The Academy Awards, also officially and popularly known as the Oscars are awards for artistic and technical merit in the film industry. Given annually by the Academy of Motion Picture Arts and Sciences (AMPAS), the awards are an international recognition of excellence in cinematic achievements as assessed by the Academy's voting membership. The tradition of rewarding for the greatest contributions in the film industry initiated on May 16, 1929 at the Hollywood Roosevelt hotel, where the first ever Oscars ceremony took place.

The faith of the films is in the hands of The Guild of Academics , as imagined by many, a group of old men who wear monocles and like to smoke expensive cigars in their spare time. Contradicting common opinions, The Academy is actually an organisation that includes 6687 members that are separated into 17 branches (directors, actors, producers, technicians and others).

The hype around the Oscars has grown and many gambling companies are using this as an opportunity to earn some money. I have always been interested in the field of data science and machine learning. When combined, these factors made me think:

How can Random Forest machine algorithm be used to predict the outcomes of the Academy Awards.

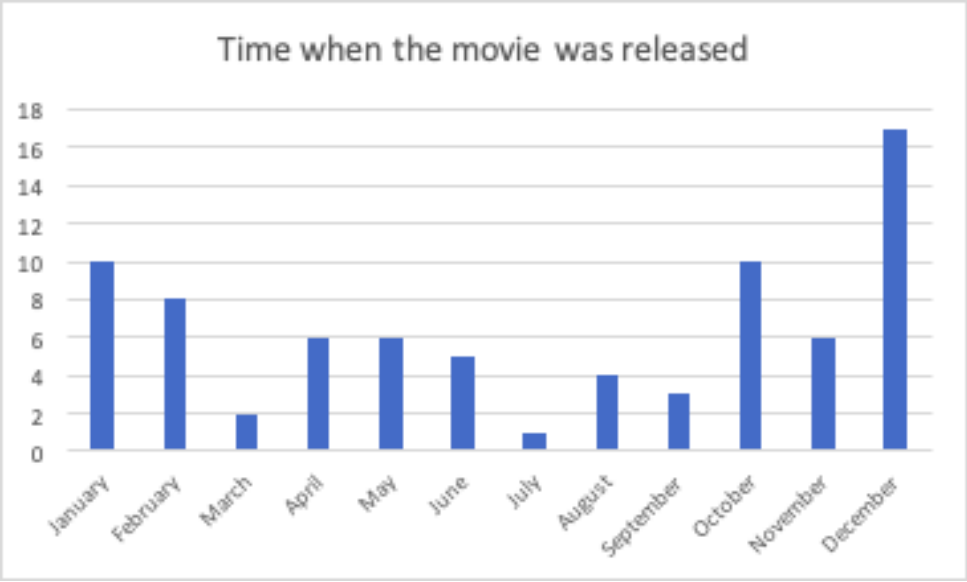
The data I will be looking at is diversified. In it, I collected categories starting from the name of the movie and going all the way to its rating on streaming websites. The main part is the list of all the winners starting from the year 1929 (the first academy awards). The other columns will include characteristics such as rating of the movies, the number of nominations, and a language of the movie.

Before starting to analyse the data there are a few things that need to be taken into account. The process of data collection is a very significant part of any research. Since we are trying to predict

outcomes by looking at the patterns that were found in the dataset previously, the size of the dataset will play a huge role in the accuracy of our model. However, there are often categories that can not only be useless but will “harm” our results. This is why I started my research by collecting data from various websites and combining them into a single dataset. I tried to include data that will increase my chances of getting correct results from the model.

There are many interesting patterns that I have observed during my investigation. A lot of them are characteristics of an Oscar bait, a term generally used in the movie industry to indicate a movie that what made solely to capture as many golden statuettes as possible, with its central aim towards the best picture nominations.

1. The importance of the time frame in which the movie was released. A research showed that movies that were released in the from august to December received the main award with the probability of more than 30% while movies released before may have had less than 20% chance. Consequently, in the chart we can see that the most winners were released in January and December. So the information about the release date of the picture will come in handy in the analysis.



