# 1 Part A

I started off by loading the dataset into a data frame.

I also tried to get the d types of the data just to see the types of the columns in the dataset for my own study as I thought to use this information to convert the year of birth to a numerical value. The year of birth was extracted from the birth date and then converted to a numerical value because I couldn’t get the award age given that the year of birth was an object.

In the original data frame, we have only, a few numerical fields/columns and those are the confidence levels for some of the categorical variables and I am not entirely sure what it means by, for e.g., “religion: confidence”.

The shape of the original data frame shows that we have 441 rows/observations and 27 columns/fields.

I have also tried to find a list of all the columns that contain null values and it turns out that '\_last\_judgment\_at', 'birthplace\_gold', 'date\_of\_birth\_gold', 'race\_ethnicity\_gold', 'religion\_gold', 'sexual\_orientation\_gold' and 'year\_of\_award\_gold’ all contain null values.

### Considering the Subset

I got the subset of the original dataset, the subset includes the birth place, the date of birth, the race, the year at which the award was granted and the award category itself. I showed the first three rows of this subset as instructed to me by the convener in the assignment sheet.

### Unique Award Subset

I showed the unique values in the award category column because this helped me to see the distribution of religion based on the award category. This also helped me later for part 2 when I tried to see the impact of each of the country categories and races

This gave the output “['Best Director', 'Best Actor', 'Best Supporting Actor', 'Best Actress', 'Best

Supporting Actress']”. The output you see was given as a list because I used to the “to list”

method.

### Adding the length of date of birth

I added a new column for the length of the date of birth. I just needed to get the length of the string using the length function.

Then I showed the unique values of this column using the same “to list” commands I used in 1.2. This gave me the output “[11, 10, 8, 9, 15, 4]”.

### Cleaning date of birth

As you can see from the code, I first checked if the length of the date of birth is four which means that only the year is given and return the date of birth by giving the first of January as the standard DOB. If the length of the year part is greater than four which is in one of the cases, then I exclude whatever is it that comes after the year and return it in a cleaned form. At last but not the least, I try to change the year if it is given as two digits. I change it by appending 19 before it. In this whole function, I try to make the split based on the hyphen. Then I use a lambda function to apply the whole function onto every row of the column of the date of birth.

### Adding Country

I added the country USA where it wasn’t given. I first split the birth place based on the comma. Then I check the last element of the array whether it has a length of 2. If so, I append USA to the end of the birth place, and this is the trick. There is an exception for the New York City, so I included that too. I then apply a lambda function like 1.4.

### Adding Award Age

To add the award age of, a sequence of commands was used. I first made a function to extract the year of the date of birth. The year of birth column was used to get the years in one column which were in object form and then I turned them into and integer by using the second command. I then used the third command to subtract the year of birth from the year of award. This is how I got the award age with three simple commands.

### Adding Country of Origin

Getting the country of origin was simple as I had to extract the last element of the array that we get form splitting the place of birth based on a comma. I then apply the function using a lambda function.

### Most Oscar Winners are from USA

This is part of the data exploration part and the hypothesis that most Oscar winners are from USA is true. I haven’t included the output because it was to large for the page, but results show that 290 out of all people are from USA which is more than any other number of people based on the country. So, this hypothesis is correct. Please see the graph below for proof.A close up of a logo

Description automatically generated

### Most Oscar Winners are White

This is also part of the data exploration section and the hypothesis that most Oscar winners are White is true. This is because the results show that 411 of the total people are White which is more than the number of people from other races. Please see the graph below for proof.

A screenshot of a social media post

Description automatically generated

### Best Directors are Older than Best Actors or Actresses

This is also part of the data exploration section and the hypothesis that best directors are older than the best actors or actresses is true. The proof of this can be seen by the one command I ran and the output I got which is given below. You can also see the results from the box plot below where the median for best director is higher than the median for best actress and best actor. Moreover, I also ran the commands “*df\_subset.groupby('award').mean()*” which shows that the mean age for the best director is higher than the age for the best actor and actresses. This is below the box plot.

A close up of text on a white background

Description automatically generated

### Discretizing the Age Using the Cut Function

For creating the buckets for the age, I used the cut function as shown below to create partitions for the numerical age. The cut function is used to bin values into discrete intervals.

Here you can see that the cuts made are between four categories which are 0-35, 35-45, 45-55, 55-83. The 83 comes from the max age in the award age column.

