Classification Modeling to Predict Annual Income

Data Science Course 3 – Machine Learning

Presented By Group 7

Jing Tang

David Graham

Maxim Smetanin

Panthea Saffarzadeh

Ramila Mudarth

Sivi Rakaj

Contents

OBJECTIVE

In this project, we intended to evaluate different machine learning classification models to research the likelihood that an individual would earn an income of $50,000 a year annually or less given certain demographics and social features. The data set used is from a 1994 Census dataset (Census Income Data Set, n.d.).

We reviewed the data set to determine the data quality required for analysis. We however performed additional tasks of data validation and processed feature engineering to proceed with the analysis and modeling.

INTRODUCTION

There is a hypothesis that demographics and socioeconomics status determines an individual's annual income. Using US census data from 1994 we attempted to derive various predictive modeling to assess the possibility of an individual's income, given their social status and demographics, to have the ability to earn over 50K. Through feature importance analysis we can determine if features like education, gender, age or immigration status influence an individual's income. The data set is 70% categorical than numeric.

Within a community, these features can then be focused on to stabilize socioeconomic disparity. (Machine Learning Mastery , 2020)

In our case study, we will begin with transformation of the data using data scalers and encoders. This data will be subjected through various modeling algorithms and evaluation of model performance. We also will apply techniques like dimensionality reduction and hyperparameter tuning to tighten the efficiency of the model and achieve optimal accuracy.

DATA PREPARATION

Dataset has 32,561 entries. The original dataset provides 14 input variables that are a mixture of categorical and numerical data types. The Target Data is the income level i.e . >50 K or <50K. Since there are only two options, this will be a binary classification. The data set had no null values except some irregular data “?” in Work class, Occupation and Native Country. We replaced the "?" values with “Other”. The following features were re-engineered into broader categories for age, working hours, education and marital status. Further to that we created classification features >60 hours or <60 hours called Hour Class.

With Feature importance analysis , we determined the Top 5 Features - Final Weight(fnlwgt), Capital Gain, Marital Status Education, and Age.

Categorical	Numeric	Target

• Work Class
• Education
• Martial Status
• Occupation
• Relationship
• Hour Class
• Race
• Sex
• Native Country
	• Age.
	• Hours-per-week
	• Education Number
	• Fnlwgt (Weight by Demographics)
		• Income Level
		  • >50K
		  • <50K

Figure 1 Data Classification

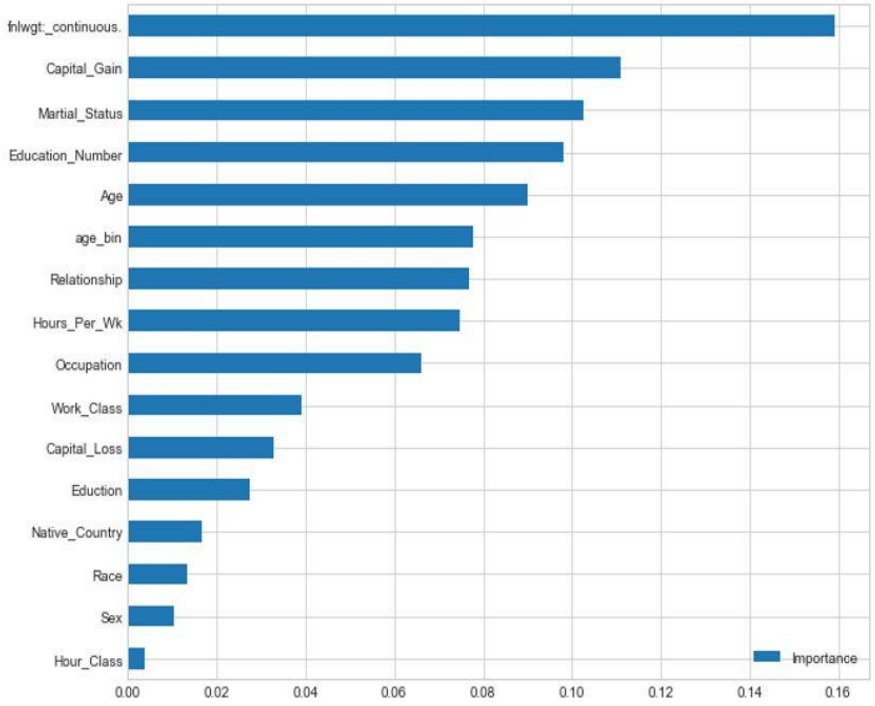


Figure - 2 : Feature Importance Evaluation - Using Random Forest Classifier

DATA VISUALIZATION AND ANALYSIS

We proceeded to review the proportion of the target variable in the dataset. From the graph below (fig 3), we can clearly observe that the Income level less than 50K is almost 3 times of those above 50K. This is a mildly imbalanced data set. However, since there is no data on the upper limit of adult's income above 50K, it's premature to conclude that the total amount of wealth is skewed towards high income groups.

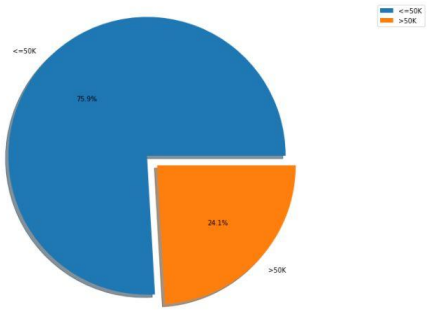


Figure - 3 : Data Composition - Imbalance Data Set

Figure - 4 : Disparity of observation by Gender

We used histogram to visualize the gender of the person based on (the number of records and weight distribution) and the total income of that specific record number (fig 4). This plot shows that there are more male than female participants. When we compare the two genders corresponding income distribution, more percent of males have an Income above $50K compared to females.

Observations from Data Visualizaton	
(Visualization in Appendix)	
Numerical (Histogram Analysis)	**Categorical (Frequency Distribution Analysis)**
Education level higher than a 'Masters/Doctorate/Professional school' with a greater representation for Income >50K, there is still a good amount for <50K	Based on occupation, 3600 records had their occupation in sales and less than 200 records were in the Private-house-servers category.
'White'/'Asian-pac-islander' has a greater likelihood of earning >50K	Most records on education were high-school grads or college educated. We happen to notice that this is the same as the education-num attribute in the data set as well. This could be used as an ordinal
Ratio of individuals earning more than 50K is higher in Work Class - Self Employment	Aligning to statistics, married individuals make greater than 50K annually.
Males have a higher representation of earning more than 50K	Work class features have most records in private sectors(self incorporated, self non-incorporated, private organizations), and the rest were evenly distributed among public sectors(state, federal, local government employees)
For relationship status - 'Husband or Wife', there is a higher representation of earning more than 50K	
Approximately individuals greater than 30 years have to work > 60 hours to earn more than 50K	

Figure - 5 : Inferences from Data Visualization

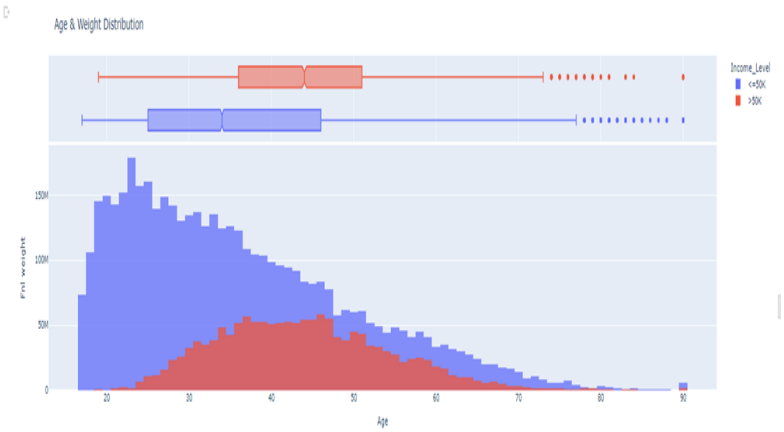


Figure - 6 : Box Plot Distribution of Feature Weight and Age against Income

As per the box plot analysis, 75% of Individuals in the >50K classification are within the age group of 36 -51 years, with a median at 44. 75% of Individuals in the <50K classification are with the the age range of 25 -46 with median at 34.