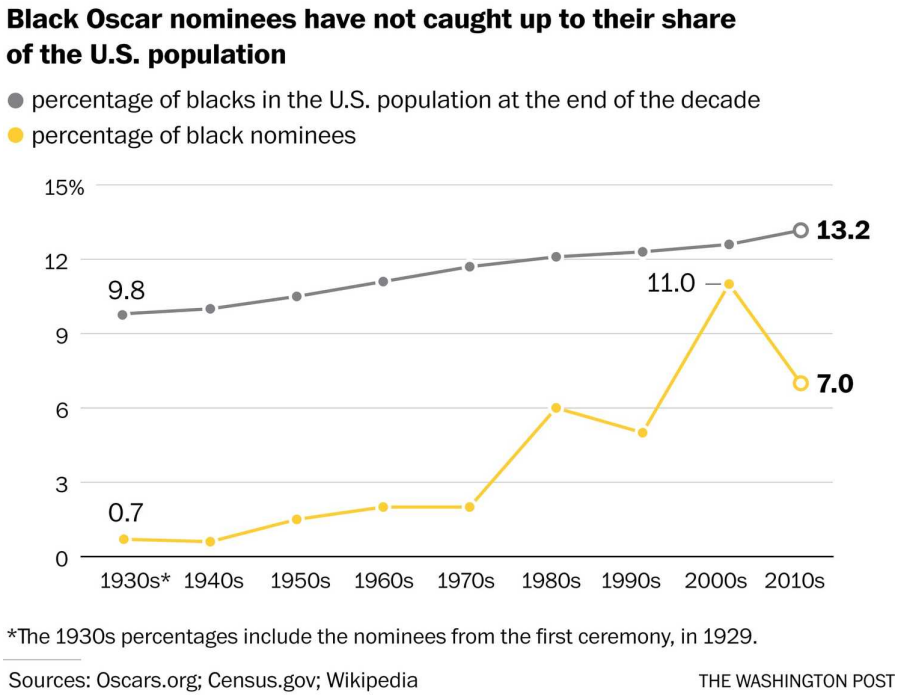
2. Often times, winners of other movie awards, such as the golden globes, end up taking the Oscars statuette as well. In this case the largest correlation can be seen with categories like the best actor/actress (0.90 and 0.70 correlation respectively), or the best screenplay(0.80). However, for the best picture, this number has dropped from an impressive 0.86 in the 2000s to the depressing 0.33 average in the past 2 decades. On top of that, there are two categories for best picture in the golden globes - best comedy and drama, while there is only one category in the oscars. So, simply putting all the money into the Golden Globe winner basket will not do our budget any good.
3. In addition to #2, there are several other movie awards happening throughout the year that have an even higher success rate of "predicting"[1] the following Oscars award winners.
    - Directors guild of America
    - Producers guild of America
    - Writers guild of America
    - Screen actors guild of America

    This is only a small portion of the "successful" predictors out there. Their success rate can go up to 85% with multiple categories which is no different than the golden globes. However, there is one significant difference between them, and it is a 70% success rate of predicting the best picture winner.
4. The word choice. Well, no movie actually won an Oscars because of its release date, the critics votings, or the golden globes. No, it won because of the story, the plot, the atmosphere, and the image. The way to combine these features in one is to look at the key

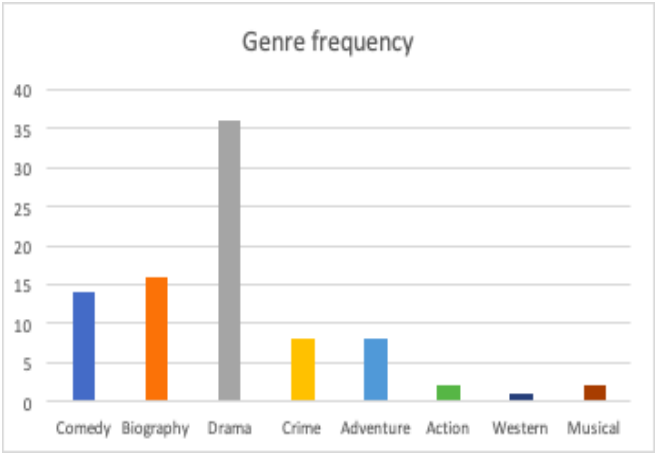
[1] Giving the award to the same movie that later went on to win the Oscars

