****

**Adult dataset Final report**

Mentor:

Muppidi Srikar

By:

Ajay M

Anil Gugulothu

Harika Kanthi P

Laxminarayana Adepu

Utthejh Gade

Vivek Menon

**table of contents**

**1. Introduction** 3

**2. dataset** 4

**2.1 Hierarchy of features** 5

**3. data exploration (EDA)** 7

**3.1 Univariate Analysis** 7

**3.2 BIVARIATE Analysis** 8

**3.3 Multivariate Analysis** 11

**4. Data Pre-processing** 6

**4.1 Data Cleaning** 2

**4.1 Data Preparation** 2

**5. Model Building** 2

**5.1 Introduction to models** 3

**5.2 Model building** 3

**5.1 Model building** 3

**5.1 Model building** 3

**1. Introduction**

The adult dataset was extracted by Barry Becker from 1994 Census Data which is collected by the US Census Bureau. The Census Bureau is responsible for producing data about the American people and economy.

The objective of this dataset is to predict the income (target variable). This is a binary classifier. An individual is classified to have income above $50K or below $50K. Typically, income of an individual depends multiple factors such as age, education, occupation, marital status, etc. All of these are present in the dataset along with other factors which shall be discussed later.

One of the possible use cases is to help government agencies make decisions during developing income-based policies. For example, Affordable Care Act. Affordable Care Act was signed into law by President Obama in March, 2010 to help low income families afford healthcare insurance by government subsiding the premium payments. This can be also applied by the government during the COVID-19 crisis where the government can find out who individual can potentially afford to pay for the treatment.

It is also important to understand from economics perspective and how this dataset applies to today’s economic situation. In 1994, when the dataset was collected the median household income was $34,076. In 2019, the median household income is $ 63,030. The Affordable Act Care cut off for Household income is $85,320 for a family of three. That means, for a family of three the as long as the household income is below $85,320, they will qualify for this government assisted program.

**2. Dataset**

The dataset contains 32,561 observations with 15 features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1) Age | 2) Capital Gain | 3) Capital Loss | 4) Education | 5) Education Num |
| 6) Fnlwgt | 7) Hours per Week | 8) **Income** | 9) Marital Status | 10) Native Country |
| 11) Occupation | 12) Race | 13) Relationship | 14) Sex | 15) Workclass |

1. Age: Numerical continuous feature which refers to the age of individual. From 17 years to 99 years.
2. Capital Gain: Numerical continuous feature which refers to the profit from selling investments. Only a few individuals have capital gains.
3. Capital Loss: Numerical continuous feature which refers to the loss from selling investments. Only a few individuals have capital loss.
4. Education: Categorical discrete feature which refers to the highest education level of the individual. This feature contains values from pre-school to doctorate.
5. Education num: Numerical discrete feature which refers to the number of years in education the individual has.
6. Fnlwgt: Numerical continuous feature which refers to the weighted value calculated by the Census Bureau.
7. Hours per Week: Numerical continuous feature refers to the number of hours the individual worked per week.
8. **Income: Target Variable**. Categorial discrete. Takes on two values <=50K and >50K.
9. Marital Status: Categorial discrete feature which refers to the marital status of the individual. Values such as
10. Native Country: Categorical discrete feature which refers to the native country of the individual. Values such as: United States, Mexico, India, etc.
11. Occupation: Categorical discrete feature which refers to the occupation of the individual. Values such as: Admin-clerical, Armed Forces, Executive, etc.
12. Race: Categorical discrete feature which refers to the race of the individual. Values such as White, Black, American-Indian-Eskimo, Asian Pacific Islander, Other, etc.
13. Relationship: Categorical discrete feature which refers to the relationship of the individual in the family. Values such as Husband, Wife, Own-child, Not-in-family, Other-relative, or Unmarried.
14. Sex: Categorial discrete feature which refers to the sex of the individual. Values: Male or Female.
15. Workclass: Categorical discrete feature which refers work status of the individual.
16. These are features in our dataset. Let’s have a look at the hierarchy of the values to understand them further.

**2.1 Hierarchy of Values**

**1) Marital Status**

* Hierarchy in Marital Status tells us about which income class they belong to4
* Never Married tend to be young crowd and typically end up in class 0
* Married will dominate class 1 as married tend to earn more than never married

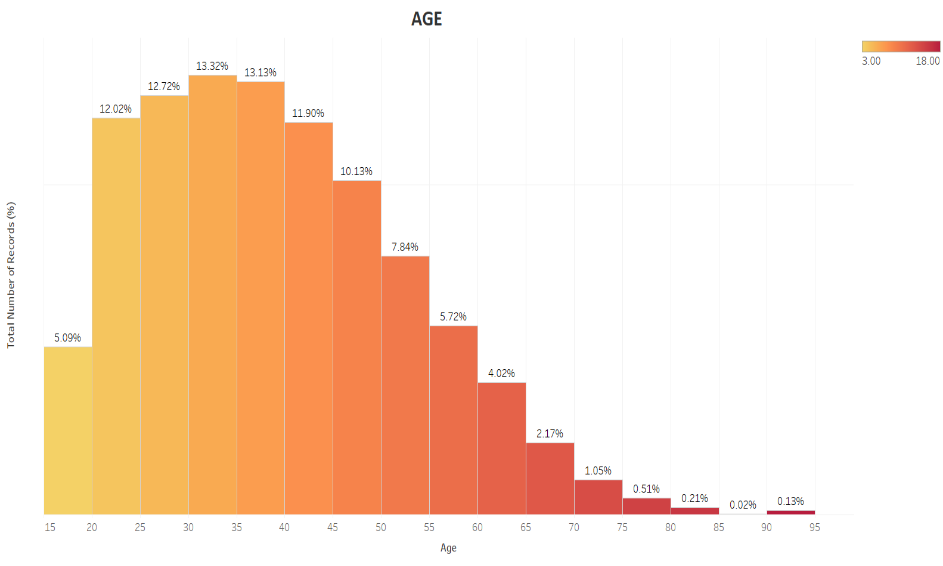
**2) Occupation**

* White-Collar refers to salaried professionals who work in office and management
* Blue-Collar refers to members of the working class who perform manual labour and earn hourly wages
* Gold-Collar refers highly skilled knowledgeable people who combine intellectual labor with manual labour

**3) Education**

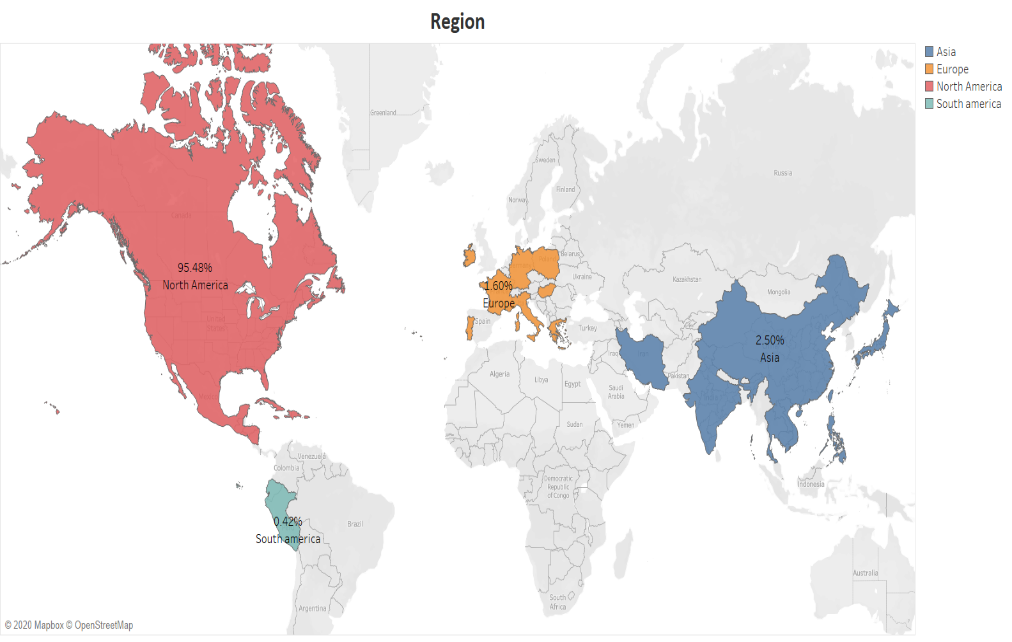
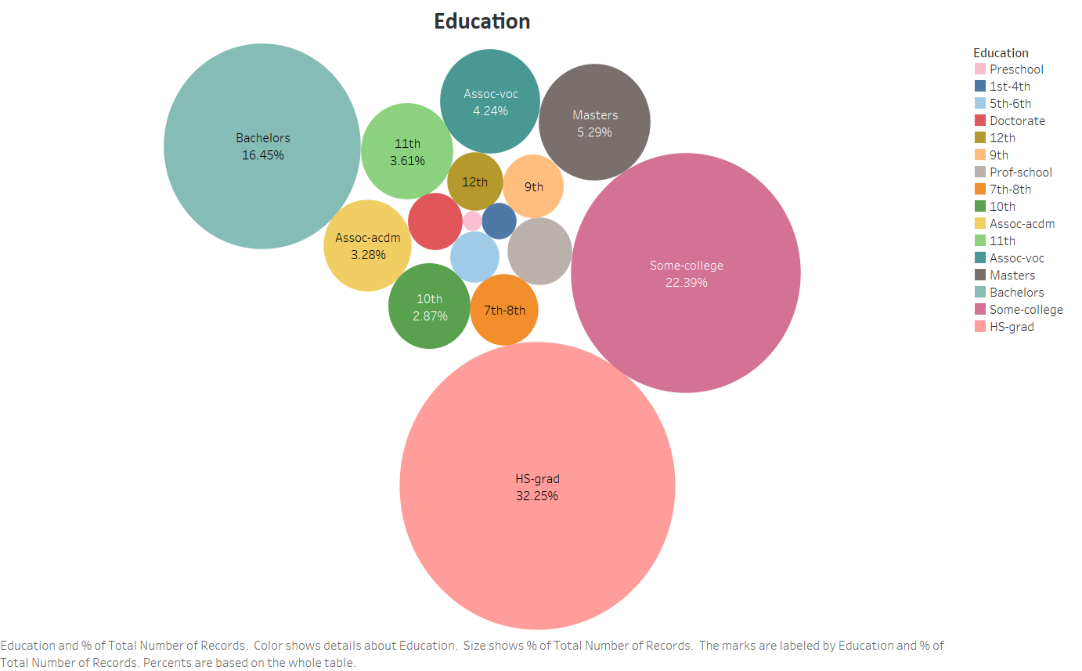
* Hierarchy in Education is important to understand the relationship between education and Income
* University > College > High School Grad > Non-High School Grad > Pre-school
* University: Professor-school & Doctorate > Masters > Bachelors
* College: Associate Academic > Associate Vocational > Some-college
* Non-HS Grad: 12th > 11th > 10th > 9th > 7th – 8th > 5th – 6th > 1st – 4th

