**INCOME PREDICTION OF AN INDIVIDUAL**

**Hari Kiran Pendurthi**

**PROBLEM STATEMENT AND BACKGROUND:**

An enormous problem, particularly in the United States, is the stark wealth and income inequality. The likelihood of diminishing poverty is one valid reason to reduce the world's surging level of economic inequality. The idea of universal moral equality promotes nationwide economic stability and guarantees sustainable development.

So, the ability to predict income is crucial for governments in several areas of the nonprofit and corporate sectors. This will have a significant impact on a variety of areas, including marketing, business, and urban development, where the population's income distribution is an important element.

In order to address the issue of income inequality, this study demonstrates how data mining tools can be used. This has been accomplished using the UCI Adult Dataset. Based on a particular set of characteristics, classification has been done to determine whether a person's yearly income in the US falls in the income category of greater than 50K Dollars or less equal to 50K Dollars.

This study seeks to determine a person's earning potential based on a number of contributing factors. The struggle against economic inequality will be aided by this. The use of this model helps in the completion of a careful analysis that identifies the most important elements in improving an individual's revenue.

**DATA SET:**

The Census Income dataset has 48,842 entries. Each entry contains the following information about an individual:

**Age:** the age of an individual

* Integer greater than 0

**Work class:** a general term to represent the employment status of an individual

* Private, self-emp-not-inc, self-emp-inc, federal-gov, local-gov, state-gov, without-pay, never-worked.

**Final Weight:** Final weight in other words, this is the number of people, the celsus believes the entity represents

* Integer greater than 0

**Education:** The highest level of education achieved by an individual.

* Preschool, 10th Grade, Bachelors, Masters, etc.

**Education Num:** The highest level of education achieved in numerical form.

* Integer greater than 0

**Marital Status:** Marital status of an individual. Married civ spouse corresponds to a civilian spouse while Married AF spouse is a spouse in the Armed Forces.

* Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

**Occupation:** The general type of occupation of an individual

* Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-speciality, Handlers-Cleaners, Machine-op-inspect, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

**Relationship**: represents what this individual is relative to others. Each entry only has only one relationship attribute and is somewhat redundant with marital-status.

* Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

**Race**: Descriptions of an individual’s race

* White, Asian-Pac-Islander, Armer-Indian-Eskimo, Other, Black.