words. Key words can show what the movie is about, its main characters, the setting, and many other attributes. Using key words is a great way to describe a movie without using full sentences that may be hard to break down through machine learning. The key words make it possible to look into the artistic part of the movie, that is incomprehensible for a pile of metal plates and wires. They are a simplified version of the movie and that is what makes them such wonderful data for machine learning.

5. Let's talk more about the academy. There are a few "problems" with the guild that consequently have a huge effect on the outcome of the predictions. Of the 6687 members, a colossal 96% are white, causing a shift toward white nominees and winners. As you can see in the Figure 1, the percent of non-white nominees is even lower than their share of the total population. There was a lot of controversy around this issue and, in attempts to fix this problem, the academy added an extra 600 non-white members at the end of last year, however, in comparison to the other 6000 members, their opinions are not weighted that highly. So, even though there is more representation of the non-white people in the academy, it will not significantly change the results of voting and therefore predictions of the algorithm lean towards movies with all or mostly white actors.



6. As the industry is developing, the public becomes more and more demanding. Pictures that have received the golden statuette 60 years ago would not even be nominated in 2019. The preferences of the Academy change and so do the winners. This will restrict my dataset only to the past 20 years. This is to ensure that movies from 1930s do not pull the results in the wrong direction. However, there is one genre that has dominated the Academy Awards Best Picture nomination. As can be fairly concluded from the graph below, it is the drama. This also underlines the importance of the genre in the final results of the voting.



This is only a few of the major relationships that need to be taken into account when collecting the data. I will include information that I think will positively[2] influence the outcomes of the predictions. The process of identifying the "right" data is very subjective because it is all based on probability. However, one thing that can be done is to check if the collected data has correlations within it.

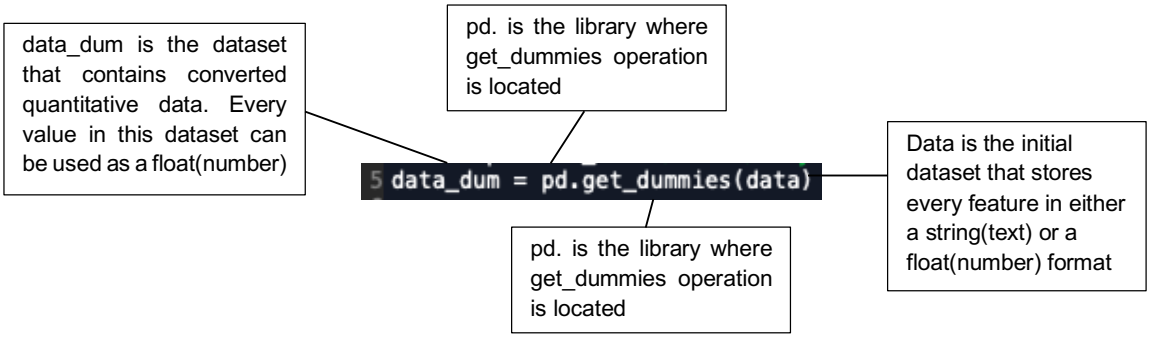
Collected data can be found in the appendix section of the essay

An important part of data preparation is dealing with categorical data. In this case, it is the movie genres, language, and country. When creating a model, the data has to be understood by a computer. Since it can only understand numbers or Boolean values, categorical values have to be converted into a comprehensible form. A simple but wrong solution would be to give a value to each of the categories. It is wrong because some values will be higher than the other which will lead to a creation of correlations that were not initially there. This is why dummy variables have to be created. These variables can only take two states: 1 and 0. This simplicity allows their usage without creating any non-existent correlations. Below you can see an example of how categorical data gets converted to Boolean.

