University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

SriHarsha Gullapalli
Masters in data science
Person ID- 50414483

Abhishek Cheekatimarla
Masters in Data Science
Person ID- 50367388

REPORT
PART1-NETFLIX DATASET

1. Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? How many entries and variables does the dataset comprise?

The given dataset is a Netflix dataset. In this dataset we have the Tv shows and Movies that has been released till the year 2022 having 12 variables such as id, type, title, director, cast, country, date added, release year, rating, duration, listed in, description. Besides this, we have entries of 8806 rows of various movies, TV shows. We have both the numerical and the categorical values in this dataset along with the NAN values which can be replaced or delete. In total we have 4307 NAN values in various variables, in this assignment we have replaced the NAN values with the categorical values.

2.Provide the main statistics about the entries of the dataset (mean, std, number of missing values, etc.)

In our given dataset we have only one numerical value that is the release year. So, the mean, standard deviation , min, 25%, 50%, 75%, max of the release year is as follows:

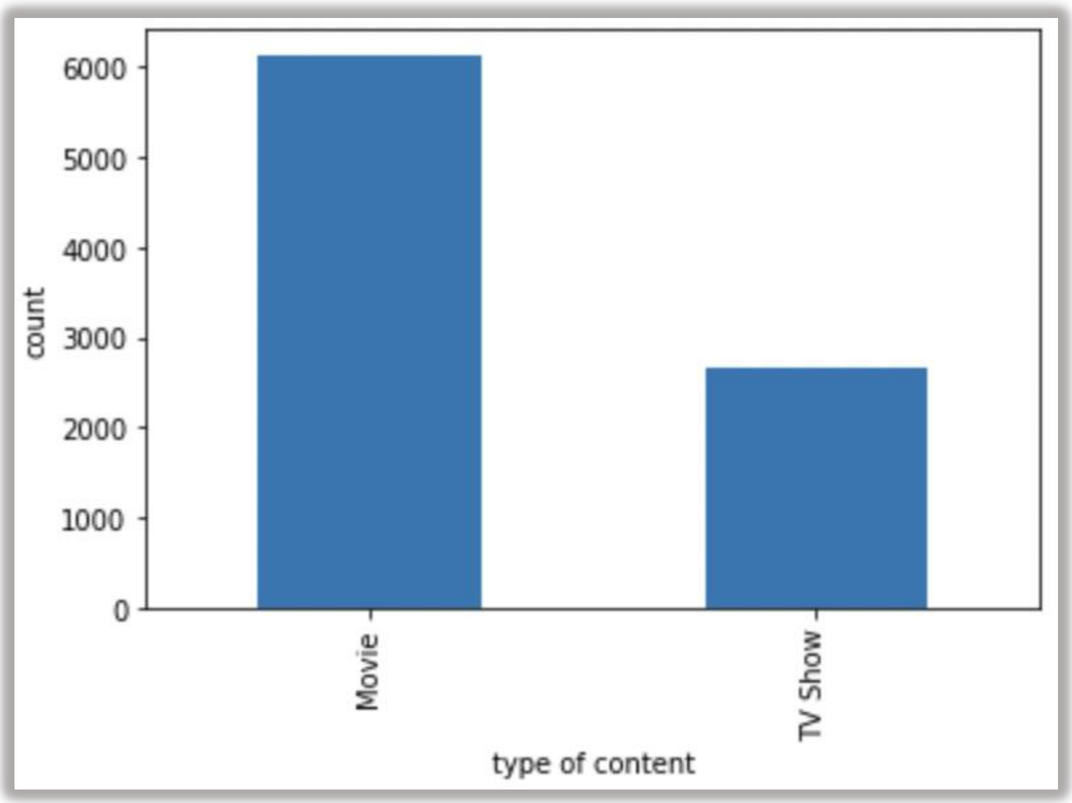
	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

As explained in the above question, we have a total of 4307 missing values of the given 12 variables. The following is the number of missing values with respect to variables is as follows in the below images.

```
show_id         False
type            False
title           False
director         True
cast             True
country          True
date_added       True
release_year    False
rating           True
duration         True
listed_in       False
description     False
dtype: bool
```

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

3.Provide at least 5 visualization graphs with short description for each graph, e.g. discuss if there any interesting patterns or correlations.

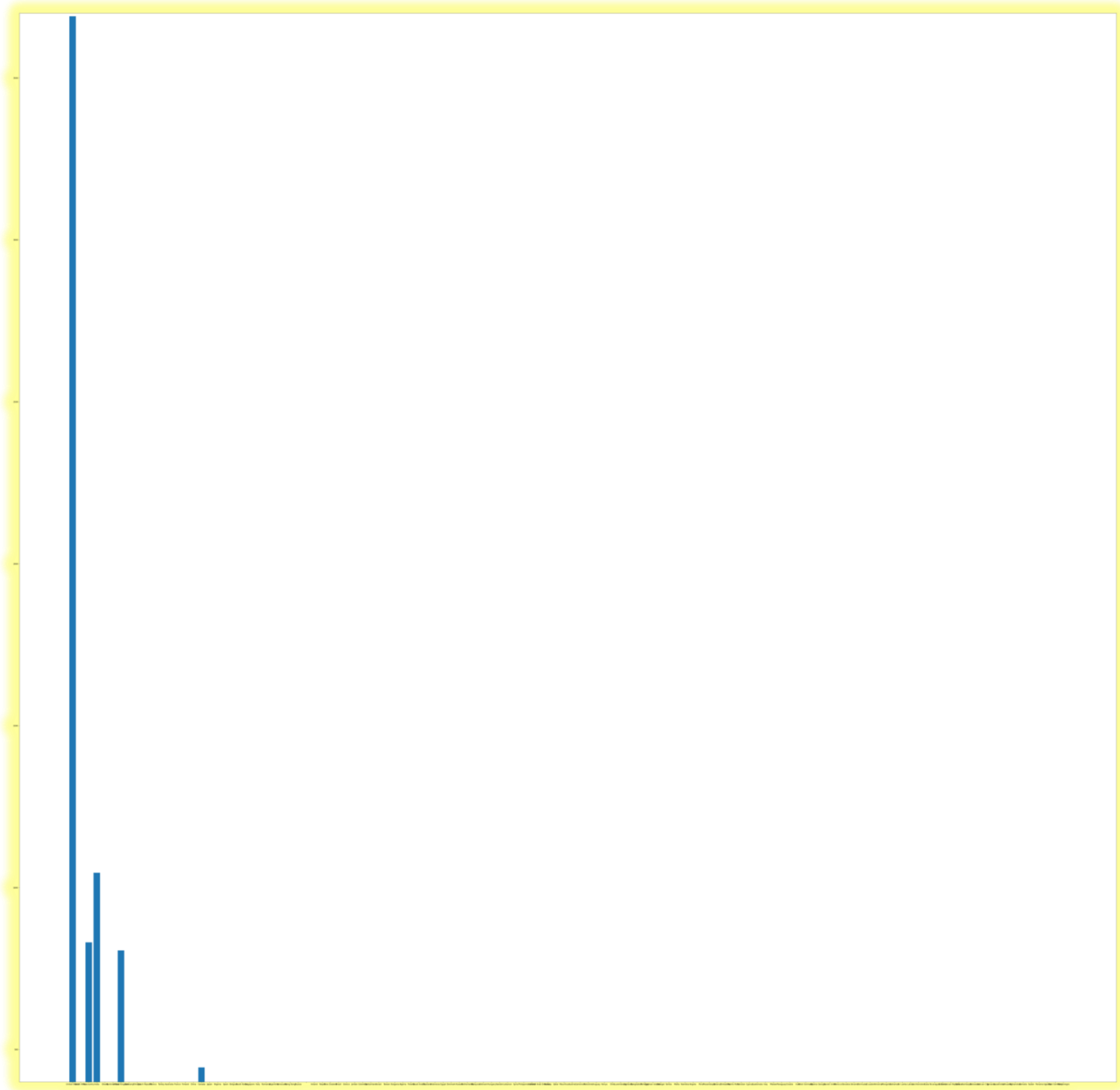


Which type of content is more in the Netflix data?

