Capstone Project Report
Factors for Success on Broadway

Loni J. Wood

Northwest Missouri State University, Maryville MO 64468, USA
S212226@nwmissouri.edu

Abstract. Broadway is considered a major contributor to New York City's economy, generating nearly $15 million dollars between 2018-2019. [7] To better understand what factors make Broadway shows a success, this project used Excel and Python machine-learning models to predict gross sales. By using exploratory data analysis (EDA) and regression analysis, this project found that musicals performed in the second quarter of the season had the highest gross sales. Further analysis concluded that, although inflation was a major driver in the surging gross sales, attendance and increasing average ticket prices also played a significant role. The quadratic polynomial model performed the best among all the independent variables, which were Year, Attendance, Capacity, Gross_Potential, and Performances. Since attendance had the strongest significance to predicting gross, but would not be generally known in advance, the models were also tested without using attendance as an independent variable. When attendance was removed, the models were still able to successfully predict gross sales using the remaining independent variables.

Keywords: Broadway · machine learning · theatre · predictive analysis · musicals

1 Introduction

Known as the "highest form of theatrical entertainment in the world", Broadway theatre in New York City is host to a variety of musicals, plays, and even concerts between the 41 theatres.[4] The domain chosen for this project was about Broadway shows. While millions of people attend Broadway shows annually [1], outside of the plot, what makes a show successful?

In the dataset search, a variety of websites such as Kaggle, BroadwayLeague, BroadwayWorld, Playbill, GitHub, and Google were reviewed. The search found a dataset at Austin Cory Bart's Corgis DataSet Project titled "Broadway CSV File" [2]. The data was originally collected by The Broadway League and "is the national trade association for the Broadway industry"[2].

The dataset that was used in this project can be found at https://corgis-edu.github.io/corgis/csv/broadway/.

2 Project Plan

The two main tools that were used in the project are:

- Excel
  - Data Cleaning
  - Visualizations
  - Modeling
- Python
  - Data Cleaning
  - Visualizations
  - Machine Learning

The steps needed to implement the project are shown in Figure 1.

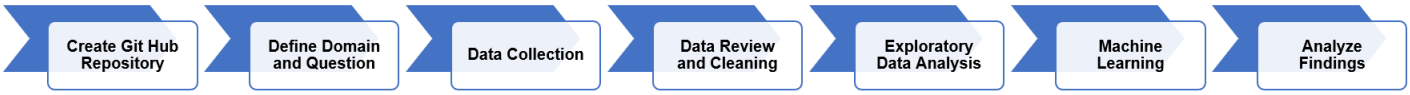


Fig. 1: Project Steps Outline. L.Wood 2023

2.1 Project Plan Steps Defined

A more detailed definition of the project steps is listed below.

1. **Create GitHub Repository**: This will be the centralized location for all parts of the project. It will store items such as code, visualizations, and the dataset.
2. **Define Domain and Question**: This will be the overall focus of the project.
3. **Data Collection**: This is where the dataset is located to analyze and answer the question.
4. **Data Review and Cleaning**: This is where the initial scrub of the data occurs.
5. **Exploratory Data Analysis**: This is where features will be analyzed and identify any correlations.
6. **Machine Learning**: This is where modeling techniques will be implemented such as regression analysis.
7. **Analyze Findings**: This is where all the machine-learning exercise results will be reviewed.

3 Dataset

The dataset for this project was originally gathered from the Broadway League. It is a seemingly structured dataset containing 31,296 rows across 12 columns or attributes. There appears to be a good mix of categorical and numeric data. The dataset that was found for this project is not real-time and was already located in a CSV format. Since the data is in a CSV file, it will be uploaded using batch processing to handle the information.

3.1 Attribute Descriptions

Four of the attributes were related to dates and three of those attributes were numeric. These numeric dates break the week into day, month, and year. The month attribute was using the numeric number to represent the month (1 = January, 2 = February, etc.) Other numeric attributes included attendance, capacity, gross, gross potential, and performances. Capacity and gross potential are percentages and will need to be reviewed. Under capacity, there appear to be some data integrity issues where capacity percentages are well over 100%. Under gross potential, there are zeros due to a lack of information available. The non-numeric attributes include show name, show theatre, show type, and date full.

Each of the attributes refers to the ending week of each performance and will be described as the "performance week" throughout. The descriptions of the attributes are listed below. [2]

- **Date Day** This is the day of the performance week.
- **Date Full**: This is the performance week of the show listed in the entire date format.
- **Date Month**: This is the month of the performance week listed in a numeric equivalent (1 = January, 2 = February, etc.)
- **Date Year**: This is the year of the performance week.
- **Show Name**: The name of the show.
- **Show Theatre**: The name of the theatre where the show was held.
- **Show Type**: The type of show that was performed. It could be a Musical, Play, or Special.
- **Statistics Attendance**: This number indicates the total number of people in attendance during the performance week.
- **Statistics Capacity**: This number indicates how full the theatre was during the performance week. Capacity is measured as a percentage.
- **Statistics Gross**: This number indicates the total gross sales of the performance week. Total gross sales are measured in dollars for this project. This is also the variable to be predicted in the project.
- **Statistics Gross Potential**: This is the max potential that a show can gross during the performance week. This is a percentage comparing Statistics Gross/Statistics Gross Potential.
- **Statistics Performances**: This is the number of shows that were held during the performance week.

3.2 Data Cleaning

Data cleaning is essential in any data analysis. If poor information is located in the dataset and then analyzed, the outcome could result in poor decision-making. Poor information could be anything from missing data, duplicates, or outliers. All of these anomalies can skew results. If there is missing data, decisions will need to be made to determine the best way to handle it. One option would be removing columns or rows that have missing values over a specific threshold. For example, if the percentage of missing values in a column is 80%, it might be best to remove the entire column. Another way would be to replace the missing values with the mean or median of that specific attribute.

For this project, data cleaning occurred with two different tools. Excel was used to prepare the non-numeric data for analysis. It was used to analyze the non-numeric attributes which include date full, show name, show theatre, and show type. These non-numeric attributes were reviewed to determine if there was any significance to the gross. Python was used to analyze the numeric attributes which include date day, date month, date year, statistics attendance, statistics capacity, statistics gross, statistics gross potential, and statistics performances. A name clean-up was used in both tools. The new names are shown in Table 1.