[2] By positively I mean that the model will not try to create non-existent patterns because of irrelevant information

Language		Language_English	Language_Chinese	Language_French	Language_Spanish	Language_Japanese
English		1	0	0	0	0
Japanese		0	0	0	0	1
Spanish	⇨	0	0	0	1	0
English		1	0	0	0	0
French		0	0	1	0	0
Chinese		0	1	0	0	0

This is the line of code that allows to perform this operation



There are multiple algorithms that can be used when working on a machine learning problem. All of them have different applications and work better in specific situations. In this research I will look at how well Random forest Algorithm can predict outcomes of the Academy Awards. In the process of my research I will try to identify whether random forest algorithm is a good choice when predicting the outcomes of Oscars.

When working with on data analysis there is no single method that will perform equally well with all datasets. While some algorithms will return a nice and relevant predictions, others may skew the data and cause problems. A wrong choice of the algorithm may cost a company

millions of dollars. To avoid these issues, the data scientist has to evaluate the pros and cons of a particular method and take them into account.

The first method I will be looking at is random forest. Below you can see some of the advantages of this algorithm over the others.

- This algorithm can be applied to both categorical and numerical data, which is the case with my dataset.
- Can be used as a combination of classification[3] and regression[4] tasks.
- This tool can also be used to fill out missing values in the dataset without losing its integrity and accuracy
- Random forest can be applied to **highly dimensional datasets[5].**
- With a larger combination of trees in the model, there will be a smaller chance of overfitting it
- This tool provides high accuracy predictions

  At first, we need to have a concrete understanding of decision trees work and what they are.

[3] A process in which the data is put into categories, clusters
[4] A process that allows to create a model of the data and apply it in ML based on relations between the variables
[5] A dataset that contains many independent variables

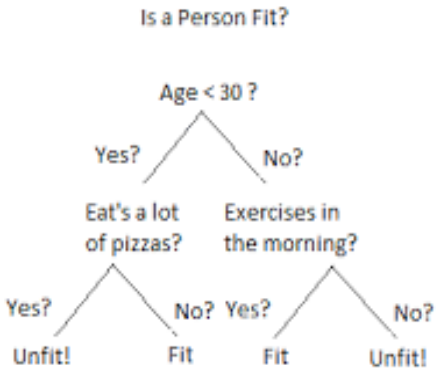


Figure 1 Example of a single decision tree

Shortly speaking, it is a flowchart that allows us to place data into categories(if we look at the example above, the categories are fit and unfit). To be placed in a group, a data point needs to go through a sequence of identifying tags that will determine its path down the tree. Each of the "tree" branches represents the outcome of the test while the nodes are showing what test will be performed on the data. Their simplicity to visualise and understand makes decision trees useful in the field of machine learning. On the other side, they can be not so helpful when working with larger sets of data, such as mine.

However, data scientists were facing a problem when implementing decision trees to the datasets: in many cases, deep trees would overfit[6] the training sets[7] (the bias is very high while variance is low), this can lead to wrong predictions due to high reliance on the training set.

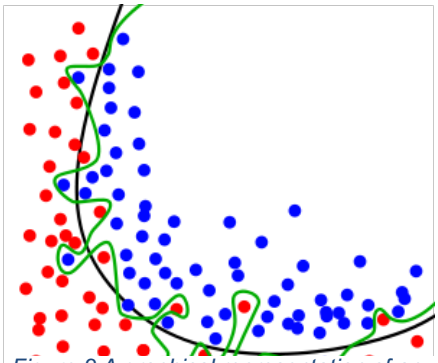


Figure 2 A graphical representation of an overfitted dataset. Black line shows the actual relationship and green line shows the overfitted relationship predicted by the algorithm

Driven by the intentions of making Decision trees more applicable in the field of data science and with larger datasets, in 1995, Tim Kam Ho (a computer scientist at IBM Watson) promoted the idea that collections of trees can perform accurate predictions without overfitting the model. Essentially, random forest is a combination of decorrelated[8] decision trees, hence where the word "forest" comes from.