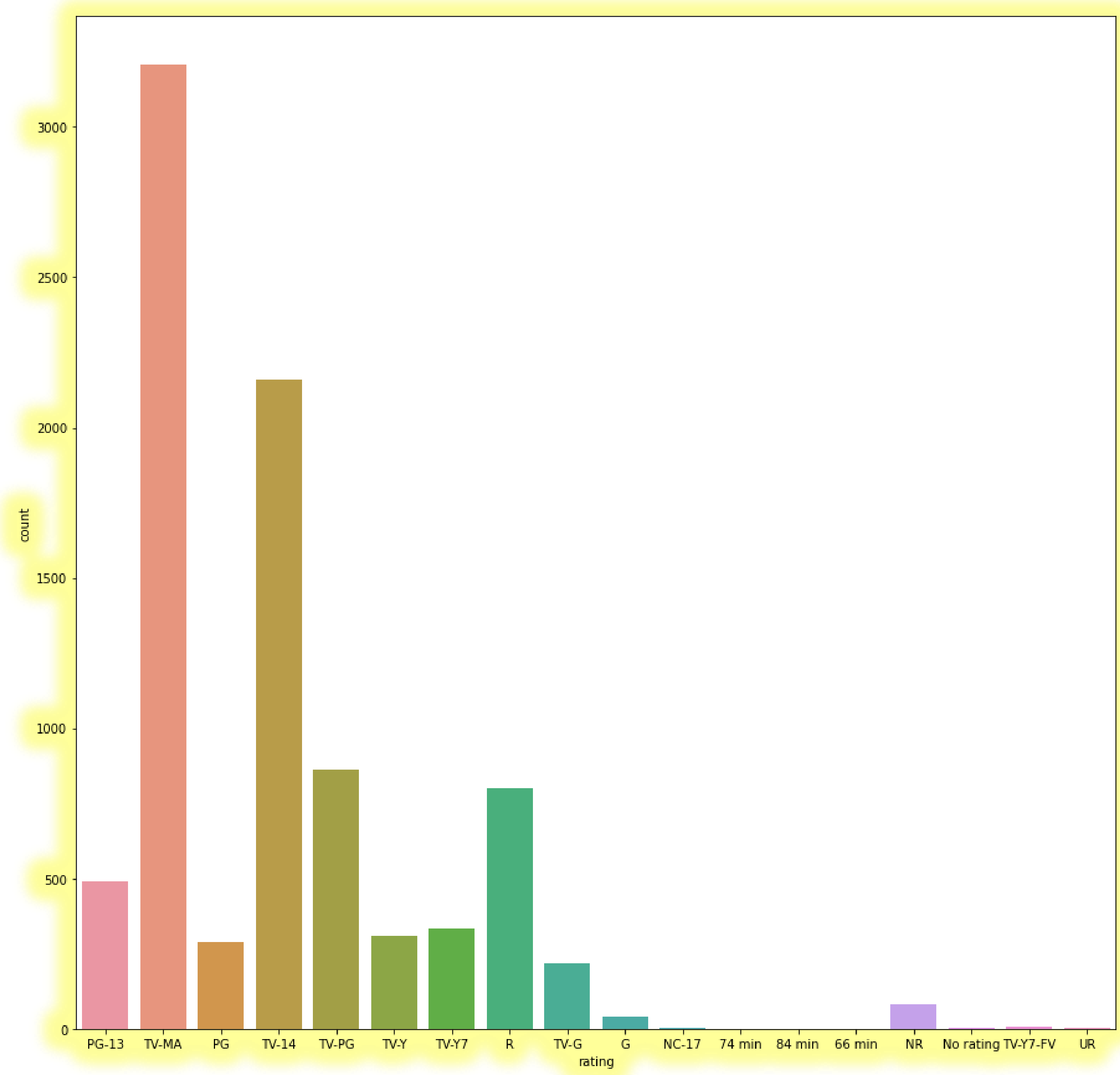
In above graph, we visualized the type of content that is type 'movie' is more in the Netflix when compared to the Tvshow.

Which country has produced more than 400 movies, TV shows?

In the below graph, we visualized the countries which has produced more than 400 type of movies, TV shows.

