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....................................................................................................3

Dataset Description....................................................................................................................3

Literature Review.......................................................................................................................4

Data Preprocessing

Handling Missing Values...............................................................................................4

Handling Duplicate Entries............................................................................................4

Feature Engineering.......................................................................................................4

Exploratory Data Analysis.........................................................................................................6

Preparing data for Modeling......................................................................................................9

Models Trained

Logistic Regression......................................................................................................10

Decision Tree...............................................................................................................12

Random Forest.............................................................................................................13

Gradient Boosting........................................................................................................14

Overall Results and Conclusion...............................................................................................15

Challenges................................................................................................................................15

Learnings..................................................................................................................................15

If we had more time..................................................................................................................15

References................................................................................................................................16

**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.

1. 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. |

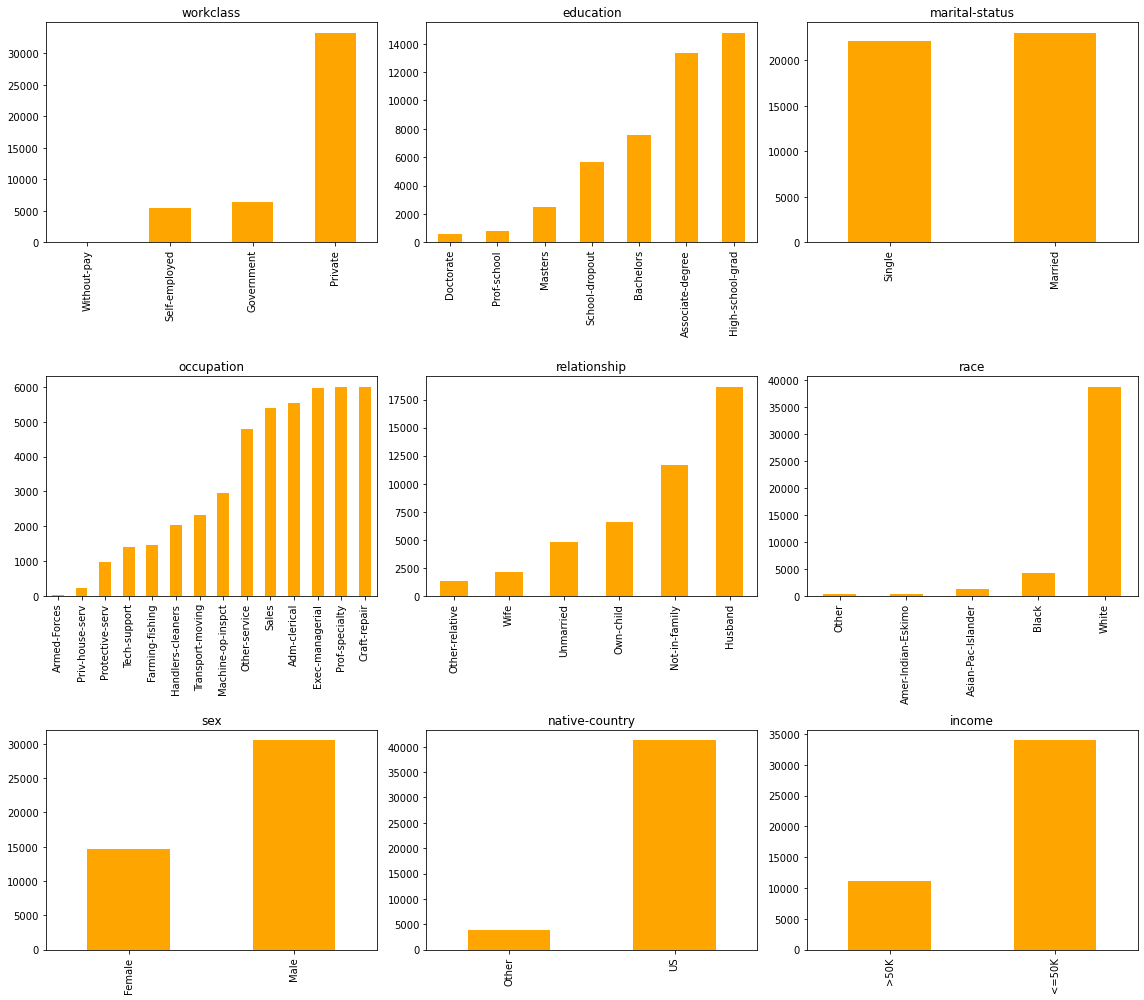
* Feature - ‘native-country’  
    
  41256 out of 45175 observations (91%) had native-country as US.