### Fixing Country Category

From the plot shown in 1.8, we can see that the top 5 countries to pick which will have an impact on the award category are USA, Italy, France, England and Canada. I renamed the rest as “Other Countries” and the motivation behind that comes from the fact that there are too many categories and to class them like I had is a better idea. Not because, it is hinted in the assignment sheet but because all the countries which have been classed as “Other Countries” have less records and thus will not have a major impact on the award category if we compare them, individually, to the top five I have selected. I created a function for this and used a lambda function to apply that to all the rows in the country category column.

### Adding Religion and Sexual Orientation

I added religion and sexual orientation from the original data frame to the subset because I will need that later for the improved model and the improved model subset will be derived from this one. I will shed more light on this when I discuss the improved model. Moreover, I will also need this for 1.14 as you will see next.

### Analyzing the Religion Column

I first tried to make a count plot for the whole religion column just to see the variation and as you can see below, the dataset is dominated by the “Na” values in the religion column. This means that we don’t have data for majority of the values inside the religion column.

A close up of a logo

Description automatically generated

Now that we know that there are missing values inside the religion column, the question that comes to mind is that “What are we going to do about it?”. One possibility is to just do a backward fill or a forward fill for fixing the missing values by converting them to actual values. The other possibility is to replace them with the most dominated value which is “Roman Catholic” Judging by the chart, it would be bad to replace all missing values with Roman Catholic. This is because replacing all missing values with Roman Catholic would bias the whole religion column in a sense that it would be highly probable to predict the award category given that the religion is Roman Catholic. So, we need to investigate in creating a relatively more even distribution for this. So, I tried to investigate the distribution of religion based on each category of the award. And my charts are show below.

Best Director

A screenshot of a cell phone

Description automatically generated

Best Actor

A screenshot of a cell phone

Description automatically generated

Best Actress

A screenshot of a cell phone

Description automatically generated

Best Supporting Actor

A screenshot of a cell phone

Description automatically generated

Best Supporting Actress

A screenshot of a cell phone

Description automatically generated

As you can see that all the five categories are dominated by missing values and there seems to be diversity in the religions of all the five categories. In some categories Jewish religion is the second most dominant and, in some categories, the Roman Catholic religion dominates. There is also some diversity in the religion’s presence in a sense that Best Supporting Actor is missing some religions like the Hindu religion and Best Supporting Actress is missing the Sufism religion. So, I have decided not to replace the missing values with the most dominant value due to the diversity in between the categories with reference to the religion.

So, I have decided to do a forward fill and then a back fill. The reason for the order (forward fill first and then a backward fill) is because the plots that I got for this case was more diverse and gave out an even distribution for all religions. The other case (backward fill first and the forward fill) didn’t give me an even distribution for all religions and was biased more towards the Roman Catholic religion. Moreover, the reason to do both is because the dataset shows that the first few rows and the last few rows are missing for the religion column so using just a single backward fill or forward fill would not be enough.

One thing I would like to point out that I had to replace all “Na” values with the None keyword because with “Na”, Python thinks that this is still actual data, so it doesn’t count it as missing. So that’s why I decided to create function that return None if “Na” is in the column. Then I applied the function as a lambda function. And after this I applied the forward and backward fill.

This is what the cleansed religion chart looks like below. If you compare it to the uncleansed chart, this looks like a better visualization because the previous lower level (quantitively) religions have increased as can be seen by the bars for each one of the religions.

A picture containing screenshot

Description automatically generated

### Analyzing and Cleansing Sexual Orientation

The sexual orientation had missing values as shown in the bar chart below.

A screenshot of a cell phone

Description automatically generated

One thing to notice here is the fact that there are very few values that are missing here, eleven to be exact. Therefore, I have treated the missing values by replacing them with the most frequent value which is “Straight”. I didn’t use a backfill or a forward fill in this case because it wouldn’t have much of an impact on the number of other values because they are very less in number. For e.g. if a backward fill or a forward fill resulting a slight increase in the “Matter of Dispute” orientation or the “Gay” orientation, it wouldn’t make much of a difference because the number of “Straight” orientations is very high and it would still dominate the predictions for the award category in a sense that it would be highly probable to predict the award category given that the sexual orientation is “Straight”. So, I used a lambda function to apply a function to every row.

### Reducing Religion Categories

As we can see, there are almost 20 religion categories, so we need to reduce these in order to get better predictions. How we do the reduction is very simple and I did this by picking the top 4 categories for religion and replace all others with the value “Other Religions”. The reason to do this is to combine all other values that would not have much of an impact, individually, on the award category. This, I believe, is the real reason why the convener has hinted this in the assignment sheet too.