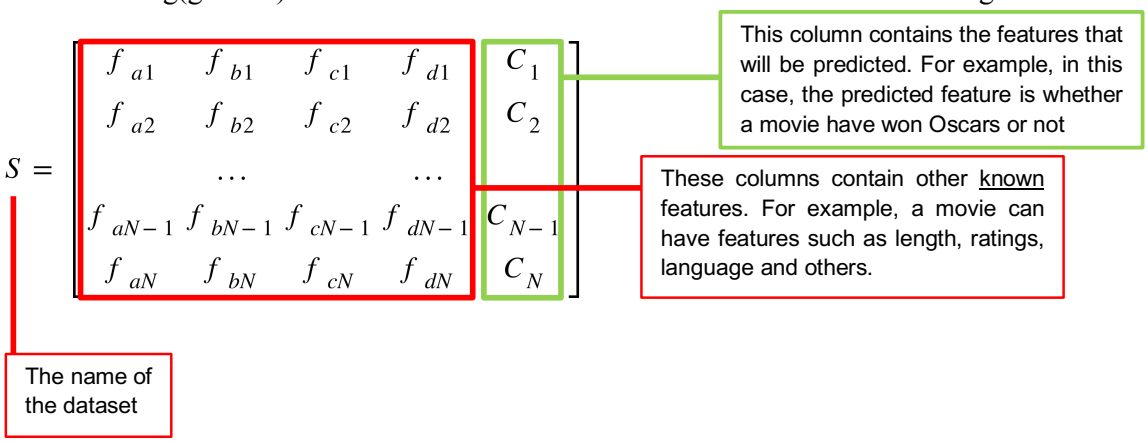
---

[6] Production of analysis that follows a particular dataset too closely, not allowing it to be applied to other datasets