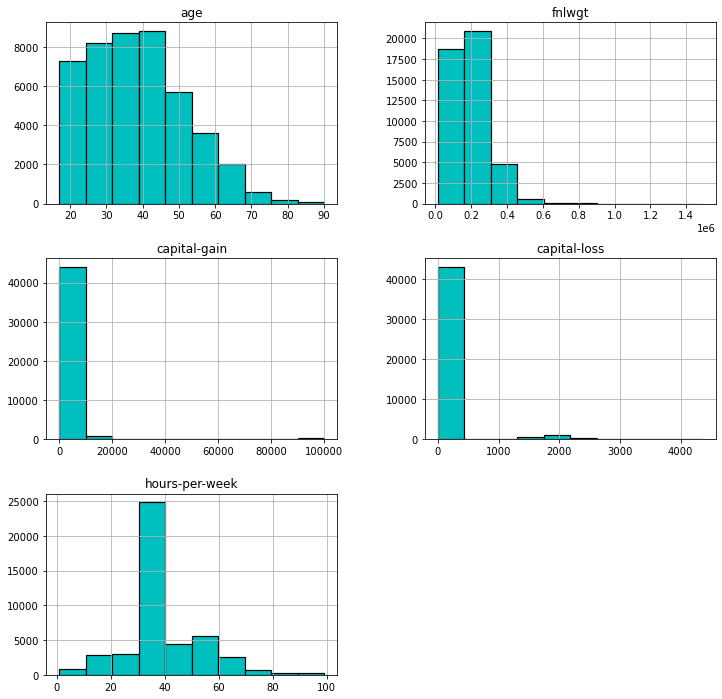
|  |  |  |
| --- | --- | --- |
| **Old Category** | **New Category** | **Logic** |
| United-States | US | Just for better naming |
| Philippines, Germany, Puerto-Rico, Canada, El-Salvador, India, Cuba, England, China, South, Jamaica, Italy, Dominican-Republic, Japan, Guatemala, Poland, Vietnam, Columbia, 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’

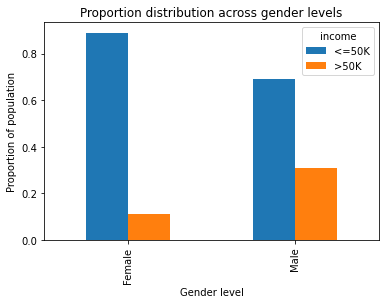
|  |  |  |
| --- | --- | --- |
| **Old Category** | **New Category** | **Logic** |
| Private | Private (No Change) |  |
| Without-pay | Without-pay (No Change) |  |
| Local-gov  State-gov  Federal-gov | Government | All the 3 older categories are government jobs. |
| Self-emp-not-inc  Self-emp-inc | 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