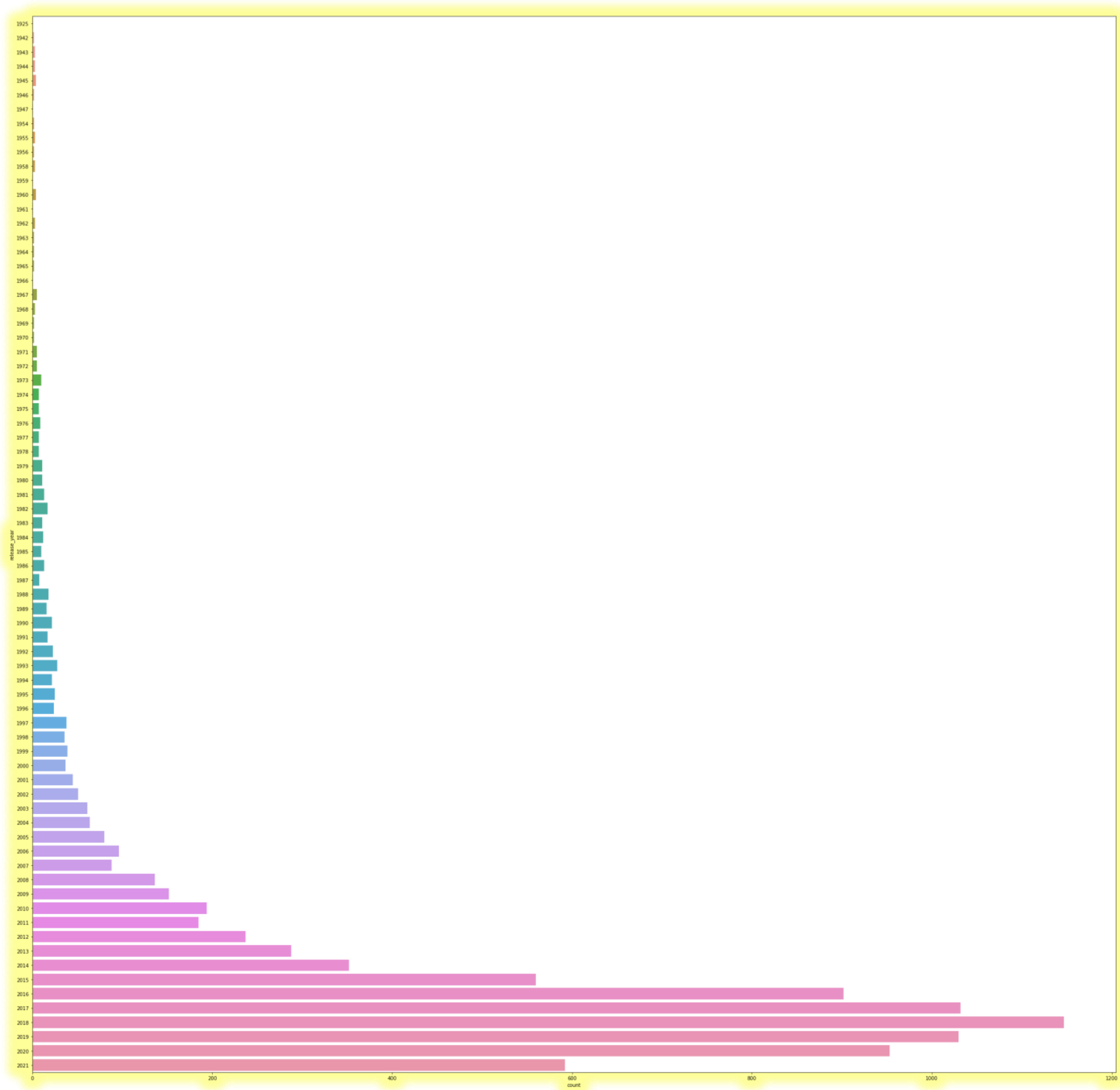


In the below graph, we visualize the ratings of movies, TV shows present in the Netflix data.



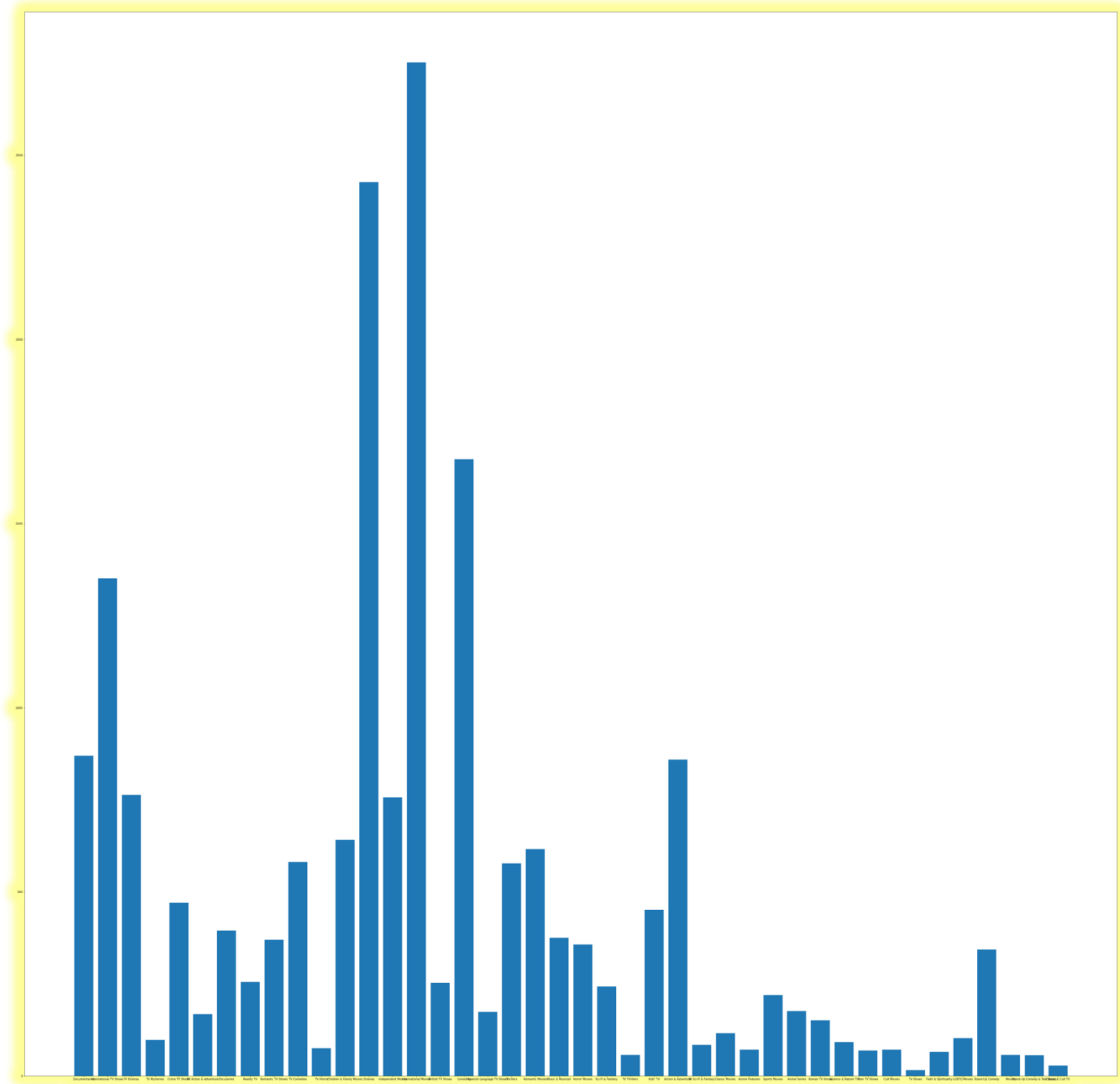
In which year the maximum movies has been released in Netflix?

In the below graph, We visualized and observed that in the year 2018 most number of the movies has been released and the count is around 1100-1200.



Which type of genre is produced more in Netflix dataset?

In the below graph, we visualized that the international movie genre has produced more in the Netflix dataset.



2500
2000
1500
1000
500
0
Documentaries
International TV Shows
TV Dramas
TV Mysteries
Crime TV Shows
Docuseries
Reality TV
Romantic TV Shows
TV Comedies
Children & Family Movies
Dramas
Independent Movies
International Movies
British TV Shows
Comedies
Spanish-Language TV Shows
Thrillers
Romantic Movies
Music & Musicals
Horror Movies
Sci-Fi & Fantasy
TV Thrillers
Kids' TV
Classic Movies
Anime Features
Sports Movies
Anime Series
Korean TV Shows
Teen TV Shows
Cult Movies
TV Shows
Faith & Spirituality
LGBTQ Movies
Stand-Up Comedy
Movies
Classic & Cult TV

PART-1- TITANIC DATASET

1. Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? How many entries and variables does the dataset comprise?

The given dataset is the Titanic dataset. In this dataset we have 8 variables that contains survived, Pclass, Name, sex, age, siblings/spouses aboard, parents/children aboard, fare. We have both the numerical, decimal and categorical data in this dataset. The dataset comprises of total 887 entries and the size of the data is about 7096. This data is used to know how many have survived in the titanic with respect to the names and the passenger class they have booked. The dataset does not have missing values. The ordinal nature is pclass and the continuous nature is age.

df_titanic.shape	df_titanic.size
(887, 8)	7096

2.Provide the main statistics about the entries of the dataset (mean, std, number of missing values, etc.)