Table 1: Attribute Names

Original Name	New Name	Numeric vs Non-Numeric
Date Day	Day	Numeric
Date Full	Full_Date	Non-Numeric
Date Month	Month	Numeric
Date Year	Year	Numeric
Show Name	Name	Non-Numeric
Show Theatre	Theatre	Non-Numeric
Show Type	Type	Non-Numeric
Statistics Attendance	Attendance	Numeric
Statistics Capacity	Capacity	Numeric
Statistics Gross	Gross	Numeric
Statistics Gross Potential	Gross_Potential	Numeric
Statistics Performances	Performances	Numeric

3.3 Data Cleaning using Python

To clean the numeric attributes in the dataset, Python was used in a Jupyter Lab notebook. Matplotlib.pyplot, re, and pandas were imported to assist in the data-cleaning process.

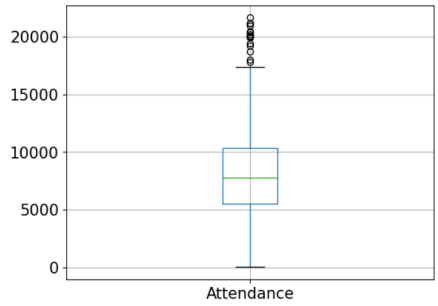
To begin the data-cleaning process the original_broadway.csv used pandas to read the dataset into a data frame. Once the dataset was read, the next step in the data-cleaning process was to address any null, missing, or duplicate values.

It was determined that the dataset did not include any of these anomalies. To cleanup the column headers, the headers were split on a delimiter and the Gross Potential was renamed to Gross_Potential due to the space in the name. To narrow the data frame down to numeric attributes only, the non-numeric attributes were dropped. This narrowed the dataset to 8 columns and 31,296 rows.

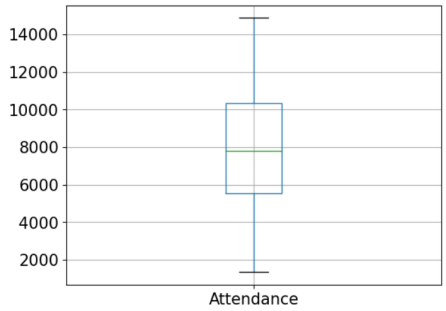
The next step in the data-cleaning process was to address any outliers. Each of the eight numeric attributes was reviewed for outliers. Attendance, capacity, gross, and gross_potential were the only attributes where outliers appeared. A calculation using a quantile of 0.995 and 0.005 were used to remove the outliers. Once the outliers were removed, the numeric attributes remaining consisted of 8 columns (including Gross) and 29,010 rows. Box plots showing the before and after the outliers were removed are shown in Figure 2.

3.4 Data Cleaning using Excel

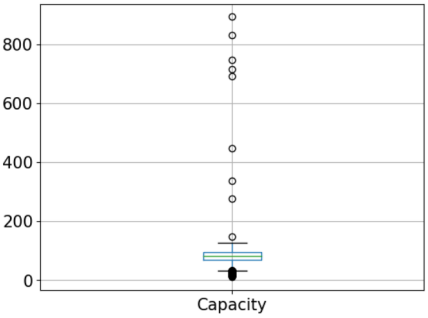
The data cleaning that occurred in Python already determined that there were no missing values or duplicates. In Excel, a new copy of the broadway_original.csv file was saved and renamed as broadway_cleaned.xlxs. A power query was built to prepare the non-numeric attributes for analysis. The query dropped all numeric columns except for gross. The only non-numeric that was dropped was the Name attribute. The reason is that the show's name would be unknown when predicting the effect on gross sales in advance. This kept the columns to 3 (not counting Gross) and the number of rows unchanged at 31,296. The query was then duplicated three times, one query for each non-numeric attribute (Full_Date, Theatre, Type). Both Theatre and Type queries were grouped and transposed so the Theatre and Type instances are now the column headers and the gross attributes are now the instances. Figure 3 shows an example of the new layout.



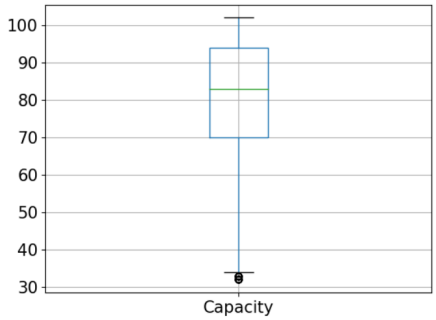
(a) Boxplot of attendance before outliers were removed



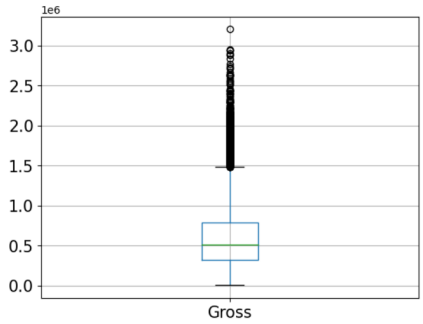
(b) Box plot of attendance after outliers were removed.



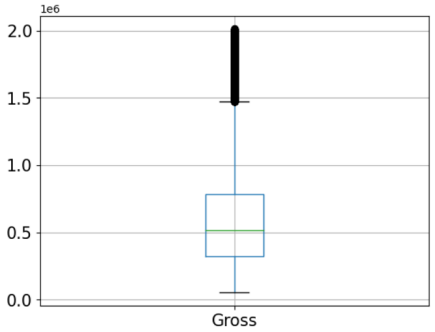
(c) Box plot of capacity before outliers were removed.



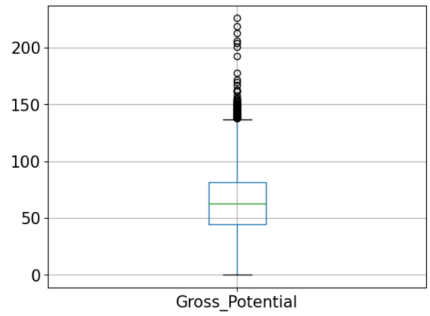
(d) Box plot of capacity after outliers were removed.



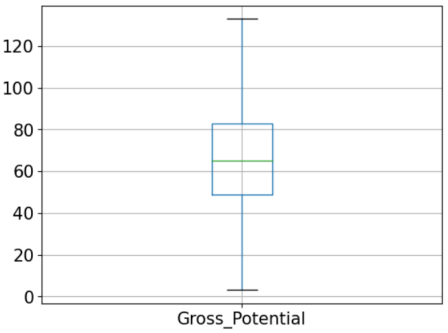
(e) Box plot of gross sales before outliers were removed.