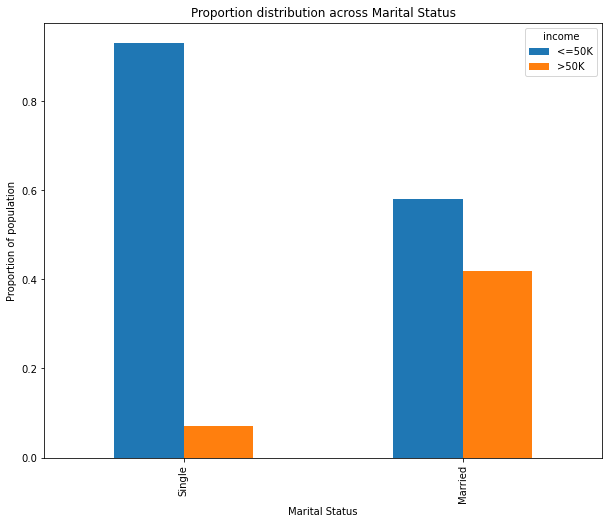


1. Gender v/s Income

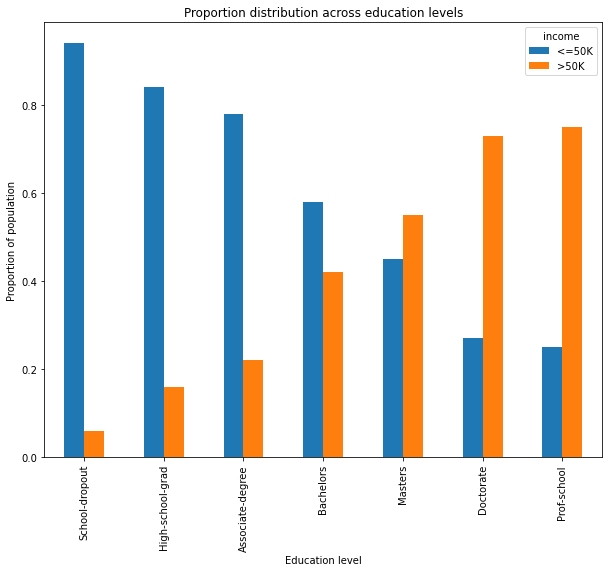
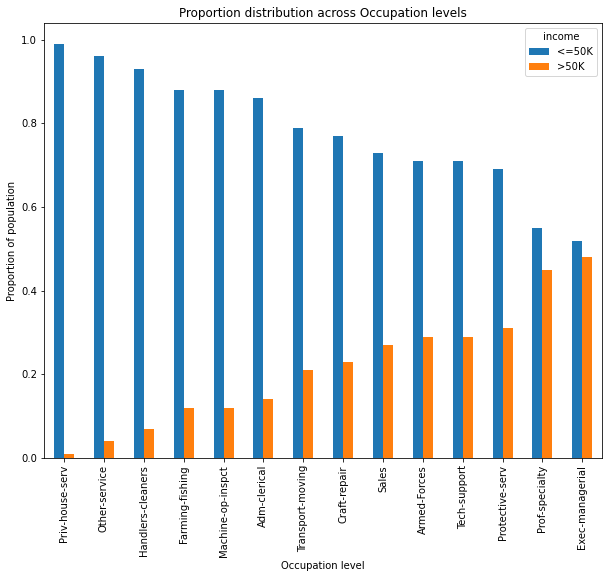
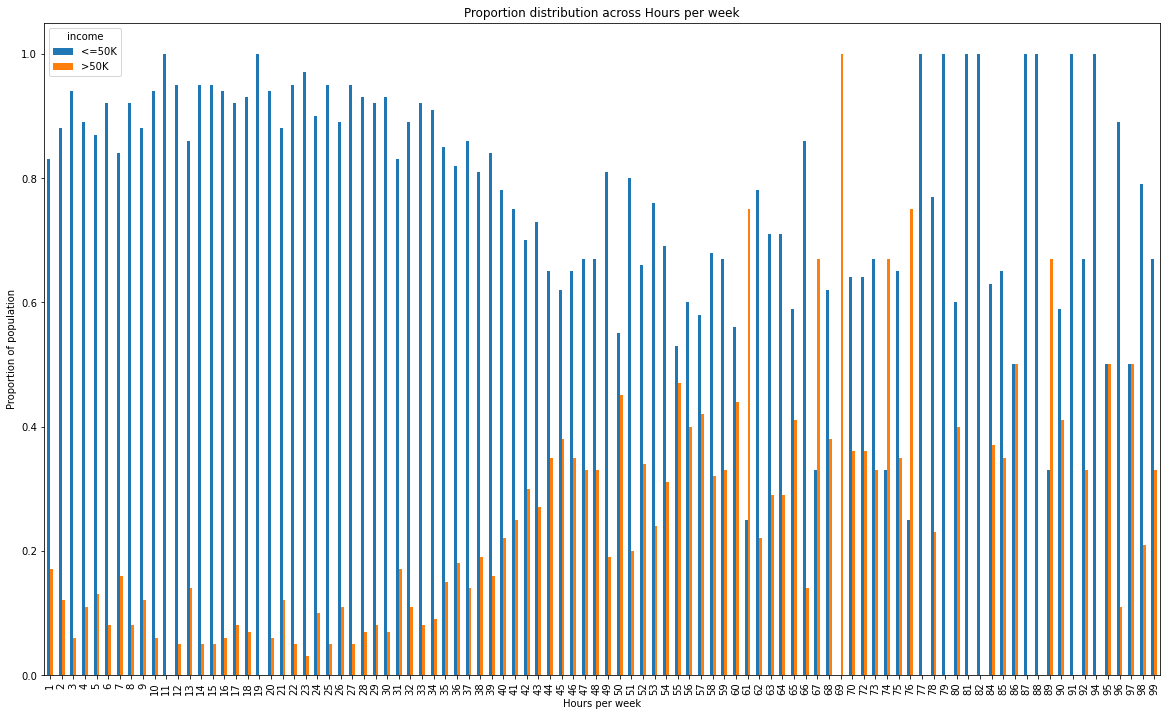


