Analysis on The influence on Adult Income by Different Variable With Machine Learning

As society progresses, the cost of living increases dramatically. Even when people are generating more and more income, their salaries often merely cover their daily consumption, let alone their desires for extravagant items. People with different lifestyles have different backgrounds. I am curious to find out what really causes the difference in people's income. Hence, I want to build a model that will accurately predict an adult's income based on different variables. From Kaggle, I found an excellent dataset that is organized by the University of California Irvine. The data is originally "extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker."

Part 1: Data Processing

My first step in creating the model is data processing. After downloading and reading the dataset into my google notebook, I begin by looking at the overview of the data.

```
# Display the data
adult.head()
```

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native.(
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female	0	4356	40	Unit
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	0	4356	18	Unit
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Female	0	4356	40	Unit
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female	0	3900	40	Unit
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female	0	3900	40	Unit

The dataset is relatively clean, composed of two data types, object and int64. The shape of the dataset is (32561, 15). It is crucial to pinpoint the target variable before beginning any analysis. As I want to investigate the adult income, the attribute “income” is evidently my target variable.

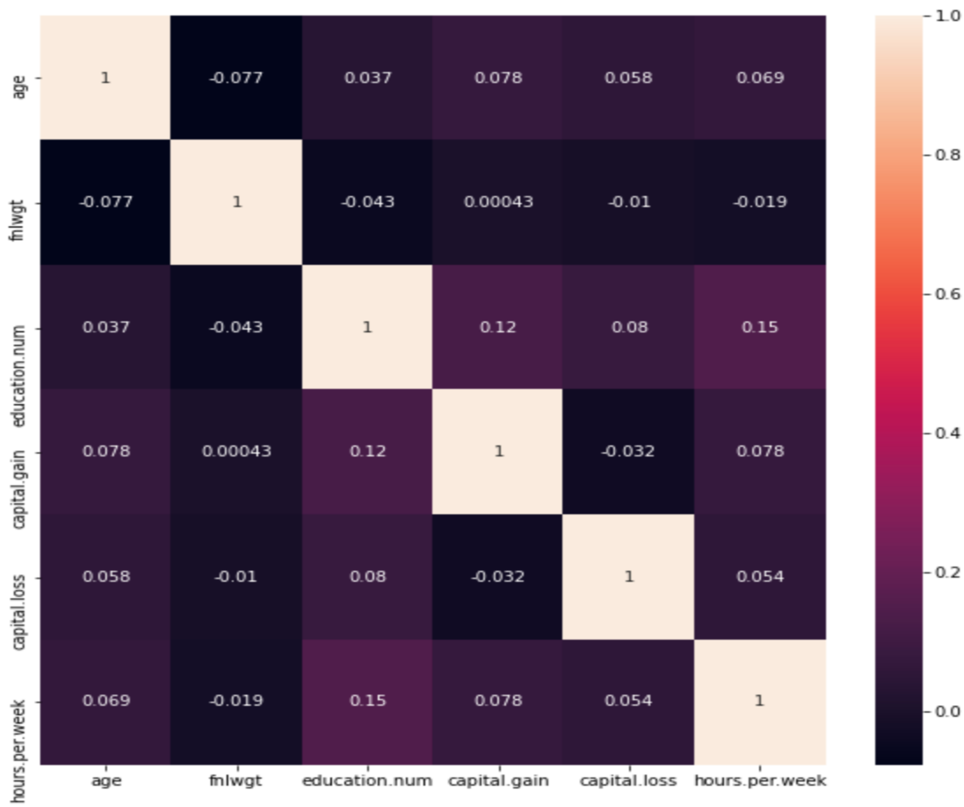
```
# Overview of dataset
adult.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             32561 non-null  int64 
