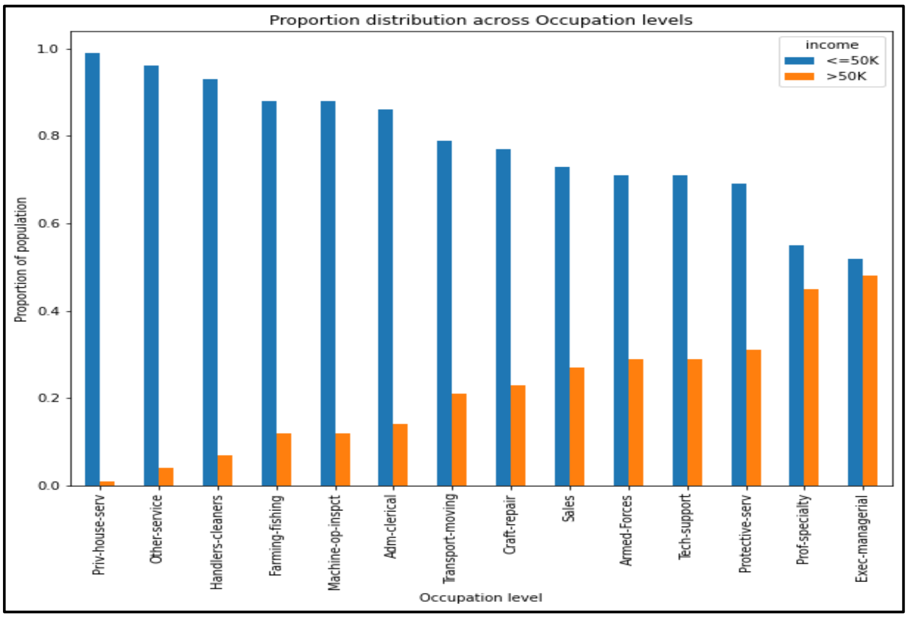
We observe that the proportion of married people who earn over $50K per annum is much higher than the proportion of single people who earn more than $50K per annum.

5. Education Level v/s Income



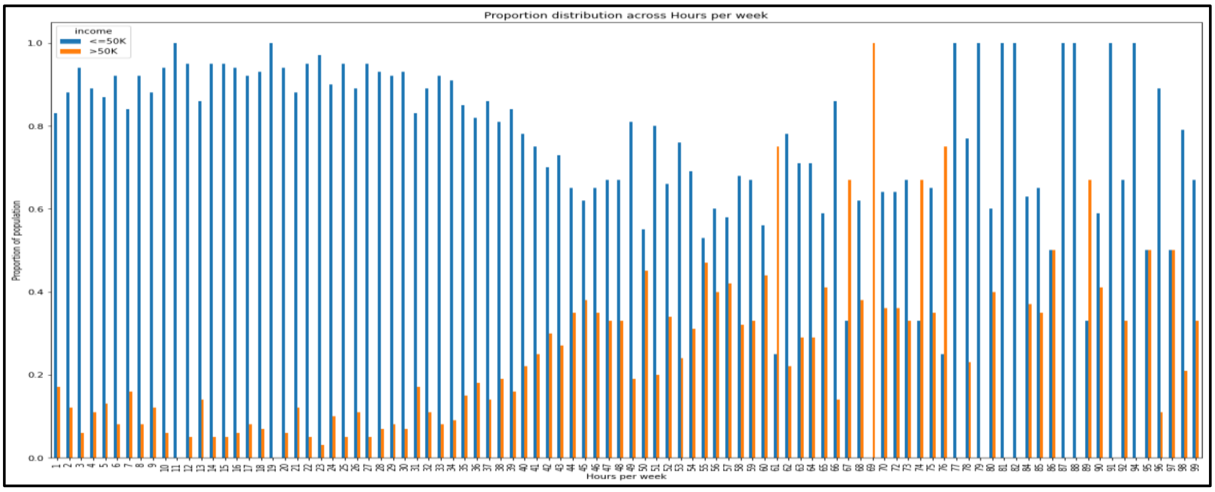
As expected, as the education level increases, so does the proportion of people who earn more than $50K a year. However, what is interesting is that only after a masters degree the proportion of people earning more than $50K a year is a majority. Another interesting statistic is to note that the proportion of people who earn more than $50K a year is almost the same for those who did a Doctorate v/s those who went to a professional school.