The main statistics such as mean, std, count, min, 25%, 50%, 75%, max are as follows:

	Survived	Pclass	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
count	887.000000	887.000000	887.000000	887.000000	887.000000	887.00000
mean	0.385569	2.305524	29.471443	0.525366	0.383315	32.30542
std	0.487004	0.836662	14.121908	1.104669	0.807466	49.78204
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.00000
25%	0.000000	2.000000	20.250000	0.000000	0.000000	7.92500
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.45420
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.13750
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.32920

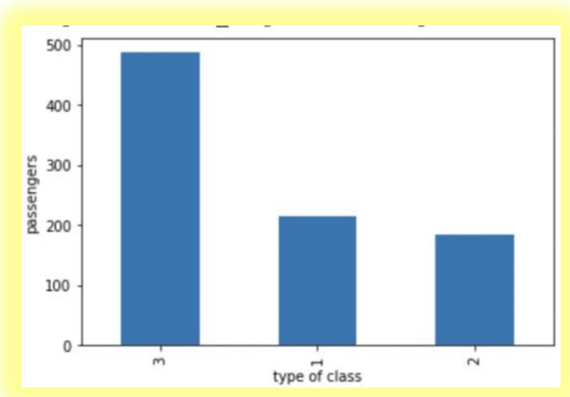
The number of missing values in the given dataset is zero as there are no mull values in the titanic dataset.

```
Survived                   False
Pclass                     False
Name                       False
Sex                        False
Age                        False
Siblings/Spouses Aboard    False
Parents/Children Aboard    False
Fare                       False
```

3.Provide at least 5 visualization graphs with short description for each graph, e.g. discuss if there any interesting patterns or correlations.

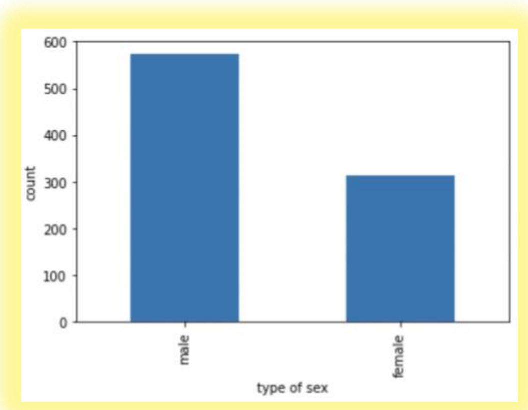
-How many passengers have booked different type of passenger classes?

From the below visualization we can say that the passengers have booked and are more in the third class when compared to the first and second classes.



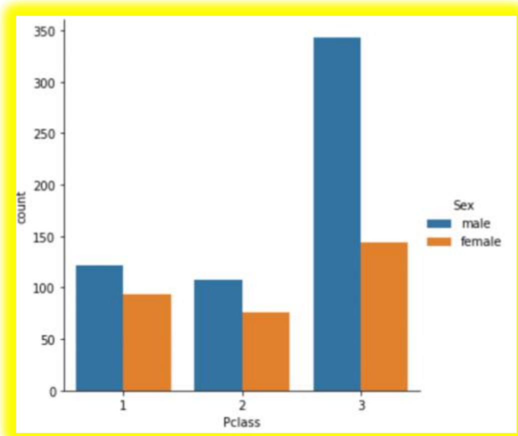
-Which gender of passengers are more on the titanic?

From the below visualization using bar chart, we came to know that the male passengers are more when compared to the female.



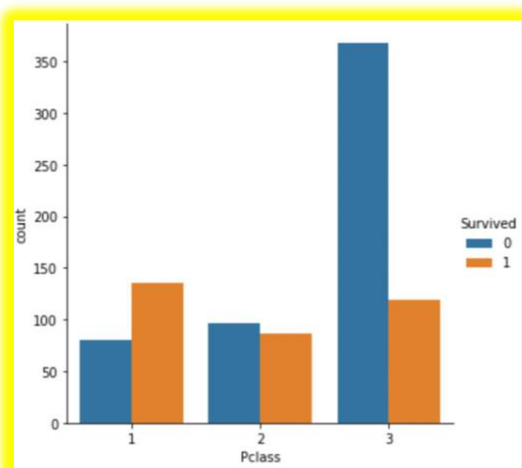
-How many male and female passengers are present on the different types of classes?

From the below visualization using bar graph, we can say that the third class has highest male and female population on titanic.



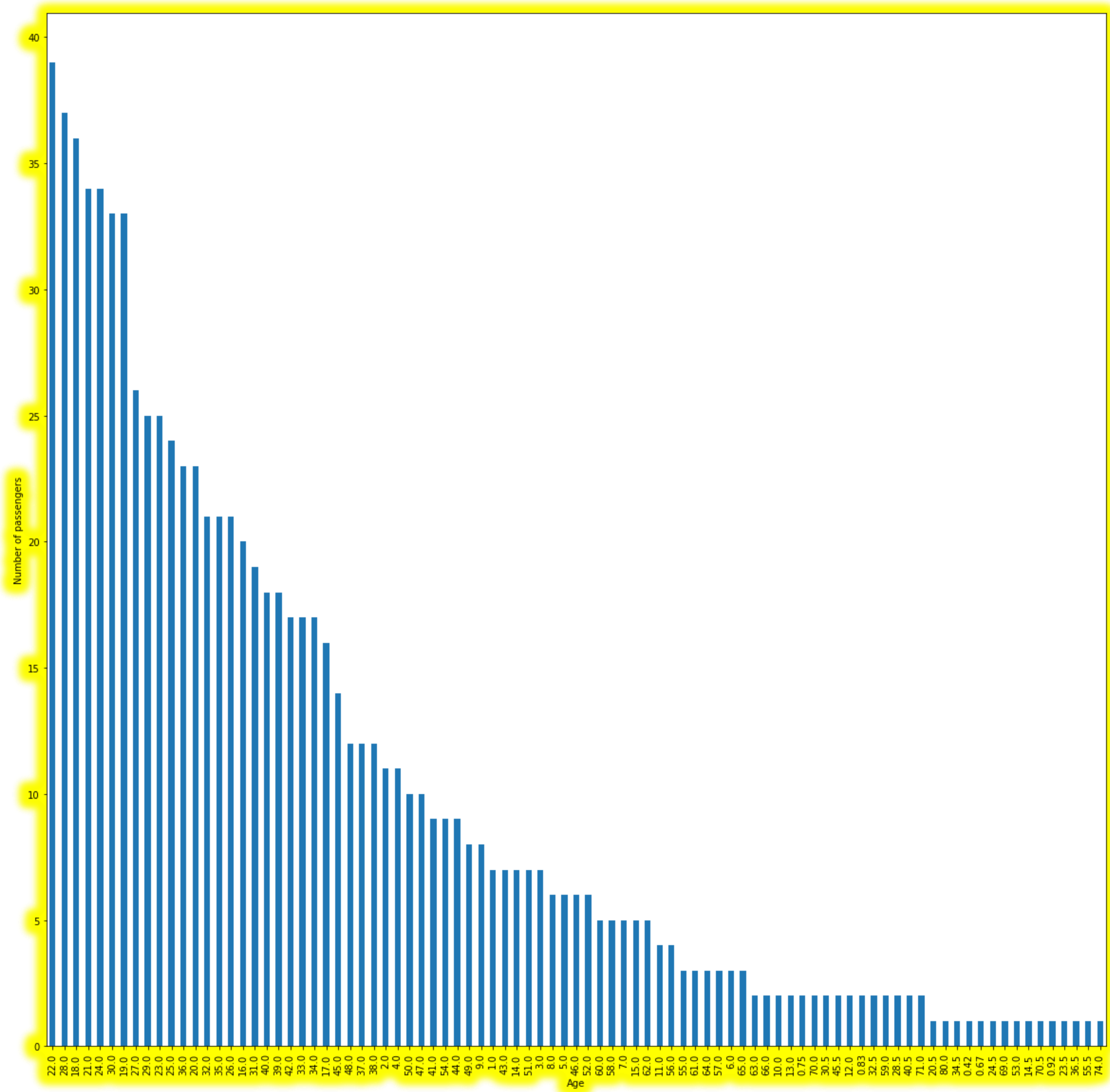
-In which type of classes, passengers have survived more?