[7] A data set that is used as a base for finding patterns

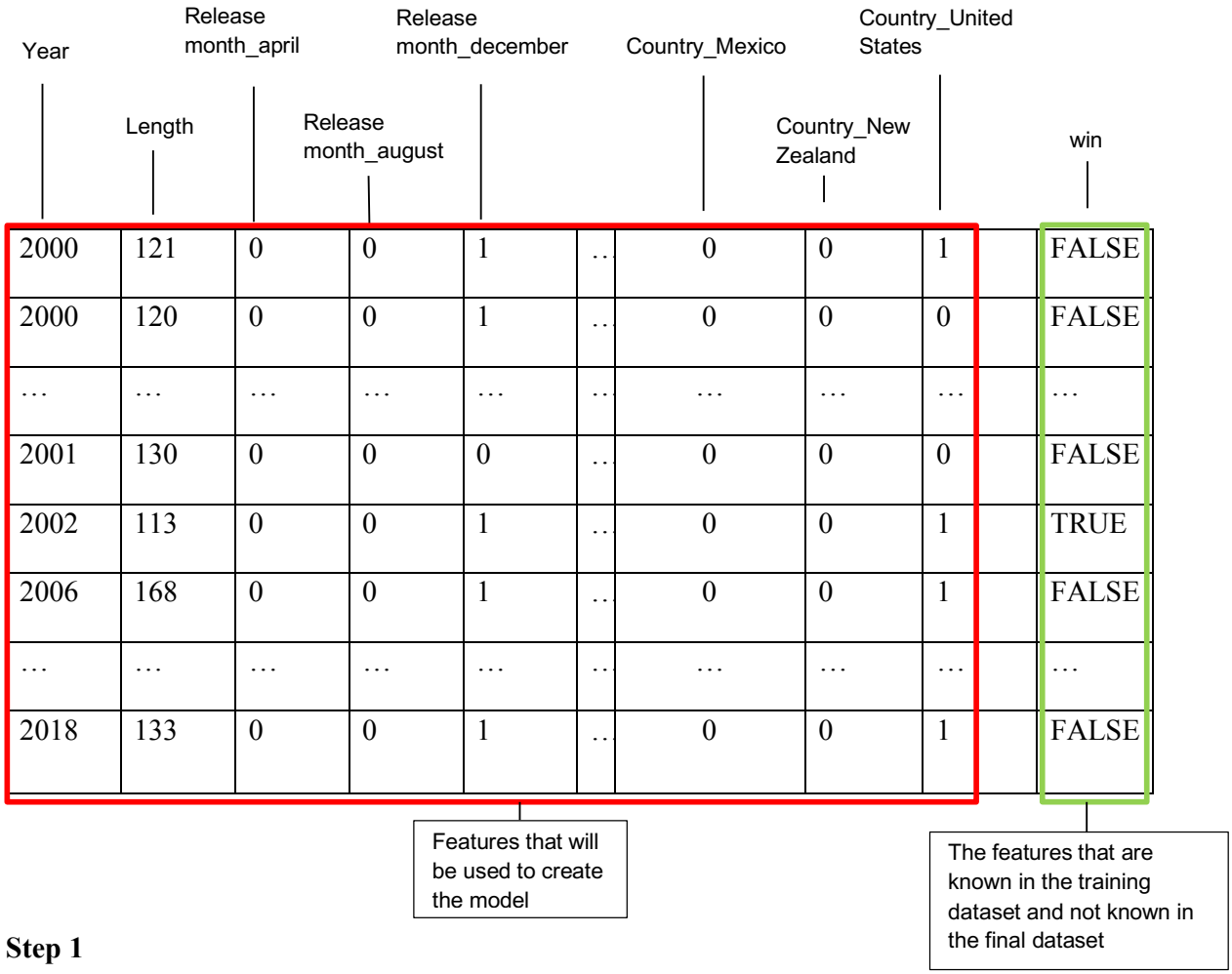
[8] Uncorrelated, lacking relationship

The following(general) dataset can be used to show the work of random forest algorithm.



In the dataset above, the "f" columns represent features of the dataset; rows represent all samples from 1 to N. And finally, the last column is the feature that will be predicted from the model, in this problem, this column will contain true or false values. True – a winner, False – a nominee. The distinction between f-columns and C-columns is that C-columns will be predicted while f-columns are known and contain the information that will be used in the predictions.

In the following dataset, the triple point sign indicates that there are rows /columns in between that are not displayed. When applied to the collected data:



Step 1

When the dataset is ready, **random** subsets, also known as bootstrapped datasets, that are the same size(contain the same columns but randomly selected rows) as the original dataset are created. Because the subsets are chosen randomly, there is a possibility of selecting the same entrees more than once which is allowed to ensure the randomness of the selected subsets. The following datasets are examples of bootstrapped datasets. In reality, there is more of them.