6. Occupation v/s Income



It is observed that for occupations such as Prof-specialty and Exec-managerial, the proportion of people whose salary is more than $50K a year is more than the same proportion for other occupations.

7. Hours per week v/s Income



It was anticipated that there would be a strong positive correlation between the number of hours worked per week and the percentage of the population earning over $50K annually, but this was not consistently observed in the graph. Specifically, there were instances (such as when individuals worked 77, 79, 81, 82, 87, or 88 hours per week) where nobody earned more than $50K per year. However, as we can see, the general trend is that the proportion of people earning more than $50K per year is increasing as they work longer hours.

Preparing data for Modeling - One-hot encoding categorical variables & data splitting

Dummy variables were created for each categorical variable (‘education’, ‘workclass’, ‘sex’, ‘race’, ‘native-country’, ‘occupation’, ‘marital-status’ and the response variable ‘income’) and a reference category was dropped. The reference categories that we chose to drop were based on easy interpretability.

Categorical Column	Reference Category Dropped
education	School-dropout
workclass	Government
sex	Female
race	Other
relationship	Unmarried
native-country	Other
occupation	Craft-repair
marital-status	Single
income	<=50K

We separate the predictors and the response variable, and split the data into training (80%) and testing. Also, we scale the numerical columns using standard scaler to improve the convergence for the Logistic Regression Model we build (only for the statsmodels one)

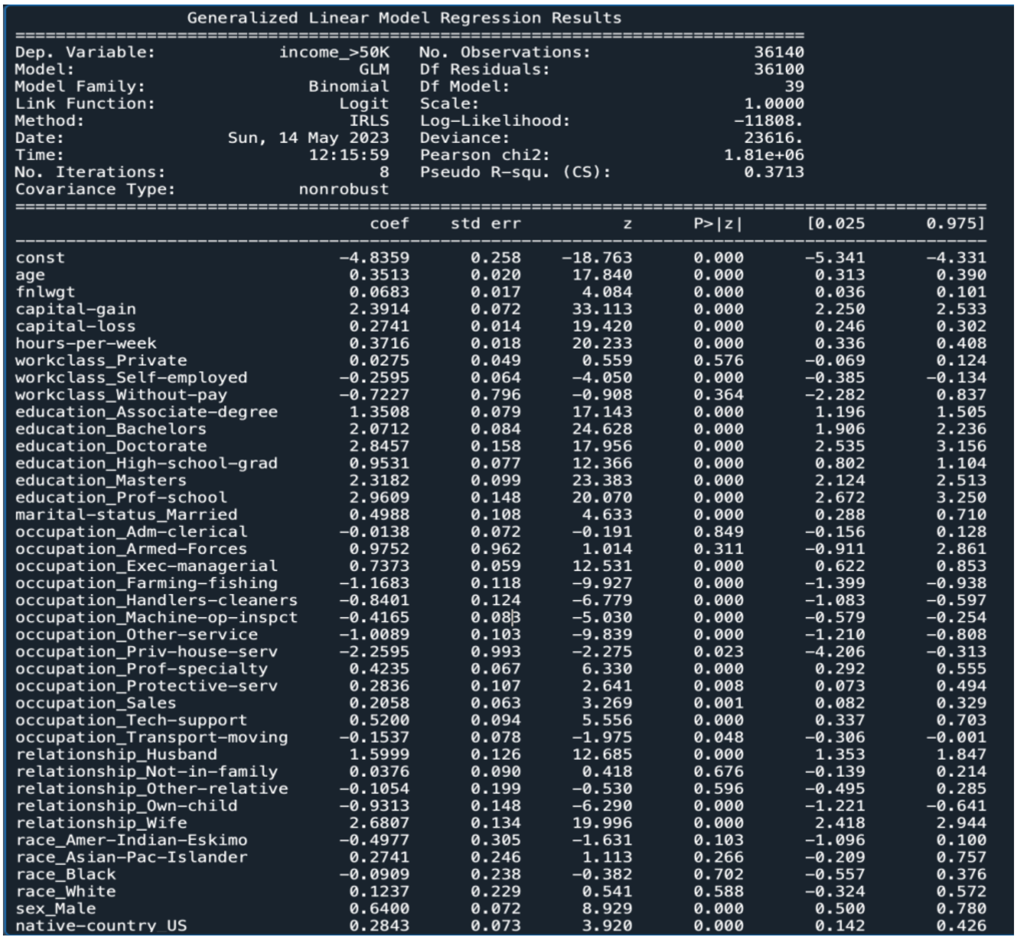
Models Trained

1. Logistic Regression

The first model we decided to train our dataset on is Logistic Regression. We decided to keep the Logistic regression model as our base to compare with the remaining three models based on performance evaluation metrics, especially the F1-score since our dataset is imbalanced.