### Simple Random Forest Model

To create a simple random forest classifier based on the race, date of birth and country of origin, I had to create dummy variables by which I mean that I had to convert these (race, date of birth, and country of origin) categorical variables to numerical ones. This is done by using the get dummies function from the Panda library. Once I created the dummy variables, I then had to include inside the predictors all the features that were numerical and not categorical because that would give an error. I did this by making a custom function that takes all values from the first list that exist in the second. After that I randomly split the dataset into random training and testing subsets. Note here that I have used 10,000 trees for the random forest classifier. I don’t think it was necessary to specify the depth of the tree. The reason for this is because according to the Sci Kit library [1] if the maximum depth of the tree is not specified then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split (the minimum number of samples required to split an internal node) samples. The default value then taken for this argument is taken as “None”. Then we build the forest of trees using the fit function on the training sets. Notice also that I have set Out of Bag score to true and this is because the out-of-bag (OOB) error is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample. This allows the Random Forest Classifier to be fit and validated whilst being trained [2].

Moving on, we test the model by making it make predictions on the predictors test set. We use this prediction to build a confusion matrix. This helps us to understand how many **hits** we got for each of the categories. I haven’t included the confusion matrix here because it is different every time, we run the algorithm. We’re basically computing a cross tabulation for all the award categories with the actual values against the predicted ones. Somewhere, we see zeros in the matrix, and this is because there were no right predictions made for that category (we had miss for all those instead of a hit).

I then calculated the accuracy for this model which revolves around 30% and 40%. It gives different results every time I run it but these (30 and 40) are the values around which the values for the percentage of the accuracy of the model revolve.

I then try to get the features which had the impact on the award category in ascending order. It turns out that the most impacting factors are the age buckets with the age bucket from 0 to 35 having the largest impact.

A screenshot of a cell phone

Description automatically generated

# Part B (Improvised Model)

For the improvised model, I also tried to describe the data in order to describe the numerical fields and it turns out that the mean award age is 43. A category wise break down for age is shown below. This data has been acquired from the “df\_subset1” which I have specifically used for the improved model and I have cleansed it with the best of my ability.

A screen shot of a computer

Description automatically generated

One other thing I would like to point out here is that the predictors given in the simple model which was suggested by the module convener were not enough, so I thought it was important to include more predictors to see what effect it has on the accuracy. Therefore, I have used more predictors than the ones suggested for the first part.

Now, I will move on to improving the model that we currently have. I started off by cleansing the outliers for all the categories for the award category with reference to the award age. These outliers can be clearly seen in the box plot the we built in 1.10. It is important to know here what outliers are in order to make sure we know what we’re doing. Outliers are any values inside a column that are 1.5 times the Inter Quartile Range above and below the upper quartile and the lower quartile. I cleaned the outliers for all the five categories that exist in the award column. After cleaning the outlier’s category wise, I also wanted to make sure that all these cleaned categories are reflected in the subset. So, I concatenated them into the original data frame.

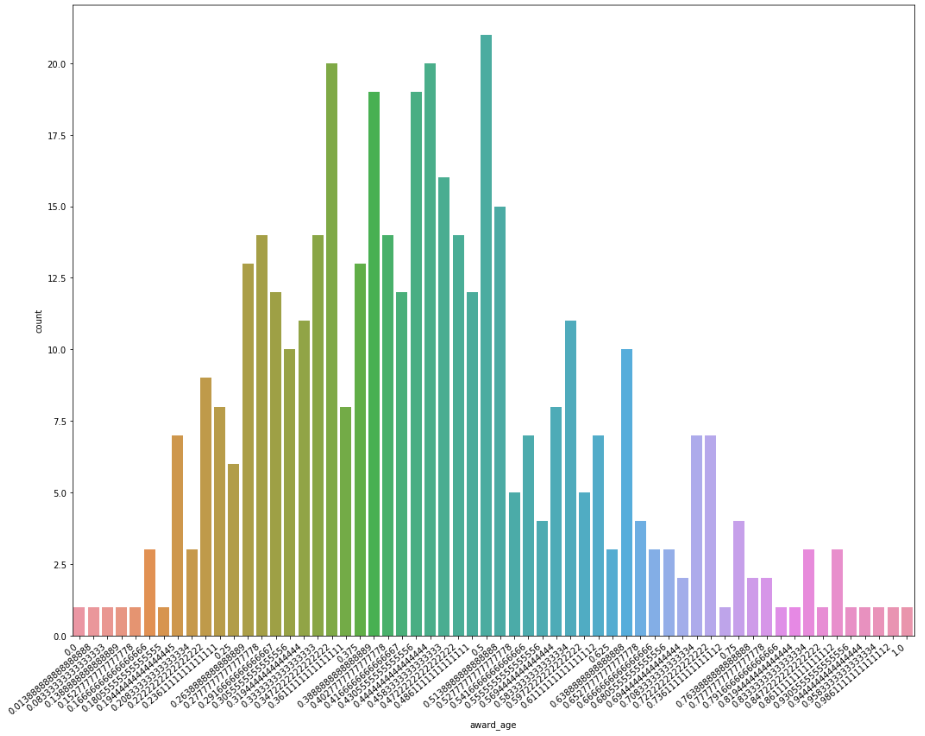
For the improvised model, I did try to use the numerical age instead of the age buckets and it gave me a bad accuracy. I also tried normalizing the age of the award, but it gave me a bad accuracy too. This is because of the variation inside the numerical age whether it’s normalized or not is too high and this can be seen from the plots I have drawn below.