**3. Exploratory Data Analysis**

**3.1 Univariate Analysis**

Age: Majority of the dataset lies from age 25 to 45. Which represents most of the individuals in our dataset and it is representative of the workforce.

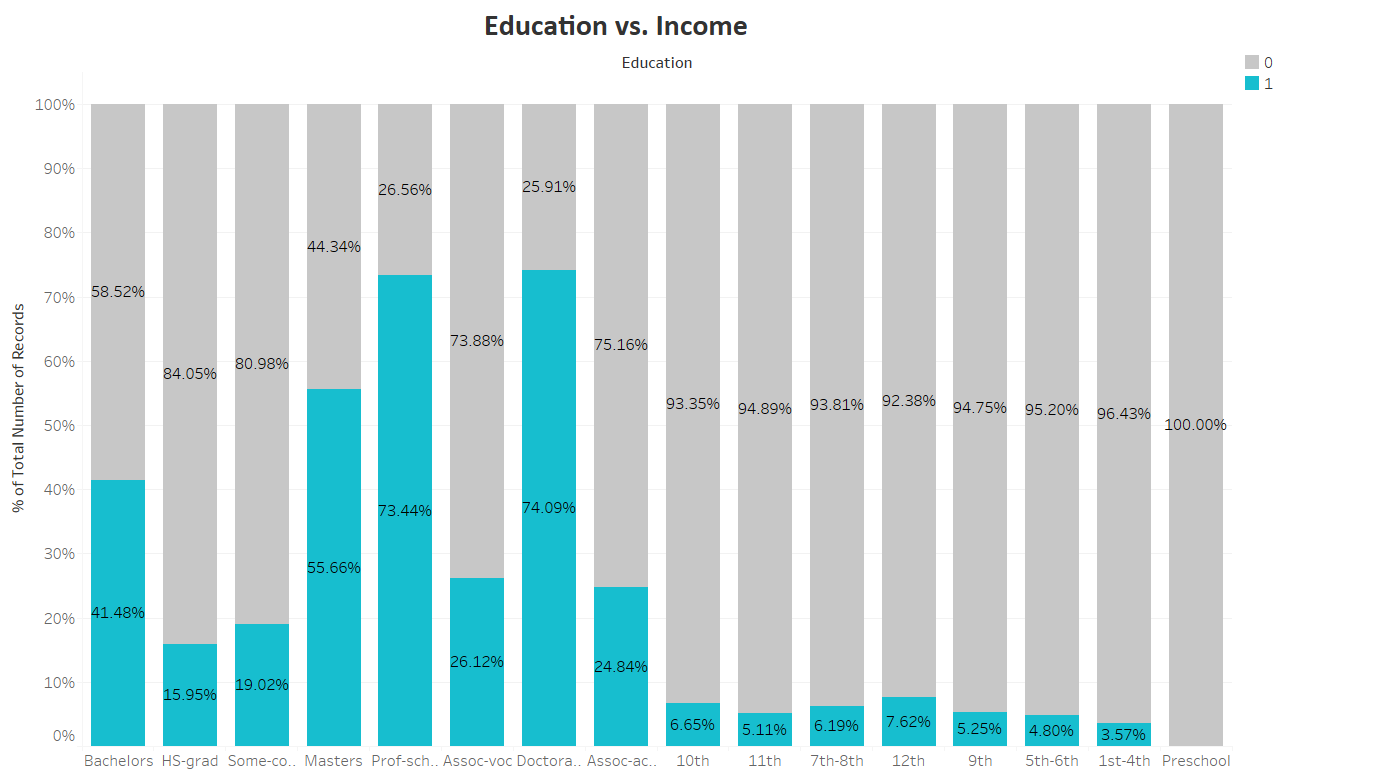
Gender: In our dataset we can see that the distribution for males is 67% and females is 33%.



Region: Native country data was divided according to the region as it would be easier to visualize and understand where the majority of the individuals belong. Since, this is US dataset majority of the individual’s native country is North America (95.48%), followed by Asia at 2.50%, Europe at 1.60%, and South America at 0.42%.

Education: Education is a critical metric to determine an individual’s income. The majority of the data points belong to HS-grad (32.25%), followed by some-college (22.39%), Bachelors (16.45%), higher degrees such as Masters are at 5.29%, and doctorate (1.27%).