- Using statsmodels (train-test split and standardised numerical features)

  We used statsmodels to have a better picture of coefficients and significant features.



```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:            income_>50K   No. Observations:                36140
Model:                            GLM   Df Residuals:                    36100
Model Family:                Binomial   Df Model:                           39
Link Function:                  Logit   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -11808.
Date:                Sun, 14 May 2023   Deviance:                       23616.
Time:                        12:15:59   Pearson chi2:                 1.81e+06
No. Iterations:                     8   Pseudo R-squ. (CS):             0.3713
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                           -4.8359      0.258    -18.763      0.000      -5.341      -4.331
age                              0.3513      0.020     17.840      0.000       0.313       0.390
fnlwgt                           0.0683      0.017      4.084      0.000       0.036       0.101
capital-gain                     2.3914      0.072     33.113      0.000       2.250       2.533
capital-loss                     0.2741      0.014     19.420      0.000       0.246       0.302
hours-per-week                   0.3716      0.018     20.233      0.000       0.336       0.408
workclass_Private                0.0275      0.049      0.559      0.576      -0.069       0.124
workclass_Self-employed         -0.2595      0.064     -4.050      0.000      -0.385      -0.134
workclass_Without-pay           -0.7227      0.796     -0.908      0.364      -2.282       0.837
education_Associate-degree       1.3508      0.079     17.143      0.000       1.196       1.505
education_Bachelors              2.0712      0.084     24.628      0.000       1.906       2.236
education_Doctorate              2.8457      0.158     17.956      0.000       2.535       3.156
education_High-school-grad       0.9531      0.077     12.366      0.000       0.802       1.104
education_Masters                2.3182      0.099     23.383      0.000       2.124       2.513
education_Prof-school            2.9609      0.148     20.070      0.000       2.672       3.250
marital-status_Married           0.4988      0.108      4.633      0.000       0.288       0.710
occupation_Adm-clerical         -0.0138      0.072     -0.191      0.849      -0.156       0.128
occupation_Armed-Forces          0.9752      0.962      1.014      0.311      -0.911       2.861
occupation_Exec-managerial       0.7373      0.059     12.531      0.000       0.622       0.853
occupation_Farming-fishing      -1.1683      0.118     -9.927      0.000      -1.399      -0.938
occupation_Handlers-cleaners    -0.8401      0.124     -6.779      0.000      -1.083      -0.597
occupation_Machine-op-inspct    -0.4165      0.083     -5.030      0.000      -0.579      -0.254
occupation_Other-service        -1.0089      0.103     -9.839      0.000      -1.210      -0.808
occupation_Priv-house-serv      -2.2595      0.993     -2.275      0.023      -4.206      -0.313
occupation_Prof-specialty        0.4235      0.067      6.330      0.000       0.292       0.555
occupation_Protective-serv       0.2836      0.107      2.641      0.008       0.073       0.494
occupation_Sales                 0.2058      0.063      3.269      0.001       0.082       0.329
occupation_Tech-support          0.5200      0.094      5.556      0.000       0.337       0.703
occupation_Transport-moving     -0.1537      0.078     -1.975      0.048      -0.306      -0.001
relationship_Husband             1.5999      0.126     12.685      0.000       1.353       1.847
relationship_Not-in-family       0.0376      0.090      0.418      0.676      -0.139       0.214
relationship_Other-relative     -0.1054      0.199     -0.530      0.596      -0.495       0.285
relationship_Own-child          -0.9313      0.148     -6.290      0.000      -1.221      -0.641
relationship_Wife                2.6807      0.134     19.996      0.000       2.418       2.944
race_Amer-Indian-Eskimo         -0.4977      0.305     -1.631      0.103      -1.096       0.100
race_Asian-Pac-Islander          0.2741      0.246      1.113      0.266      -0.209       0.757
race_Black                      -0.0909      0.238     -0.382      0.702      -0.557       0.376
race_White                       0.1237      0.229      0.541      0.588      -0.324       0.572
sex_Male                         0.6400      0.072      8.929      0.000       0.500       0.780
native-country_US                0.2843      0.073      3.920      0.000       0.142       0.426
```