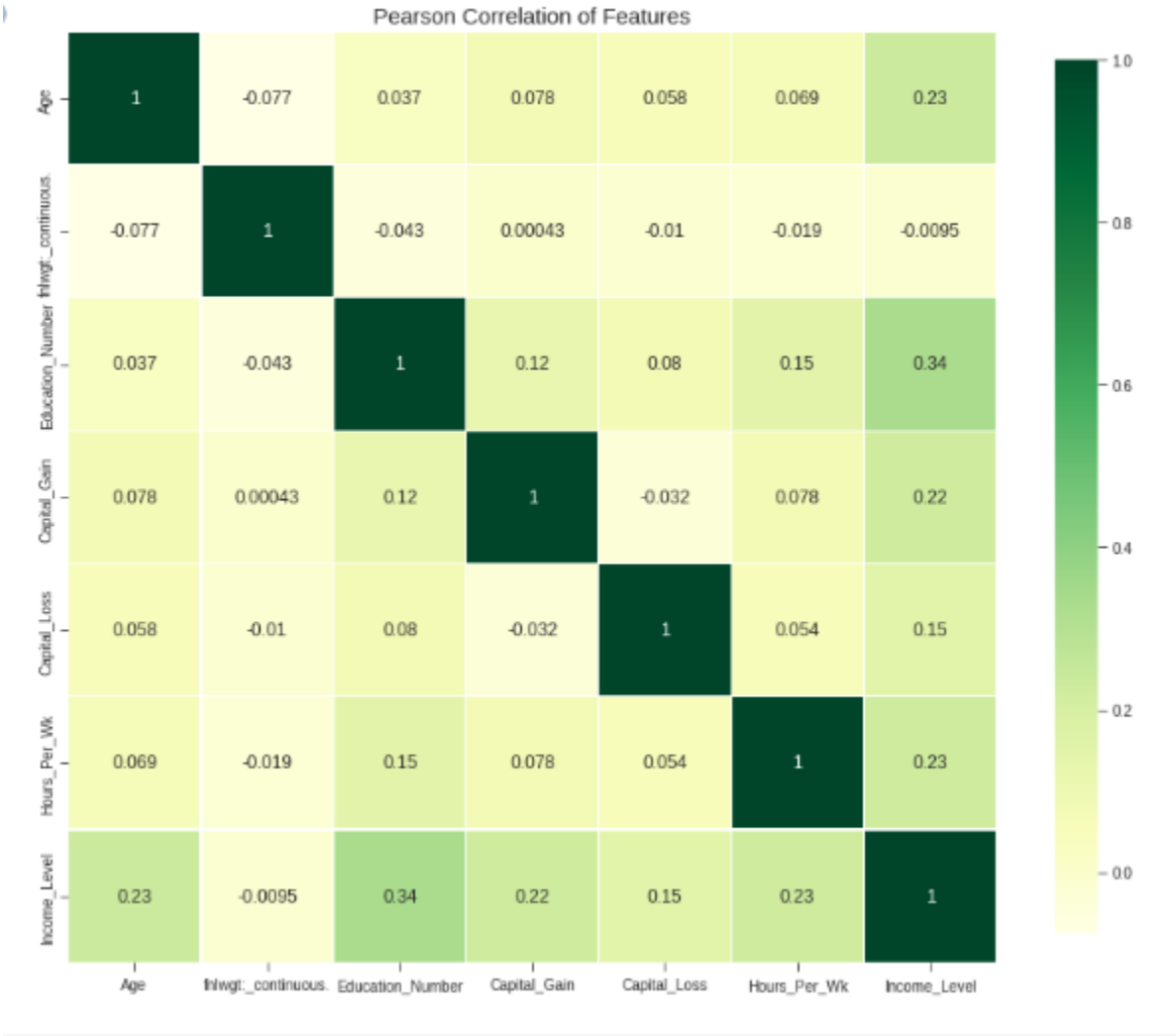


Figure - 7 : Correlation Matrix against Income Category

There is moderate to low correlation between target Variable (Income) and other numerical features. Since there is no linear correlation, we can proceed to apply data modeling.

DATA TRANSFORMATION

Prior to modeling, the numeric and categorical data were scaled and transformed.

Categorical Variables were transformed using OneHot Encoder.

Numerical Variables were scaled using MinMax Scaler. Both of these variables were then fitted into a pipeline.

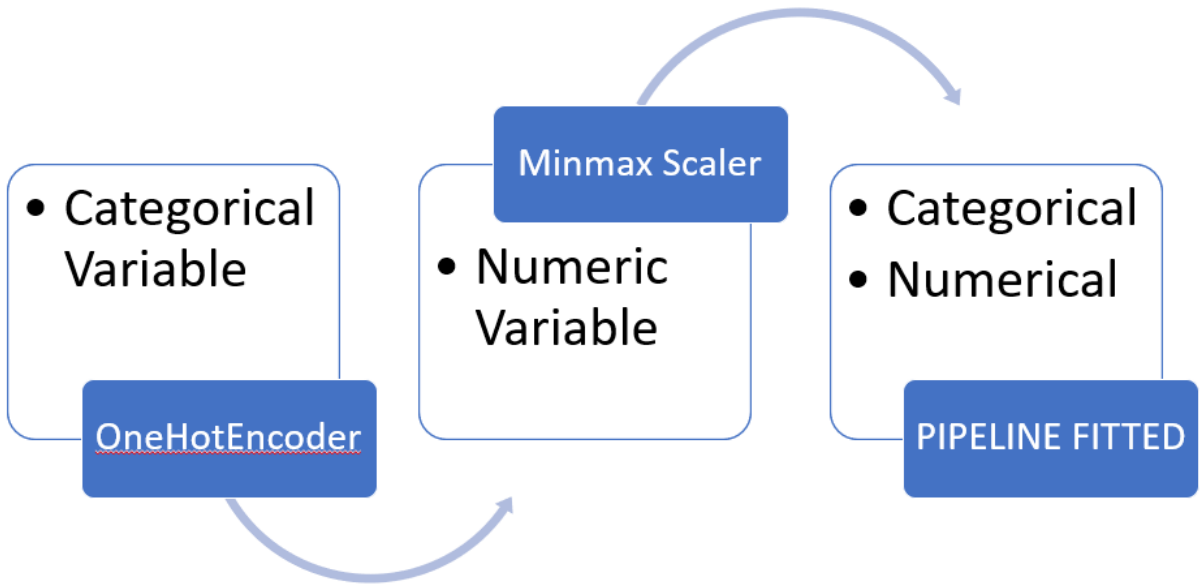


Figure - 8 : Process Flow for Pipeline

The data was then classified into Train (80%) and Test (20%)

Data Array	Rows	Original Shape	Transformed Shape
X TRAIN	26048	11	86
X TEST	6513	11	86
Y TRAIN	26048	1	1
Y TEST	6513	1	1

DATA MODELING

Ensemble

The dataset was processed through an ensemble classifier - Voting Classifiers

Hard Voting

KNN, logistic regression and Random Forest Classifier were applied with a voting type : Hard.