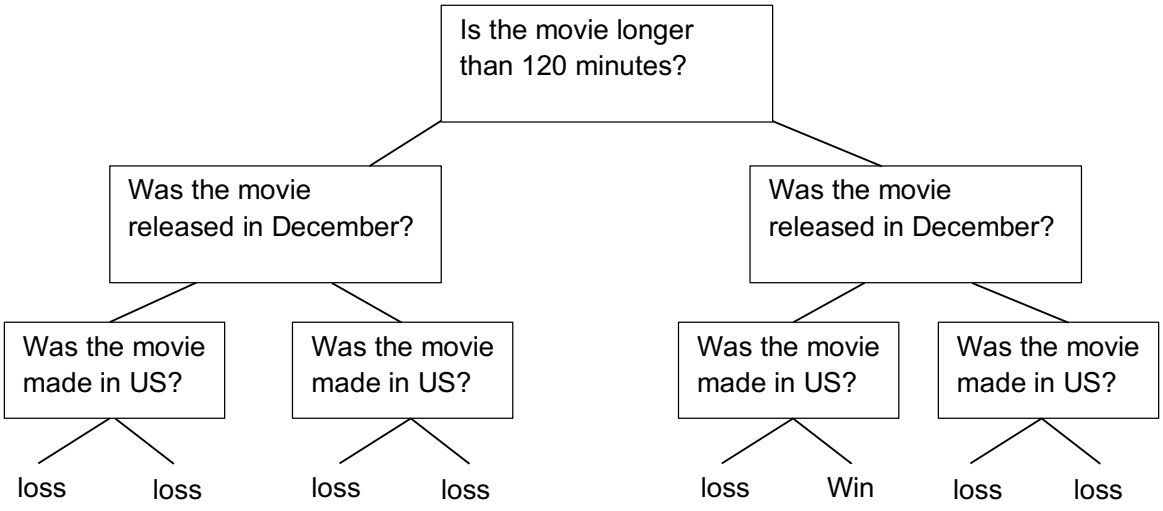
2001	137	0	0	1	…	0	0	0	FALSE
2003	138	0	0	1	…	0	0	1	FALSE
2007	123	0	0	0	…	0	0	1	TRUE
2017	106	0	0	0	…	0	0	1	FALSE
2017	104	0	0	0	…	0	0	1	FALSE

This bootstrapped dataset only has 5 randomly-selected rows but the amount of columns is the same

As mentioned above, the subsets have to be **random** for the tool to give accurate predictions. Otherwise, some dependencies that are non-existent in original dataset may pop up. This will not only create incorrect results but will also lead further research in the wrong direction.

Step 2

A decision tree is created based on each bootstrapped dataset. In the following diagram, right is yes and left is no. At this point, we do not use all of the columns to compute the results. **Random** features are selected and only they will be used in computing the results of a random tree.



At the end of this step, the results produced by each individual tree will be highly biased (because only one tree was used which means that the results are highly related to that particular tree) and will be a very poor representation of the overall patterns in the dataset. This underlines the significance of the next step in achieving accurate results.

Step 3

Now, that we have created a random forest model, how do we actually use it?

At this point we look at the dataset whose C-values we would like to predict.

$$S_{predict} = \begin{bmatrix} f_A & f_B & f_C & f_D & ? \\ & \cdots & & \cdots & \\ f_{Ak} & f_{Bk} & f_{Ck} & f_{Dk} & ?k \end{bmatrix}$$

We run this subset down every single decision tree that was created in the first two steps. At the end we will have multiple values for the final prediction. The results will not be the same for each of the decision trees because of the random element involved in their creation.

Step 4

"Averaging**"** the results from multiple random trees. A very simple way to identify what the "true" answer actually is, based on the ALL predictions, is to see which prediction was the most common.

For example, if, based on the results of all random trees, a nominee has 20 wins and 43 losses, then the overall result is a loss.

This method allows us to combine the results from multiple decision trees. Also, there is no need for complicated formulas in the process of "averaging" the final results because every all decision trees have the same weight. No result from one tree is more important than results from the other.

There are other application of this algorithm when solving real world problems

Some of them include:

Medicine: To create a medicine, a complicated combination of chemicals is created. Thus, to identify the great combination in the medicines, Random forest can be used. With the help of machine learning algorithm, it has become easier to detect and predict the drug sensitivity of a medicine. Also, it helps to identify the patient's disease by analysing the patient's medical record.

Stock Market: Machine learning also plays a significant role in the stock market analysis. When you want to know the behaviour of the stock market, with the help of Random forest algorithm, the behaviour of the stock market can be analysed. Also, it can show the expected loss or profit which can be produced while purchasing a particular stock.

Now that I have selected the method and collected the data, there is one step left to find out the results. I will create a model and train it on the data from the past 20 years. Unlike decision trees, random forest takes a long time to go through and compute, so I will be using python 3[9]. This will allow to achieve truly(or very close to it) random decision trees and also speed up the process of their creation.

In python, there are several steps that need to be taken to get the results:

1. All necessary libraries must be imported. By default in python, we can only access functions from the main library; to access more advanced features we need to import those libraries.