It can be observed that there exists an income gap between females and males. The proportion of males earning more than $50K per annum is more than double of their female counterparts.

1. 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.

1. 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.
2. 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.
3. 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 |

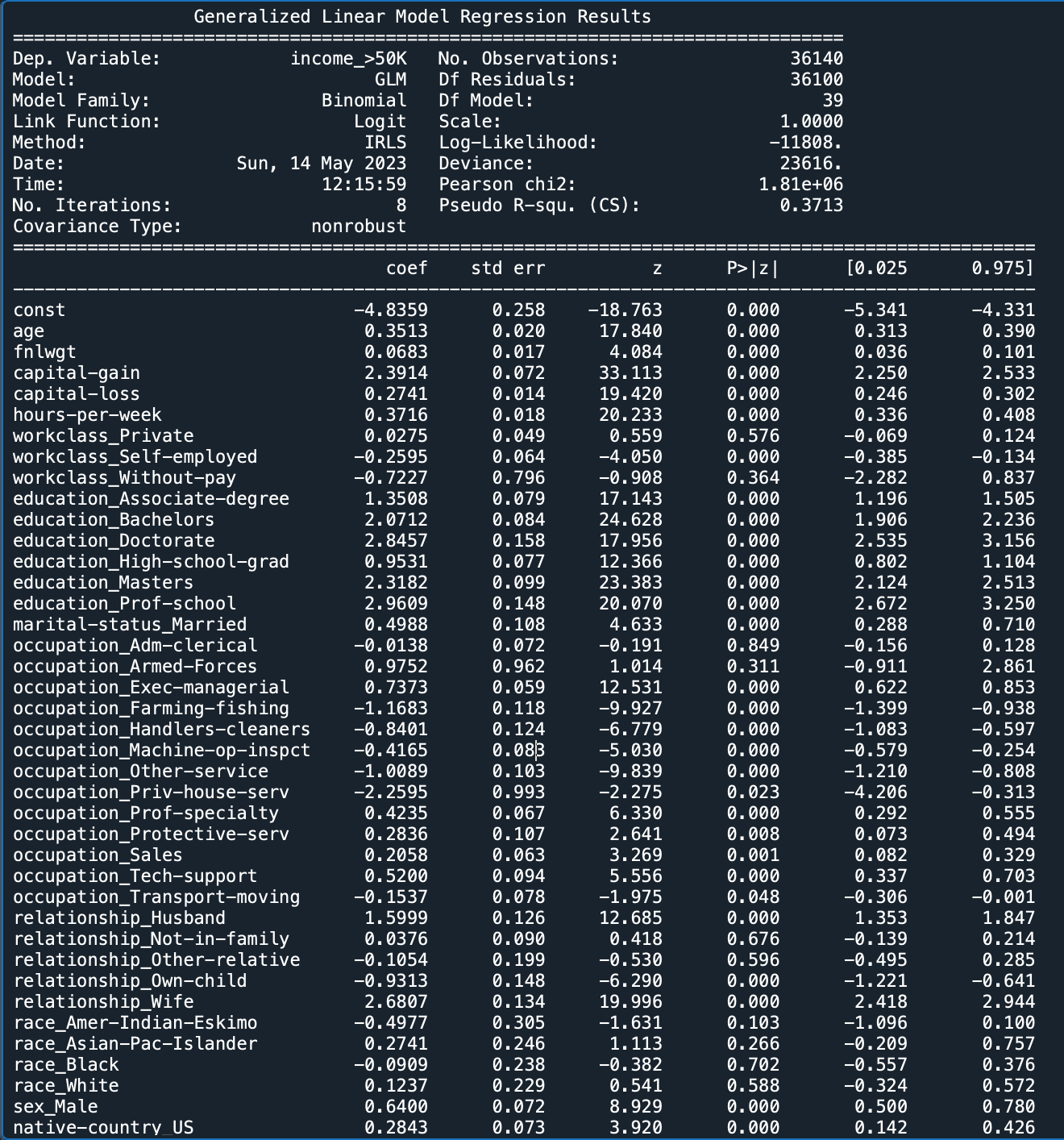
We separate the predictors and the response variable, and split the data into training (80%) and testing. Also, we scale the numerical columns using standard scaler to improve the convergence for the Logistic Regression Model we build (only for the statsmodels one)