(f) Box plot of gross sales after outliers were removed.



(g) Box plot of gross_potential before outliers were removed.



(h) Box plot of gross_potential after outliers were removed.

Fig. 2: Box plots of numeric attributes before and after outliers were removed. L.Wood 2023

Play	Musical	Special
268910		19830
	214109	22074
245815		22427
165668		23528
	67578	23853
259924		24480
276314		24480
294749		24480
		24512
263564		25117
264171		25558

(a) Type layout

	Booth	Broadway	Ethel Barrymore
1		597026	
2	137387		175704
3		581270	
4	144713		182854
5		608810	
6	128673		199905
7		603264	
8	134999		179420
9		550033	
10	110708		140373
11		573191	
12	134698		154988
13		554559	
14	103884		118313
15		614889	
16	155897		153316
17		552597	
18			155138

(b) Theatre layout

Fig. 3: Cleaned Layout Examples (Type & Theatre). L.Wood 2023

The third non-numeric attribute, Full_Date, was cleaned using Power Query as well. To begin the cleanup, the Full_Date column was duplicated. One column was converted into quarters and renamed to Quarter. The Quarters range from 1-4 and are broken up into 3-month increments. A description of the Quarters can be found in Table 2. The duplicated Full_Date column was narrowed down to the year and renamed as Year. The Full_Date query was grouped and transposed so the Quarter and Year instances are now the column headers and the gross attributes are now the instances. Figure 4 shows an example of the new layout. After the initial data-cleaning process, the dependent variable for this project is Gross. This will leave the independent variables as Day, Full_Date, Month, Year, Theatre, Type, Attendance, Capacity, Gross_Potential, and Performances.

Table 2: Quarter Descriptions

Quarter	Month
1	January, February, March
2	April, May, June
3	July, August, September
4	October, November, December

Sum of Gross	Quarter				
Year	1	2	3	4	Grand Total
1991	735071	9604320	10172809	10153473	30665673
1992	9682543	9136607	8881083	8615002	36315235
1993	7668634	8341136	8084173	8195774	32289717
1994	8401986	16199154	17059384	34576760	76237284
1995	38167388	44518931	42344522	46612398	171643239
1996	23497373	66237053	98080199	105661671	293476296

Fig. 4: Cleaned Layout Examples (Quarter & Year). L.Wood 2023

4 Exploratory Data Analysis

Exploratory data analysis, or EDA, is a critical part of data analysis. In this step, a deeper review will occur to better understand the dataset. The EDA process will help gain insights into any patterns, outliers, and relationships using visualizations and statistics. If EDA is not thoroughly accomplished, then it could impact future modeling accuracy and lead to poor decisions.

For this project, there were various EDA techniques used to prepare the data. To begin the EDA process, the variables need to be understood and the data cleaned. Data cleaning steps include looking for any null, missing, or duplicate values. Also in the data cleaning step, any outliers will need to be discovered and removed. Once this has been completed, univariate, bivariate, and multivariate visualizations can be used to start the analysis. Correlations will be used to help identify relationships between variables.

4.1 Exploratory Data Analysis using Python

Data Cleaning is a necessary step when analyzing data. When initially reviewing the data, it was discovered that there were no missing, null, or duplicate values. For the numeric data, box plots were created for each of the variables. Outliers were identified and a quantile calculation of 0.995 and 0.005 was used to remove the outliers. Please see Figure 2 for a visual of the variables that had outliers. The figure reflects the box plots before and after the outliers were removed.

Once the outliers were removed, a correlation heat map was used to analyze the relationships between the variables. Since the goal is to predict gross sales (dependent variable), the focus was to identify which variables were correlated to gross. When compared to the gross, both day and month have little to no correlation at 0.026 and 0.021. Performances, year, and capacity have minimal to moderate correlation. The gross_potential and attendance have the strongest correlation at 0.72 and 0.81. Both day and month were dropped since there was little to no correlation. See Figure 5 to see the correlation heat map.

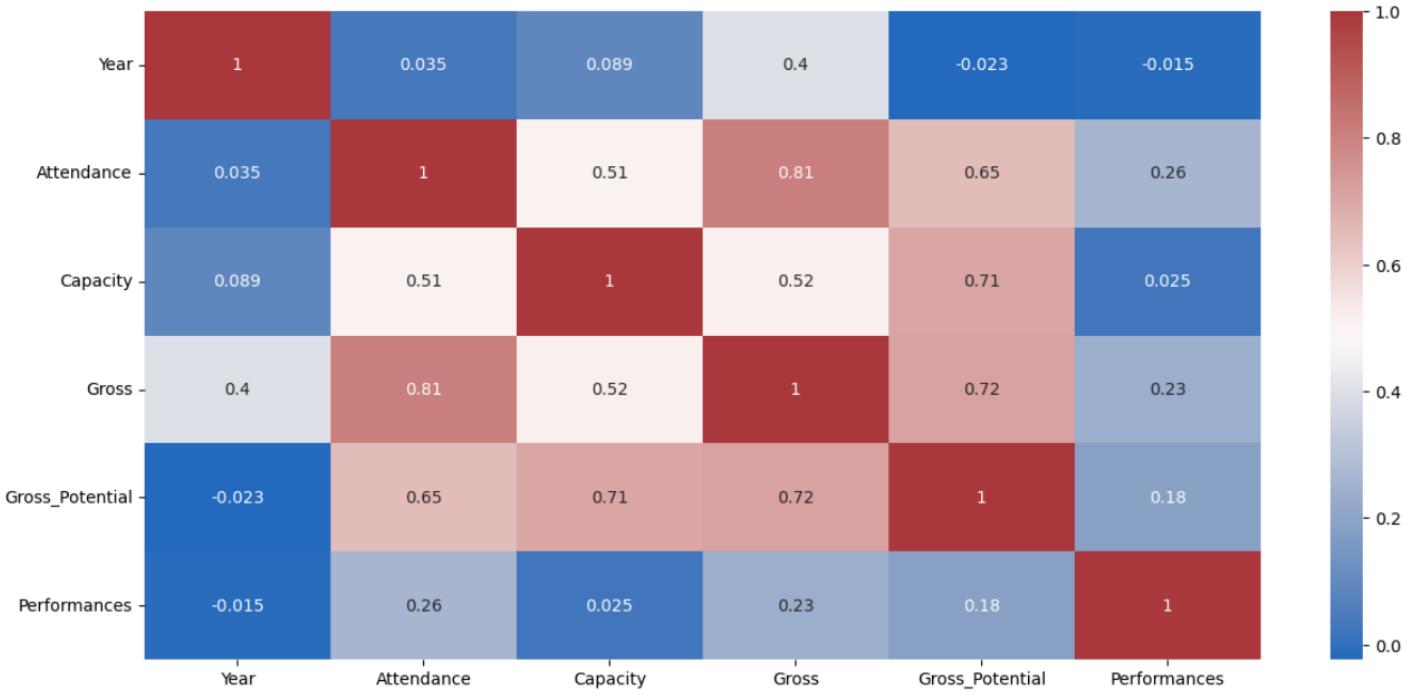
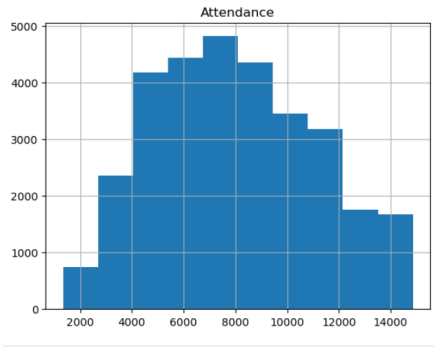
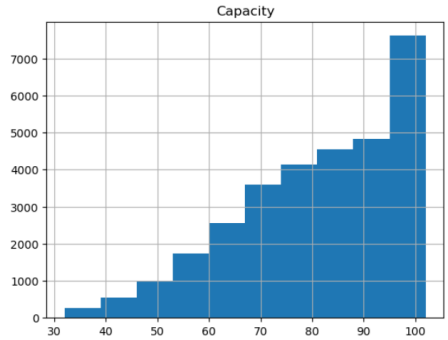