**Sex**: The biological sex of the individual

* Male, Female

**Capital Gain**: Capital gains for an individual

* Integer greater than or equal to 0

**Capital Loss**: Capital loss for an individual

* Integer greater than or equal to 0

**Hours per week**: the hours an individual has reported to work per week

* Continuous

**Native Country**: Country of origin for an individual

* United-States, Cambodia, England, India, Japan

**The Label**: Whether or not an individual makes more than $50,000 annually.

* <=50k, >50k

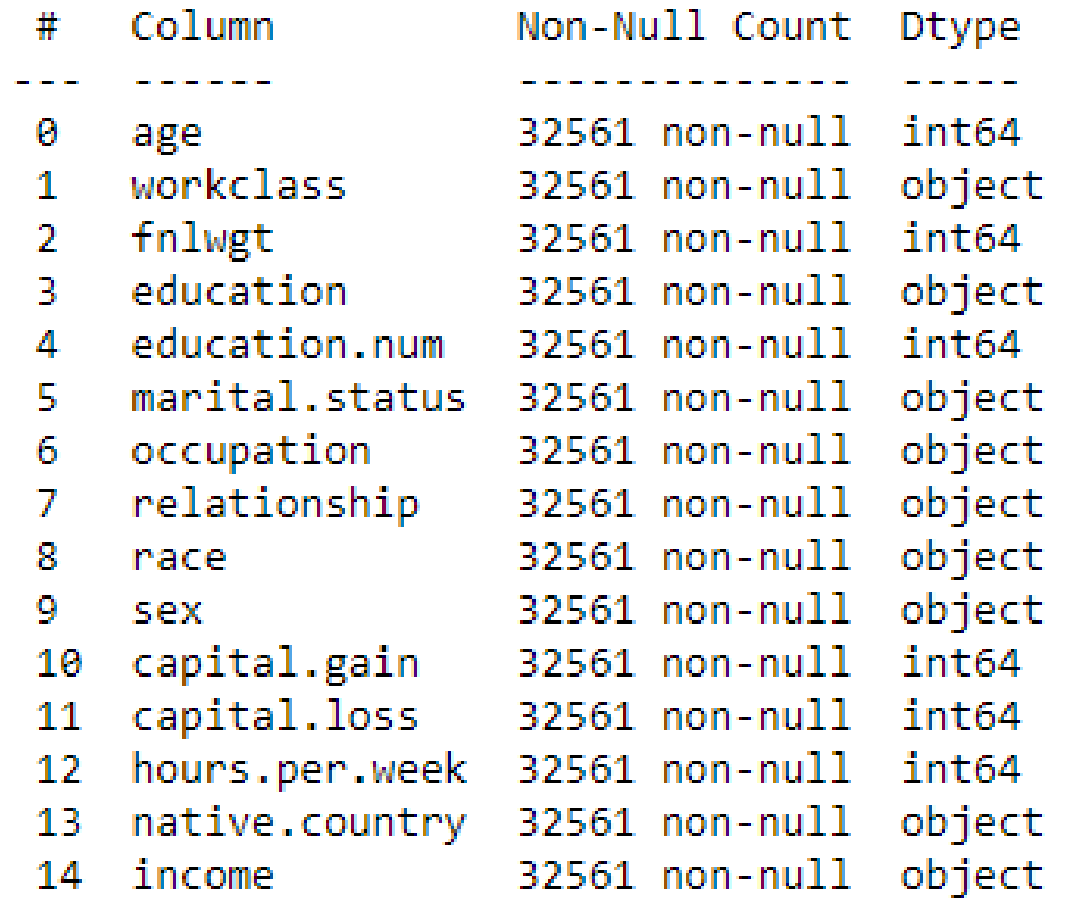
**METHODS**:

**DATA PREPROCESSING:**

1. **Data Selection:** After analysis of all rows and columns in the data we can see that the ‘education’ column and ‘education.num’ column represent the same thing. Hence we can remove one of the values. Later we have also removed the fnlwgt column since the values of fnlwgt are very much scattered and there is no proper correlation between the fnlwgt and income classification.



After the data selection, we started to look if we have any missing data(null values) in our adult census dataset. Looking at the summary we see that we do not have any null values.

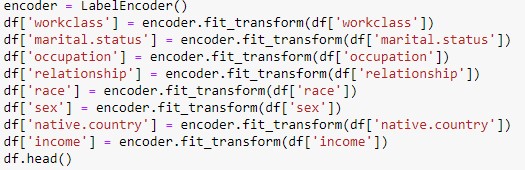


Upon further analysis we see that instead of null values we have ‘?’ as our data. All the cells which contain ‘?’ can be considered as missing values.

1. **Missing Data Imputation:** We initially started with removing all the rows with missing values but we have observed that we might lose some important information available in those rows. We have then used the ‘forward filling’ approach to replace the missing values with values from the next row.



1. **Handling Categorical Values:** Majority of the data in our dataset is categorical. Since algorithms do not handle categorical data well, we used the label encoder from sklearn to handle them.



1. **Data Normalisation:** Since the values of capital.gain and capital.loss range is high compared to all the other values in the dataset. We normalize these columns so that the algorithms can handle them well.



**ALGORITHMS USED:**

We have used the following 4 classification algorithms and analyzed their performance on the test data.