**3.2 Bivariate Analysis**

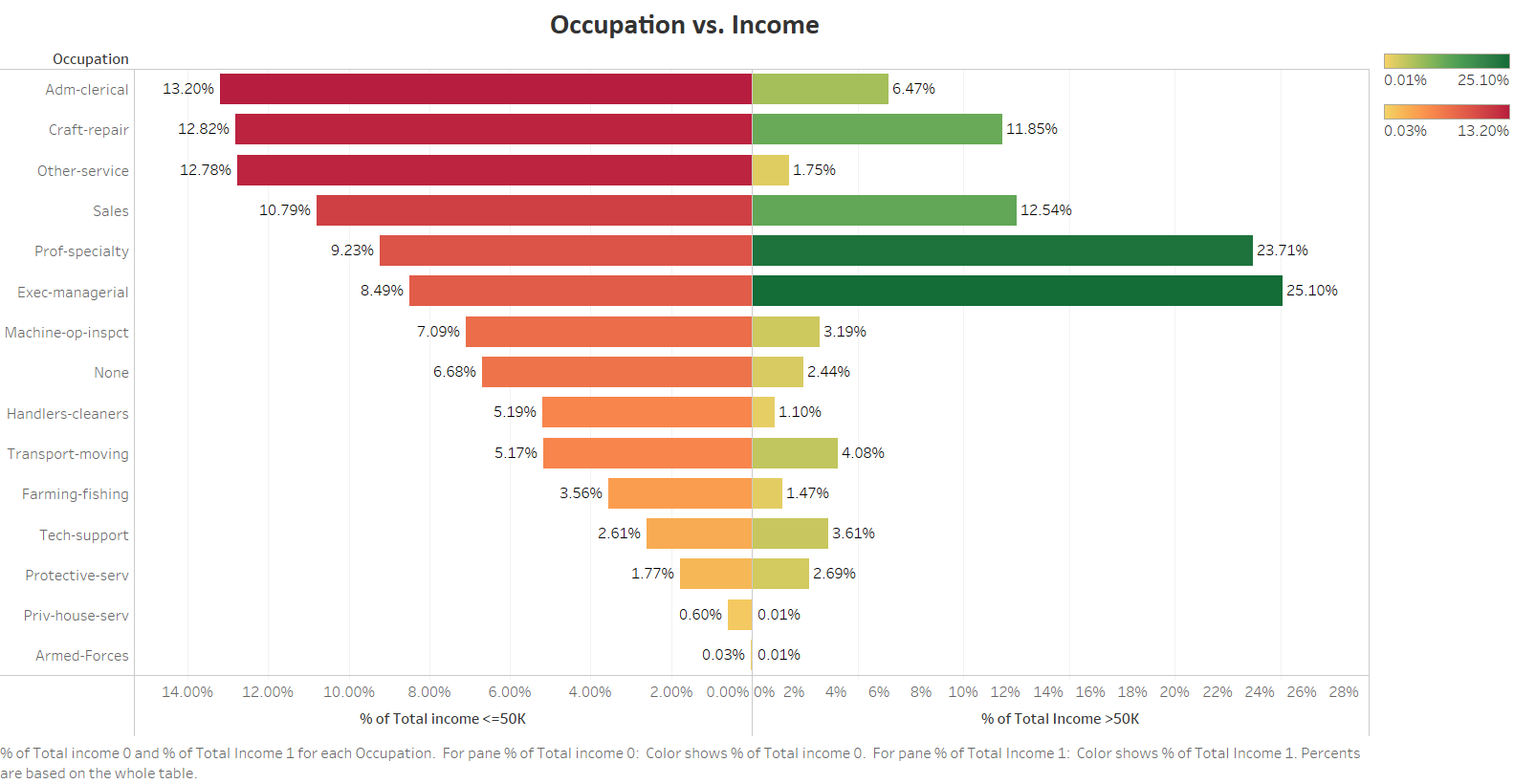


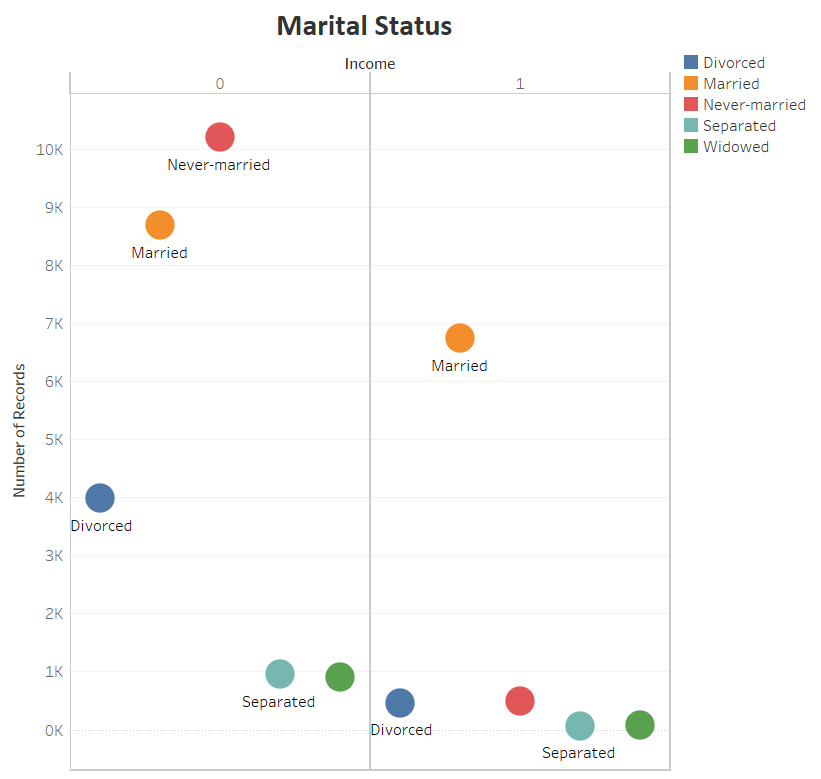
**Education vs. Income**:

This stacked bar displays the relationship between Education and Income. We have education on the x-axis and % of records in the y-axis. Grey area indicates records <=50K (class 0) and blue represents records >50K (class 1).

From the graph we can observe that individuals who have Doctorate as their highest education, 74.09% of their records belong to class 1 and 25.91% belong to class 0. For Professors at universities, 73.44% of the records belong to class 1 and 26.56% to class 0. High School Grad (HS Grad) has the highest number of records and only 15.95% records belong to class 1 and 84.05% records to class 0. For Some-college, 19.02% of the records belong to class 1 and 80.98% of the records belong to class 0.

This proves that there is a strong relationship between the highest education level and potential to earn above 50K. This also go with logic as the level of education increases so does the earning potential.

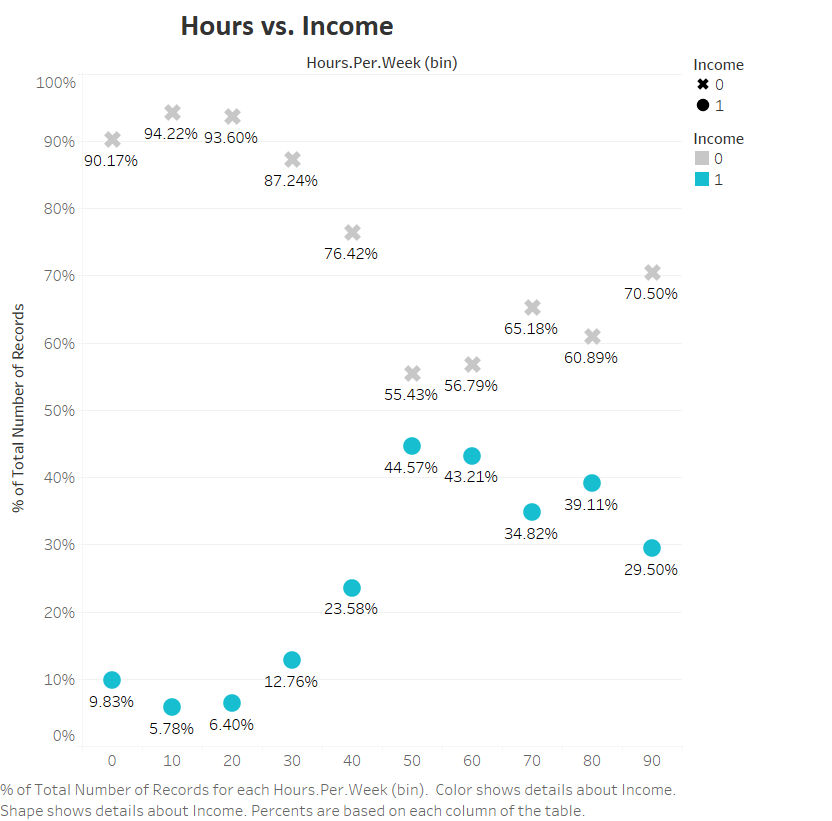
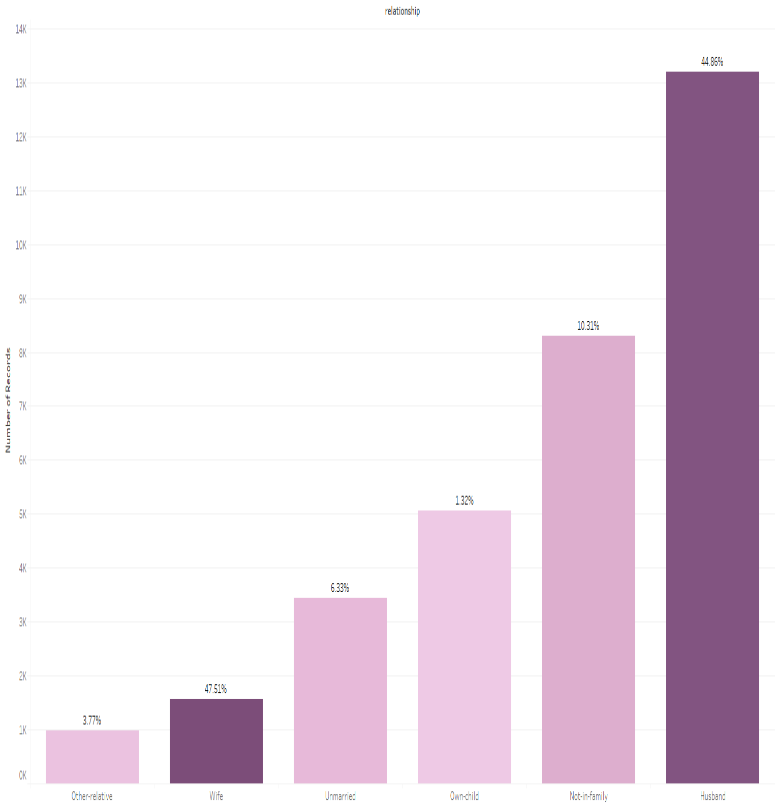
O

**Occupation vs. Income:**

Higher the occupation (according to the hierarchy previously mentioned) higher the number of records belong to class 1. On x-axis we have records belonging to class 0 (<=50K) on the left and records belonging to class 1 (>50K) on the right. We can see Exec-Managerial leading class 1 at 25.10%. Exec-managerial includes positions such as CEO, CFO, Vice-Presidents, Managers etc. Followed by Prof-Specialty at 23.71%, this includes college professors who tend have high degrees such as doctorates and typically are high earners. In Class 0, we have adm-clerical leading the number of entries at 13.20% of class 0 records. These people typically perform administrative work at office. Followed by Craft-repair and Other services which require manual labour work and tend to earn less. The pattern we see here is similar to education. Higher job positions more records in class 1.

**Marital Status vs. Income:**

On x-axis we have class 0 on left side and class 1 on right side and on y-axis we have number of records. Majority of the records in class 1 belong to Married, as married individuals tend to earn more than never-married members. Class 0 is dominated by Never-married as this group tend to be younger people.



**Hours vs. Income:**

On x-axis we have number of hours and on y-axis we have % of number of records. First, we notice an upward trend as the number of hours increase the number of records which belong to income class 1 (>50K) also increases. The same can be said for class 0 as well, as the number of hours increase the number of records for class 0 (<=50K) decrease. Which goes with the general notion that as the number of hours increase the more income a person makes.

In USA, 40 hours is the average number of hours an individual works on a typical work week. For example, individuals who works 20 hours a week, 6.40% of the records belong to class 1 and 93.60% of the records belong to class 0. For 40 hours, 23.58% records belong to class 1 and 76.42% records belong to class 0. For 60 hours, 43.21% belong to class 1 and 56.79% records belong to class 0.

Clearly, there is an upward trend as the number of hours the percentage of records belonging to class 1 also increase. However, if you notice after 70 hours there is a drop and there is even a steeper drop at 90 hours. This is due to self-employed individuals and manual workers tend to work more hours. As manual labour job tends to pay less than typical corporate job. Self-employed salary could be low due to it being a startup as well.