The model accuracy score for voting classifier was 83.4%, while the individuals were,

- KNeighbors Classifier - 83.8%
- LogisticRegression Classifier - 82.3%
- RandomForestClassifier - 82.3%

Soft Voting

KNN, logistic regression and Random Forest Classifier were applied with a voting type: Hard.

The model accuracy score for voting classifier was 83.57%, while the individuals were as follows.

- KNeighbors Classifier - 83.8%
- Logistic Regression Classifier - 82.0%
- RandomForestClassifier - 82.3%

Between Hard and Soft Voting Classifiers, there was no difference in model accuracy without any hyperparameter tuning. Hyperparmeter tuning on this classifier tends to be time consuming but there is an accuracy increase that can be achieved when applied. When applied, the model accuracy improved by 1% to 84%.

Bagging Classifier

Bagging classifier for untransformed data produce an accuracy score of 81.94%.

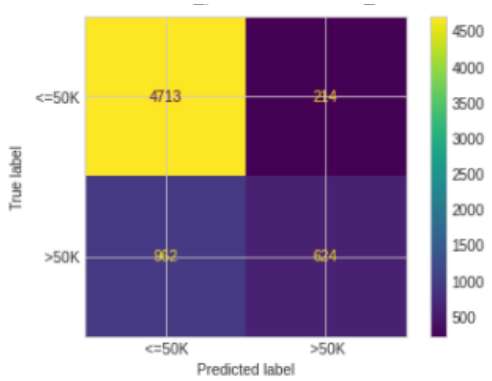


Figure - 9 : Confusion Matrix - Bagging Classifier (No Tuning)

Classification Report

	precision	recall	f1-score	support
0	0.83	0.96	0.89	4927
1	0.76	0.39	0.51	1586
accuracy			0.82	6513
macro avg	0.79	0.67	0.70	6513
weighted avg	0.81	0.82	0.80	6513

Figure - 10 : Classification Matrix - Bagging Classifier (No Tuning)

Gradient Boosting

Gradient Boosting classifier defined 100 trees for the model.

The overall accuracy score for the model is 84%

Classification Report

	precision	recall	f1-score	support
0	0.87	0.93	0.90	4927
1	0.71	0.56	0.63	1586
accuracy			0.84	6513
macro avg	0.79	0.74	0.76	6513
weighted avg	0.83	0.84	0.83	6513

Figure - 11 : Classification Report - Gradient Boosting (No Tuning)

ADA Boosting

Results using AdaBoostClassifier(DecisionTreeClassifier(max_depth=1), n_estimators=200, algorithm="SAMME.R", learning_rate=0.5)) was 83.96% accuracy.

Classification Report

	precision	recall	f1-score	support
0	0.87	0.93	0.90	4927
1	0.71	0.57	0.63	1586
accuracy			0.84	6513
macro avg	0.79	0.75	0.77	6513
weighted avg	0.83	0.84	0.83	6513

Figure - 12 : Classification Report - ADA Boosting

Random Forest Classifier

By running the random forest classifier, the accuracy score was 82.1% with no tuning.

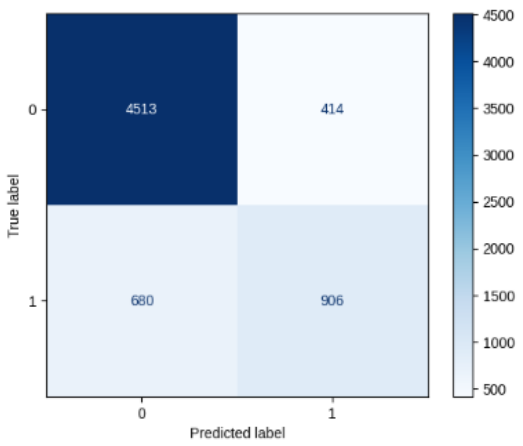


Figure - 13 : Confusion Matrix - Random Forest Classifier (No Tuning)

	precision	recall	f1-score	support
0	0.87	0.92	0.89	4927
1	0.69	0.57	0.62	1586
accuracy			0.83	6513
macro avg	0.78	0.74	0.76	6513
weighted avg	0.82	0.83	0.83	6513

Figure - 14 : Classification Report - - Random Forest Classifier (No Tuning)

Hyperparameter Tuning on Random Forest Classifier

A grid search CV had been run on the hyperparameter - n estimator, min sample Leaf, min weight fraction leaf, max features and criterion with values below.

```
rnd_clf = RandomForestClassifier()
params = {
    'n_estimators': [5, 10, 20, 50, 100, 1000],
    'min_samples_leaf': [5, 10, 20, 50, ],
    'min_weight_fraction_leaf': [0, 1, 2],
    'max_features' :["auto","sqrt", "log2"],
    'criterion': ["gini", "entropy"]
}
grid_search = GridSearchCV(estimator = rnd_clf,
                           param_grid=params,
                           cv=4, n_jobs=-1, verbose=1, scoring = "accuracy")
```

Figure - 15 : Hyperparameter Tuning Input

The optimal values were,

- max_features='sqrt'
- min_samples_leaf=5,
- min_weight_fraction_leaf=0
- n_estimators=1000