**Logistic Regression**: A classification model called logistic regression uses input variables (features) to forecast a categorical outcome variable (label), which can only have one of a small number of class values. As there are two alternative values to predict the adult income, i.e. <=50k and >50k , we employed logistic regression for classification. This statistical analysis method predicts a dependent data variable based on the prior observations of the data.

|  |  |  |
| --- | --- | --- |
| **Method** | **Accuracy** | **Training mean squared error** |
| Logistic Regression | 0.8109 | 0.1948 |

**Naive Bayes**: This classification method is based on the Bayes theorem with an assumption of independence among predictors. The Bayes Theorem determines the probability of an event given the probability of an earlier event.

|  |  |  |
| --- | --- | --- |
| **Method** | **Accuracy** | **Training mean squared error** |
| Naive Bayes | 0.8079 | 0.1976 |

**Decision Tree Classifiers**: This classification algorithm is used to compute a decision tree. Even with a wide range of training examples and a significant number of attributes in large datasets, this technique performs effectively. The trees are easy to modify and can be expressed as a set of rules that predicts the adult income.

|  |  |  |
| --- | --- | --- |
| **Method** | **Accuracy** | **Training mean squared error** |
| Decision tree classifier | 0.8134 | 0.0206 |

**Gradient Boosting Classifier**: Gradient Boosting Classifiers are algorithms which are used to develop a powerful predictive model using many weaker learning models. The main aim of using these classifiers is to reduce the loss and discrepancy between the actual class value and the training data. Especially for big complex data such as ours, Gradient Boosting Classifiers are essential. This algorithm is used to minimize bias error of the model. This model can be used to predict both continuous target variable and categorical target variable i.e. regression and classification.

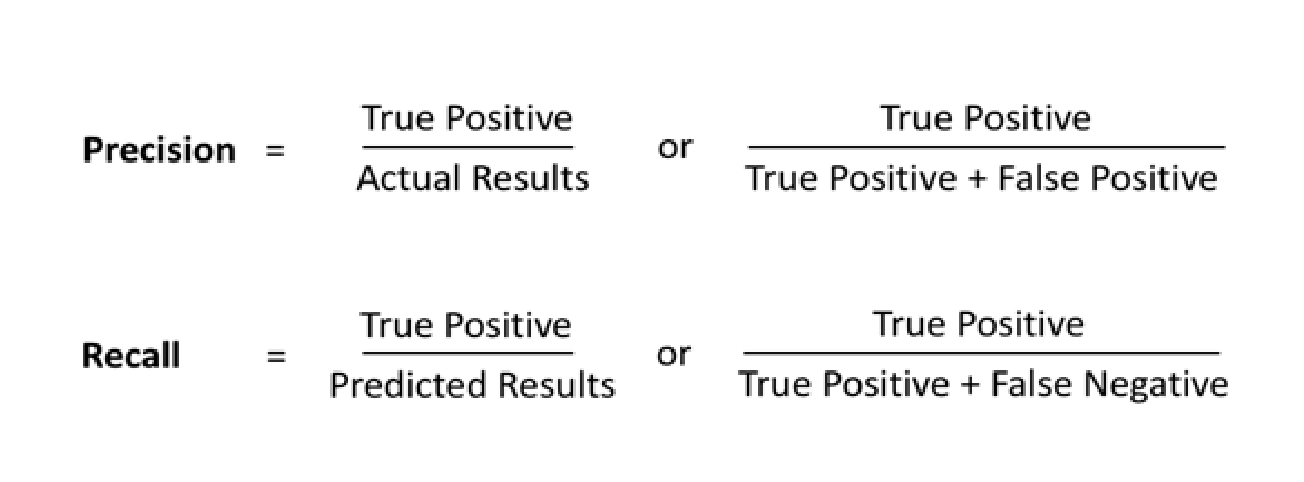
|  |  |  |
| --- | --- | --- |
| **Method** | **Accuracy** | **Training mean squared error** |
| Gradient boosting classifier | 0.8688 | 0.1319 |

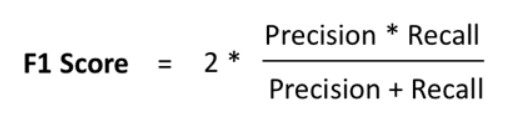
**RESULTS:**

After the analysis of the above 4 algorithms we can see that the baseline model for Logistic Regression, Naive Bayes and Decision Tree Classifier provide an accuracy of 80-81% on the test data. However the baseline model of Gradient boosting classifier provides an accuracy of 87% on the test data. Also, we can see that in the decision tree the training error is 0.02 but the accuracy is less. This means that the model has been overfitted and it might not work well for new and unseen data.