From the below visualization, we can say that the passengers in the first class have survived more when compared to the other two classes. Where ‘0’ indicates the passenger has died and ‘1’ indicates that passenger is alive.



-Amongst all the passengers, which age type of passengers are more on board?

From the below visualization we can say that the passenger of age 22 are more when compared to the other passengers.



Number of passengers
40
35
30
25
20
15
10
5
0
22.0
28.0
18.0
21.0
24.0
30.0
19.0
27.0
29.0
23.0
25.0
36.0
20.0
32.0
35.0
26.0
16.0
31.0
40.0
39.0
42.0
33.0
34.0
17.0
45.0
48.0
37.0
38.0
2.0
4.0
50.0
47.0
41.0
54.0
44.0
49.0
9.0
1.0
43.0
14.0
51.0
3.0
8.0
5.0
46.0
52.0
60.0
58.0
7.0
15.0
62.0
11.0
56.0
55.0
61.0
64.0
57.0
6.0
65.0
63.0
66.0
10.0
13.0
0.75
70.0
30.5
45.5
12.0
0.83
32.5
59.0
28.5
40.5
71.0
20.5
80.0
34.5
0.42
0.67
24.5
69.0
53.0
14.5
70.5
0.92
23.5
36.5
55.5
74.0
Age

PART-1- PENGUIN DATASET

1. Provide brief details about the nature of your dataset. What is it about? What type of data are we encountering? How many entries and variables does the dataset comprise?

The dataset is about the different species of penguins having different body structure, their sex and year they were born. We encounter Numerical, categorical, decimal and NAN values. In this dataset we have 8 variables such as species, island, bill length, bill depth, flipper length, body mass, sex and year. Besides this, there are 19 NAN values present in different variables of the dataset. The dataset has 345 rows and the size of the data is 2752. The nature of the dataset is continuous.

```
penguin_df.size

2752
```

```
penguin_df.shape

(344, 8)
```

```
penguin_df.isna().any()

species              False
island               False
bill_length_mm        True
bill_depth_mm         True
flipper_length_mm     True
body_mass_g           True
sex                   True
year                 False
```

2.Provide the main statistics about the entries of the dataset (mean, std, number of missing values, etc.)

The main statistics of the dataset are as follows:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
count	344	344	342.000000	342.000000	342.000000	342.000000	333	344.000000
unique	3	3	NaN	NaN	NaN	NaN	2	NaN
top	Adelie	Biscoe	NaN	NaN	NaN	NaN	male	NaN
freq	152	168	NaN	NaN	NaN	NaN	168	NaN
mean	NaN	NaN	43.921930	17.151170	200.915205	4201.754386	NaN	2008.029070
std	NaN	NaN	5.459584	1.974793	14.061714	801.954536	NaN	0.818356
min	NaN	NaN	32.100000	13.100000	172.000000	2700.000000	NaN	2007.000000
25%	NaN	NaN	39.225000	15.600000	190.000000	3550.000000	NaN	2007.000000
50%	NaN	NaN	44.450000	17.300000	197.000000	4050.000000	NaN	2008.000000
75%	NaN	NaN	48.500000	18.700000	213.000000	4750.000000	NaN	2009.000000
max	NaN	NaN	59.600000	21.500000	231.000000	6300.000000	NaN	2009.000000

The data has a total of 19 NAN values listed below on the left side figure(1) and we filled those value with the most frequent one using fit transform. Now we have zero NAN values listed below on the right-side figure (2)

```
species              0
island               0
bill_length_mm       2
bill_depth_mm        2
flipper_length_mm    2
body_mass_g          2
sex                 11
year                 0
dtype: int64
```

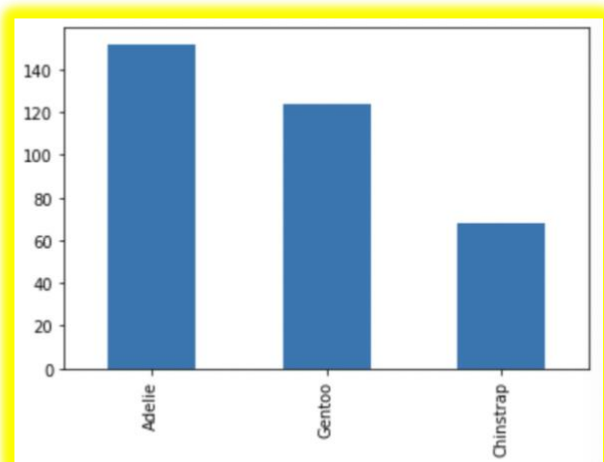
```
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
sex                  0
year                 0
dtype: int64
```

Figure (1) Figure (2)

3.Provide at least 5 visualization graphs with short description for each graph, e.g. discuss if there any interesting patterns or correlations.