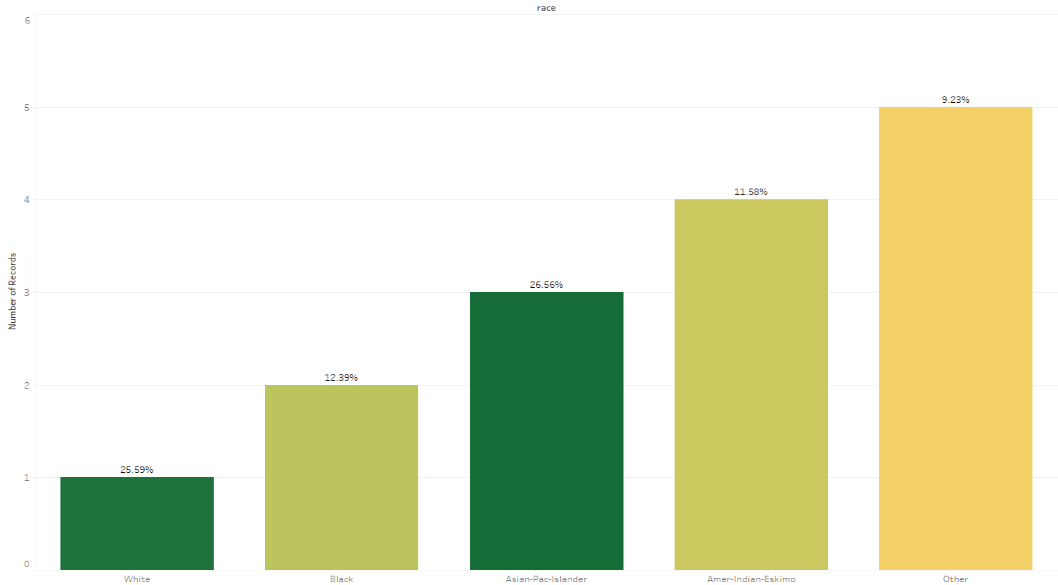
**Relationship vs. Income:**

The graph plotted with number of records against relationship shows that husbands occupy about 40% of the data. Husbands and wives earn more as compared to any other relationship. This again shows that the married people tend to be in the higher income group. Of the wives 47.51% are in the higher group and of the husbands 44.86%. The wives and husbands are usually the bread winners of the family and can be seen to earn a better income as compared to other relations.

|  |  |
| --- | --- |
|  |  |

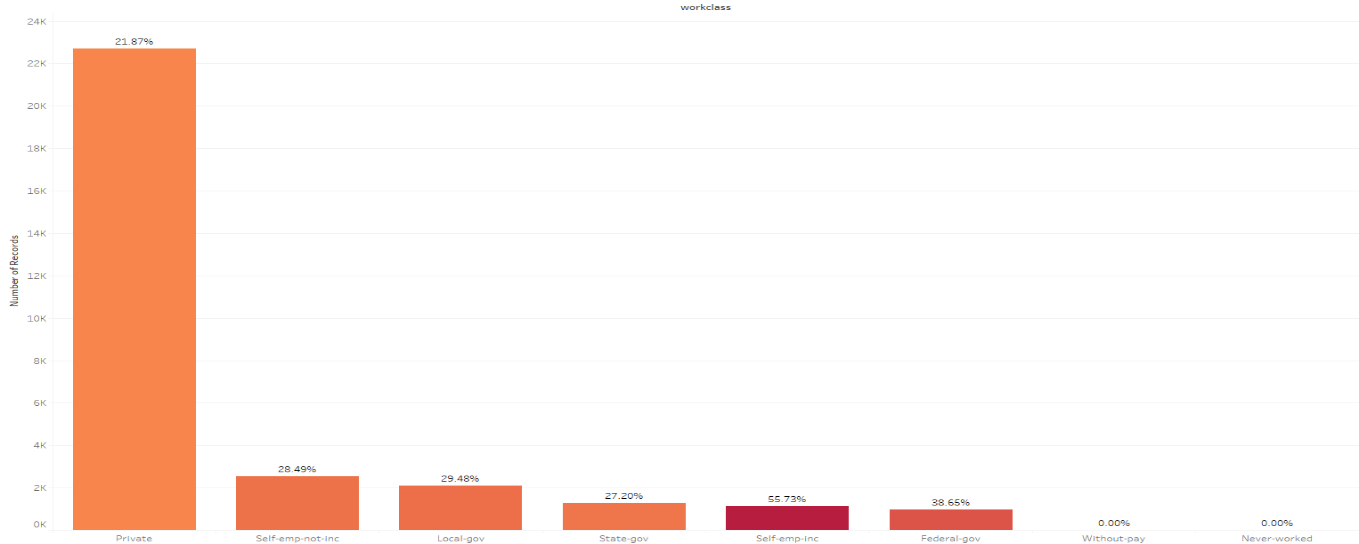
**Capital loss vs. income (left) & Capital Gain vs. income (right):**

show a similar pattern. Most of the capital gains and capital losses are zero while very few individuals earn this kind of income. As the level of capital gains and capital losses increase the income class percentage i.e. the percentage of people with income greater than $50000 also increases. For any individual earning more than $8000 as capital gains will be in the higher income group.



**Race vs. Income :**

Whites occupy the greatest number of records in the dataset followed by blacks. Whites are about 85% and of which 25.68% accounts to the higher income class. Others are the least both in terms of number of records and the income class percentage. Asian-Pac-Islanders have the highest income percentages followed by whites with 25.59%.

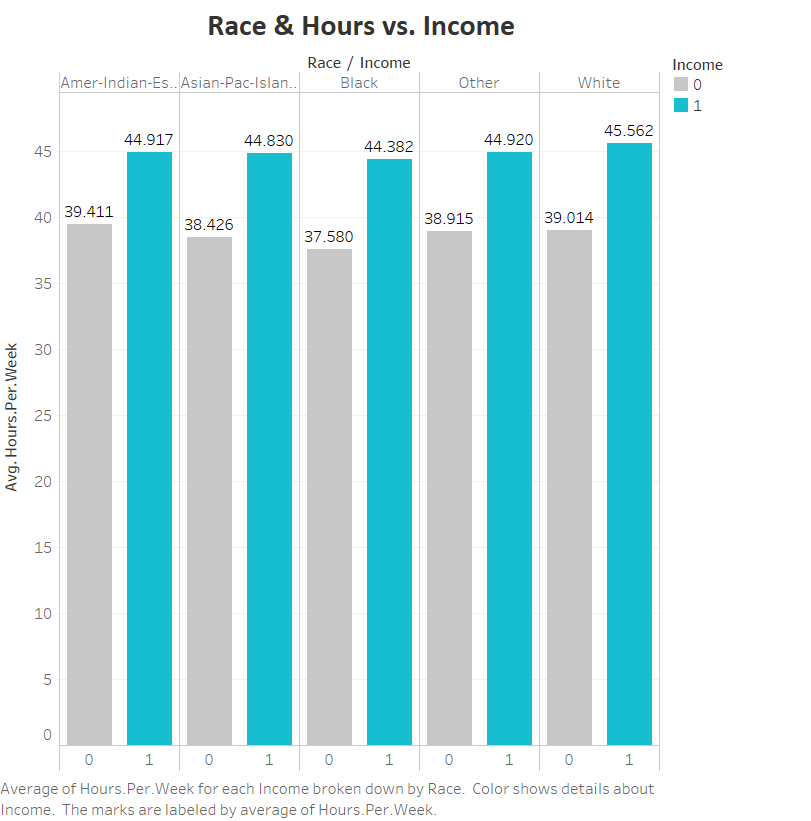


**Work class vs. Income**:

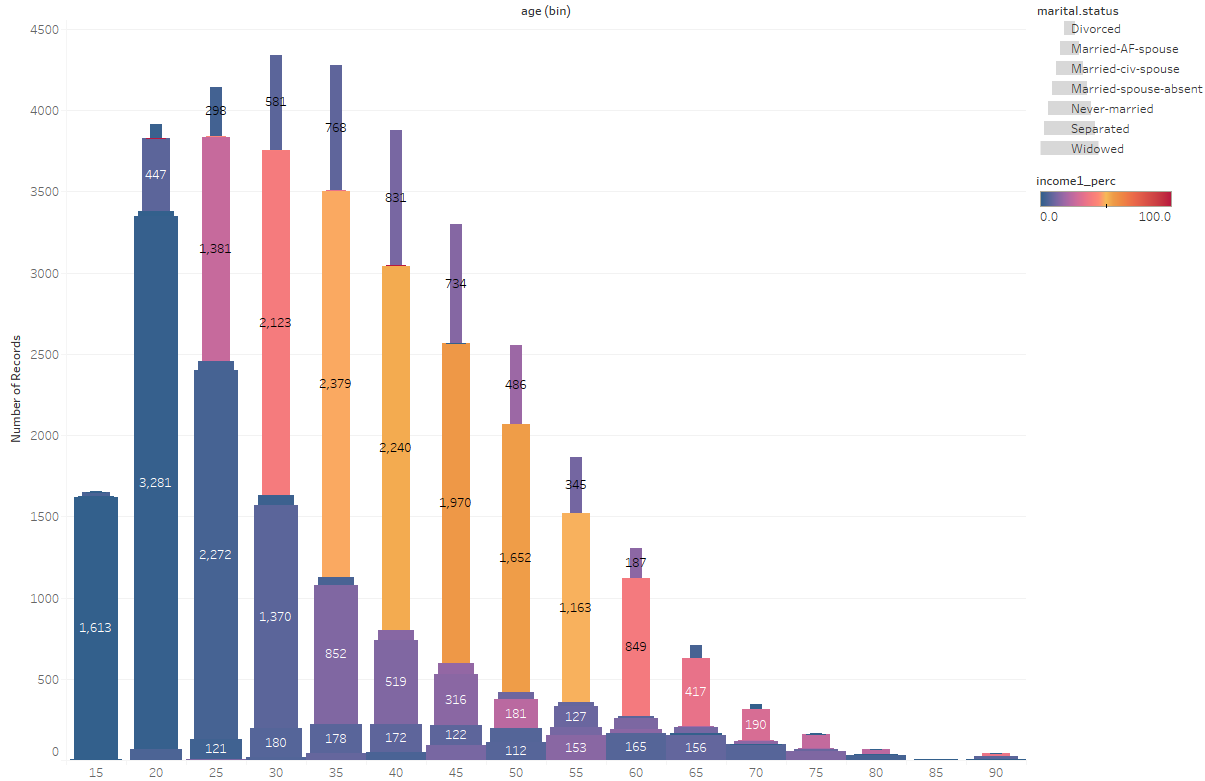
The private sector is the highest because of all individuals prefer a private company instead of government company for the availability and the income earned. Similarly, they prefer to work in already existing companies rather than start a new one.