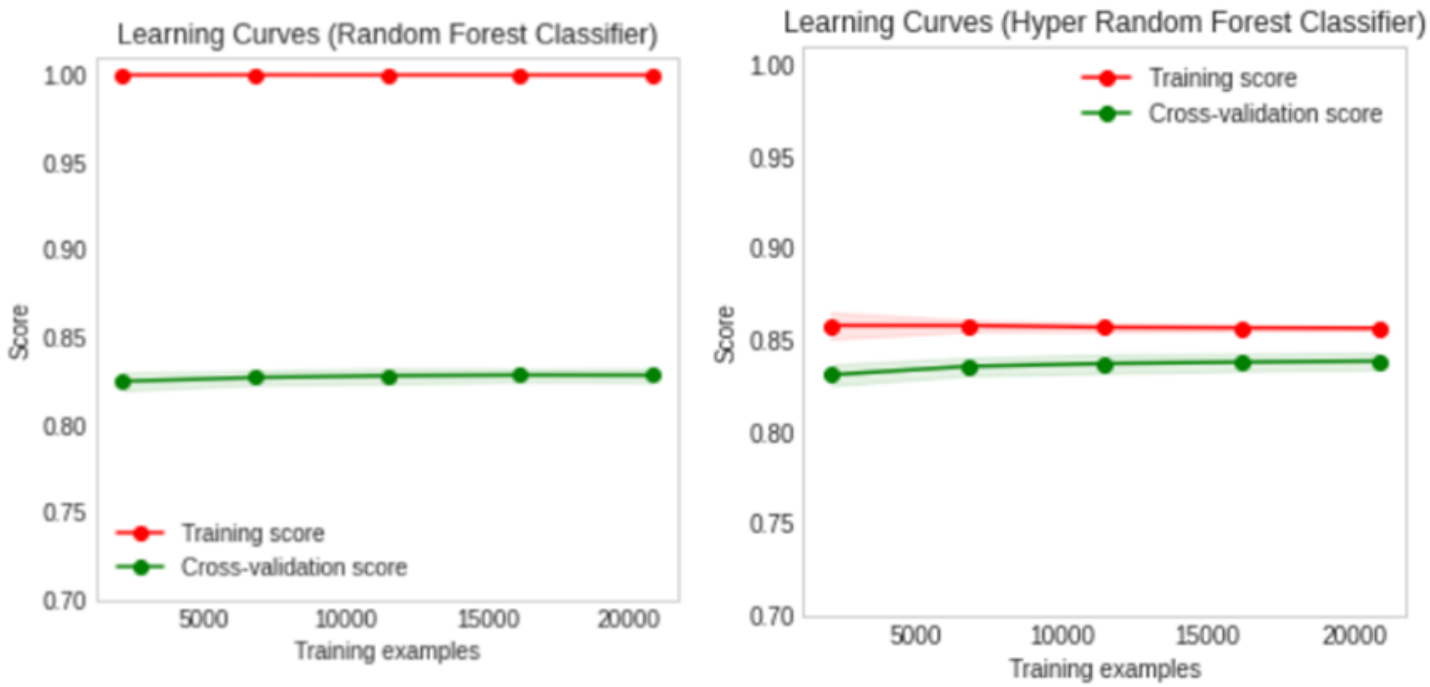


Figure - 16 : Learning Curve Before and After Hyper Parameter Tuning for Random Forest Classifier

```
Classification Report
              precision    recall  f1-score   support

       <=50K       0.87      0.93      0.90      4927
        >50K       0.73      0.55      0.63      1586

    accuracy                           0.84      6513
   macro avg       0.80      0.74      0.77      6513
weighted avg       0.83      0.84      0.83      6513

Confusion Report
[[4605  322]
 [ 707  879]]
CPU times: user 1.51 s, sys: 5.07 ms, total: 1.51 s
Wall time: 1.51 s
```

Figure - 17 : Classification Report after tuning for Random Forest Classifier

Accuracy increased to 84%

With Dimensionality Reduction

When PCA dimensionality reduction was applied, it reduced the data set to 1 dimension. The Random Forest Classifier was applied again and results were shown below.

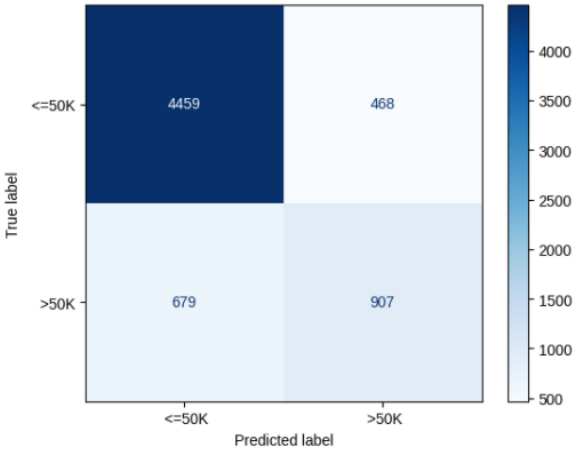


Figure 18 - Confusion Matrix With Dimensionality Reduction

```
              precision    recall  f1-score   support

       <=50K       0.87      0.91      0.89      4927
        >50K       0.66      0.57      0.61      1586

    accuracy                           0.82      6513
   macro avg       0.76      0.74      0.75      6513
weighted avg       0.82      0.82      0.82      6513
```

Figure 19 - Classification Matrix With Dimensionality reduction

The accuracy score was 82% , which did not improve the models accuracy rate.

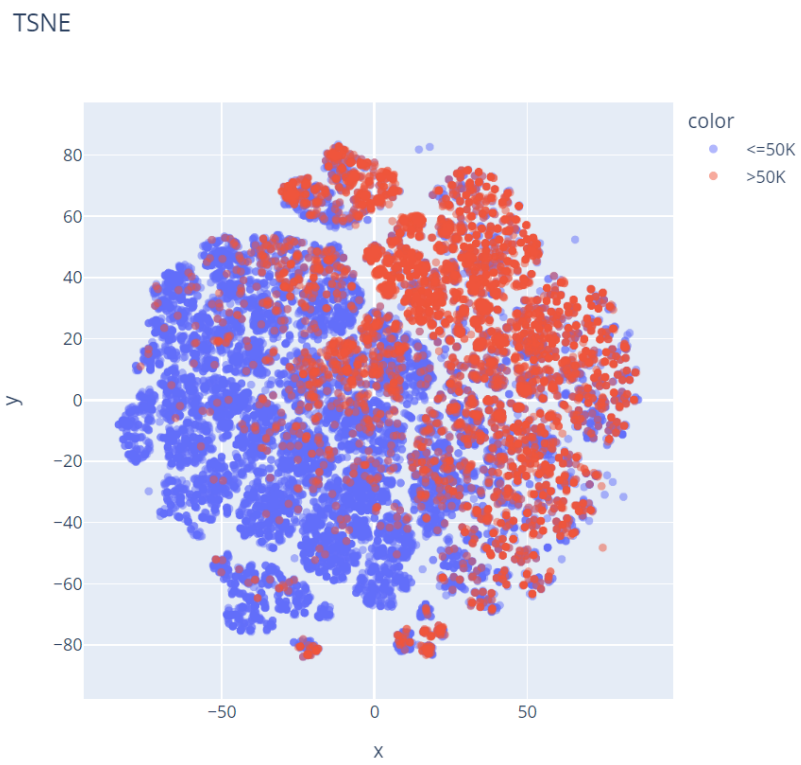


Figure 20 - TSNE reduction

Dimensionality reduction using the t-distributed stochastic neighbor embedding (TSNE) reduction method. TSNE splits the dataset in two distinctive parts It shows no clear boundary between income level.

Tensor Flow

We have applied tensor flow to our dataset, using three hidden layers with 30 neurons each and activation function ReLU. For the final layer we have used Sigmoid activation, this is well suited for datasets with binary targets. Model was compiled using Adam optimizer, and binary cross entropy for loss calculations. Starting from 20 epochs, batch size 32, we have set callback monitoring validation loss value with patience set to 3. The calculation was interrupted at the seventh epoch. The accuracy 83.96 %. It was the best result so far.

Decision Tree

For running the decision tree classifier, a randomized search CV classifier was used with the following parameters.

The optimal tuning parameters obtained by Random grid search

- criterion: gini',
- max_depth: None,
- max_features: 7,

- min_samples_leaf: 3

	precision	recall	f1-score	support
<=50K	0.86	0.92	0.89	4927
>50K	0.67	0.53	0.59	1586
accuracy			0.82	6513
macro avg	0.76	0.72	0.74	6513
weighted avg	0.81	0.82	0.81	6513

```
[[4515  412]
 [ 748  838]]
```

Figure 21 - Classification Report for Decision Tree Classifier

Gives mode score 82%

Extremely Randomized Trees

Extremely Randomized Trees classifier created an accuracy model of 80.31% with no hyperparameter tuning.