Fig. 5: Heat map correlation matrix for the numeric variables. The strongest correlated attributes to gross were year, attendance, capacity, gross_potential, and performances. L.Wood 2023

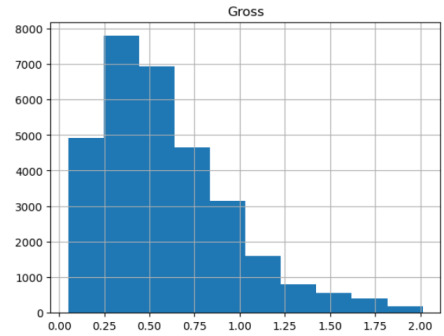
Histograms of the remaining variables can be found in Figure **??**. Attendance and gross_potential have a normal distribution. Gross is right-skewed whereas capacity is left-skewed. Performances is a uni-modal distribution with the main peak around 8-9 performances. The year is multi-modal distribution, with peaks around 1998-2000, 2003-2007, and 2011-2016.



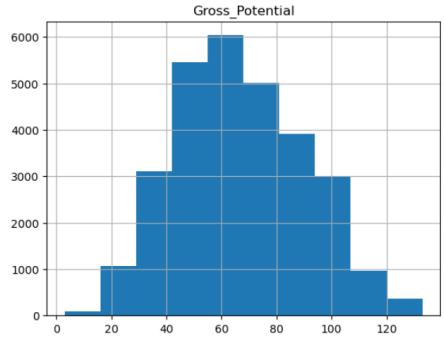
(a) Attendance histogram showing a normal distribution.



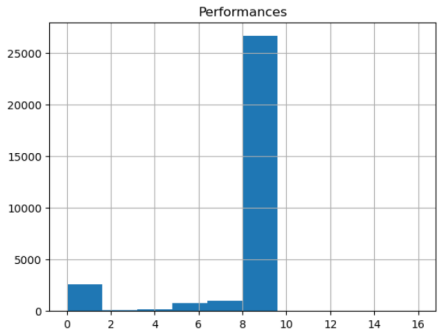
(b) Capacity histogram showing a left-skewed distribution.



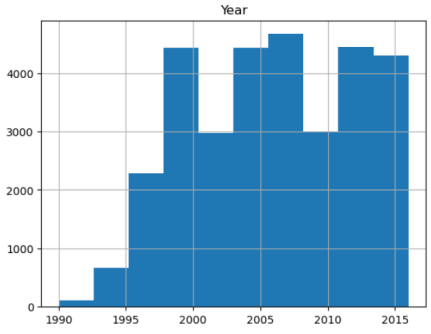
(c) Gross sales histogram showing a right-skewed distribution.



(d) Gross_Potential histogram showing a normal distribution



(e) Performances histogram showing a unimodal distribution.



(f) Histogram of Year showing a multimodal distribution with peaks around 1998-2000, 2003-2007, & 2011-2016.

Fig. 6: Histograms of the remaining numeric attributes. L.Wood 2023

A closer look was done to compare the years to the number of attendance. The reason was to see if there are any peaks over the years or if there has been a steady increase. The results of the comparison are in Figure 7. 2016 can be explained since the dataset did not contain all of 2016. There is a significant increase from the years 1996-1998. After 1998, attendance appears to stay relatively consistent.