This pattern can be seen in the above graph which is plotted with relationship against the number of records. Self-employed individuals tend to earn more as compared to those in other sectors like government and private. Government also has a higher income level percentage as to private. The without pay and never worked are always shown to be in the lower income group.

**3.3 Multivariate Analysis**



**Race & Hours vs. Income:** The plot race against hours seperated by the income levels , <= $50000 (grey) and > $50000(blue) shows that the individuals irrespective of their race tend to put in more than 45 hours per week for the higher income while those with lower income have an average of less than 40 hours per week



**Age & Martial Status vs. Income**:

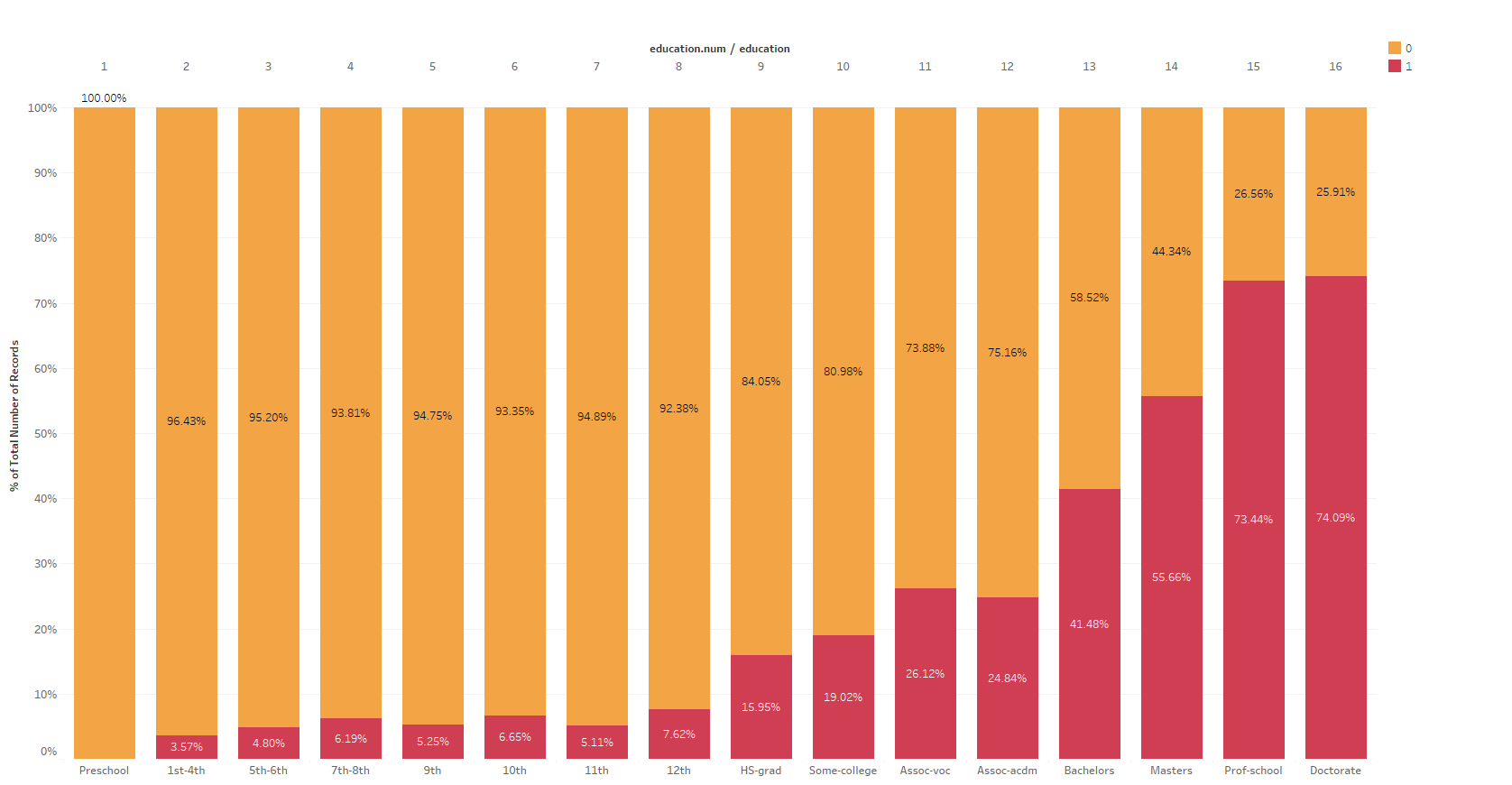
The age distribution plot which is divided by the different status of their marriage (marital status) shows never married part of the data lies below 30 years and very few remain remain unmarried after 30 years. They also had lower levels of income than compared to the married part of the dataset. The divorced rate also did increase towards 45 years and then gradually decreased as well. The married people earn a higher level of income as seen. The married civ spouse had higher income level percentages as the years progressed and always had 25% of them always in the higher income class.All the individuals below the age of 20 always had a lower income it is because they have just started earning income.

**4. Data Pre-processing**

**4.1 Data Cleaning**

**I. Education number vs. Education**

Adult Census Dataset consists of 15 features namely age, work class, occupation, marital status, relationship, education, education num, hours per week, capital gain, capital loss, race, gender, fnlwgt and income. Income is the target variable which is to be predicted.

The features education num is described to be a number assigned to the educational levels in the feature education. From the data analysis of education against income shows that as the level of education increases the level of income does too. The education number would be a better feature to be used for further analysis since the number plays a role in determining the income level.

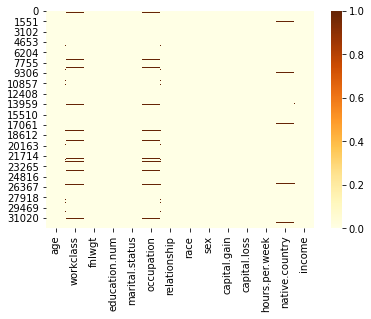
**II. Misclassification**

|  |  |  |
| --- | --- | --- |
| RELATIONSHIP | GENDER | NUMBER OF RECORDS |
| *Husband* | *Female* | *1* |
| *Wife* | *Male* | *2* |

The dataset contains instances where the gender and the relationship of the individual did not make sense. Husband being a female is not a possible combination and similarly neither is a wife and male. The gender of these observations have to be classified according to their relationship.

**III. Null Values**

There is no duplicity in the dataset. The dataset also contains some missing values. The missing values are represented as “**?** “ in the dataset. The graph of missing values in each feature shows about that there are missing values in three features. The features work class and occupation contain 5.6% missing values and are simultaneously missing. The native country also contains around 2% missing values in the dataset.



There a few approaches of dealing with these null values.

1. Dropping them from the dataset
2. Selecting only the USA part of the data because the census consists of 90% of USA individuals
3. Most frequent value imputation, in other words mode imputation
4. Imputation using K nearest neighbors

**4.2 Data Preparation**

1. **Data Scaling**

The numerical features age, education, hours per week are around the same line while capital gain and capital losses have very high values. Though most individuals tend towards zero of those who earn capital gains and losses have very high values. The highest value in capital gains was 99,999 while in case of losses was 4,356. We scaled down these two features using standardization.

Standardization is the process of putting different variables on the same scale. This allows us to compare the variables and will not emphasize more only on those with higher scales.

Usually, to standardize variables, we calculate the mean and the standard deviation of the variable. We then, for each observation in the variable subtract the mean and divide by standard deviation. This process produces values that are in terms of its standard deviations. For instance, if a variable has mean 10, standard deviation 10, and the observation has a value 50, that means the variable is (50-10/10) = 4 standard deviations above the mean. The interpretation is the same irrespective of which variable we standardize.

**II. Feature Scaling**

This process is where we select those features which contribute most to our target variable or output which in our case is the income class. Having irrelevant features in the data can decrease the accuracy of the models and adds unnecessary noise to the data. This makes our model to be equally dependent on those irrelevant features which in turn affects the model performance as a whole.

The statistical testing of the features was one of the feature selection methods. P value in simple words is the probability of null hypothesis being True. It is a measure to prove whether the statistical null hypothesis we have assumed is True or False. It is also dependent on level of error we can tolerate i.e. Level of significance. For the dataset we can accept about 5% error i.e. Level of significance or 95% level of confidence. That means any feature that has a p value of less than level of significance (0.05) is considered to be statistically significant else not. The common hypothesis we assume for our dataset:

*Null Hypothesis (H0):* The feature has no significant importance in explaining the predictive feature

*Alternate Hypothesis (HA):* The feature has a significant importance in explaining the predictive feature

*Level of confidence*: 0.95

*Level of significance*: 0.05

Of all the features only final weight (fnlwgt) had a p value of 0.1198 which is greater than the level of acceptable error which is 0.05 and hence is proven to be a feature which is not statistically significant i.e. the feature is irrelevant and should not be considered further.

|  |
| --- |
| SELECTED FEATURES |
| * *Age* |
| * *Capital gains* |
| * *Capital losses* |
| * *Education num* |
| * *Hours per week* |
| * *Sex* |
| * *Race* |
| * *Work class* |
| * *Occupation* |
| * *Marital status* |
| * *Relationship* |
| * *Native country* |

**III. Feature Creation**

Feature ARSC is a weighted product of age, race, gender and native country where each of them was assigned weights based on their income class percentage. The income class percentage was calculated by considering only those observations where income is greater than $50000 against total number of observations for that particular category in each feature.

***ARSC = Age \* Race \* Sex \* Native country***

|  |  |  |
| --- | --- | --- |
| *RACE* | *INCOME1\_PERC* | *RANK* |
| *Other* | *9.23* | ***1*** |
| *Amer-Indian-Eskimo* | *11.58* | ***2*** |
| *Black* | *12.39* | ***3*** |
| *White* | *25.59* | ***4*** |
| *Asian-Pac-Islander* | *26.56* | ***5*** |

Race was assigned weights based on their individual category income class percentage. Higher the percentage, higher the rank as we can see in the above table. Similarly, native country was also assigned ranks on the income class percentage in each country.