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64 
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64 
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64 
 11  capital.loss    32561 non-null  int64 
 12  hours.per.week  32561 non-null  int64 
 13  native.country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

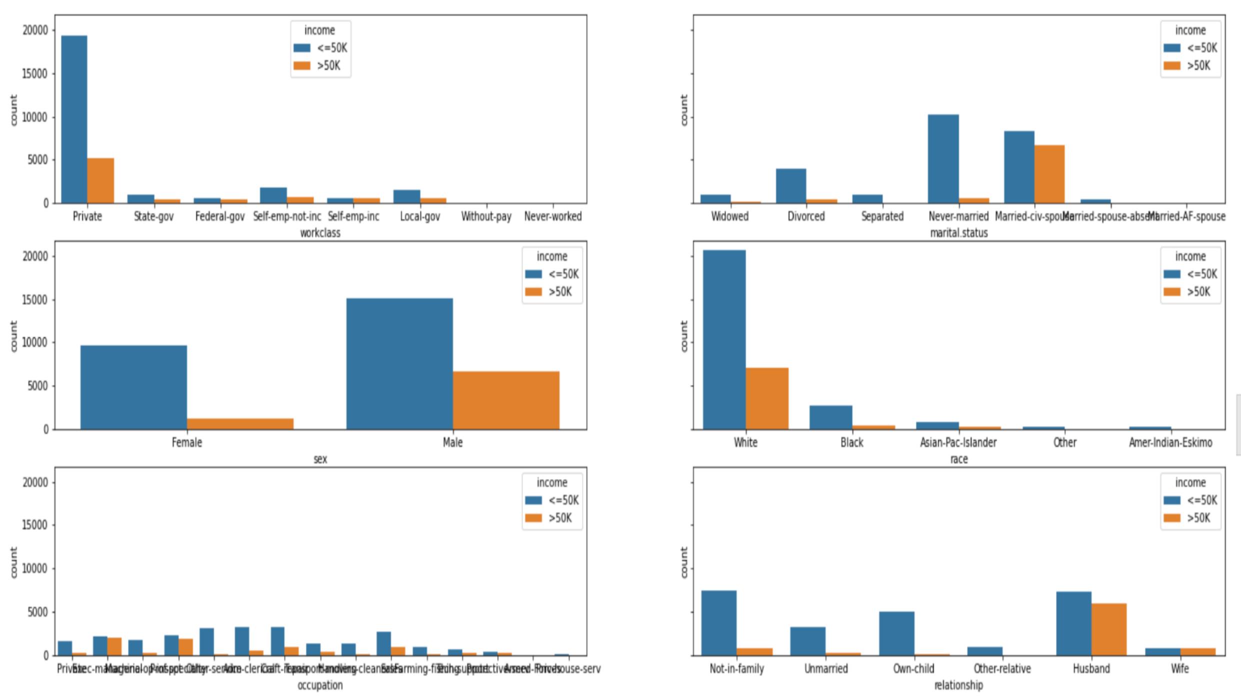
I noticed that some of the value is missing, so I have to assign them values so they do not interfere with the latter analysis. With a brief clean up, the data is ready to be utilized in the next phase.

Part 2: EDA

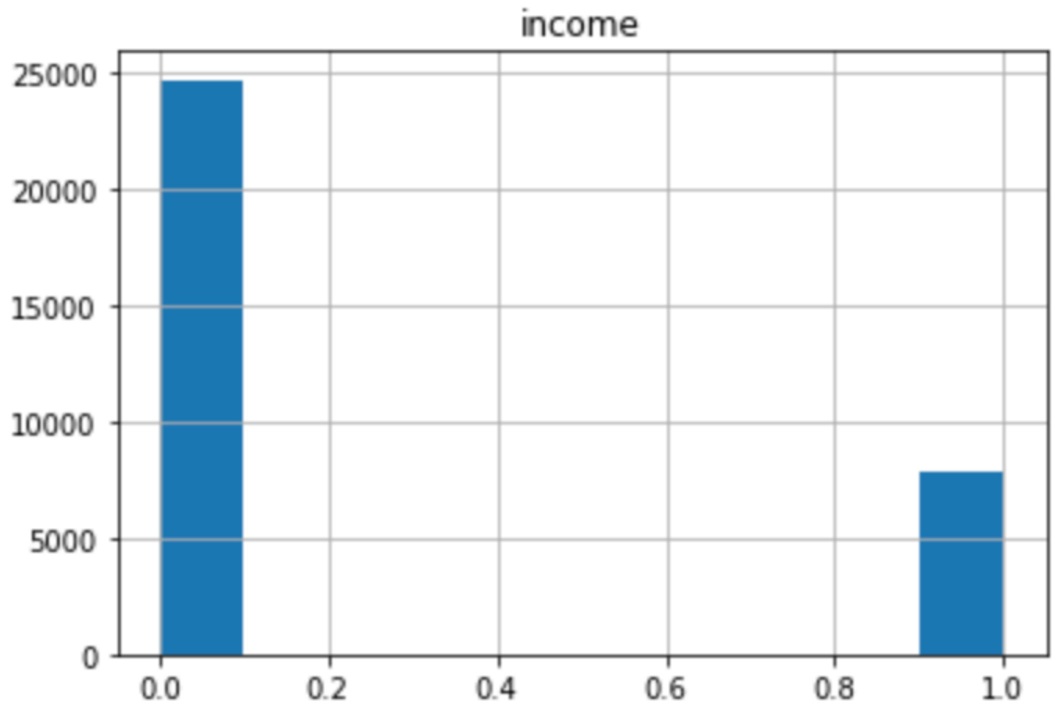
Since my data is split into two data types, I will do EDA on each of them separately. First up, we have type int64. With Seaborn, I created a correlation graph.



Interestingly, education.num can hours.per.week is the most correlated with an index of 0.15, and the second highest is capital.gain and education.num, which is 0.12. This does not depict too much useful information. However, I just want to know my dataset a bit better at this point.



On the other hand, for the categorical variables, I did 6 subplots to investigate the distribution of income above or below 50k within each category. Some of the interesting discoveries I noticed is the significant difference in income between females, unmarried, and own-children. Most of the other attributes are generally evenly split while the distribution of these three attributes includes some outliers.



The last graph that I am going to include is the distribution for my target variable "Income." The bar standing on zero represents the number of people with an income lower than 50k. The amount of people above 50k is on 1.0.

Part 3: Machine Learning

After studying the data with graph analysis, I began my machine learning process. Since the dataset has different data types, Hence, my first step is to format them into numerical values to make the data synchronized.