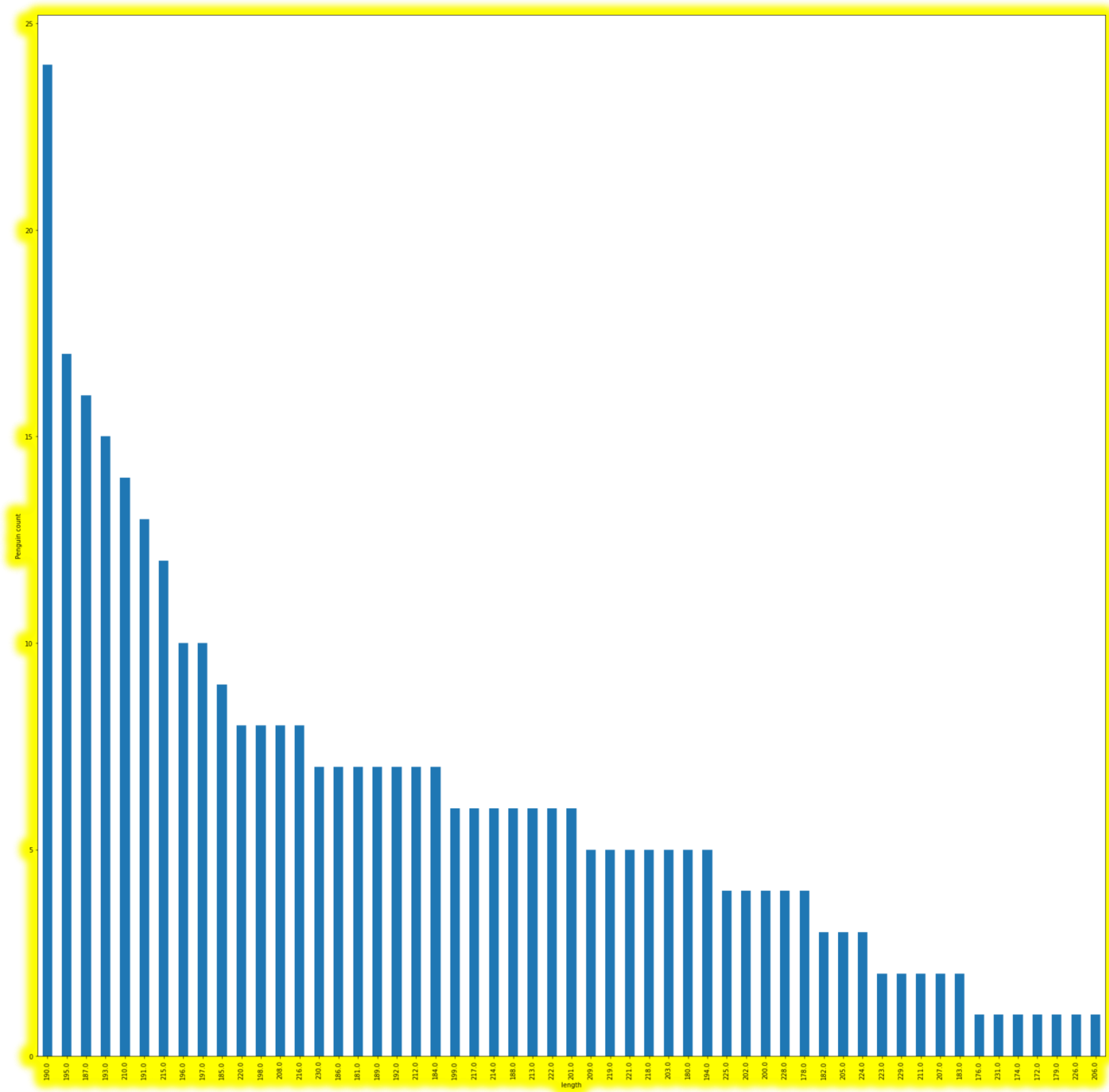
-Which type of penguin species are more?

From the below visualization we can say that the species Adelie are more when compared to the other two species Gentoo, chinstrap.



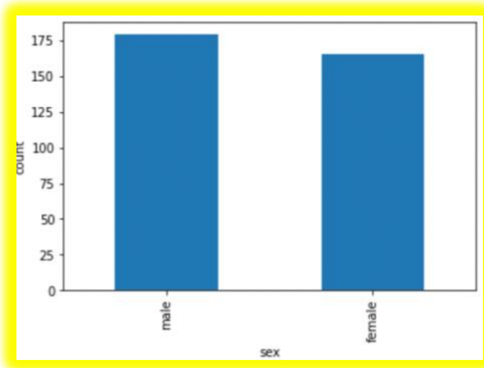
-What flipper length in mm does most of the penguins have?

The penguins having the flipper length of 190.0 are more and highest when compared to the other flipper lengths in mm. more than 20 penguins have the same flipper length in mm.

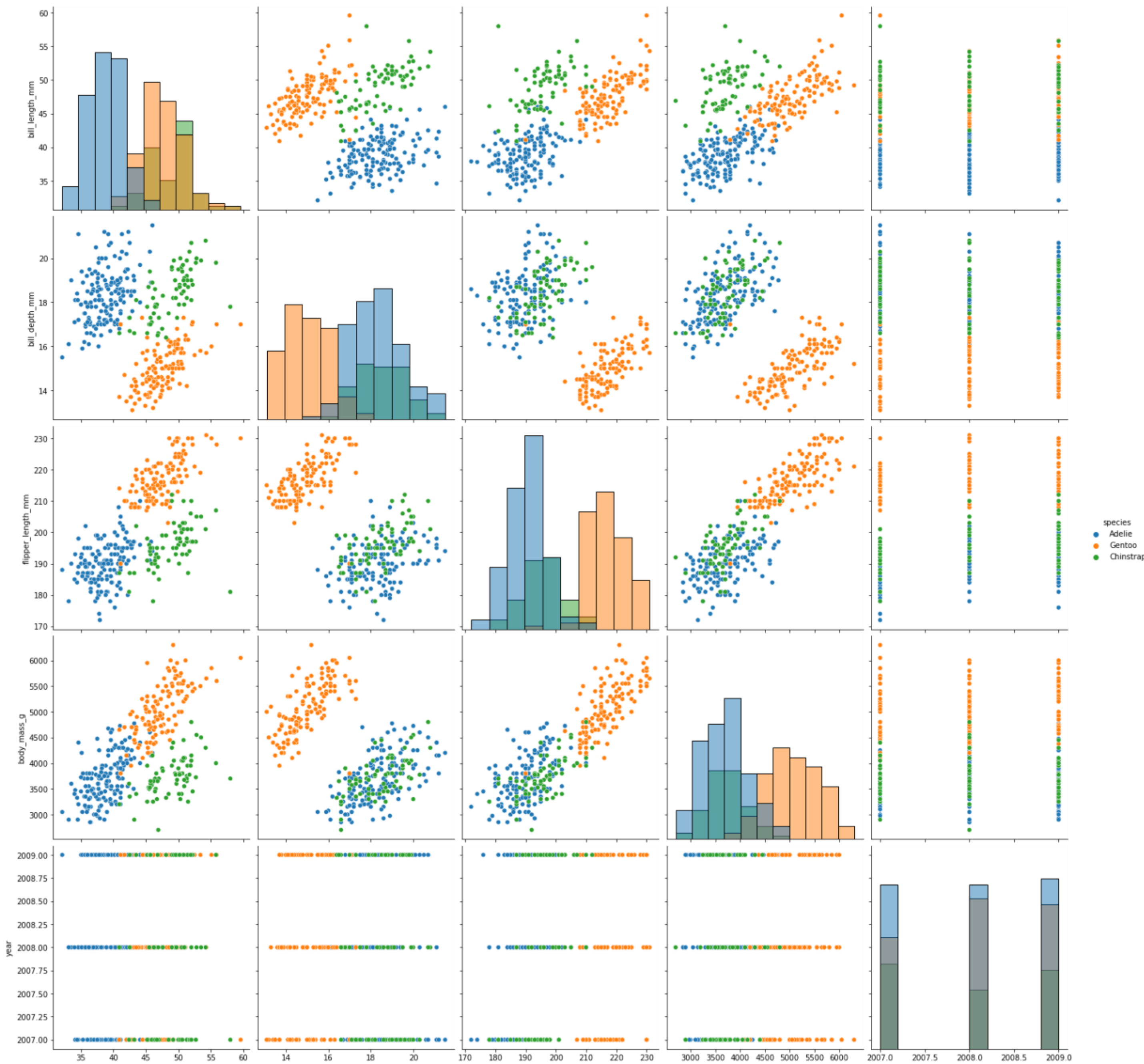


-Which type of gender are more in penguins?

From the below observation we came to know that the male and female penguins are almost same but comparatively male penguins are slightly more than the female ones.



