Project Report

Predicting Ad Revenue

By Amrutha Venkatesha

Table of Contents

Problem statement:

XYZ Studios is a leader in original TV and radio content and wants to launch a streaming platform for an ad revenue of $7M over the next year. They have a new ad-based subscription model at $5.99 a month and they need a forecast of how that will perform when compared with competitors.

Context:

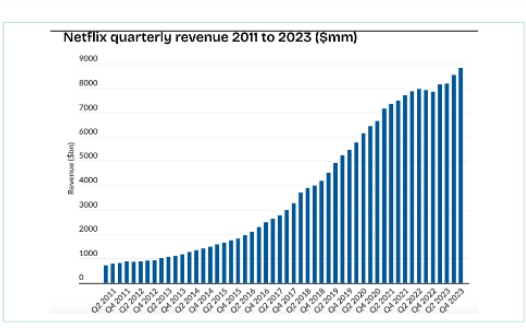
XYZ has a huge list of original programming in TV and radio formats. They also have a big customer base in these two fronts along with a long line of advertisers. There has been a delay in entering the digital space and there are now at least 7 big competitors who completely own the online streaming platform, Amazon being the biggest since they have a footing in Radio, TV, retail and original content. XYZ would like to attract digital customers with their new subscription model. It is a rewards based model for both customers and advertisers. Customers accumulate points for ads watched and they can redeem these points either to reduce next month's subscription price or to purchase a product from the advertiser (instore or online).

Data Cleaning

Different sources presented data differently

Twitch annual revenue 2016 t0 2022 ($mm)

Year	Revenue ($mm)
2016	275
2017	300
2018	880
2019	1230
2020	1890
2021	2675
2022	2800



Some revenue data was arranged by Months, instead of year. The ‘agg’ function was useful in arranging data by year

YtRevAd = YtRev.groupby(YtRev["Date"].dt.year)['Ads Revenue (Mn)'].agg(sum)

Slicing was useful when only the Year needed to be stored

disRev['Year'] = disRev['Year'].str[-4:]

The data from different companies came in different csv files and the merge operation was used a lot to combine multiple data frames to create a master data frame

df1 = disRevdf.merge(NetRev, on="Year", how="outer", suffixes=('_disney','_netflix'))

Most missing data was fixed by looking up various websites for revenue and subscriber numbers.

Any more null and not available data were replaced with 0 which just meant the company was non-existent then.

df.replace(np.nan, 0, inplace=True)

Exploratory Data Analysis

Explore the data using describe(), info(), dtypes and other methods and attributes available on the data frame

Melt and realign the data in a plottable format

df_T_clean = pd.melt(df_T,id_vars= ['company data'])

df_T_clean = df_T_clean.rename(columns={0:'Year', 'value': 'Revenue($)/Subscribers(count)'})

df_T_clean['type'] = df_T_clean['company data'].str.split('_').str[0]

df_T_clean['company'] = df_T_clean['company data'].str.split('_').str[1]

This is how the cleaned up Revenue and Subscribers data looked like:

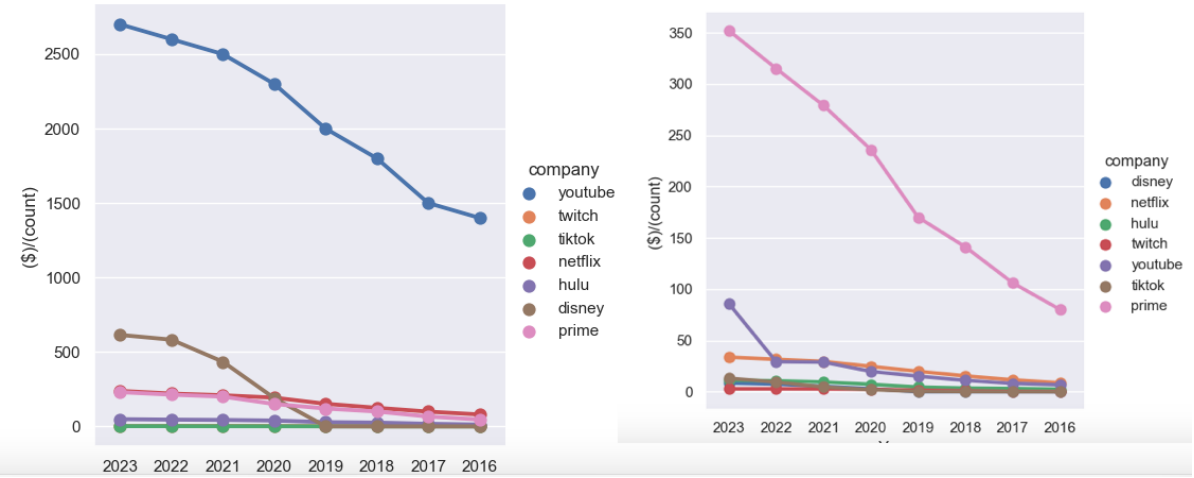
[92]:

	Year	Revenue_disney	Revenue_netflix	Revenue_hulu	Revenue_twitch	Revenue_youtube	Revenue_tiktok
0	2023	8.400	33.70	4.4	2.8	86.000	13.200
1	2022	7.400	31.60	4.1	2.8	29.243	9.400
2	2021	6.200	29.60	3.8	2.7	28.845	4.600
3	2020	2.802	24.72	1.5	2.3	19.772	1.900
4	2019	0.000	19.83	1.6	1.5	15.149	0.350
5	2018	0.000	15.39	1.5	0.9	11.100	0.150
6	2017	0.000	11.60	1.0	0.4	8.100	0.063
7	2016	0.000	8.80	0.9	0.1	6.700	0.000

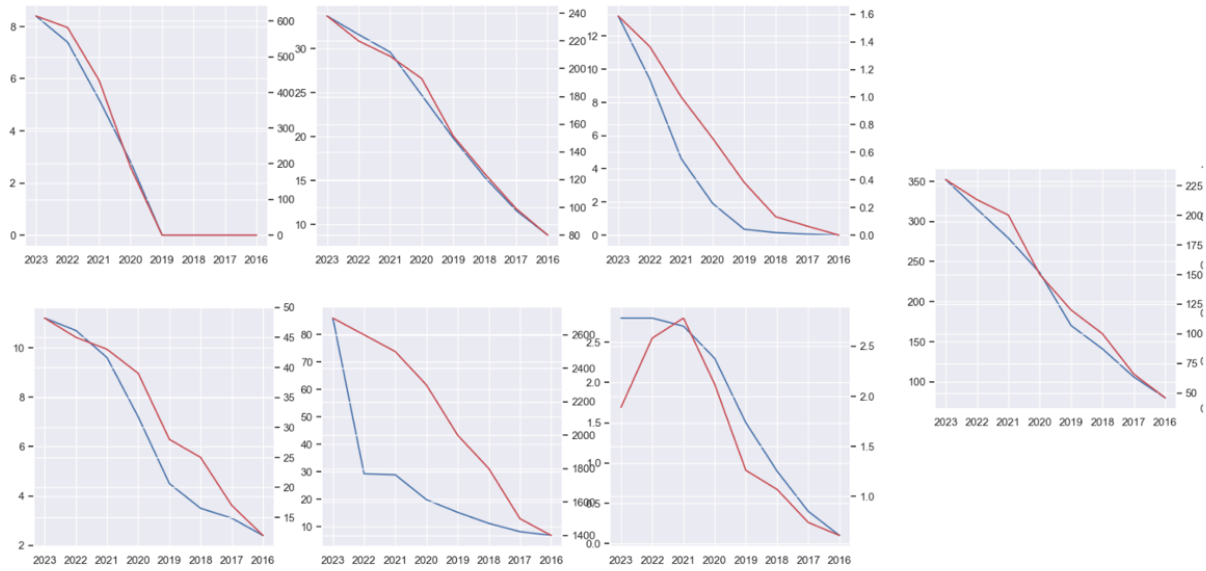
[151]:

	Year	Subscribers_youtube	Subscribers_twitch	Subscribers_tiktok	Subscribers_netflix	Subscribers_hulu	Subscribers_disney
11	2023	2700.0	1.89	1.587	238.0	48.2	614.0
10	2022	2600.0	2.58	1.366	220.0	45.0	582.0
9	2021	2500.0	2.78	1.000	209.0	43.0	434.0
8	2020	2300.0	2.12	0.700	192.9	39.0	189.9
7	2019	2000.0	1.26	0.381	151.6	28.0	0.0
6	2018	1800.0	1.07	0.133	124.3	25.0	0.0
5	2017	1500.0	0.74	0.065	99.0	17.0	0.0
4	2016	1400.0	0.61	0.000	79.9	12.0	0.0

This chart shows the revenue and subscribers growth by the year and for each company. We can look more closely at each company as shown in the next plot



The next chart shows how each of the company grew over the past 7 years



The above charts and data helps us formulate the following hypothesis

Statistical Analysis

Hypothesis

Based on the above plots, it seems like # of subscribers and revenue generated are related positively.

Null Hypothesis - Number of subscribers does not influence the revenue generated

We can calculate the p-value to see if this holds

Pearson's co-efficient shows a negative correlation between the two variables as shown below. The [0,1] array gives the results we want

corr_mat = np.corrcoef(x,y)

Permutation replicates on Pearson's co-eff also gives a very high p-value.

Next the linear regression to find the slope and intercept to establish a relationship is used. This is no clear relationship

bs_slope_reps, bs_intercept_reps = draw_bs_pairs_linreg(df_T_sub_series, df_T_rev_series, 1000)

Compute and print 95% CI for slope

print(np.percentile(bs_slope_reps, [2.5, 97.5]))

Hypothesis Conclusion:

The p-value is very high and so we cannot reject the null.

Also, the slope and intercept graph does not show strong correlation between the two variables.

Assumption for our model:

The revenue listed is from various sources, especially for Amazon Prime, whose parent company is a conglomerate, unlike the other companies listed but much like our company XYZ.

This means XYZ should be safe to implement a new subscription model given its vast customer base and availability of original content.

Model Building

Pre-Processing

The first step in this was to create dummies for categorical data

```
dfo=df.select_dtypes(include=['object']) # select object type columns

df = pd.concat([df.drop(dfo, axis=1), pd.get_dummies(dfo)], axis=1)
```

Based on the data set obtained, divide the data into test and training data sets

Pick the dependent and independent variable. Revenue, here is the dependent variable

```
X_train, X_test, y_train, y_test=train_test_split(X, y,
                                                  test_size = 0.25,
                                                  random_state = 246)
```

K-means clustering to predict revenue and subscribers

Cluster analysis is a technique used in data mining and machine learning to group similar objects into clusters. K-means clustering is a widely used method for cluster analysis where the aim is to partition a set of objects into K clusters in such a way that the sum of the squared distances between the objects and their assigned cluster mean is minimized.