Based on this initial analysis we can figure out that the decision tree classifier might not be suitable for this problem. We then carried on to compare the precision, recall and F1 scores for the remaining 3 algorithms(Logistic Regression, Naives Bayes, Gradient Boosting Classifier).

Precision is nothing but the ratio of useful (relevant data) data to the actual results obtained, whereas the recall indicates the ratio of useful data to the results which are obtained by the algorithms. Recall calculations are based on the specific algorithm used. F1 Score is calculated by the harmonic mean of both precision and recall.





The above figure is the formulas used to calculate precision, recall and F1 score.

The obtained calculations of each algorithm(Logistic Regression, Naives Bayes, Gradient Boosting Classifier) based on our data is mentioned below:

* **Logistic Regression**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 score** |
| <=50k | 0.84 | 0.93 | 0.88 |
| >50k | 0.64 | 0.42 | 0.50 |

* **Naive Bayes**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 score** |
| <=50k | 0.83 | 0.95 | 0.88 |
| >50k | 0.67 | 0.33 | 0.44 |

* **Gradient Boosting Classifier**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 score** |
| <=50k | 0.89 | 0.95 | 0.92 |
| >50k | 0.77 | 0.61 | 0.68 |

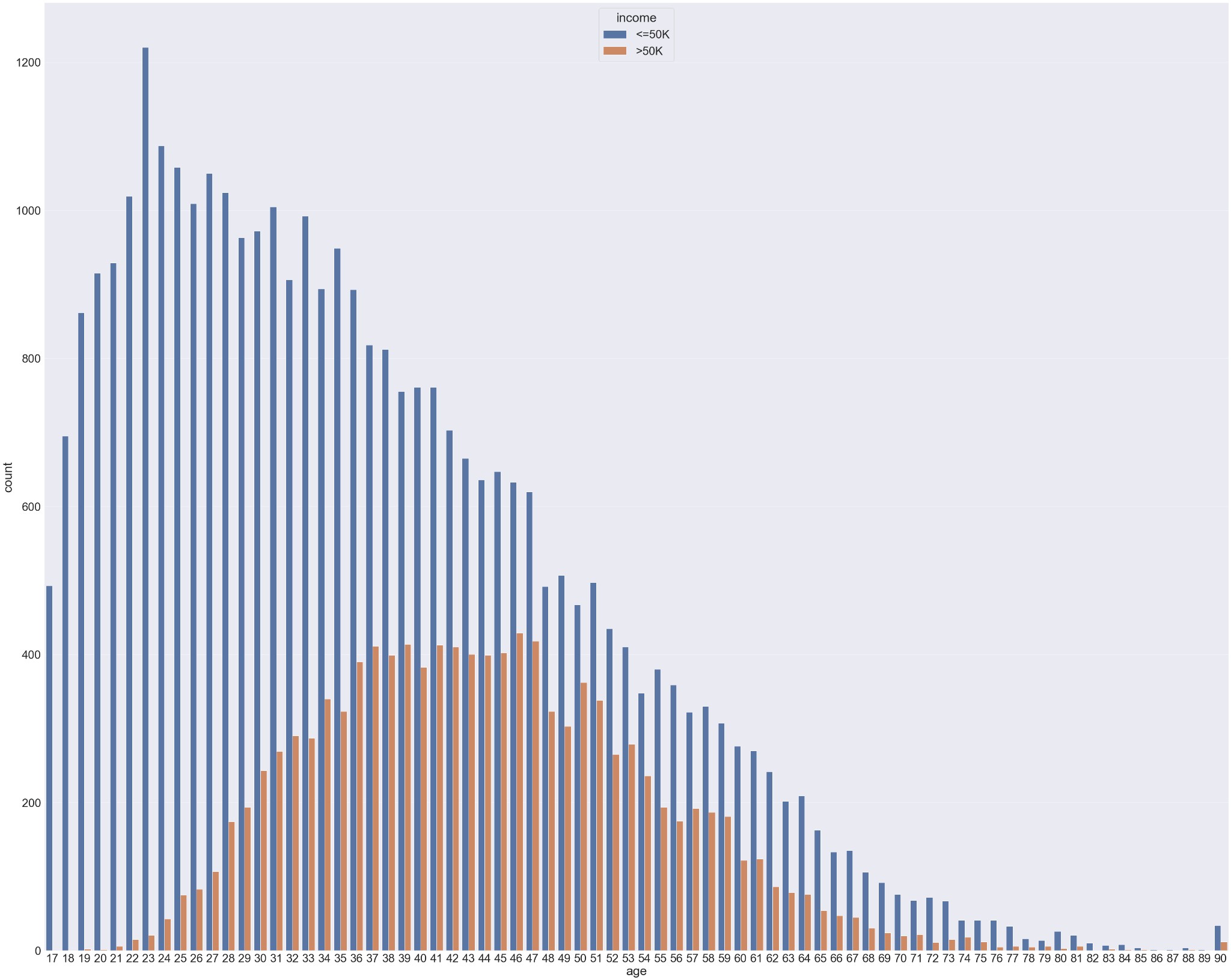
The precision and recall for each class are highest for the gradient boosting classifier. The same pattern is shown during the analysis of F1 score as well. An F1 score of 0.92 for gradient boosting classifier is the highest among the 3 models. As we know that a high F1 score represents an efficient model and since the other metrics also incline positively towards Gradient boosting classifiers we have decided to tune the Gradient boosting classifier and try to get a better accuracy.