Learning Curve Analysis

Random Forest, ADA, Gradient Boost Classifier

Default Random Forest Classifiers overfit the model. However with hyper parameter tuning, the model fitting has improved. Interestingly, gradient and ADA boosting show good fittings.

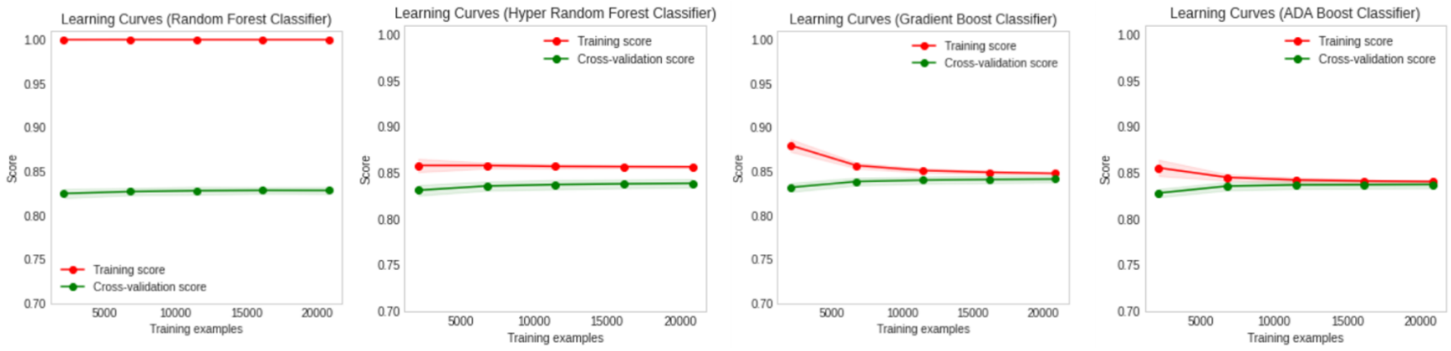


Figure 22 - Learning Curves for Random Forest Classifier(Before and After Tuning), Gradient Boost and ADABoost Classifier

The validation score and the training score converge . Hence we can conclude increasing the size of the training set will not be of any benefit. Hence tuning the model is a better option.

CONCLUSION

Based on the data visualization, it can be inferred that the data set is mildly imbalanced for Income Levels as well as Gender observations. This could create some level of bias in the overall projections and in the modeling. However based on feature importance for a classification model, the top 5 features were Final Weight(fnlwgt) , Capital Gain, Marital Status, Education, and Age. The data features are 70% categorical and 30% numeric. Hence the need for transformation through one hot encoder. This identifier does influence the overall accuracy of the classification modeling.

For the modeling purposes, we used a 80-20 train test split. The data was then run through Random Forest Classifier with and without hyperparameter tuning as well as dimensionality reduction. We were able to fine tune the accuracy score using Hyperparameter tuning but dimensionality reduction did not influence the scoring. The modeling was then tested with the classifier listed below and it can be inferred that Random Forest Classifier - Tuned Hyperparameters, Gradient Classifier and ADA Boost have the most accurate score for our Classification problem. The Decision Classifier with Tuning had the lowest performance. Some models have an F1 score of 0.9 so be confident that our models are efficient in prediction <50K.

Classifier Type	Accuracy Score	Target	Precision	Recall	F1 Score
Random Forest	83.63%	Less 50,000$:	0.86	0.88	0.89
		More 50,000$:	0.6	0.54	0.61
Random Forest Tuned	84.12%	Less 50,000$:	0.87	0.94	0.9
		More 50,000$:	0.73	0.55	0.63
Decision Tree Tunned	81.37%	Less 50,000$:	0.83	0.94	0.88
		More 50,000$:	0.69	0.41	0.52
ADA Boost	83.95%	Less 50,000$:	0.87	0.93	0.9
		More 50,000$:	0.71	0.57	0.63
Gradient Classifier	83.96%	Less 50,000$:	0.87	0.93	0.9
		More 50,000$:	0.71	0.56	0.63
Ensemble Voting	83.54%	Less 50,000$:	0.86	0.94	0.9
		More 50,000$:	0.72	0.52	0.61
Tensor Flow	83.53%	Less 50,000$:	0.88	0.91	0.89
		More 50,000$:	0.68	0.61	0.62
Wide & Deep Neural Network	83.45%	Less 50,000$:	0.87	0.92	0.89
		More 50,000$:	0.7	0.59	0.64

Figure 23 -Performance Report of Machine Learning Algorithm

All the models were able to run with good performance for the task at hand. The learning curves showcased that increase in observations would not improve the model's accuracy score but rather focus on fine tuning the parameters for the modeling.

References

Cencus Income Data Set. (n.d.). Retrieved from UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Census+Income

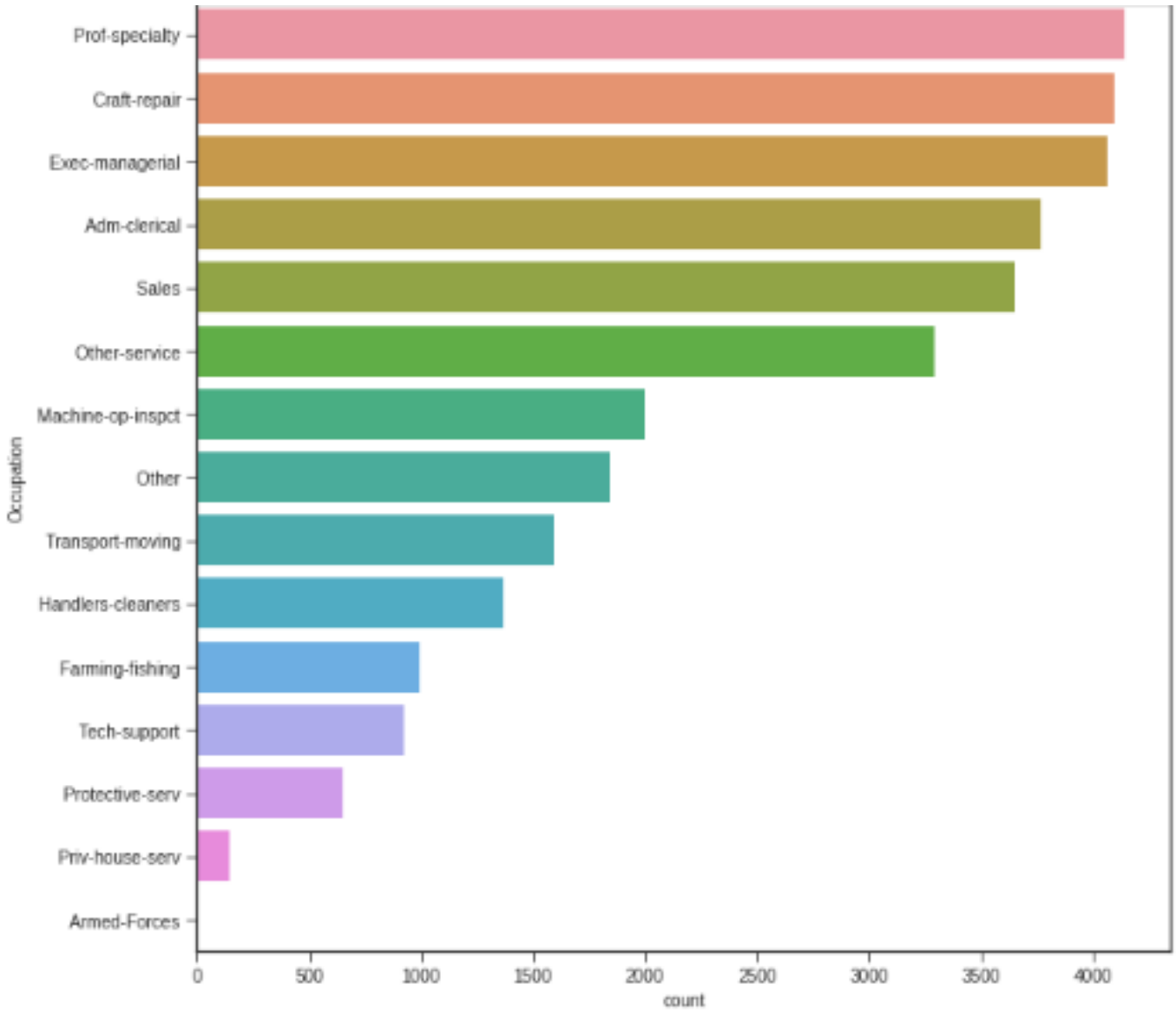
Machine Learning Mastery . (2020, October 27). Retrieved from Imbalanced Classification with the Adult Income Dataset: https://machinelearningmastery.com/imbalanced-classification-with-the-adult-income-dataset/