**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.



**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).

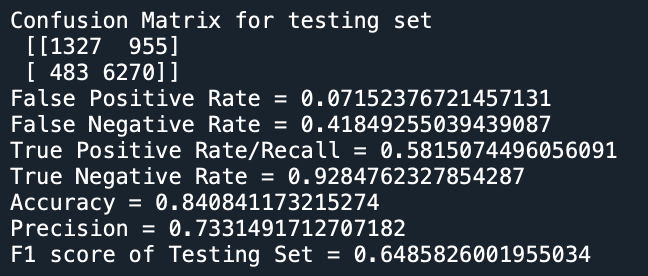
– Odds of income being >$50K increase by 159%, 286%, 693%, 915%, 1831%, 1621% for HS grad, Assoc. degree, Bachelors, Masters, Prof-school, and Doctorate respectively (compared to school dropout)

– 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:

****

**–** 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)

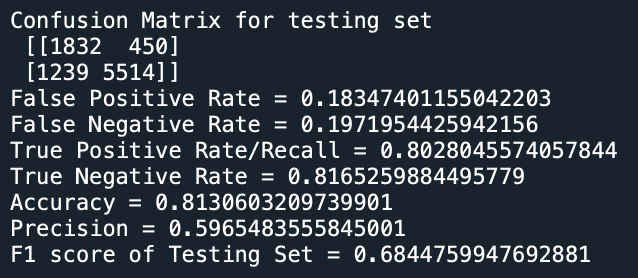
* Using sklearn (with 5-Fold CV, threshold tuning and standardising all features)

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:



– 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.