We have tuned the following parameters for this gradient boosting algorithm

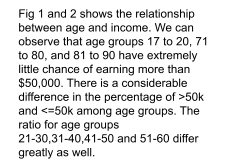
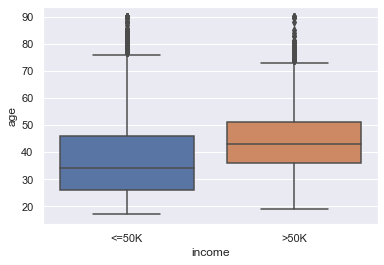
* Learning\_rate
* N\_estimators
* Max\_depth

The accuracy seems to bump up a bit while changing the max\_depth to 7 or 8. When we use the parameters as depth = 3, learning\_rate=0.1 and n\_estimators=500 the accuracy on the new and unseen data tends to bump a bit.

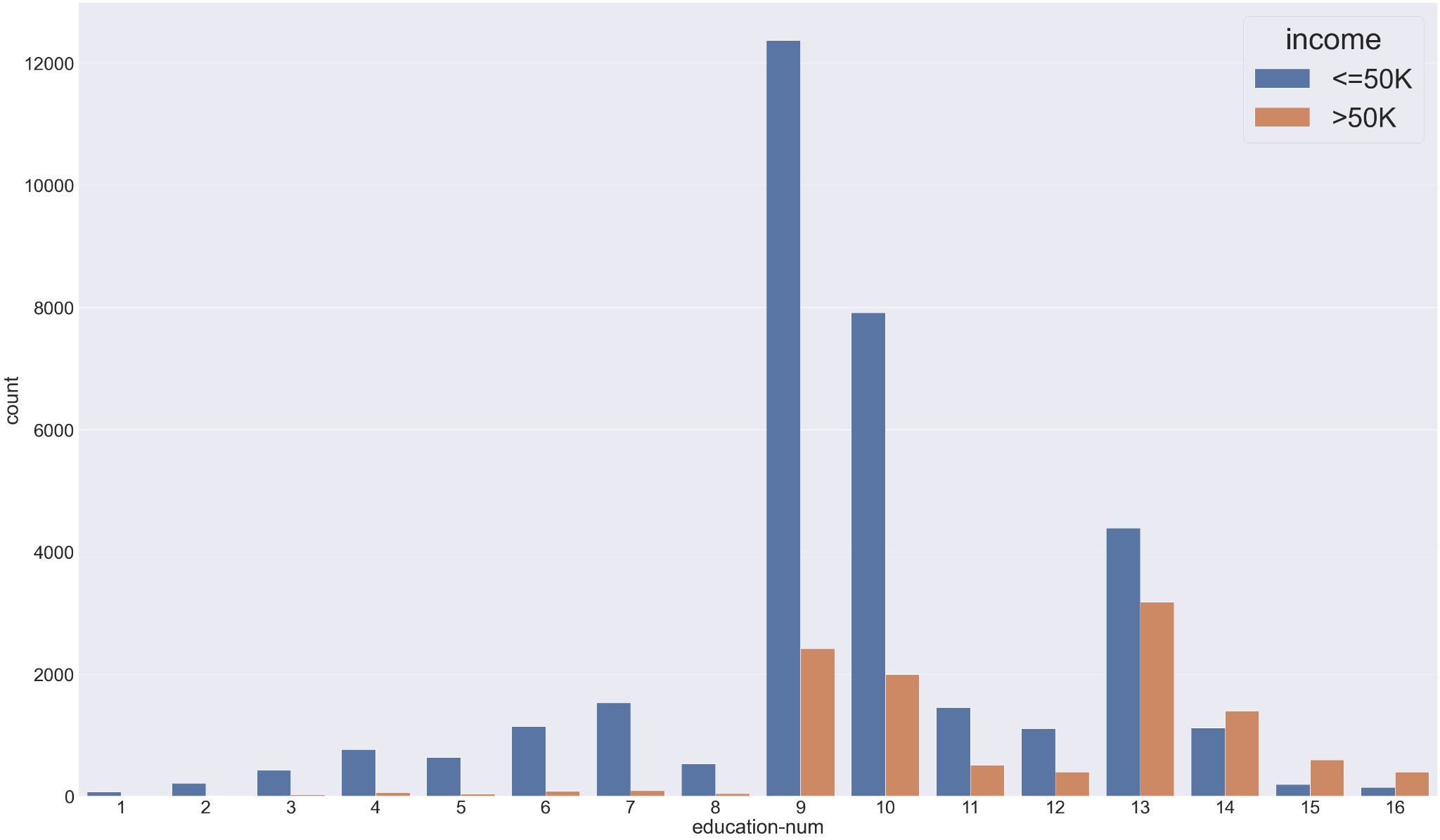
**DATA VISUALIZATION:**



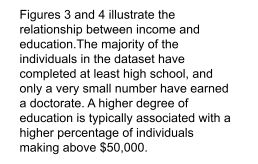
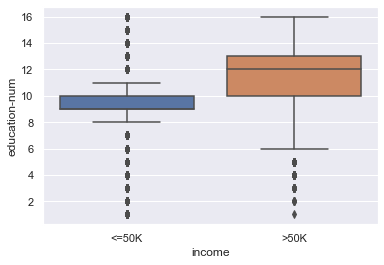
**Fig-1**



**Fig-2**



**Fig-3**



**Fig-4**

**LIBRARIES USED:**

1. Pandas - Pandas is an open-source Python bundle that is most broadly utilized for information science/information examination and AI undertakings. It is based on top of another bundle named NumPy which offers help for multi-dimensional clusters.
2. NumPy - NumPy is a library for the Python programming language, adding support for enormous, multidimensional clusters and lattices, alongside a huge assortment of significant level numerical capacities to work on these exhibits.
3. Matplotlib - Python scripts can be used to create 2D graphs and plots using the Matplotlib module. It has a module named pyplot which makes things easy for plotting by providing features to control line styles, font properties, formatting axes etc.
4. Sklearn - Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. Through a Python consistency interface, it offers a variety of effective tools for statistical modeling and machine learning, including classification, regression, clustering, and dimensionality reduction.
5. Seaborn - A matplotlib-based Python data visualization library is called Seaborn. It offers a

sophisticated drawing tool for creating attractive and instructional statistical visuals.

**LESSONS LEARNED:**

* 1. Despite being easily available, data isn’t always simple to use. In order to use data to make any predictions, there are a number of pre-processing steps that need to be performed.
  2. Every algorithm can be used for the dataset but there will be few algorithms that are accurate for that specific dataset.
  3. Hyperparameter tuning might increase the efficiency of an algorithm.
  4. While visualizing the data, the correlation between the independent and dependent data can be identified.
  5. Measures like accuracy on the test data and training error can be used to determine whether the model is to be over-fitted or under-fitted.