From the below pair plot visualization, we found some strong co relations between the variables bill length, bill depth, flipper length and body mass of different species of penguins.



PART2 -PENGUIN DATASET

1.Provide your best accuracy and the weight vector?

By taking hyper parameters alpha = 0.001 and number of iterations as 10000 we get the accuracy as **86%.**

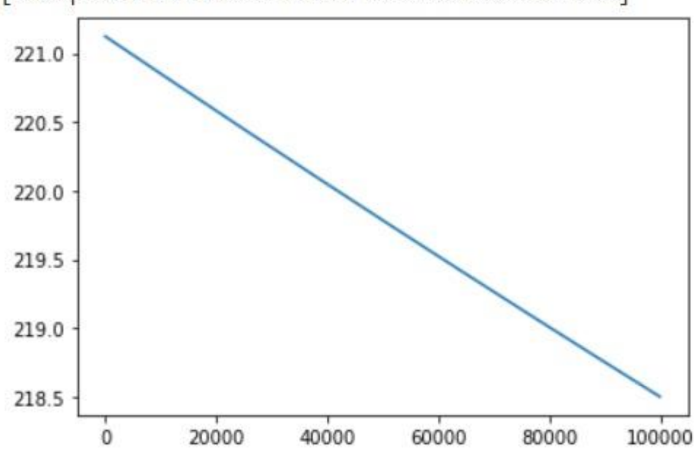
The weight vector for the above hypermeters is

```
[619] 1 trained_weights

array([[0.3549868530379999],
       [0.6384337474027506],
       [-0.4371722553347996],
       [-1.4868311325239696],
       [0.19729545182856445],
       [-2.41491084671798]], dtype=object)
```

2.Include loss graph and provide a short description?

The loss is getting reduced if we use gradient

```
1 plt.plot(loss)

[<matplotlib.lines.Line2D at 0x7fbdb84fa3d0>]
```



From the graph we can say that the loss did not get significantly reduced. Although the number of iterations are more.

3.Explain how hyperparameters influence the accuracy of the model. Provide at least 3 different setups with learning rate and #iterations and discuss the results?

For training accuracy by taking hyper parameters alpha = 0.000001 and number of iterations as 10000 we get the accuracy as 44.56521739

```
m,n = X_train.shape
weight = np.random.randn(1,n)
lr = LogitRegression(x=X_train,y=y_train, alpha = 0.000001, num_iter = 10000,weights=weight)
trained_weights,loss = lr.fit()
pred = lr.pred()
# test_lr = LogitRegression(x=X_test,y=y_test, alpha = 0.000001, num_iter = 1000,weights=trained_weights)
# result = test_lr.pred()
```

[33]

```
count =0
for i in range(len(pred)):
    if pred[i] == y_train[i]:
        count+=1
accuracy = count/276
print(accuracy*100)
```

```
44.565217391304344
```

For Testing accuracy :

[37]

```
test_lr = LogitRegression(x=X_test,y=y_test, alpha = 0.000001, num_iter = 1000,weights=trained_weights)
result = test_lr.pred()
```

```
count =0
#print(result)
for i in range(len(result)):
    if result[i] == y_test[i]:
        count+=1
test_accuracy = count/len(y_test)
print(test_accuracy*100)
```

```
50.0
```

For training accuracy by taking hyper parameters alpha = 0.000001 and number of iterations as 100000 we get the accuracy as:

```
m,n = X_train.shape
weight = np.random.randn(1,n)
lr = LogitRegression(x=X_train,y=y_train, alpha = 0.000001, num_iter = 100000,weights=weight)
trained_weights,loss = lr.fit()
pred = lr.pred()


```

```
count =0
for i in range(len(pred)):
    if pred[i] == y_train[i]:
        count+=1
accuracy = count/276
print(accuracy*100)
```

```
63.40579710144928
```

For Testing accuracy:

[605]

```
test_lr = LogitRegression(x=X_test,y=y_test, alpha = 0.000001, num_iter = 1000,weights=trained_weights)
result = test_lr.pred()

```

[608]

```
count =0
for i in range(len(result)):
    if result[i] == y_test[i]:
        count+=1
test_accuracy = count/len(y_test)
print(test_accuracy*100)
print(count)
```

```
45.588235294117645
31
```

For training accuracy by taking hyper parameters alpha = 0.001 and number of iterations as 10000 we get the accuracy as:

[617]

```
m,n = X_train.shape
weight = np.random.randn(1,n)
lr = LogitRegression(x=X_train,y=y_train, alpha = 0.01, num_iter = 10000,weights=weight)
trained_weights,loss = lr.fit()
pred = lr.pred()


```

[618]

```
count =0
for i in range(len(pred)):
    if pred[i] == y_train[i]:
        count+=1
accuracy = count/276
print(accuracy*100)
```

```
75.0
```

The weight vectors for the above hyperparameters:

[619]

```
trained_weights
```

```
array([[0.3549868530379999],
       [0.6384337474027506],
       [-0.4371722553347996],
       [-1.4868311325239696],
       [0.19729545182856445],
       [-2.41491084671798]], dtype=object)
```

For Testing accuracy: 86%

[620]

```
test_lr = LogitRegression(x=X_test,y=y_test, alpha = 0.001, num_iter = 10000,weights=trained_weights)
result = test_lr.pred()

```

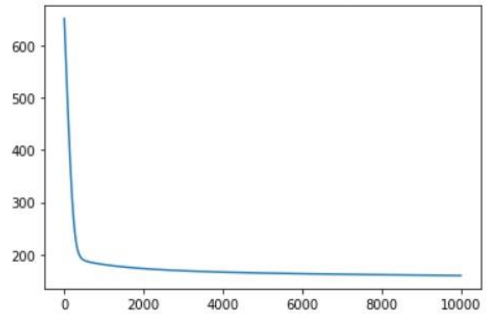
```
count =0
for i in range(len(result)):
    if result[i] == y_test[i]:
        count+=1
test_accuracy = count/len(y_test)
print(test_accuracy*100)
print(count)
```

```
86.76470588235294
59
```

The loss graph: (The loss got decreased significantly from iterations 0 to 2000 after that there is no significant decrease in the loss)

```
plt.plot(loss)
```

```
[<matplotlib.lines.Line2D at 0x7fbdb80b9ed0>]
```



The accuracy score having hyper parameters as 0.001 and iterations as 10000 gives the best accuracy score for our model. The optimal hyperparameters are to be recognized by trial and error method.

4.Discuss the benefits/drawbacks of using a Logistic Regression model.

Advantages	Disadvantages
Logistic regression is more straightforward to apply, analyze, and train.	Logistic Regression should not be done if the number of observations is smaller than the number of features; otherwise, overfitting may occur.
Overfitting is less likely with logistic regression, but it can happen in high-dimensional datasets. In these cases, regularization (L1 and L2) techniques may be used to avoid over-fitting.	Independent variables must be linearly connected to the log odds (log(p/(1-p)) in order to use logistic regression.
It performs well when the dataset is linearly separable and has good accuracy for many simple data sets.	The average or no multicollinearity between independent variables is required for logistic regression.