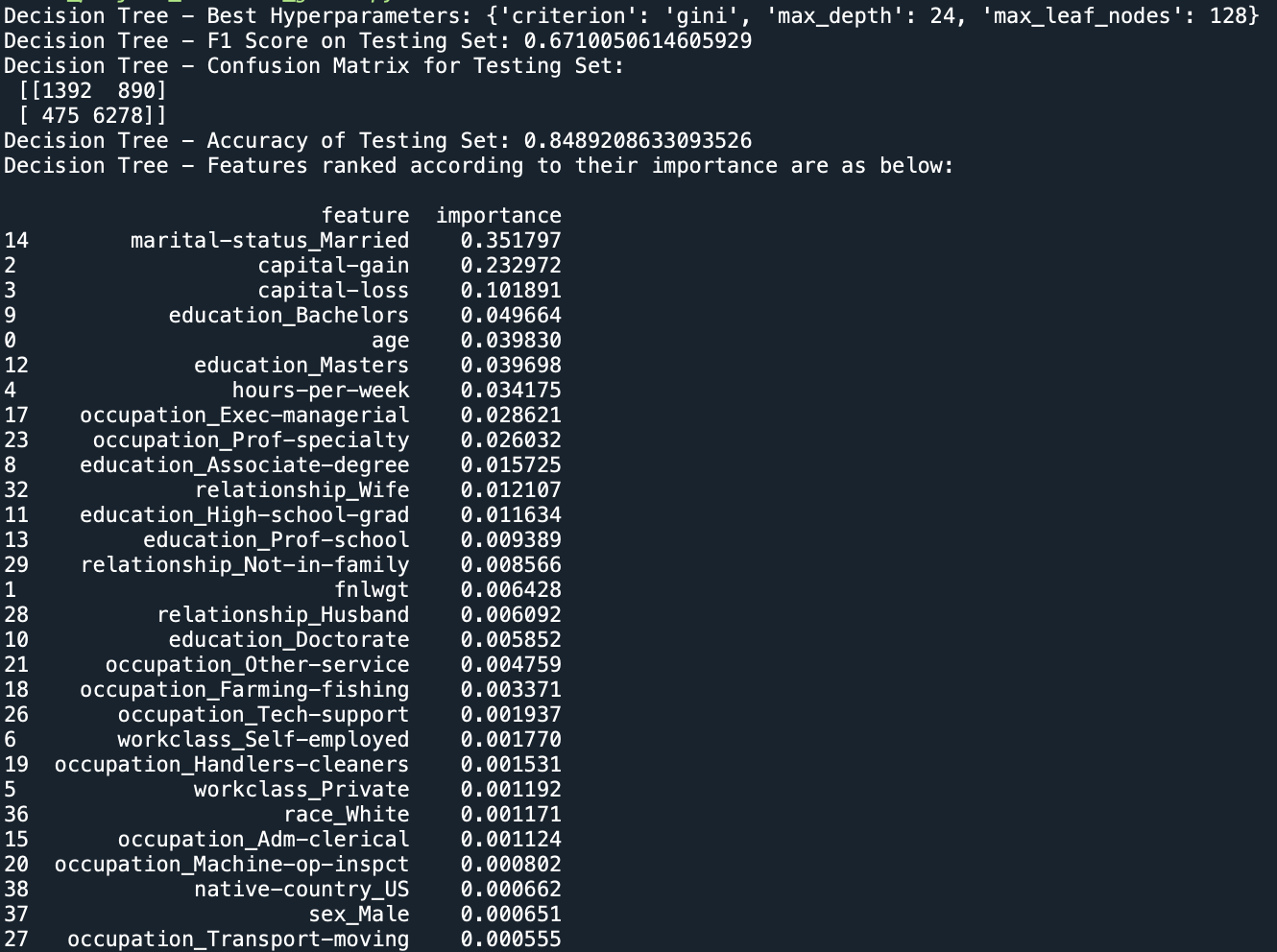
1. **Decision Tree Classifier (with 5-Fold Cross Validation)**

Thesecond 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:

****

– **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

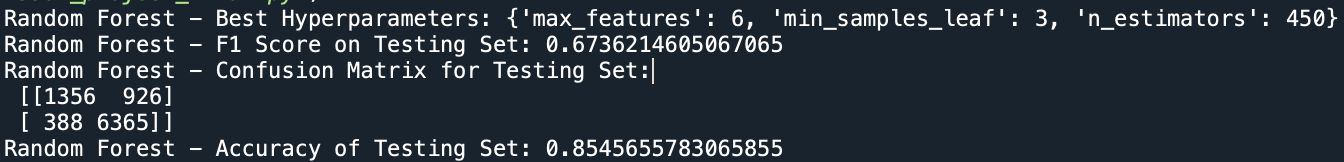
1. **Random Forest Classifier (with 5-Fold Cross Validation)**