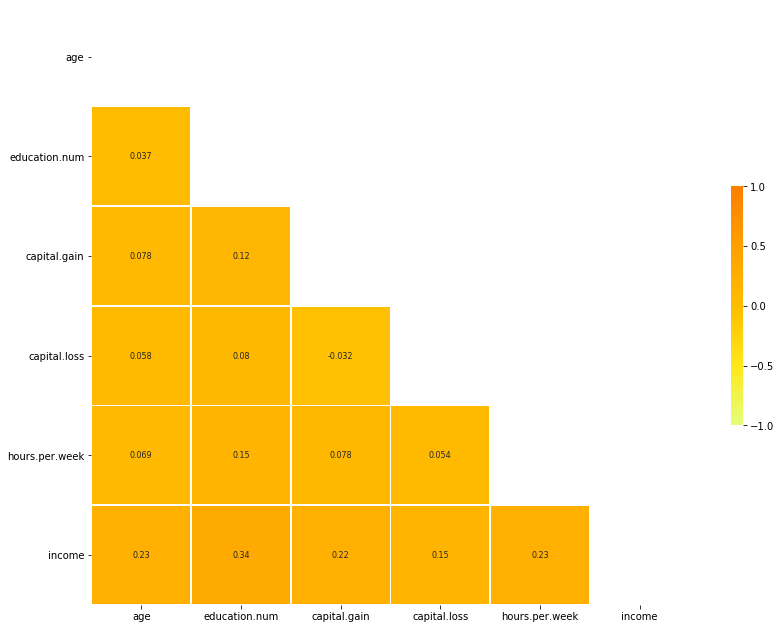
**5. Model Building**

**5.1 Non-Linear Models**

To understand nonlinear models, we first need to know what linear models are. Let’s dive into that for the next few lines. Algorithms that can classify a given dataset based on a linear decision boundary, which just means that the data can be divided into two separate groups with the help of a line.

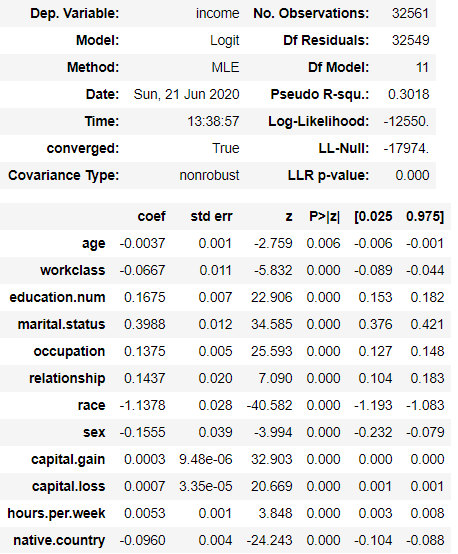
By doing so, we are able to classify the dataset into two distinct classes. In this case, the data points which lie above the line belong to one class whereas the data points below the line belong to another class. This task can be easily accomplished by using a statistical model called ***Logistic Regression.***

**Note:** Here since features have no much co-relation to the independent variable, therefore we can assume that there is no linearity between the dependent and independent features.



Sometimes correlation plot does a bad job of confirming the linearity of the model. For Logistic Regression, from statsmodels.api we import a function called Logit, this function helps us summarize the potential effectiveness of Logistic Regression and the Pseudo-RSquared basically tell us if the model is fit for Logistic Regression or not.

This is the Logit summary of our data. As we can see the Pseudo-RSquared is 0.30, which is a very low value and thus Logistic Regression is not possible with our dataset. So here we understand the reason why we take non-linear models as our model for this dataset.



There are certain instances where Logistic Regression cannot perform its best, this happens when the data cannot be by a linear decision boundary. What do we do in this case?

This is where non-linear models come in to save the Machine Learning world with its amazing behavior.

**5.2 Decision Tree Classifier**

**What is a Decision Tree Classifier?**

A Decision Tree is a non-statistical model which uses a tree like structure to classify the data depending on the features of the data. Here the Algorithm splits features based on certain decision criteria till no more split are possible. The feature the algorithm first uses to split the data is called the root node and the features which are split after the root node is called leaf nodes or just nodes. When no more splits are possible and we attain a node where no more separation is possible, this is called leaf node.

**How is a Decision Tree Classifier made?**

To understand how decision trees are made we first need to understand the terminology used here.

1. Root Node: The feature the algorithm picks to be our initial split is called a Root Node.
2. Node: The node which can be further split is called Leaf Node.
3. Leaf Node: The node at which no more splits takes place, which means that ,this leaf node consists of only one class (in the case of classification).

First, we need to figure out how the nodes are being calculated. Decision trees have various algorithms they work on; the primary aim of the algorithm is to choose a feature which gives the best split for the data. One of the algorithms being CART and the other ID3. These two algorithms do just about the same thing but they have their own advantages and disadvantages.

**CART**

**S**tands for **Classification and Regression Trees**, is an algorithm used by decision trees to find the best feature that provides the best split. This algorithm is also called Gini index.

**Gini index = ,**

Where ‘i’ represents a distinct class.

We first try to find the index given that P(A|B) and then by taking the weighted average of the gini index we can see if the feature is suitable to be used as a Root node or not.

Here, the lower the gini index the better the split, another way to think of gini index is a measure to find the impurity in the feature, so lower the impurity, better the split. This is the idea behind CART.

The default algorithm for a Decision Tree is the gini index or the CART algorithm because this algorithm is computational time taken is very less compared to the other algorithm.

**ID3**

We’ve talked a fair bit on how CART algorithm helps the model to find a suitable feature for its splits. There exists another algorithm which does the same job of find the best feature to split on. Another name for the ID3 algorithm uses information gain, and like the CART algorithm the best feature is found by first finding the entropy of the given feature and then find the information gain, pretty simple.

Let us see how this is really done and implemented in the decision tree model. The algorithm first aims to find the root node, in this case the entropy of the entire dataset is calculated.

***Entropy(S)*** =

Here Entropy (S), also can be written as H(S), indicates the entropy of the entire dataset. The entropy of a data is found by plugging in the values of positive class and negative class into the equation and compute the entropy. Here we also notice a negative sign at the start of each section of the formula. This negative sign is implied because the value for a fraction is a negative value, so to sum it up we apply a negative sign.

The maximum value for entropy depends on the number of classes,

|  |  |
| --- | --- |
| Number of classes | Maximum value for entropy |
| 2 | 1 |
| 4 | 2 |
| 8 | 3 |
| 16 | 4 |

Once H(S) is calculated, we now try to find the entropy of each attribute (H(t)). The entropy is calculated using the same formula as stated above. The total entropy of a categorical variable is calculated by taking the positives and negative classes for each unique class present in the feature. For Regression, the same concept is followed but here each value is taken a threshold and the entropy is found.

After calculating H(t), we need to find the information gain. The information gain IG is calculated by;

**IG =**

Here, **t** is the number of classes present in the feature;

represents the number of positive samples present in that attribute class;

represents the number of negative samples present in that attribute class;

This process is repeated till there are no more splits, this means that the model has perfectly fitted into the data, or till the max depth criteria is fulfilled. Here the higher the information gains the better the split. Again, we can break this down into simple meaning, that is, we are dealing with gain and we always need to maximize our gains so there for we always expect the split to have a high gain value.

So, let’s summarize what we have done here:

* Step-1: We first calculated the entropy of the entire dataset.
* Step-2: We then calculated the entropy of each feature.
* Step-3: We then found the information gain.
* Step-4: This process is repeated till a stopping criterion is met.

Now you may think, compared to the gini index, this method seems a tad bit exhausting and you are right to think so as well. This method of computation for finding the best split is a tad bit exhausting. The machines also tend to believe that, this method is a bit too tiresome and therefore increases computational time.

So, a question may pop in your mind, “Wait! So, when do we know what algorithm to use?” Well, this is purely on trial and error basis, in some cases ID3 may prove better or CART.

**Food for thought:**

In **Decision Tree** as we have no probabilistic model, but just binary split, we don’t need to make **any assumption at all**. That was about Decision Tree, but it also applies for **Random Forest.** The difference is that for Random Forest we use Bootstrap Aggregation. It has no model underneath, and the only assumption that it relies is that **sampling is representative**. But this is usually a common assumption.

*Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to****decrease******variance****(bagging),****bias****(boosting), or****improve predictions****(stacking). Hence these too do not have any separate assumptions of their own.*

**Advantages of Decision Tree:**

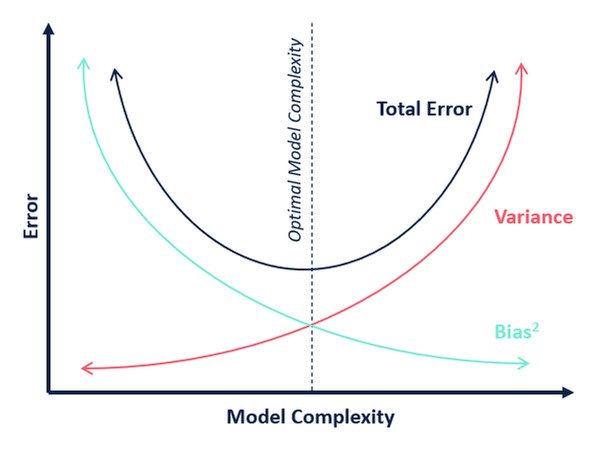
The best thing about Decision tree based models is that we do not require scaling.

Decision tree models are more interpretable and gives us the kind of data that represents the class.