PART-3 LINEAR REGRESSION

As we are using the linear regression, we are finding the **flipper_length_mm** using other feature values.

```
[365]
Y = penguin_df.loc[:,"flipper_length_mm"]
X = penguin_df.drop("flipper_length_mm",axis=1)
xval = X.values
yval = Y.values
X_train, X_test = xval[:276], xval[276:]
y_train, y_test = yval[:276], yval[276:]
```

1.Providing the LOSS VALUE and WEIGHT VECTOR?

The loss value is calculated using the mean square error and the weights are calculated using the **OLS method** with the formula given in the class.

$$w = (X^T X)^{-1} X^T y$$

```
def OLS(self):
    n = np.ones(len(self.x))
    x = self.x
    x = np.array(x)
    y = self.y
    y = np.array(y)
    x_len,x_width = x.shape
    x_mat =x
    if x_width ==1:
        x_mat = np.stack((n,x.self),axis = -1)
    x_trans = np.matrix.transpose(x_mat)
    z = np.linalg.inv(np.matmul(x_trans,x).astype('float32'))
    return np.matmul(z,np.matmul(x_trans,y))
def MSE(self):
    N=len(self.y)
    result =[]
    weights = np.transpose(self.OLS())
    for i in self.x:
        result.append(np.matmul(weights,i))
    result = ((np.array(self.y)-np.array(result))**2).mean()
    return result
```

The result from the above snippet are the **weight vector** and the **loss value**.

The resultant output for the Loss value and the weight vector is:

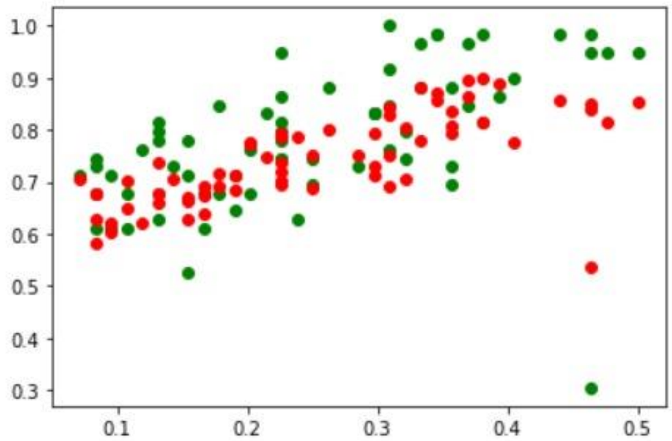
```
1 linr.OLS()

array([0.13840320763273484, -0.01364163348376582, 0.014285825009678899,
       0.17679761200816035, 0.6092359921323506, 0.045243744060526114],
      dtype=object)

[370]  1 linr.MSE()

0.00929606785606843
```

2.Show the plot comparing the predictions vs the actual test data?

The below scatter plot is comparing the **predictions vs the actual test** data:

```
import matplotlib.pyplot as plt

x = penguin_df.loc[:,"culmen_depth_mm"]
x=x.values
x=x[276:]
plt.scatter(x,y_test,c="green")
plt.scatter(x,result,c="red")
# plt.legend("actual","predicted")
```

```
<matplotlib.collections.PathCollection at 0x7fbdb9cd34d0>
```



3.Discuss the benefits/drawbacks of using OLS estimate for computing weights?

BENEFITS	DRAWBACKS
The statistical method elucidates cost structures and differentiates the roles of various variables in determining output.	As with OLS, a large data set is necessary in order to obtain reliable results.
Cost drivers or how inputs contribute to output are two ways to interpret coefficients.	When dealing with limited samples, estimating weights might have surprising outcomes.
It is very easy to explain and to understand	In comparison to the rest of the training data, some points in the training data have excessively large or small values for the dependent variable.

PART-4 RIDGE REGRESSION

1.Provide your loss value and the weight vector?

The loss value is calculated using the mean square error and the weights are calculated using the **OLS method** with the formula given in the class.

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\lambda I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

```
class RidgeRegression():
    def __init__(self,x,y):
        self.x = x
        self.y = y
    def OLS(self):
        n = np.ones(len(self.x))

        x = self.x
        x = np.array(x)
        y = self.y
        y = np.array(y)
        x_len,x_width = x.shape
        x_mat =x
        if x_width ==1:
            x_mat = np.stack((n,x.self),axis = -1)
        x_trans = np.matrix.transpose(x_mat)
        lam = np.identity(6)
        inter =np.matmul(x_trans,x)+lam
        z = np.linalg.inv(inter.astype('float32'))
        return np.matmul(z,np.matmul(x_trans,y))
    def MSE(self):
        N=len(self.y)
        result =[]
        weights = np.transpose(self.OLS())
        for i in self.x:
            result.append(np.matmul(weights,i))
        error = ((np.array(self.y)-np.array(result))**2).mean()
        return error,result
```

The result from the above snippet are the **loss value** and the **weight vector**.

The resultant output for the Loss value and the weight vector is:

```
[441]  1 ridr_error

    0.009412904430244465

[442]  1 ridr_weights = ridr.OLS()

       1 print(ridr_weights)

    [0.15284325356185624 -0.01490576521074534 0.029362604910945
     0.19779357685790766 0.5366923931594774 0.04202403868745974]
```

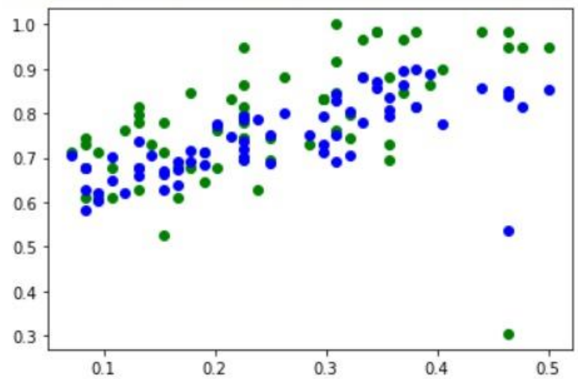
2.Show the plot comparing the predictions vs the actual test data?

The below scatter plot is comparing the **predictions vs the actual test** data:

```
[444]
ridge_result =[]
for i in X_test:
    ridge_result.append(np.matmul(lin_weight,i))
ridge_result
```

```
plt.scatter(x,y_test,c="green")
plt.scatter(x,ridge_result,c="blue")
# plt.legend("actual","predicted")
```

```
<matplotlib.collections.PathCollection at 0x7fbdb9b9cc50>
```



3.Discuss the difference between Linear and Ridge regressions. What is the main motivation of using l2 regularization?

LINEAR REGRESSION	RIDGE REGRESSION
Linear regression uses a best-fit straight line to establish a link between a dependent variable (Y) and one or more independent variables (X).	Ridge Regression is a technique for dealing with multicollinear data.
For linearly separable data, linear regression performs extremely well.	It results in a large level of variance among the independent variables; we can adjust the value of the independent variable, but this will result in information loss.
Although linear regression is a useful tool for analyzing correlations between variables, it is not recommended for most practical applications since it oversimplifies real-world situations by assuming a linear relationship between variables.	The model gets into issues like overfitting or underfitting.

Main motivation for using l2 regularization:

The purpose of L2 regularization is to lower the likelihood of model overfitting. It makes model more robust and decreases the complexity of the model.

Team Member	Assignment Part	Contribution (%)

Abhishek Cheekatimarla	Part-1,2,3,4 discussed and collabarated	50%
SriHarsha Gullapalli	Part-1,2,3,4 discussed and collabarated	50%