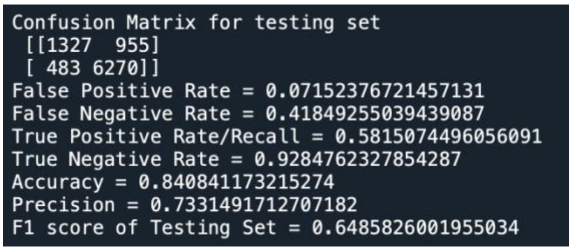
Insignificant features (p-value > 0.05): workclass_private, workclass_Without-pay, occupation_Adm-clerical, occupation_Armed-Forces, relationship_Not-in-family, relationship_Other-relative, race_Amer-Indian-Eskimo, race_Asian-Pac-Islander, race_Black, race_White. It was interesting to note that all races were insignificant.

Some (not all) interpretations of significant features:
– Odds of income being >$50K increase by 42% for each 1 standard deviation increase in age. (since we scaled the numerical features)
– For men, the odds of income being >$50K increase by 90% (compared to women).
– Odds of income being >$50K increase by 159%, 286%, 693%, 915%, 1831%, 1621% for HS grad, Assoc. degree, Bachelors, Masters, Prof-school, and Doctorate respectively (compared to school dropout)
– For married people, the odds of income being >$50K increase by 65% (w.r.t single).

- Using sklearn (train-test split and standardised numerical features)

  We also used sklearn model library logistic regression model to get a better understanding of model metrics especially confusion matrix and F1-score.

  Results & Interpretations on the testing set:



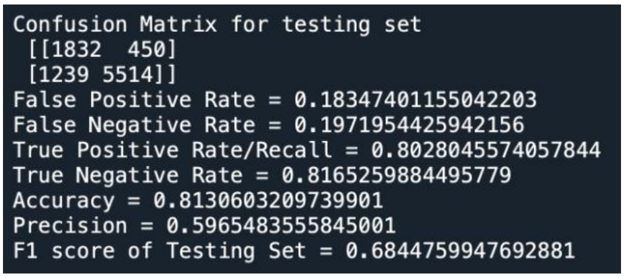
```
Confusion Matrix for testing set
 [[1327  955]
 [ 483 6270]]
False Positive Rate = 0.07152376721457131
False Negative Rate = 0.41849255039439087
True Positive Rate/Recall = 0.5815074496056091
True Negative Rate = 0.9284762327854287
Accuracy = 0.840841173215274
Precision = 0.7331491712707182
F1 score of Testing Set = 0.6485826001955034
```

  – FN (955) > FP (483): Higher tendency to miss positive samples than incorrectly classify negative samples.
  – Low FPR
  – High FNR - difficulty identifying positives
  – **Recall** (proportion of actual positives that are correctly identified as positive by the model) - **0.58** (not a high recall rate)
  – Accuracy - 84% - may not be a good metric since class dist. is imbalanced
  – **Precision** (proportion of positive predictions that are correctly classified by the model) - **0.733** (decent)
  – F1 score - 0.648 (can be improved)

- Using sklearn (with 5-Fold CV, threshold tuning and standardising all features)

  In order to improve the model's performance, we decided to tune the threshold parameter of logistic regression. Moreover, since the response variable is imbalanced, we tune the threshold value to remove the bias towards the majority class. We did this using 5-Fold cross-validation.
  50 candidate values for threshold were taken from 0 to 1.
  Best threshold value that maximised the validation set F1 score was **0.28**

  Results & Interpretations on the testing set:



```
Confusion Matrix for testing set
 [[1832  450]
 [1239 5514]]
False Positive Rate = 0.18347401155042203
False Negative Rate = 0.1971954425942156
True Positive Rate/Recall = 0.8028045574057844
True Negative Rate = 0.8165259884495779
Accuracy = 0.8130603209739901
Precision = 0.5965483555845001
F1 score of Testing Set = 0.6844759947692881
```

  – Performed better than the previous model (in terms of F1 score)
  – **Recall** is **higher (0.803)** → Better at correctly identifying a higher proportion of positive cases.
  – **Precision** is **slightly lower (0.597)** → This is expected because the threshold value is tuned to optimise F1 score.
  – **F1 score** is **significantly higher (0.684)** → Overall a better performing model.