Non-Normalized Age

A close up of a logo

Description automatically generated

Normalized Age



As you can see, there is a lot of variation between the numerical ages regardless of whether it is normalized or not so this explains why I was getting a bad accuracy for this. That is why I excluded it from using it for predicted the improvised model.

Analyzing the sexual orientation with reference to every category, tells us that in every category, the “White” race is the dominant one. This means that the “Straight” orientation will have the most effect on predicting the award category which simply means that it is highly probable that we can get a correct prediction given that the sexual orientation is straight.

Analyzing the race ethnicity with reference to every category, tells us that for best director, there is not much diversity for race ethnicity as it is dominated by “White” race with only a few records to show for “Asian” race. The same can be seen for best actress with the exception that instead of “Asian” we see only a few records of “Multicultural” But we can still see that, all the categories are dominated by the “White” race in the race ethnicity column. This means that the “White” orientation will have the most effect on predicting the award category which simply means that it is highly probable that we can get a correct prediction given that the race ethnicity is “White”.

Analyzing the country of origin with reference to every category, tells us a similar trend which we saw for the sexual orientation and the race ethnicity. Here all the categories are dominated by the people of “USA” while others being very few quantitively.

The age bucket is interesting in a sense that for best director, the 35-45 age bucket is very dominant, and we can see a slightly normal distribution for this. A similar thing can be seen for the best actor category. The age bucket 0-35 for best actress is the most dominant which means that it is highly probable to correctly predict the best actress award given that the age bucket is 0-35. For best supporting actor all age buckets, to an extent are equally distributed with the exception for the 0-35 age bucket. For best supporting actress, we can see that it would be highly probable to correct prediction given that the age bucket is 0-35. We can also see a decreasing trend as the age buckets increase for best supporting actress.

The final nail I tried to hammer in the coffin to get a better performance was derive a new column. You must’ve notice that the “age bucket”, “award age” and the “country of origin” are the other derived features. So, gender is the new column I tried to derive from the existing columns. The trick was to use the award categories because actors would be Male, and actresses would be Female. If you look at the plot before I fill any missing values, the Female and Male number is almost equal. After filling the missing values using a backfill, I got a higher proportion of males that females. This indicates that it is highly probable to get a higher accuracy for the award category given that the person is male. And this is what led to a higher accuracy in the end I believe as we will see now.

I kept everything the same as when I built the normal model except that I included more features which to name are, “sexual orientation”, “religion” and “gender”. The accuracy I got revolved around 50% and 60% for the improvised model. I also produced a classification report. The numbers change every time, but the trend stays the same in a sense that the precision for best actress, best supporting actor, and best supporting actress remains higher than 50.

I then did some cross validation that gives me a better understanding of the improvised model and it shows that apart from the two very low scoring iterations which are in the 40’s the other 8 are divided between the 60’s and 50’s. This helped me to gain a better understanding of the model because “learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called overfitting. To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a test set” [3].

I then tried to use the “Select From Model” library to get the features that had 5% or more impact on the output and it turns out that this model picks 5-6 features every time (this is just an estimate, maybe more or less). I used these features to build the model yet again and got a reduced accuracy. So, I have concluded that unless I use all the features, I get a bad performance (revolves around 40%, see code) and this is exactly what happened in this case when I used the features which had 5% or more impact on predicting the output.