**Disadvantage of Decision Trees:**

Decision Trees tend to overfit the data. Remember when I said that the model ‘perfectly’ fits into the data, if you felt that was a good thing then you are wrong. This tends to pave ways for another topic called bias and variance. Let’s talk about that a bit.

Generally speaking, bias is nothing more than the error made in training whereas variance is the error made during testing. Certain models tend to have high variance and other high bias. This concept PALGUES the supervised machine learning world, basically we are employed as data scientist or engineers just to fix this problem.

The idea here is to catch the point which perfectly hits that sweet spot where bias and variance are just in the right amount. This is called the bias variance trade-off.

Why is this a major problem?Why is it important to find the optimum model complexity?

Here comes the concepts of **overfitting** and **underfitting**.

**Overfitting** is a term used when there is low bias and high variance.This basically means the model performs well in the training and does a very poor job in testing. Now you may think, “**How?**” The model learns all the points present in the data, basically memorizes the data and does a good job in predicting the training data. This is because the data may contain a lot of noise which does not allow the model to learn from the important features is in the data. Allowing the model to learn from the important features in the training phase, which gives the model to predict acurately on unseen data is called Generalization. The success of the model depends on how well it able to generalize the data.

The opposite of Overfitting is called **Underfitting**. This is happens when the bias and variance is low. This means the model has learnt nothing from the data and performs very poorly. This usualy occurs when we have less data or the number of features present in the data are very low.

Now, we understand why overfitting is a major issue and I would like to point out again that decision trees are very prone to overfitting. So we need to tackle this problem and luckily there is a solution to this.

**5.3 Ensemble Techniques (Bagging and Boosting)**

Let me illustrate what ensemble means with help of a general life example. Let me give you one stick and ask you to break it into two halves and I am pretty sure that you’d be able to break it. I hand you another two sticks and I ask you to break it in half and again with full assurity I can say you will be able to break it but with this time you would have to put in a bit more effort than before. I hand you 10 sticks this time and ask you break all of them togther in half and I am sure you would struggle with this given task.

The point of the story is that, having to break 10 sticks seems hard, but you were able to break a single stick with ease. This is what ensemble means in a certain perpective. Ensemble means taking weak models and transform them into strong models by creating multiple weak models just as how we combine 10 sticks to create a really strong stack.

There are two types of ensemble modelling techniques which are known as **Bagging** and **Boosting**. Both are very effective models that allows us to use weak models such as decision tree and creates multiple decision trees to produce a strong model. But bagging computes the data parallely and boosting computes the data sequentially. To understand the previous line, we need to dive a little bit into the geneal concepts of Bagging and Boosting.

**Bagging**

There are two concepts we need to understand when it comes to Bagging, first is the model and then the data. Let’s talk about the model first.

The Model: Like we mentioned earlier , a weak learner is taken and ‘n’ multiple weak learners are create to create a strong model. Here , each learner does not depend on the output of the previous learner but instead each output is treated like a vote and majority of the vote gives you the which class does the data point belong to.

The Data: When the data is feed into the model, there are ‘n’ number of multiple weak learners , a ‘n’ number bootstrap samples of the data is taken and each sample is feed into each model and an output is generated. **Bootstrap sampling** means choosing a random sample from the data with replacement. This means the random data points are selected and those selected points are then replaced back. This allows the same data point to be chosen multiple number of times. The number of samples depends on the number of samples present in the original data. Here not only the data points are sampled but also the features.

Bagging is designed to reduce the variance of the model which makes it ideal to use Decision Trees as it’s weak learner since Decision Trees have very high variance (which means they overfit). Using Bagging along with Decision Trees gives birth to a new model called **Random Forest**.

**Boosting**

Boosting also works on the same principles of Bagging but the only difference is that each weak model depends on the data provided by the previous model. Same like Bagging, Boosting has two concepts that is the data and the model.

The Model: The model is built sequentially , the model is built on the data set produced by the previous tree.

The data: The data, when fed into the model, the first tree is created and the data is fed into the tree, once the prediction is done, those data points are given weights which were predicted wrong by the model. Before the data passes into the next tree, the data set is sampled from the original data giving more weight for the data point which were wrongly classified by the previous model and this procedure repeats till the maximum number of weak learners are made.

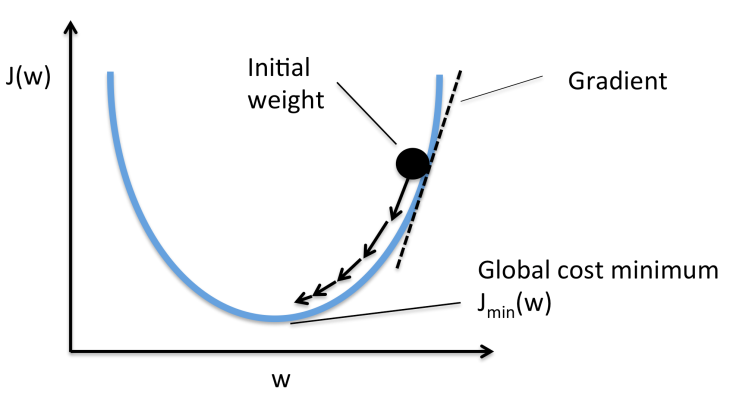
Generally both Boosting and Bagging models use Decision Trees as their default model/estimator. There are different kinds of boosting methods, such as;

* Gradient Boost
* XGBoost
* AdaBoost

Here, since Gradient Boost that performed best in our dataset, we would be talking more on that.

**5.4 Gradient Boost Classifier**

Gradient Boost works on the concept of gradient decent. We will talk a little bit about gradient decent before we get into gradient boosting. Gradient Decent uses a differitable loss function, by substituting a range of values and the losses are plotted onto a graph. The idea behind is to find the global minima of the curve. The weights of the model is updated using a learninig rate and step size, that is, learning rate allows the model to learn in a slow manner. This allows the model to increase its computational time but the time is sacrficable for the expected accuracy of the model.



For Gradient boosting, the principles of gradient decent is used to find an output prediction. Gradient boosting uses various steps to produce an appropriate prediciton.The steps are:

First we provide the model with a dataset and a differentiable loss function. For classification we use negative log likelyhood as its loss function. We need to understand a little bit more about the loss function for this model. So let’s discuss.

For gradient boosting model, we need to understand first log of odds and how to calculate the log of odds. Log of odds is just basically the log of the odds of the an event being in a postive class divided by the the odd of an event being in the negative class.

, here p is the probablility

that an event belongs to a certain class

The idea behind gradient boosting is to minimize the negative log likelyhood. What is Negative log likelyhood ? Log likelyhood can be computed using the formula;

This is how log likelyhood calculated and the idea here is to maximize the likelyhood, but since machine learning is all about minimizing the loss, lets not disrupt that norm. By multiplying the loss function with -1 we get what we call a **negative log likelyhood**. This loss function is what we try to minimize here.

The above equations is in terms of probabilities which we need to convert in terms of log of odds. After using certain transformations we get the above equation as;

This is now our new loss function.

**STEP 1**: Intialize the model with a constant value.

We need to allow the algorithm to start working on something and therefore we need to intialize the the model with an intial value. Let’s illustrate the steps with an example from the dataset we have been given, a manual caluclation of how gradient boost works. For simplicity, we have taken three records and only categorical features. First let’s see how the table is.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Workclass | Education | Marital status | Occupation | Relationship | race | Sex | income |
| private | Bachelors | Never-married | Exec-managerial | Not-in-family | White | Male | >50k |
| private | HS-grad | Divorced | Sales | Unmarried | White | female | <=50k |
| State-gov | Assoc-acdm | Divorced | Adm-clerical | Unmarried | White | Male | >50k |

Now, Since our variable are pretty much self-explainatory. Lets now intialize the model with a constant value. The mathematical form written for step 1 is;

Let’s break this formula down, we understand we need to intialize the model with a constant. We need to find a value that minimizes the loss. Here ,’i’ represents each data point, γ represents log of odds.

To find the log of odds which gives the least error can be done by taking the derivative of the loss function. Here the loss function is;

When we differeniate the loss function with respect to log(odds), we get;

In our dataset we have two positive classes(>50k/1) and one negative classes(<=50k/0). By plugin in the observered values into the above equation and as well as equating the equation to zero.

Note: The output of this function is terms of probalility so we can denote this as ‘p’.

The probability is 0.67 and using this we can calculate the log of odds using the log of odds equation mentioned earlier.

We have now completed the first step and we intialized the model with a constant value which is 0.69.

**STEP 2:** Calculate the pseudo-residual.

We calculate the pseudo-residuals by subtracting the observed values with the previous predicted probability. We tabulate these residuals.

We can mathematically represent the finding of the pseudo-residuals as;

Note:, this means that f(x) is given the log of odds value in the previous step

So here we take the derivative of the loss function f(x), which is nothing but the log of odds, with respect to log of odds. We already have computed for the derivative of the loss function.

Let’s tabulate these residuals for later use;

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Workclass | Education | Marital status | Occupation | Relationship | race | Sex | income | Residuals |
| private | Bachelors | Never-married | Exec-managerial | Not-in-family | White | Male | >50k | 0.33 |
| private | HS-grad | Divorced | Sales | Unmarried | White | female | <=50k | -0.67 |
| State-gov | Assoc-acdm | Divorced | Adm-clerical | Unmarried | White | Male | >50k | 0.33 |

**STEP3**: Building a regression tree on the pseudo residuals.

Using the concepts of Decision Tree we build a regression tree which has the residuals which we previously tabulated as it’s leaf node.

For the sake of explaination, lets take relation-ship as our root node. Now lets create a tree stump. Usually in practice is allowed to grow for a max depth of 3 but here we just take a decision stump(A tree with just one split) with 8 to 32 leaf nodes.