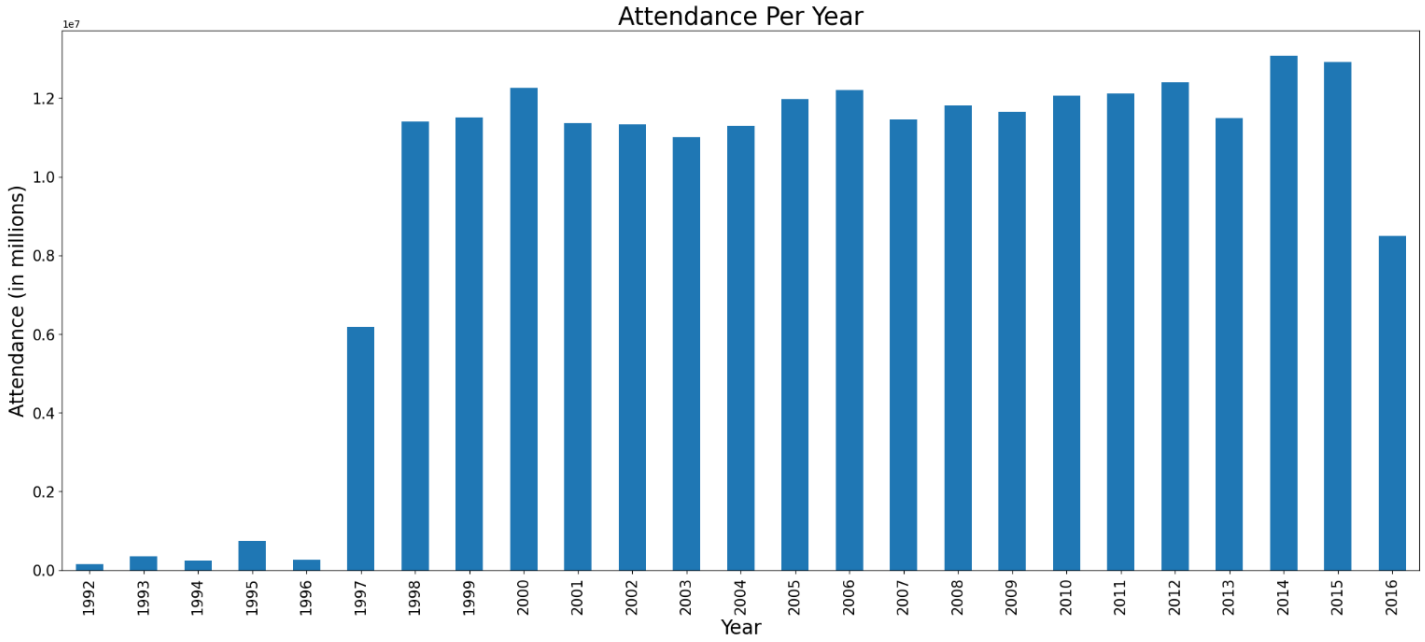


Fig. 7: Attendance per Year. L.Wood 2023

4.2 Python Codes

A variety of Python codes were used in the cleaning and exploratory data analysis of the numeric variables. The following libraries were imported to assist with the cleaning and visualizations.

- pandas
- numpy
- matplotlib.pyplot
- seaborn
- scipy
- re

To begin the initial data-cleaning process, the dataset needed to be reviewed for any missing, null, or duplicated values. The codes used for this process can be found in Listing 1.1. These codes revealed that the dataset had no missing, null, or duplicated values.

```
broadway = pd.read_csv("original_broadway.csv") #reading in
    the file

broadway.info() #checking for blanks

broadway.isnull().any() #checking for null values

duplicates = broadway.duplicated() #checking for duplicates
sum(duplicates)
```

Listing 1.1: Python Initial Cleaning Codes. L.Wood 2023

To prepare the data even more, the column headers were split to simplify the names of each attribute. One of the columns, Gross Potential, was renamed to address the space between the words. To confirm that the split was done completely and the column renamed correctly, .info() was used. The next step was to drop any of the non-numeric columns and to confirm how many columns and rows were remaining in the dataset. The specific codes used for this step in the cleaning process can be found in Listing 1.2.

```
import re
broadway.columns=[re.split(r'[.]', col)[1] for col in
    broadway.columns] #splitting the column headers

broadway.rename(columns={"Gross Potential": "Gross_Potential"
    }, inplace=True) #renaming the column Gross Potential to
    Gross_Potential

broadway.info() #confirming the column headers were split and
     the column header renamed. This also shows which columns
     are numeric vs non-numeric

broadway.drop(columns=['Full','Name', 'Theatre', 'Type'],
    inplace=True) #dropping non-numeric attributes

broadway.shape #confirming the number of columns and rows
    remaining in the dataset
```

Listing 1.2: Python Additional Cleaning Codes. L.Wood 2023

To better understand the dataset and to identify any outliers, both box plots and histograms were created. These were created for each of the numeric attributes. If outliers were identified, they were removed using a quantile calculation of 0.995 and 0.005. Once the outliers were removed, if applicable, then another box plot and histogram were created to see the change. Examples of the code used can be found in Listing 1.3.

```
#Example of box plot before outliers were removed
attendance = broadway.boxplot(column='Attendance')
attendance.plot()
plt.show()
```

```
5
6 #Example of histogram before outliers were removed
7 broadway.hist('Attendance')
8 plt.show()
9
10 #Example of code to remove outliers using quantile of 0.995
    and 0.005
11 broadway_outliers = broadway[(broadway.Attendance < broadway.
    Attendance.quantile(.995)) & (broadway.Attendance >
    broadway.Attendance.quantile(.005))]
12
13 #Example of box plot after outliers were removed
14 attendance_out = broadway_outliers.boxplot(column = '
    Attendance')
15 attendance_out.plot()
16 plt.show()
17
18 #Example of histogram after outliers were removed
19 broadway_outliers.hist('Attendance')
20 plt.show()
```

Listing 1.3: Python Box Plot, Histogram, & Outlier Example Codes. L.Wood 2023

The next step was to identify any relationships. This was done using a heat map correlation matrix and a bar chart. The attributes with low to no correlation to the gross were dropped. A closer look into the relationship between year and attendance was done by creating a bar chart. This was to see how attendance was doing over the years. The codes for the relationship analysis can be found in Listing 1.4.

```
1 broadway_outliers.shape #provide the number of columns and
    rows after outliers were removed
2
3 round(broadway_outliers.corr(),2) #correlation of the
    outliers with numbers rounded to 2 decimal places
4
5 #Creating heat map correlation matrix
6 cmap = sns.color_palette("vlag", as_cmap=True)
7 correlation = broadway_outliers.corr(numeric_only=True)
8 sns.heatmap(correlation, xticklabels=correlation.columns,
    yticklabels=correlation.columns, annot=True, cmap=cmap)
9 plt.show()
10
11 broadway_outliers.drop(columns=['Day','Month'], inplace=True)
     #removing attributes that had minimal to no correlation
    to gross
12
13 broadway_outliers.info() #confirming the attributes were
    dropped
14
```

```
broadway_outliers.shape #showing the number of columns and
    rows remaining after the drop

#Histogram showing relationship between Year and Attendance
broadway_outliers.groupby('Year')['Attendance'].sum().plot(
    kind='bar', figsize=(25, 10), rot=45)
plt.xlabel("Year",fontsize=15)
plt.ylabel("Attendance (in millions)", fontsize=15)
plt.title("Attendance Per Year", fontsize=20)
plt.show()
```

Listing 1.4: Python Relationship Codes. L.Wood 2023

The complete Jupyter Notebook with Python codes and results are located on GitHub at https://github.com/lwood7983/L.Wood_Final_Capstone_Project.git

4.3 Exploratory Data Analysis using Excel

Excel was used to perform EDA on the non-numeric variables. Data cleaning in Python already indicated there were no missing, null, or duplicated values. The data was then prepared using Power Queries. The numeric columns were dropped except for the dependent variable, gross. The Name attribute was also dropped since a show's name would be unknown when predicting the effect to gross. Once the dataset was narrowed to the three non-numeric columns, not including gross, the query was then duplicated three times. Each query was used to prepare the data for the ANOVA analysis. Both Theatre and Type queries were grouped and transposed so the dataset can be used in the analysis. Please see Figure 3 for the new layout.