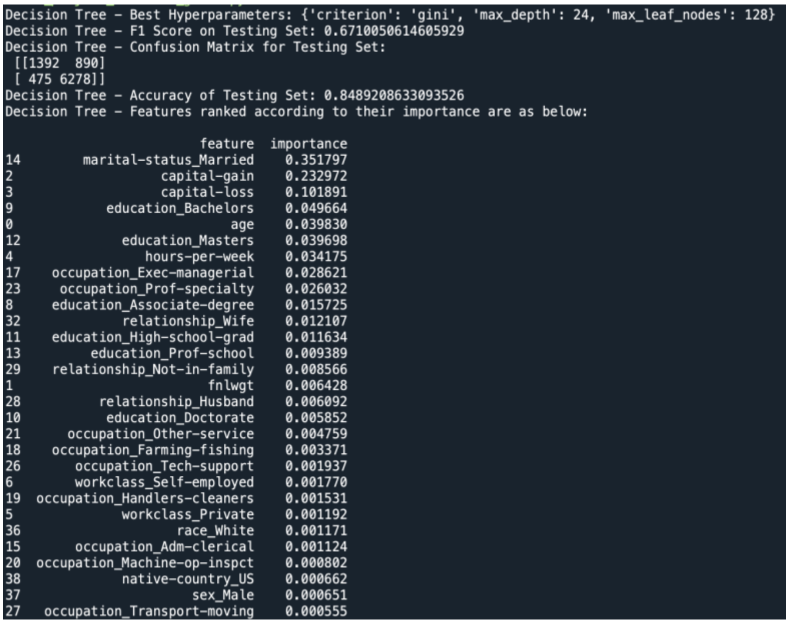
2. Decision Tree Classifier (with 5-Fold Cross Validation)

The second model we decided to train our dataset on is the Decision tree classifier. Decision Tree classifier is one of the simplest tree machine learning algorithms which tries to classify data points based by creating trees.The idea behind Decision Trees is that you use the dataset features to create yes/no questions and continually split the dataset until you isolate all data points belonging to each class. For the decision tree model we decided to tune three parameters while training the model.

Hyperparameters for tuning:

- **max_depth**: no. of levels. limits the number of splits and prevents overfitting. The deeper the tree, the more splits it has and it captures more information about the data.
  - Candidate values - np.arange(2,50,2)
- **max_leaf_nodes**: sets max. leaf nodes in the tree. Limits tree complexity by allowing the branches of a tree to have varying depths and prevents overfitting. By controlling the number of leaf nodes, we can keep the tree's structure simpler and easier to interpret.
  - Candidate values: [2, 4, 8, 16, 32, 64, 128, 256, 512, 1024]
- **criterion**: measures quality of split at each node.
  - Candidate values: gini, entropy.
    Gini impurity is the measure of variance across the different classes. Entropy is a measure of chaos within the node. And chaos, in the context of decision trees, is having a node where all classes are equally present in the data.

Results on the testing set:



```
Decision Tree - Best Hyperparameters: {'criterion': 'gini', 'max_depth': 24, 'max_leaf_nodes': 128}
Decision Tree - F1 Score on Testing Set: 0.6710050614605929
Decision Tree - Confusion Matrix for Testing Set:
 [[1392  890]
 [ 475 6278]]
Decision Tree - Accuracy of Testing Set: 0.8489208633093526
Decision Tree - Features ranked according to their importance are as below:

                        feature  importance
14       marital-status_Married    0.351797
2                  capital-gain    0.232972
3                  capital-loss    0.101891
9           education_Bachelors    0.049664
0                           age    0.039830
12            education_Masters    0.039698
4                hours-per-week    0.034175
17    occupation_Exec-managerial    0.028621
23     occupation_Prof-specialty    0.026032
8     education_Associate-degree    0.015725
32            relationship_Wife    0.012107
11    education_High-school-grad    0.011634
13        education_Prof-school    0.009389
29   relationship_Not-in-family    0.008566
1                        fnlwgt    0.006428
28         relationship_Husband    0.006092
10          education_Doctorate    0.005852
21     occupation_Other-service    0.004759
18    occupation_Farming-fishing    0.003371
26      occupation_Tech-support    0.001937
6       workclass_Self-employed    0.001770
19  occupation_Handlers-cleaners    0.001531
5             workclass_Private    0.001192
36                   race_White    0.001171
15      occupation_Adm-clerical    0.001124
20  occupation_Machine-op-inspct    0.000802
38            native-country_US    0.000662
37                     sex_Male    0.000651
27   occupation_Transport-moving    0.000555
```