**STEP4:** Finding the output values.

There maybe multiple residuals present in a single leaf node, and we need each node to contain a scalar value. If a single value or multiple values persists in a leaf node , then the output(gamma ‘γ’) value is calculated by;

For the right-hand side leaf;

For the left-hand side leaf;

**STEP4:** Update the prediction.

To update the function, we first allow an instance of the data to pass through the tree and the values are computed. After computing the values and getting a prediction which are the output values, we then add up the initial log odds and the output values of the first tree and then, we get the updated predicted value.

We update the prediction by adding the initial value along with output values of the first tree which is also multiplied by a learning rate.

*F1(x)= F\_0 (x)+ 0.8\*the output values of the decision tree*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Workclass | Education | Marital status | Occupation | Relationship | race | Sex | income | Residuals |
| private | Bachelors | Never-married | Exec-managerial | Not-in-family | White | Male | >50k | 0.33 |
| private | HS-grad | Divorced | Sales | Unmarried | White | female | <=50k | -0.67 |
| State-gov | Assoc-acdm | Divorced | Adm-clerical | Unmarried | White | Male | >50k | 0.33 |

If we allow the first data point into the model, we would get;

For the second;

For the third point;

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Workclass | Education | Marital status | Occupation | Relationship | race | Sex | income | Residuals | Predicted probablility |
| private | Bachelors | Never-married | Exec-managerial | Not-in-family | White | Male | >50k | 0.33 | 0.95 |
| private | HS-grad | Divorced | Sales | Unmarried | White | female | <=50k | -0.67 | 0.05 |
| State-gov | Assoc-acdm | Divorced | Adm-clerical | Unmarried | White | Male | >50k | 0.33 | 0.95 |

As we can see the predicted probabilities are now much better from the previous predicted probability.

This process is then repeated till a certain stopping criterion is met. This is how the gradient boost classifier functions.

**5.5 Evaluation Metrics**

For any classification problem where the classes are not equally distributed, we cannot choose accuracy as a metric to measure the performance of the model. In such cases we build a table called the CONFUSION MATRIX which is matrix created with all the correctly classified and the misclassified observations.

**Confusion matrix:**

It is matrix created using the predicted against the actual separated by the classes. A confusion matrix for a binary classification looks like

|  |  |  |
| --- | --- | --- |
|  | Predicted  ( NO ) | Predicted  ( YES ) |
| Actual ( NO ) | True Negatives **- TN** | False Positive **- FP** |
| **Actual (YES )** | False Negative **- FN** | True Positives **- TP** |

**True Positive :**

The class which is positive and is predicted to be positive

**True Negative:**

Class which is negative and is predicted negative

**False positive (Type 1 error):**

Class which is negative but is predicted positive

**False negative (Type 2 error):**

Class which is positive but is predicted as negative

The confusion matrix helps in using other evaluation metrics apart from accuracy

ACCURACY:

**Accuracy = TP+TN / TP+TN+FP+FN**

Accuracy is total correct classes predicted from the entire dataset.

RECALL:

**Recall = TP / TP+FN**

Recall is a measure of the total number of positive classes predicted correctly

Also called as True Positive Rate (TPR)

PRECISION:

**Precision = TP / TP+FP**

Precision is the measure of actual positive cases from those predicted as positive

F-MEASURE

**F1 score = recall \* precision\*2 / recall+precision**

F measure is the harmonic mean of recall and precision. In cases where both the errors need to be taken into consideration this can be a better measure.

SPECIFICITY

**Specificity = FP / TN+FP**

Specificity is the measure of the positives against the actual negatives.

**ROC AUC:**

AUC is called the area under the curve. The curve is a probability cure that tells us how much model is capable of distinguishing between the classes. Higher the value of AUC, better the model.

The ROC curve is plotted on a graph with the True Positive Rate (Sensitivity) on the Y-axis and False Positive Rate (1 - Specificity) on the X-axis. The values of the TPR and the FPR are found for many thresholds from 0 to 1. These values are then plotted on the graph.

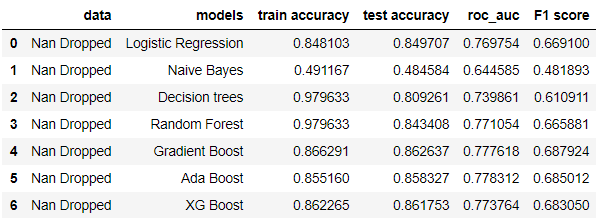
Interpretation:

The main objective here is to find a point on the ROC curve where the Area under it is the maximum. This is because it is at this point, where the model could correctly distinguish between binary classes with there being minimum overlap between them.

**5.6 Evaluating Our Model**

**I. Dropping Null values:**

In our data we took several conditions for our dataset. The first thing we did is that we dropped all the null values and this was our base model. The reason why we dropped all the null values because the number of null values was very less compared to that of the entire set of data. The data has 32561 records and the number of null values present is around 2399 which just 7.36. This would not have a significant change on data if the null values are removed.



Let’s review what we see in the above table, **data**: This just describes what transformation we have done in our data; Models: The type of models we have used; **Train accuracy**: The accuracy that is achieved in the training phase; **Test accuracy**: It is the accuracy that is achieved during the testing phase; **roc\_auc: Relative Operational Characteristic Area under the curve**, it is a metric that plots out various confusion matrices with different values as threshold which ranges from 0 to 1 and the area under this plotted line gives us the area under the curve. The area under the curve is used because it gives us a general idea on how well the model performs irrespective of the threshold; **F1\_score:** It is the weighted average of precision and recall.

As we can see from the table, models such as Naïve Bayes, Decision Trees, Random Forest fail to provide good classification metric values. The reason why I put Decision Trees and Random Forest as models that does not perform well is because of the concept of over fitting, which means there is high variance but low bias, this means these two models fitted to the noise rather than generalizing the data.

Models such as gradient boost and XGboost does a good job in classifying the data, but we wouldn’t want to go to XGboost just yet as it is a more complicated version of Gradient Boosting. Gradient Boosting performs the same way as XGBoost and therefore we do need to implement such complex models.

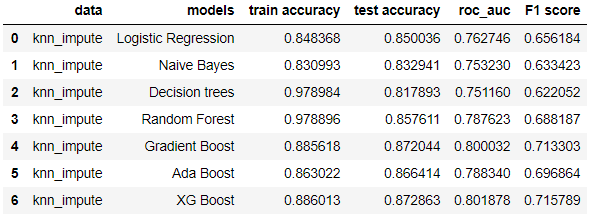
So, we conclude by putting our base model as Gradient Boosting.

**II. Imputation of Null Values using KNearest Neighbors’:**

By using KNNimputer from the sklearn.impute we impute all the null values and see how well does our model perform. We also fit the data with other models such that we can compare not only to our base model but with other models as well.

We use hyper parameters to tune the model such that the model generalizes the data efficiently. The parameters we used are:

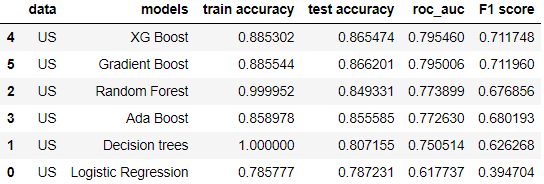
***GradientBoostingClassifier ( learning\_rate = 0.24, random\_state=90, n\_estimators=200, min\_samples\_split = 10 )***



**III. US-based native country data:**

Since majority of our data comes from those individuals whose native country is the United States of America. There are around 27000 records which belong to US as their native country. So, we fit our models onto this dataset and see how it performs. Gradient Boost continues to perform better.

***GradientBoostingClassifier ( learning\_rate = 0.24, random\_state=90, n\_estimators=200, min\_samples\_split = 10 ).***



As we can see, XGBoost and Gradient Boost almost has the same metric score and since Gradient Boosting is a little bit less complex, we continue to choose Gradient Boost as our model.

**IV. KNN Imputation along with SMOTE (Synthetic Minority Oversampling Technique):**

SMOTE here is used when there is an imbalance in the dataset and can be imported using the module called imblearn. In our data set we have 75% of the data in the class >50K and 25% of data in the class <=50K, so we use smote to see how well the model is able to classify both classes with the same number of observations in both classes.

***GradientBoostingClassifier ( learning\_rate = 0.27, random\_state=90, n\_estimators=300, min\_samples\_split = 10 )***

Here, Gradient Boost still continues to persist as the best model.

**V. Feature Engineering: Creating a feature with Age, race, sex and country:**

We create a feature that is that is based on age, race, sex and country by multiplying each of the variables with each other and drop the variables Age, Race, Sex and country. By creating such a feature, we remove a lot of noise from the data and improves the correlation between the arsc variable and target variable.

***GradientBoostingClassifier ( learning\_rate = 0.24, random\_state=90, n\_estimators=200, min\_samples\_split = 10 )***

So far, this model has given us the best output and this could be used for deployment.

These are the models were built based on different implementations of transformation in the data, so we need to do the final step of comparing all the Gradient Boosting models we have designed based on our data.

|  |  |
| --- | --- |
|  |  |

By comparing everything we can see that, the data with the new feature ARSC is better than all the models being produced and therefore the data is transformed to such a manner and evaluated using the model initialized for that particular type of data. The model built using ARSC was able to show AUC of 0.92 with an Accuracy of 0.87.

**CONCLUSION**

The goal of predicting the income class based on the census data was successful and encouraging. Since the correlation of the features was low the linear models like the logistic Regression did not do a good job in capturing the essence of the data. In such cases a non linear model like the decision trees did prove to be a better model. Since the dataset has class imbalance where the individuals that belong to income class >50000$ was only 24% which is about one third of the data, Boosting and bagging models did help in the classification. For this kind of problem, random forest classifier and gradient boost classifier is suggested.