The Full_Date attribute was converted into Quarter and Year. The quarters are represented as numbers ranging from 1-4, each number consisting of three months. Please see Table 2 for a description of the quarters. The Full_Date query was grouped and transposed to prepare Quarter and Year for ANOVA analysis. See Figure 4 for the layout example.

4.3.1 Relationship Analysis

The first non-numeric relationship analyzed was comparing type (independent variable) to gross sales (dependent variable). An ANOVA: single-factor analysis was done. The P-value was less than alpha (0.05) which means that type is statistically significant. One thing that did stand out was how much higher the average gross sales were for musicals when compared to plays and specials. Both plays and specials had approximately the same average gross. The ANOVA results for the type comparison to gross can be found in Figure 8.

Anova: Single Factor	Type Comparison					
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Play	8406	2737985361	325717.983	5.14E+10		
Musical	22551	1.5793E+10	700312.185	1.4096E+11		
Special	339	123844139	365321.944	1.6034E+11		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	8.775E+14	2	4.3875E+14	3746.34522	0	2.99602
Within Groups	3.6648E+15	31293	1.1711E+11			
Total	4.5423E+15	31295				

Fig. 8: ANOVA Analysis: Type Comparison. Musicals have significantly higher average gross sales compared to plays and specials. L.Wood 2023

A deeper look into type was done to compare the plays and specials. A t-Test: Two-Sample Assuming Unequal Variances was performed. The P-value was greater than alpha (0.05) at 0.07 which indicates that when compared to each other, plays and specials are not statistically significant. The results of the t-Test can be found in Figure 9. Based on these findings, the only thing that changes the average gross is musical vs non-musical.

	Play & Special t-Test	
t-Test: Two-Sample Assuming Unequal Variances		
	Play	*Special*
Mean	325717.983	365321.944
Variance	5.14E+10	1.6034E+11
Observations	8406	339
Hypothesized Mean Difference	0	
df	347	
t Stat	-1.80939099	
P(T<=t) one-tail	0.03562774	
t Critical one-tail	1.64925671	
P(T<=t) two-tail	0.07125549	
t Critical two-tail	1.966824	

Fig. 9: t-Test: Type Plays compared to Specials. The p-value is 0.07 is greater than alpha (0.05) meaning that plays and specials are not statistically significant when compared to each other. L.Wood 2023

The next non-numeric relationship that was analyzed was theatre. An ANOVA: single-factor analysis was performed. When looking at theatre, the p-value is less than alpha (0.05) which shows that this feature is statistically significant. It is showing that the theatre does matter when looking at gross. Looking at all of the averages, there is a minimal chance that the average mean would be the same across theatres. Figure 10 is a bar chart showing the average gross sales per theatre.

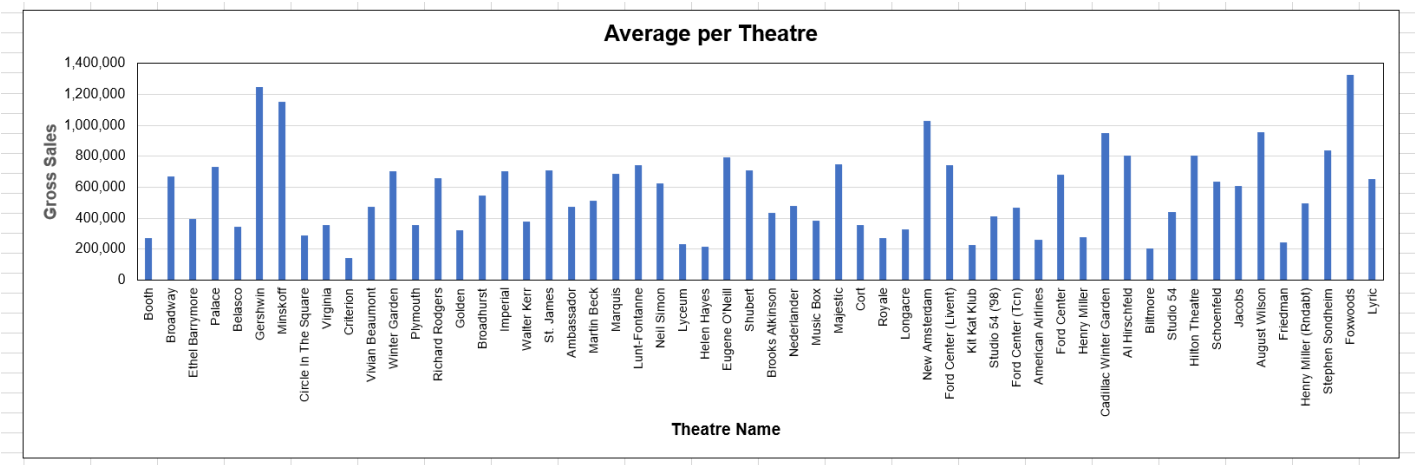


Fig. 10: Average Gross per Theatre. L.Wood 2023

The third relationship to be reviewed was comparing the quarter and type. This was to see if there were any trends based on the quarter to identify possible seasonality. A line chart was built and indicated peaks in quarter 2. Figure11 shows the line chart.