Predicting if income exceeds $50,000 per year based on 1994 US Census Data with Simple Classification Technique." UCSD CSE, https://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/048.pdf

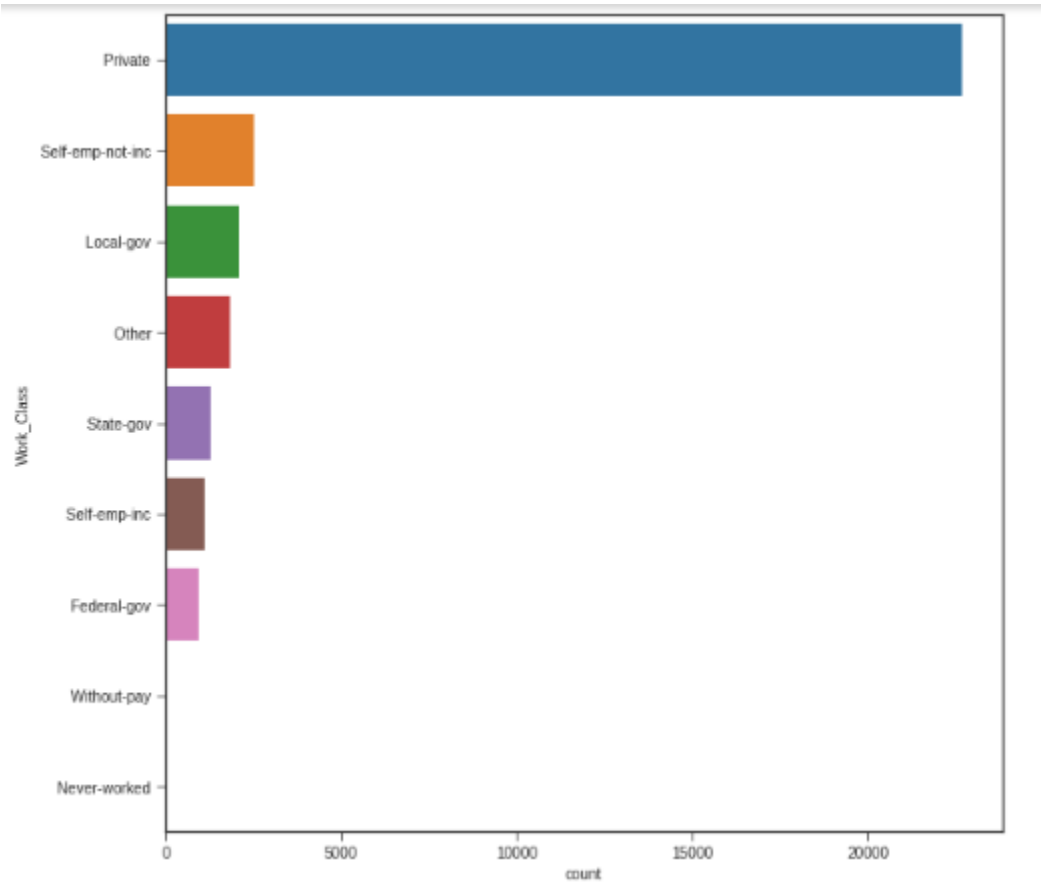
The source of the notebook: https://github.com/MaxSMCON/Group07/blob/main/Group_7_ML.ipynb