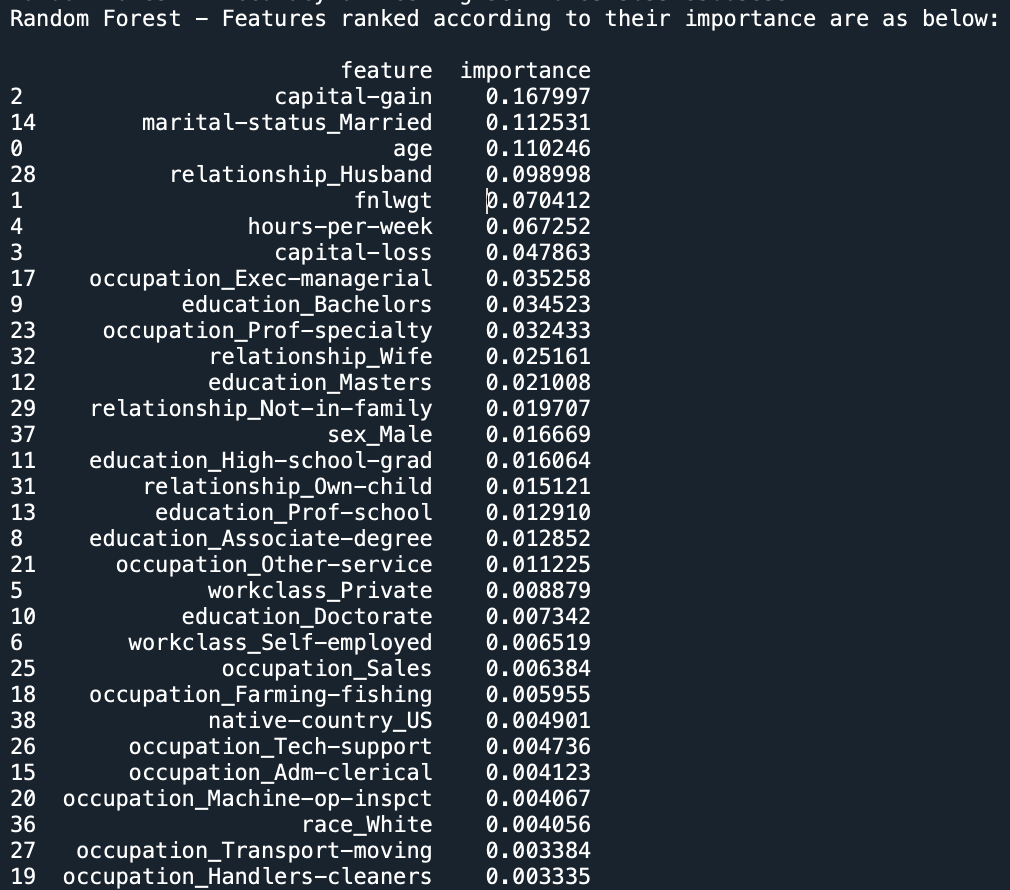
Random forests are supervised machine learning models that train multiple decision trees and integrate the results by averaging them. Each decision tree makes various kinds of errors, and upon averaging their results, many of these errors are counterbalanced.

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:

****

****

– **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

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

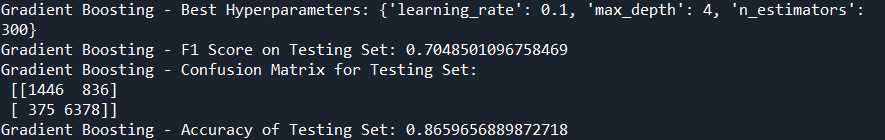
Gradient Boosting Tree Classifier, also known as Gradient Boosted Decision Tree (GBDT) is a popular machine learning algorithm that combines the concepts of gradient boosting and decision trees for classification tasks.

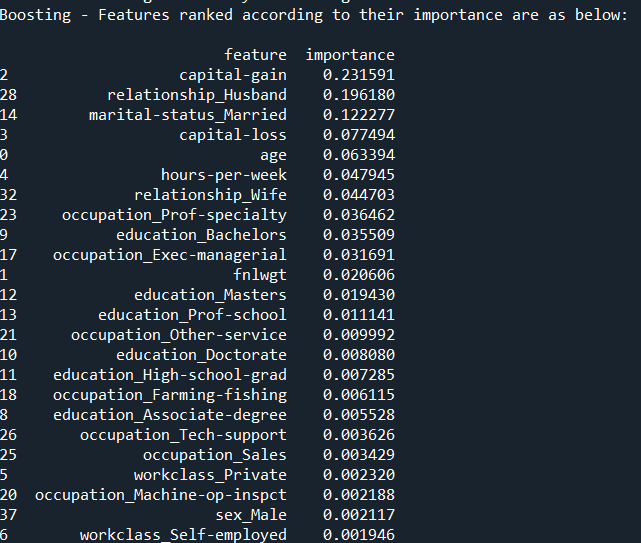
In GBDT, an ensemble of decision trees is sequentially constructed, where each subsequent tree tries to correct the mistakes made by the previous trees. The algorithm learns by minimising a loss function using gradient descent optimization. It starts with an initial prediction (often the mean or a constant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

****

**** – **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>