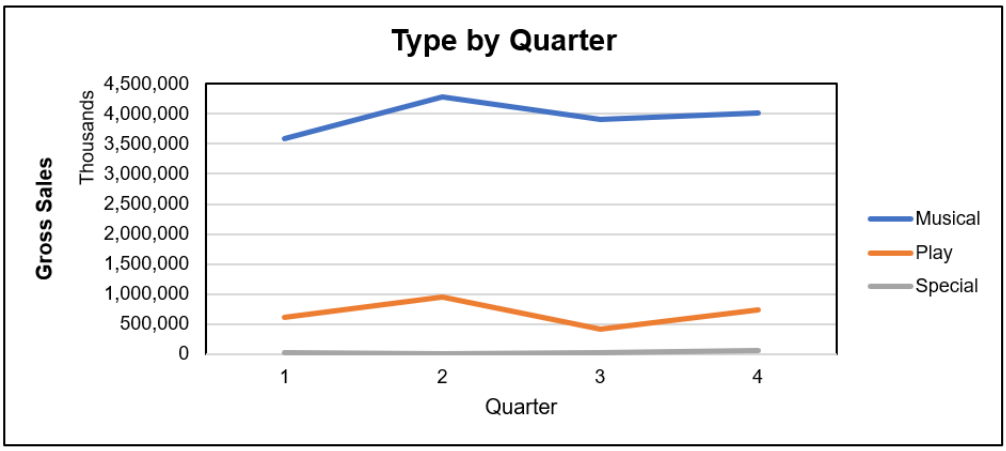


Fig. 11: Type by Quarter indicating that quarter 2 shows a peak in gross sales. L.Wood 2023

The last non-numeric relationship was comparing the quarter and year. A line graph was built to look for any trends over the years per quarter. The graph in Figure 12 indicated that there was an incline in gross sales per year. However, between 1996 and 1997 there was a significant jump in gross. This seemed to line up with what was discovered in the bar chart found in Figure 7. There was also a spike in attendance over the same years.

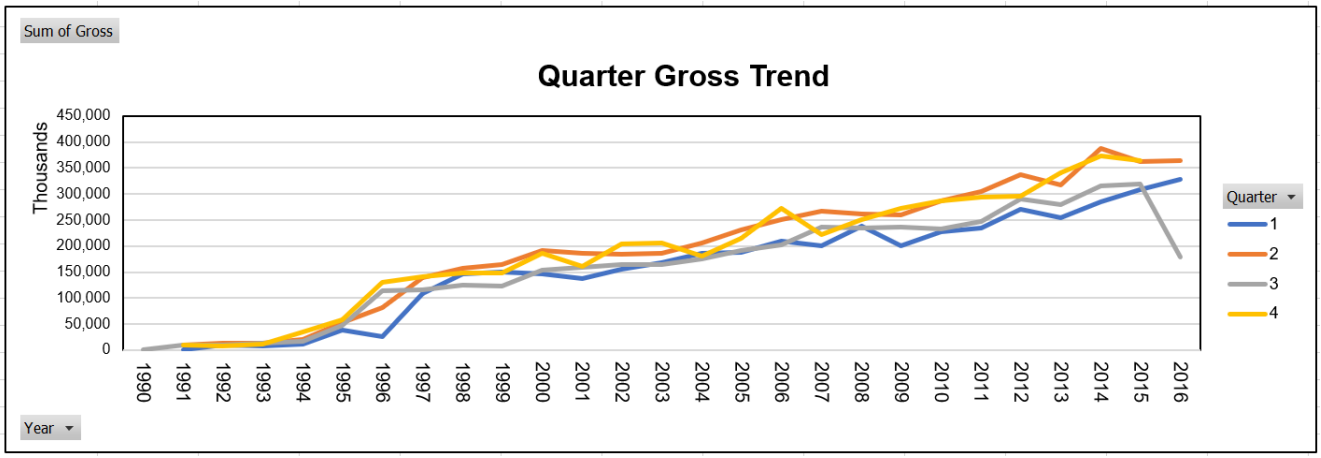


Fig. 12: Quarter Gross Trend shows that sales have increased over the years in all quarters. There was a jump in gross sales between 1996-1997. L.wood 2023

Some online searching discovered that there was a push for change in Times Square. A few key events attribute to the massive increase in gross. “The Unexpected Lessons of Times Square’s Comeback” goes into detail about the increasing push for the war on crime around 1993. [9] The article also talks about the 42nd Street Redevelopment Plan and the removal of seedy businesses.[9]. All of these changes would increase tourism, investments, and jobs contributing to the large influx of gross sales between 1996-1997.

The complete Excel workbook that includes the Power Queries and Excel visualizations is on GitHub under broadway_clean.xlxs The link to the GitHub repository is https://github.com/lwood7983/L.Wood_Final_Capstone_Project.git.

5 Machine Learning

The purpose was to determine what factors make a successful Broadway show. In order to make that determination, the goal was to not only be able to predict gross sales but to also understand what factors impact gross. For this project, a pipeline will be used for the predictive analysis. The steps used in the pipeline are shown in Figure 13.

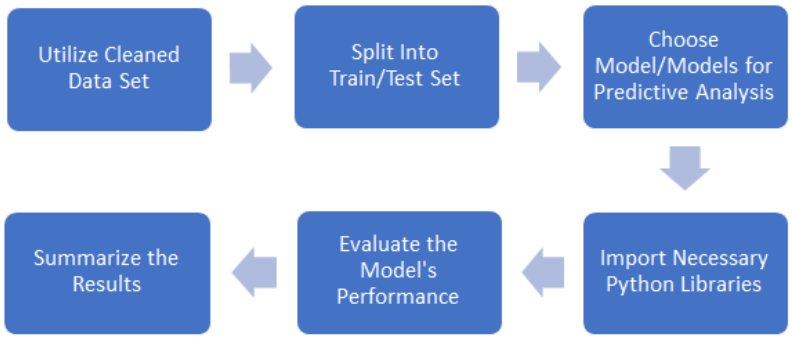


Fig. 13: Pipeline for Predictive Modeling. L.Wood 2023

The steps are explained in better detail below.

1. **Utilized Cleaned Dataset**: The data cleaning (section 3.2) and exploratory data analysis (section 4) processes have already been explained in detail. However, to recap, the dataset was relatively clean with no missing, null, or duplicated values. The data was split between numeric and non-numeric variables. The non-numeric variables were further cleaned and analyzed in Excel. The numeric variables were further cleaned in Python to remove outliers and non-correlated variables.

2. **Split Into Train/Test Set**: The numeric data will be split into a training set and a test set for the predictive analysis.
3. **Choose Model/Models for Predictive Analysis**: For the numeric variables, linear regression models will be used in Python. For the non-numeric variables, a regression model will be used in Excel.
4. **Import Necessary Python Libraries**: To complete linear regression modeling, additional libraries will need to be imported into the Jupyter Notebook. This project will be using tools from scikit-learn.
5. **Evaluate the Model's Performance**: Each of the models that are used, will also be run against the test set to evaluate the performance. This will help to determine if the model is a good fit for predicting gross.
6. **Summarize the Results**: Once all of the training and testing has been completed, a review of the results from each model will be analyzed.