Appendix

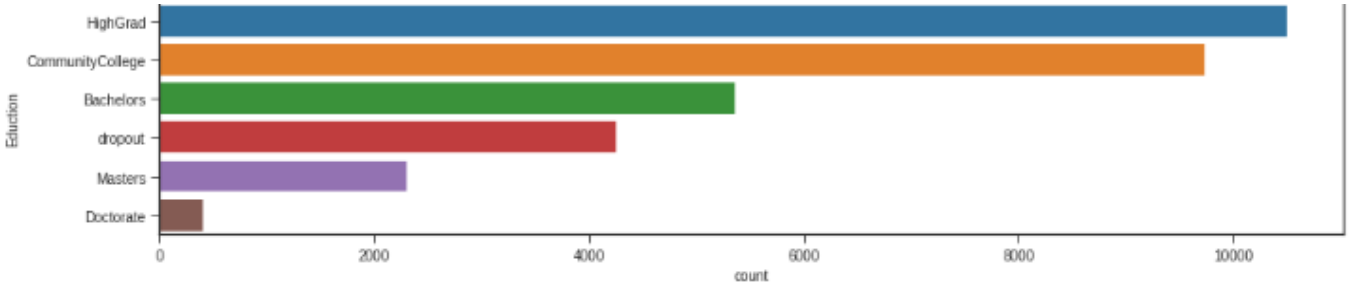
Distribution of Income Levels by sex



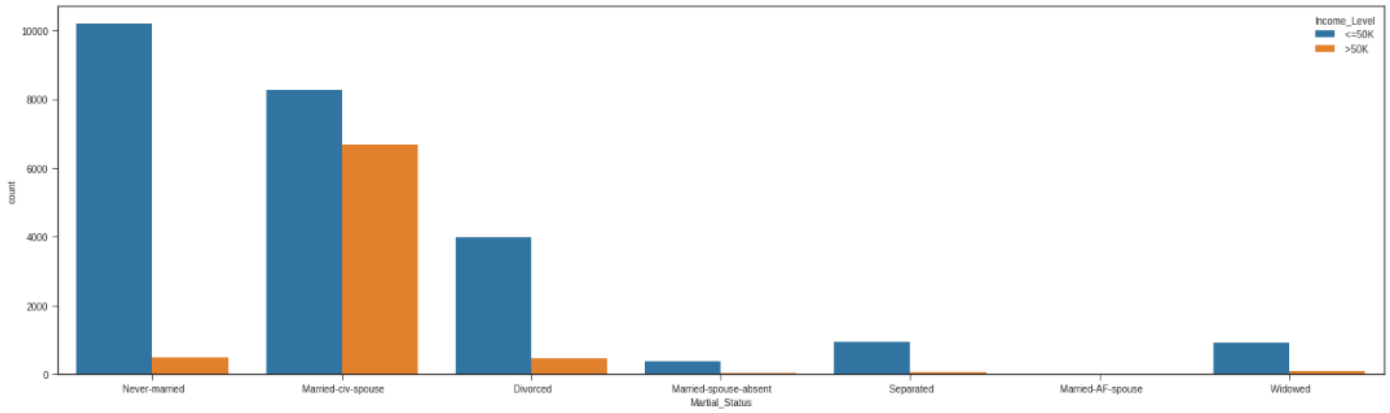
Distribution of Occupation



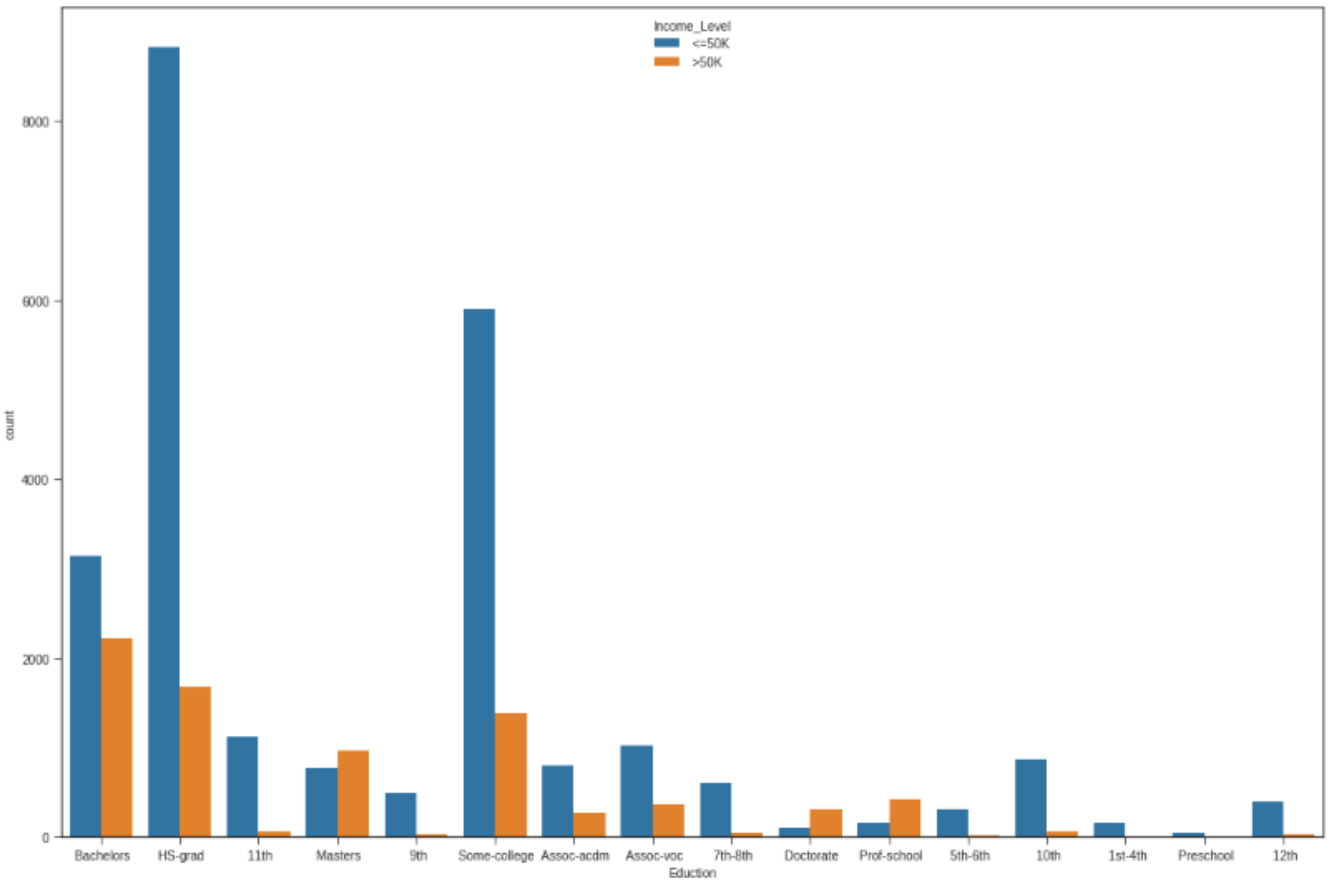
FEature weight Distribution



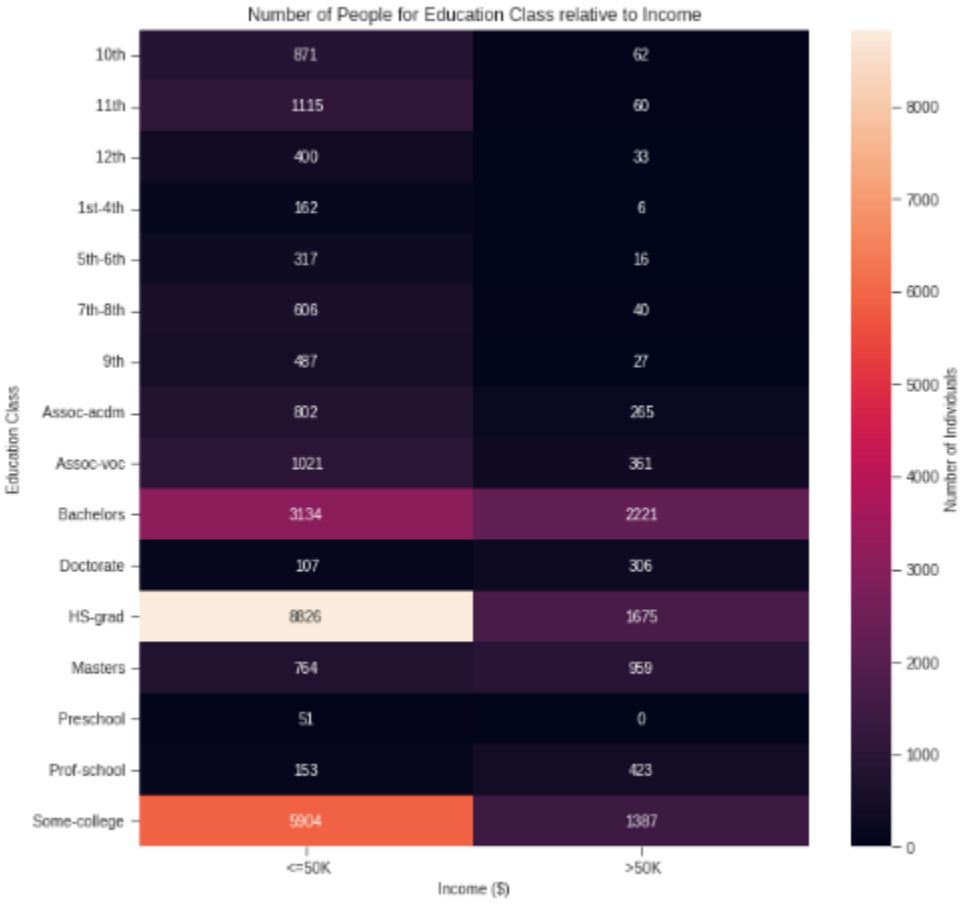
Education Distribution