– **F1 score → 0.671**
– Performed worse than Logistic Regression (in terms of F-1 score)
– **Best hyperparameters** (that maximize validation set F1 score)
→ criterion: gini, max_depth: 24, max_leaf_nodes: 128
– **Imp features** → marital-status_Married, capital-gain, capital-loss

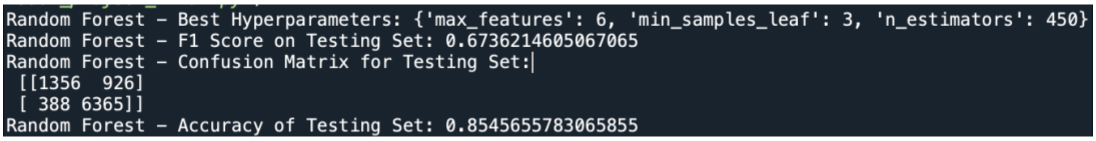
3. Random Forest Classifier (with 5-Fold Cross Validation)

Random forests are supervised machine learning models that train multiple decision trees and integrate the results by averaging them. Each decision tree makes various kinds of errors, and upon averaging their results, many of these errors are counterbalanced.

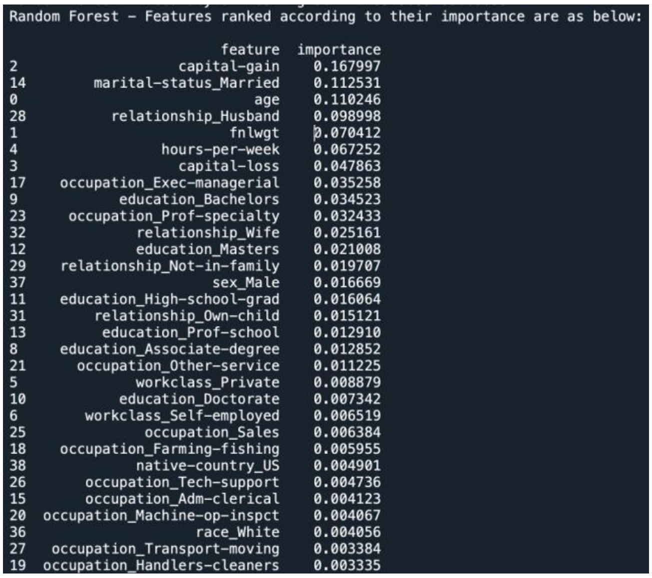
Hyperparameters for tuning:

- **n_estimators**: The number of decision trees to be included in the ensemble. The deeper the tree, it's better to try a higher number of estimators.
  - Candidate values - np.arange(100, 500, 50)
- **max_features**: Determines the maximum number of features to consider when looking for the best split at each node of the decision tree. It controls the randomness and diversity among the decision trees in the ensemble. It helps to prevent overfitting and controlling model complexity and computational efficiency.
  - Candidate values: np.arange(1, 7, 1)
- **min_samples_leaf**: Determines the minimum number of samples required to be at a leaf node. If the number of samples at a node falls below this threshold, the node is not split further, and it becomes a leaf node. Tuning this parameter helps prevent overfitting and helps in handling an imbalanced dataset.
  - Candidate values: np.arange(2, 5, 1)