# Critical Appraisal

One thing I would like to point out here that whatever I have done, I have tried to justify it to the best of my understanding and as my teacher puts it, there is no right and wrong in machine learning as long as you justify what you do in clear and succinct manner and I believe that this is exactly what I have done throughout my report. I also think that we can also experiment if timer permits to name all countries as “Other Countries” except “USA” and see whether that would yield a higher accuracy. A similar thing can be done for the religion column by renaming everything to a “Other Religions” apart from “Roman Catholic” because that surely dominates in majority of the cases that we saw out of the five.

One thing that I would like to point out that we can also try using gradient boosting classifier which is also an ensemble. Or we can go to a more advanced level by using a neural network. According to [4] Gradient Boosting Trees help us to fix errors that have been made by the previously trained trees. This builds trees one at a time. The random forest, on the other hand, trains each of the trees independently. The good thing is that random forests are relatively easier to tune than Gradient Boosting Machines. According to [5] “artificial neural networks use different layers of mathematical processing to make sense of the information it’s fed”. As the name suggests, a neural network is a concept in artificial intelligence based on the working of the human brain. Neural networks can be used both for a supervised learning approach or for an unsupervised learning approach. It’s just that implementing it is relatively harder than implementing simple a simple random forest which we did for this course work.

We can also use a clustering technique such as the K-Means clustering algorithm to get valuable insights from the data set. This isn’t as time consuming as a neural network in terms of implementation, but we would need to use “PySpark” in order to complete this which harnesses the power of parallel computing. This would be an unsupervised learning approach which would help us in exploratory data analysis for the Oscars data set through visual plots. As we have studied in the lectures, in K-Means clustering K centroids are randomly generated and clusters are developed by associating every observation with the centroid that is nearest. After which, the center(centroid) of the clusters becomes the new mean.

Here, in this assignment, we have done supervised learning by using Random Forest as opposed to clustering as we have tried to map the inputs to the outputs with enough approximation that it can predict new data.

I believe that the movie industry can benefit from this analysis as it shows that, approximately, for every two males there is one woman. The graphs/plots show that, after filling in the missing values, that males are almost twice in number than women. I believe this fact can help the movie industry to encourage women to go into the industry in order to create gender equality. This can be seen from the graph below.

A screenshot of a cell phone

Description automatically generated

One thing that is missing here is the pay for each of these people. All these people most probably get paid per movie or project they work on so pay might not be suitable but maybe the mean amount of money they earned in a year can be helpful for us to get insight into, maybe, the gender pay gap in the movie industry.

The data also shows that most of the industry is dominated by the “Roman Catholic” and “Jewish” religions so this analysis can help the movie industry to encourage inter faith diversity by bringing in more people from other faiths/religions.

We have also seen that the film industry is dominated by white people, and from the USA nationals so this fact can also help us to accept the fact that we need more diversity in terms of the people the industry brings in from other countries. This would help to create more jobs and revenue through tourism as well.

Moreover, we have also seen from the plots that many of the people in the industry are of “Straight” orientation and this fact can help us to make the recommendations to the film industry so there is clearly a need to bring in people from other orientations to help the industry become more diversified.

Plots show that we don’t have enough young directors because majority of the directors come from the 34-45 and 45-55 age buckets. So, we need more youngsters directing films to support age equality. The same goes for best actors. We also need to give chance to women that are above the age of 35 because the best actress (including supporting) is dominated by women who are less than or equal to the age of 35. Regarding, best supporting actor, we need to give younger people a chance to as the best supporting actor field as very few people who are less than the age of 35. There is a huge age gap, and this can help the industry to create more diversity in terms of age.

N.B I haven’t included graphs at some places because the graphs were too large and in number too. But I have included all code for the graphs inside the python notebook file and you would be able to see the graphs.

# Bibliography

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

[2] T. Hastie, R. Tibshirani and J. Friedman, “Elements of Statistical Learning Ed. 2”, p592-593, Springer, 2009.

[3] <https://scikit-learn.org/stable/modules/cross_validation.html>

[4] <https://medium.com/@aravanshad/gradient-boosting-versus-random-forest-cfa3fa8f0d80>

[5] <https://www.forbes.com/sites/bernardmarr/2018/09/24/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/#b12a7ee12457>