[9] general purpose programming language

The library that allows to operate with datasets in the data frame format

```
1 import pandas as pd
2
3 from sklearn.ensemble import RandomForestClassifier
```

This library contains RandomForestClassifier which is a function that allows to create the model and predict the results

2. The data must be imported and made suitable to fit into the model

This function creates a data frame from a csv file

This function(as described earlier) is used to change qualitative data into quantitative data

```
5 data = pd.read_csv('/Users/sofyamalashchenko/Documents/Sofya/EE/ee final data.csv', sep=';')
6 year_country = data[['Year', 'win', 'Length', 'Release month', 'Genre 1+C1:D135', 'Genre 2', 'Cr
7 data_dum = pd.concat([data["Movie name"], pd.get_dummies(year_country)], axis = 1)
```

At this step, a dataset without the year label is created because it is only needed for indication, not for the actual predictions

y is the dataset that contains the values that will be predicted and the x dataset contains every other feature that will be used in the predictions

```
8 x = pd.DataFrame(data["win"])
9 y = pd.get_dummies(year_country.drop('win', 1))
10 x=x.astype('int')
11 y=y.astype('int')
```

In lines 11 and 10, the values in the datasets are converted to the integer format

3. The model must be created. In this step, the training dataset is used to create the combination of decision trees which will be later used to predict the final results. These are the lines of code that allow for that:

```
12 rfc = RandomForestClassifier()
13 rfc.fit(y[0:117], x[0:117])
```

Every value from the first 17 years (2000-2016) are used as the training set. The rest are the features that will be used in the actual predictions

4. The final step is to use the created model for actual predictions. In this experiment I will be predicting the outcomes of last 2 years of the Academy Awards(2017 and 2018). After the results are received, they will be compared to the actual outcomes that are known. That way, I will be able to determine the validity of the predictions. Those years contain:

Call Me by Your Name	2017
Darkest Hour	2017
Dunkirk	2017
Get Out	2017
Lady Bird	2017
Phantom Thread	2017
The Post	2017
The Shape of Water	2017
Three Billboards Outside Ebbing, Missouri	2017
Green book	2018
Black Panther	2018
BlackkKlansman	2018
Bohemian Rhapsody	2018
The favorite	2018
Roma	2018
A star is born	2018
Vice	2018

This means that after running the code, there will be a data frame or an array that contains 17 values each of them being 1 or 0. 1 would mean that the movie won and 0 would mean that it did not.

```
15 predict = rfc.predict(y[117:])
```

This function will predict the results based on the features from dataset y, rows 117-134 which correspond to the movies nominated in 2017 and 2018

After running the code, the results were collected:

	0
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	1
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0

As described earlier:

1 – win

0 – a loss

Since we know that the movies are in the original order, the names of winning movies(in this case only 1) can be determined.

From the table on the left we see that the movie #8 (numbering starts from 0, not 1) won Oscars. This means that 8th movie from the list presented earlier won.

The 8th movie is Shape of Water. After comparing these results to the actual winners it turns out that Shape of Water(2017) and Green book(2018) won in their years.

There is a degree of randomness in these predictions which means that results may vary if running the code several times. To make sure that results do not change drastically, I performed the experiment a few more times. The results were very interesting: 90% of the time, Shape of Water would win based on the predictions, however, the Green Book was never predicted. After doing more research, I found reasons for such results:

- Green Book has comedy as its genre 1 which is highly uncommon for the Oscars winners
- It was only nominated for 5 categories at the academy awards
- Its Rotten Tomatoes score stands at a 69 which is abnormally low for the Oscars winners

In combination, these factors contributed to the wrong results.

From the performed research, I found out that Random forest is a quite good algorithm to use for Academy Awards predictions. However, it has its own drawbacks. There will always be outliers

from the main relationships that will throw any algorithm off. On the other hand, when the general relationships are followed(like in the case with Shape of Water), the results are highly accurate.