I have decided to use four different classifiers, Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting. The accuracy result is as follows.

	Accuracy
Logistic Regression	0.786581
Decision Tree	0.815907
Random Forest	0.855213
Gradient Boosting	0.866114

Evidently, the classifier that shows the best result is the slightly complicated Gradient Boosting classifier.

Features	Importance
relationship	0.387428
capital.gain	0.211737
education.num	0.202859
age	0.059260
capital.loss	0.059041
hours.per.week	0.038624
occupation	0.020671
marital.status	0.006907
fnlwgt	0.004529
sex	0.003059

To my interest, I also extracted the ranking of Importance of each attribute. Apparently, relationships are the most affecting attribute toward Income.

Part 4: Tuning Parameters

“Optuna is an open source hyperparameter optimization framework to automate hyperparameter search. “Compared to other methods such as randomizedSearch, the process of tuning could be extremely time consuming. Optuna can make this process much faster.

```
Accuracy: 0.8661139259941655
Best hyperparameters: {'n_estimators': 8, 'max_depth': 11.754432307225077}
```

Through this method, I found that the best accuracy I can receive is approximately 0.8661 with the best hyperparameters n_estimators = 8, and max_depth = 11.75.

Part 5: Conclusion

This is a very interesting project, and I feel like using it practically enhanced my understanding of the material. When I entered this class, I only had one introductory python class before, and sometimes, the material could be confusing. However, as I try to utilize the knowledge that I learned this year. I learned through my malfunctioning codes. Researching and exploring is my biggest takeaway from this project. At this point, I feel more confident applying the theoretical knowledge to practical applications in my upcoming internship.