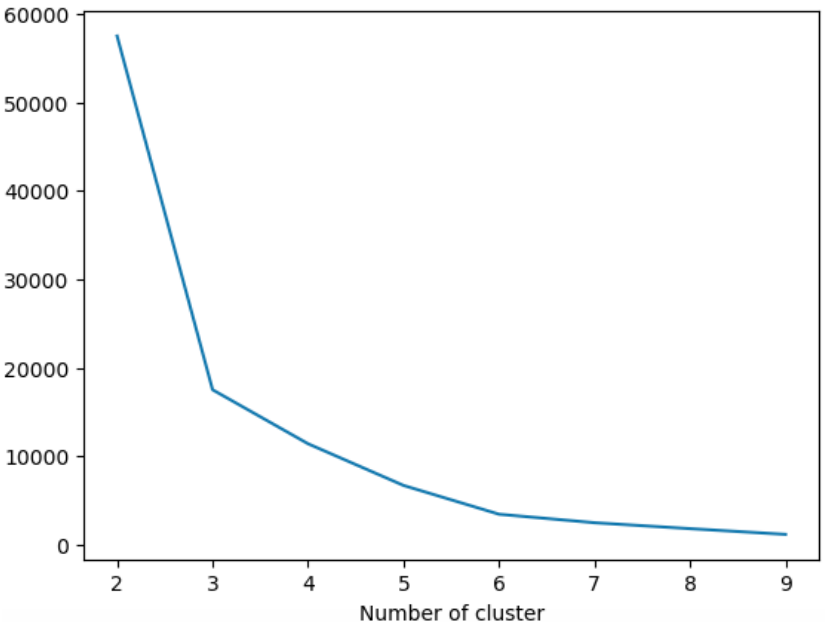
K-means is relevant here as it helps us figure out which cluster XYZ might belong in.

The knee-elbow curve which predicts the best number of clusters looks like this when either revenue or number of subscribers is the dependent variable.

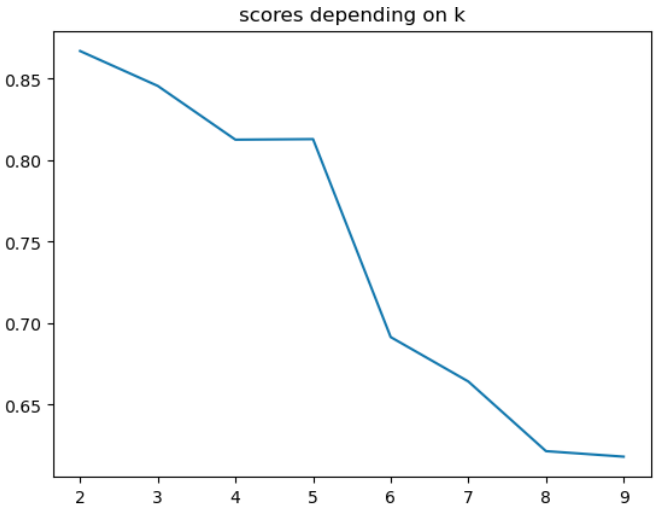
The code to generate the centroids, the silhouette scores is as below:

```
for k in range(2, 10):
    kmeans = KMeans(n_clusters=k, max_iter=1000).fit(x_cols)
    #x_cols["clusters"] = kmeans.labels_
    sse[k] = kmeans.inertia_
    score[k] = silhouette_score(x_cols, kmeans.labels_)
```

Knee-elbow curve to determine the number of clusters:



Silhouette scores:



Centroids:

centroids = model.cluster_centers_

array([[8.79391489],
[295.5],
[116.6]])

Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. Here, we build a model that establishes the relationship between subscribers and ad revenue generated.

Linear regression model is created and the RMSE score

rModel = linear_model.LinearRegression()

RMSE is around 50% for this model

np.sqrt(((predictions - targets) ** 2).mean())

Conclusion

Based on K-means

Prediction for XYZ:

Revenue of 116M with 2.56 M subscribers. Silhouette Score was 86% which is high, saying that this is a good model which shows clustering around the means.

Based on Linear Regression

RMSE is about 50 i.e if we use subscribers to predict revenue we may be right 50% of the times, which is as good as taking a random guess or a coin toss.

Number of subscribers will not be a good predictor for Ad Revenue.

Bringing it together

The models predict a 116M revenue with 2.56M subscribers for the first year and this is with 86% confidence levels. The expectation was to generate much lower revenue. The model also says subscriber count is not the primary indicator for ad revenue, and this means the subscription price of $5.99 a month is a good low end start to attract future customers.

It will be interesting to see which of the competitors will lose what percentage of their customer base to XYZ. It is also worth exploring how to entice premier advertisers based of XYZ’s rich original content.