Results on the testing set:



```
Random Forest - Best Hyperparameters: {'max_features': 6, 'min_samples_leaf': 3, 'n_estimators': 450}
Random Forest - F1 Score on Testing Set: 0.6736214605067065
Random Forest - Confusion Matrix for Testing Set:
 [[1356  926]
 [ 388 6365]]
Random Forest - Accuracy of Testing Set: 0.8545655783065855
```



```
Random Forest - Features ranked according to their importance are as below:

                    feature  importance
2              capital-gain    0.167997
14     marital-status_Married    0.112531
0                       age    0.110246
28       relationship_Husband    0.098998
1                    fnlwgt    0.070412
4            hours-per-week    0.067252
3              capital-loss    0.047863
17   occupation_Exec-managerial    0.035258
9          education_Bachelors    0.034523
23    occupation_Prof-specialty    0.032433
32         relationship_Wife    0.025161
12          education_Masters    0.021008
29   relationship_Not-in-family    0.019707
37                 sex_Male    0.016669
11   education_High-school-grad    0.016064
31      relationship_Own-child    0.015121
13       education_Prof-school    0.012910
8    education_Associate-degree    0.012852
21    occupation_Other-service    0.011225
5         workclass_Private    0.008879
10       education_Doctorate    0.007342
6      workclass_Self-employed    0.006519
25         occupation_Sales    0.006384
18   occupation_Farming-fishing    0.005955
38        native-country_US    0.004901
26     occupation_Tech-support    0.004736
15     occupation_Adm-clerical    0.004123
20  occupation_Machine-op-inspct    0.004067
36               race_White    0.004056
27   occupation_Transport-moving    0.003384
19  occupation_Handlers-cleaners    0.003335
```

– **F1 score → 0.6736**

– Performed worse than Logistic Regression but better than Decision Tree (in terms of F-1 score)

– **Best hyperparameters** (that maximize validation set F1 score)

→ n_estimators: 450, max_features: 6, min_samples_leaf: 3

– **Imp features** → capital-gain, marital-status_Married, age

4. Gradient Boosting Tree Classifier (with 5-Fold Cross Validation)

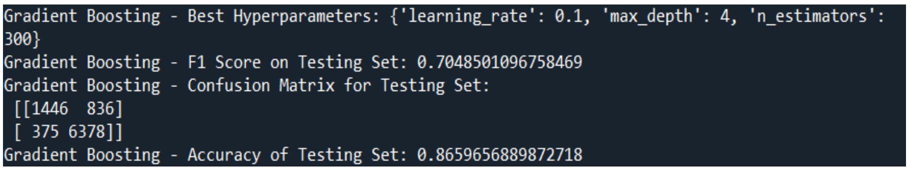
Gradient Boosting Tree Classifier, also known as Gradient Boosted Decision Tree (GBDT) is a popular machine learning algorithm that combines the concepts of gradient boosting and decision trees for classification tasks.

In GBDT, an ensemble of decision trees is sequentially constructed, where each subsequent tree tries to correct the mistakes made by the previous trees. The algorithm learns by minimising a loss function using gradient descent optimization. It starts with an initial prediction (often the mean or a constant value) and iteratively improves the predictions by adding new trees to the ensemble

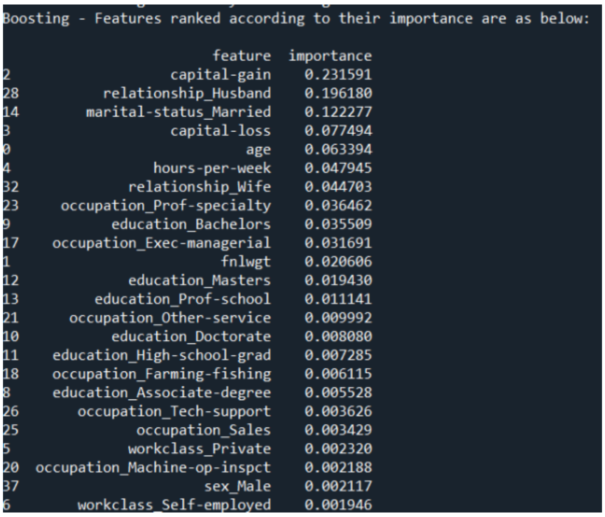
Hyperparameters for tuning:

- **n_estimators**: The number of decision trees to be included in the ensemble. The deeper the tree, it's better to try a higher number of estimators.
    - Candidate values - [100,200,300]
- **max_depth**: Number of levels. Limits the number of splits and prevents overfitting. The deeper the tree, the more splits it has and it captures more information about the data.
    - Candidate values - [2,3,4]
- **learning_rate**: Controls the contribution of each tree in the boosting process. It determines the step size at which the gradient boosting algorithm adjusts the predictions of the ensemble model to minimise the loss function. The learning rate serves as a regularisation parameter in gradient boosting. It helps control overfitting by preventing the boosting algorithm from learning the training data too quickly or being too sensitive to noise
    - Candidate values: [0.01, 0.05, 0.1]