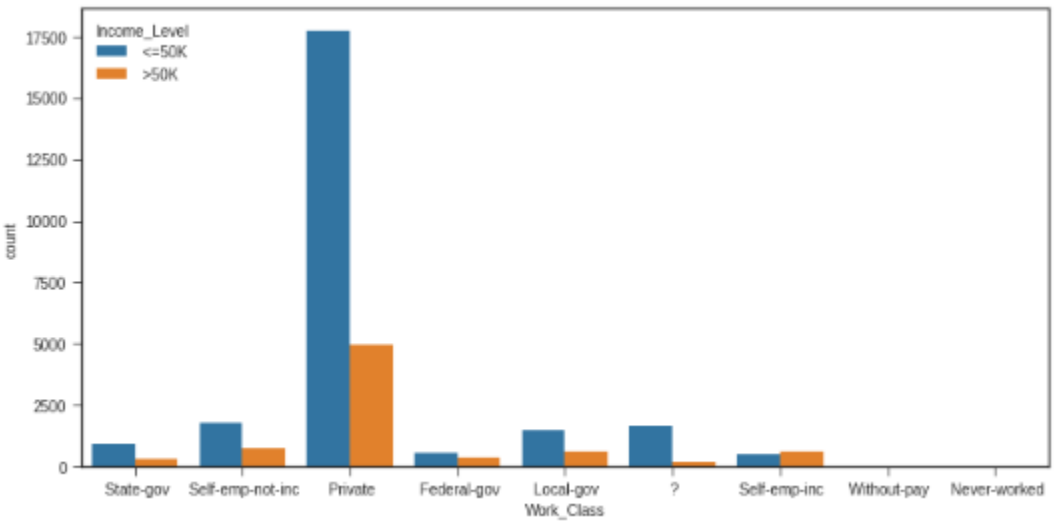
Martial Status



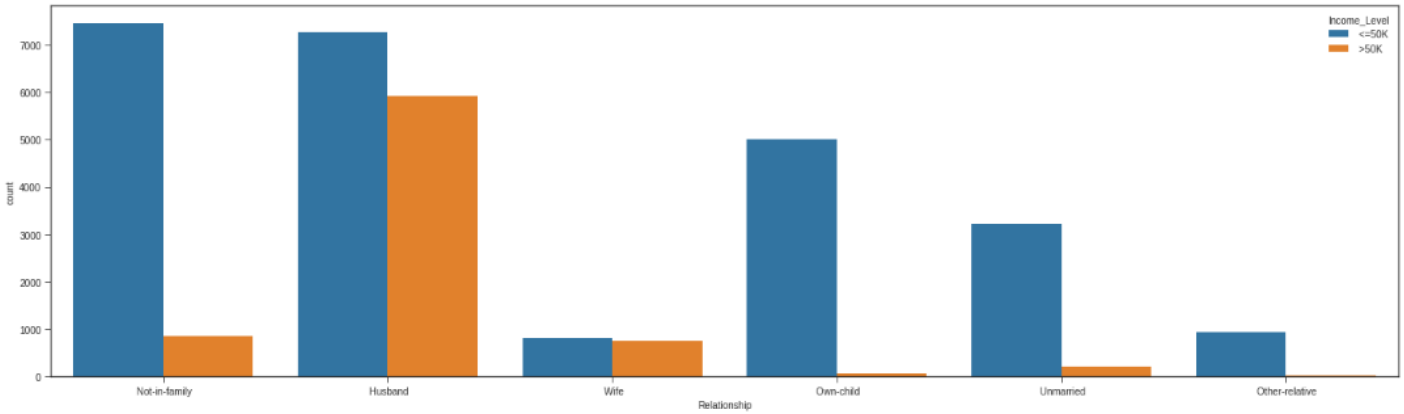
Education, Income Level Distribution



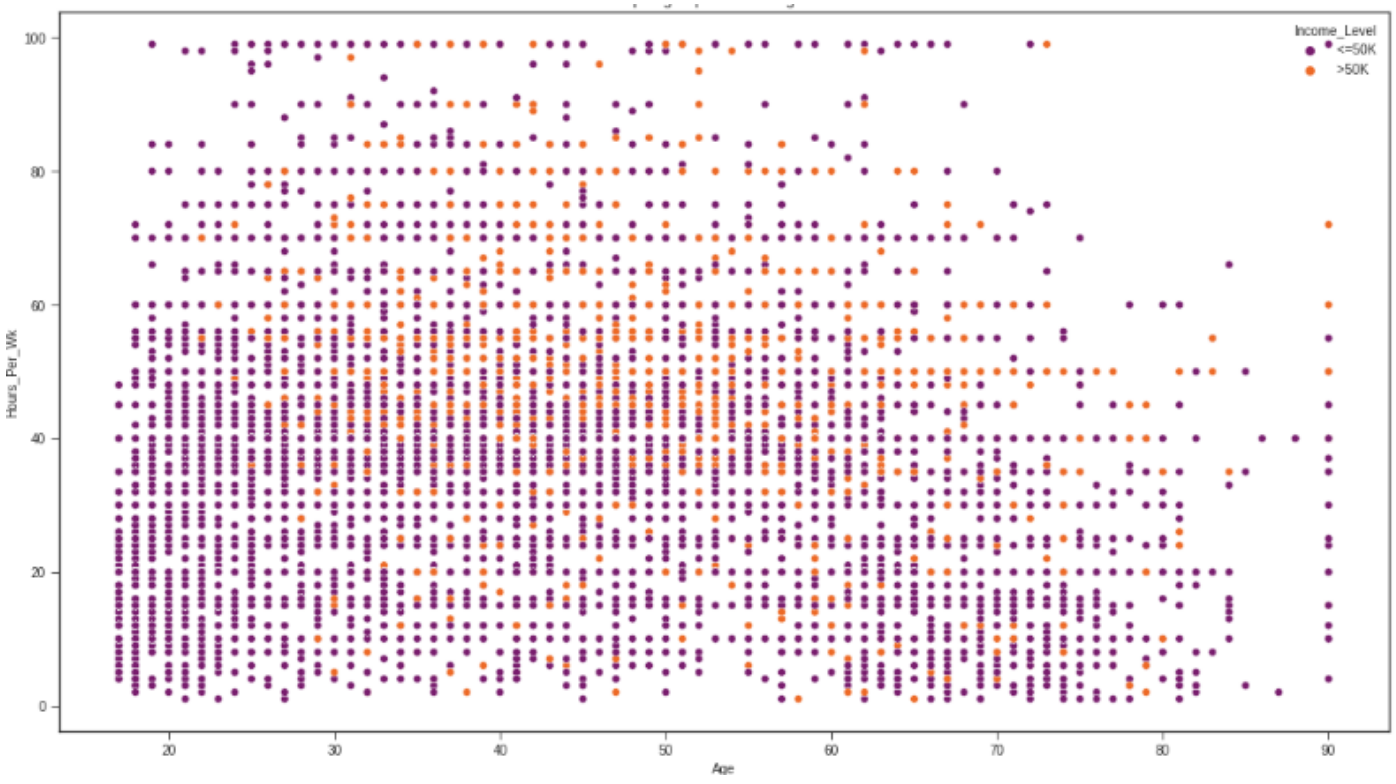
Individual Count for Education Class versus Income



Work CLass



Relationship Status Income Distribution



Scatter Plot by Income