Results on the testing set:



```
Gradient Boosting - Best Hyperparameters: {'learning_rate': 0.1, 'max_depth': 4, 'n_estimators':
300}
Gradient Boosting - F1 Score on Testing Set: 0.7048501096758469
Gradient Boosting - Confusion Matrix for Testing Set:
 [[1446  836]
 [ 375 6378]]
Gradient Boosting - Accuracy of Testing Set: 0.8659656889872718
```



```
Boosting - Features ranked according to their importance are as below:

                       feature  importance
2                 capital-gain    0.231591
28        relationship_Husband    0.196180
14      marital-status_Married    0.122277
3                 capital-loss    0.077494
0                          age    0.063394
4               hours-per-week    0.047945
32           relationship_Wife    0.044703
23    occupation_Prof-specialty    0.036462
9           education_Bachelors    0.035509
17   occupation_Exec-managerial    0.031691
1                       fnlwgt    0.020606
12            education_Masters    0.019430
13        education_Prof-school    0.011141
21     occupation_Other-service    0.009992
10          education_Doctorate    0.008080
11   education_High-school-grad    0.007285
18   occupation_Farming-fishing    0.006115
8    education_Associate-degree    0.005528
26      occupation_Tech-support    0.003626
25             occupation_Sales    0.003429
5            workclass_Private    0.002320
20 occupation_Machine-op-inspct    0.002188
37                    sex_Male    0.002117
6       workclass_Self-employed    0.001946
```

– **F1 score → 0.705**. This method **performed the best** out of all the other models.
– **Best hyperparameters** (that maximize validation set F1 score)
→ learning_rate: 0.1, max_depth: 4, n_estimators: 300
– **Imp features** → capital-gain, relationship_Husband, marital-status_Married

Overall Results & Conclusion

Overall, Gradient Boosting Classifier outperforms all the other models in terms of both F1 score and Accuracy. Further improvements can be made by fine tuning the hyperparameters.

Model	Accuracy (Testing Set)	F1 Score (Testing Set)
Logistic Regression (5-Fold CV, tuned threshold)	0.81306	0.684476
Decision Tree Classifier (5-Fold CV, tuned max_depth, max_leaf_nodes, criterion)	0.848921	0.671005
Random Forest Classifier (5-Fold CV, tuned n_estimators, max_features, min_samples_leaf)	0.854565	0.673621
Gradient Boosting Classifier (5-Fold CV, tuned n_estimators, learning_rate, max_depth)	0.865966	0.70485

Challenges

1. Handling categorical variables: We were confused about which technique to use (Label Encoding/One hot encoding) as some categorical variables had a high number of unique values and one hot encoding was leading to 107 features, which would have been tough to interpret. So we decided to recategorize (bucket) some categories based on logic, with minimal loss of information/meaning.

2. Running GBT with > 10 candidate values for each of three hyperparameters due to computational and time limitations. Despite the model running an entire night (>12 hours) and trying Google Colab (which has a dedicated GPU) we still did not get a result. So we had to restrict candidate values to 2 or 3 and over a smaller range of values.

Learnings

1. Data preprocessing takes up a large time: About 70-80% of time we spent on the project.
2. We got a chance to apply learnings from the course by training 4 models and tuning hyperparameters for each.
3. Working on assignments where direction was given v/s taking your own direction are completely different in terms of difficulty.

Things we would have done if we had more time

1. PCA to reduce dimensionality - Analyzing PCA variance ratios and dropping features step-wise with the least ratios.
2. Bagging and KNN.
3. Regularization methods to our log regression model.
4. Tuned a higher range of hyperparameters for Random Forest and Gradient Boosting.

References

[1] https://archive.ics.uci.edu/ml/datasets/adult

[2] Chockalingam, V., Shah, S., & Shaw, R. (2017). *Income Classification using Adult Census Data*. Semantic Scholar. https://www.semanticscholar.org/paper/Income-Classification-using-Adult-Census-Data-(-CSE-Chockalingam-Shah/3dd5e9f335511efbb81d65f1d6d4995019f8b5fd

[3] Lemon, C., Zelazo, C., &amp; Mulakaluri, K. (2015). Predicting if income exceeds $50,000 per year based on 1994 US Census Data with Simple Classification Techniques. UC San Diego Computer Science and Engineering. https://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/048.pdf

[4] https://www.kaggle.com/code/ipbyrne/income-prediction-84-369-accuracy