5.1 Modeling in Python

To begin running any of the models, the scikit-learn library was imported to split our numeric dataset into a training set (80%) and a test set (20%). This left the training set size at 23,208 and the test size at 5,802. The code used for the split can be seen in Figure

```
import sklearn as sklearn
from sklearn.model_selection import train_test_split

train_set, test_set = train_test_split(broadway_outliers,
                      test_size=0.2, random_state=123)
print('Train size: ', len(train_set), 'Test size: ', len(test_set))
```

Fig. 14: Code for Training and Test Split. L.Wood 2023

All codes for the Python modeling can be found in a Jupyter Notebook named Capstone_lwood.ipynb at the following GitHub link: https://github.com/lwood7983/L.Wood_Final_Capstone_Project.git

5.1.1 Linear Regression Analysis

The first machine learning model used was linear regression. This was performed on each of the numeric variables. Since the goal is to predict gross, this analysis was used to identify if there was a key variable that stood out. An example of the code used for the Linear Regression can be found in Listing 1.5

1

```
#Additional modules imported for Linear Regression and for
    model analysis.

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

#This linear regression is for Attendance
X = train_set[['Attendance']]
y = train_set['Gross']

X_test = test_set[['Attendance']]
y_test = test_set['Gross']

lr_model = LinearRegression()
lr_model.fit(X,y)

y_pred = lr_model.predict(X)
print('Results for linear regression on training data using
    Attendance')
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', lr_model.intercept_)
print('    Coefficients', lr_model.coef_)
print('    Score', lr_model.score(X,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = lr_model.predict(X_test)
print()
print('Results for linear regression on test data using
    Attendance')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test,
    y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))
```

Listing 1.5: Linear Regression Code Example. L.Wood 2023

The results of the RMSE (root mean square error) and R^2 (coefficient of determination) for the linear regression models are shown in Figure 15. The RMSE numbers at first glance appear high. However, when considering that the gross is in the millions, an RMSE in the thousands is not that high. When looking at the R^2, the attendance and gross_potential were the highest at 0.65

and 0.53. The least significant variable when looking at R^2 was the performances at 0.05. Both the training and test sets across the individual variables performed about the same.

Model	Training Feature	Set	RMSE	R2
Linear Regression	Year	Training	348,993.43	0.16
Linear Regression	Year	Test	353,875.39	0.17
Linear Regression	Attendance	Training	223,798.14	0.65
Linear Regression	Attendance	Test	228,260.93	0.65
Linear Regression	Capacity	Training	325,847.39	0.26
Linear Regression	Capacity	Test	329,217.60	0.28
Linear Regression	Gross_Potential	Training	264,288.28	0.52
Linear Regression	Gross_Potential	Test	267,398.80	0.53
Linear Regression	Performances	Training	369,642.32	0.05
Linear Regression	Performances	Test	377,287.89	0.05

Fig. 15: Linear Regression Results (RMSE & R^2) L.Wood 2023

5.1.2 Multiple Regression Analysis

The next machine learning model used to predict gross sales was multiple regression. The year, attendance, capacity, gross_potential, and performances, Independent Variables, were used for the prediction. The code used to perform the prediction can be seen in Listing 1.6.

```
1
2 #No additional modules needed to be imported
3
4 #This multiple regression using for the Independent Variables
      to predict gross
5 X = train_set[['Year','Attendance', 'Capacity','
    Gross_Potential','Performances']]
6 y = train_set['Gross']
7
8 X_test = test_set[['Year', 'Attendance','Capacity','
    Gross_Potential','Performances']]
9 y_test = test_set['Gross']
10
11 lr_model = LinearRegression()
12 lr_model.fit(X,y)
13
14 y_pred = lr_model.predict(X)
15 print('Results for multiple regression on training data')
16 print('Input: Year, Attendance, Capacity, Gross_Potential,
    Performances')
```

```
print('  Default settings')
print('Internal parameters:')
print('   Bias is ', lr_model.intercept_)
print('   Coefficients', lr_model.coef_)
print('   Score', lr_model.score(X,y))

print('MAE is  ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2    ', r2_score(y,y_pred))

y_test_pred = lr_model.predict(X_test)
print()
print('Results for multiple regression on test data')
print('Input: Year, Attendance,Capacity, Gross_Potential,
    Performances')
print('MAE is  ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test,
    y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test,y_test_pred))
```

Listing 1.6: Multiple Regression Code. L.Wood 2023

The results of the RMSE and R^2 for the multiple regression model can be seen in Figure 16. The RMSE numbers for both the test and training set were lower than any of the linear regression models. This is showing that there was an improvement in the spread of the prediction errors. Prediction errors are how far the actual values were from the predicted values. When looking at the R^2, the multiple regression model showed an improvement at 0.88 when compared to any of the linear regression models. The training and test performed the same in the multiple regression model.

Model	Training Feature	Set	RMSE	R2
Multiple Regression	Independent Variables	Training	133,737.51	0.88
Multiple Regression	Independent Variables	Test	133,737.10	0.88

Fig. 16: Multiple Regression Results (RMSE & R^2) L.Wood 2023

5.1.3 Polynomial Regression Analysis

The next model used to predict gross sales was polynomial regression. Since the majority of the independent variables had a non-linear relationship to gross, this model would help capture the relationships on a non-linear line. For this

model, the independent variables were used. The first polynomial regression used a degree power of 2, also known as a quadratic polynomial. To see if a different degree would impact the fit of the model, the polynomial regression using a degree power of 3, also known as a cubic polynomial, was also run. The polynomial regression code example is shown in Listing 1.7.

```

#Polynomial Regression using all Independent variables
#To run polynomial regression an additional module needed to
    be imported

from sklearn.preprocessing import PolynomialFeatures
power = 2 #degree power
poly_process = PolynomialFeatures(degree=power, include_bias=
    False)

X = train_set[['Year','Attendance','Capacity','
    Gross_Potential','Performances']]
y = train_set['Gross']
X_poly = poly_process.fit_transform(X)

X_test = test_set[['Year','Attendance', 'Capacity','
    Gross_Potential','Performances']]
y_test = test_set['Gross']
X_poly_test = poly_process.fit_transform(X_test)

lr_model = LinearRegression()
lr_model.fit(X_poly,y)

y_pred = lr_model.predict(X_poly)
print('Results for polynomial regression on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', lr_model.intercept_)
print('    Coefficients', lr_model.coef_)
print('    Score', lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = lr_model.predict(X_poly_test)
print()
print('Results for polynomial regression on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test,
    y_test_pred)))
```

```
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))y_test_pred))
print('R^2    ', r2_score(y_test,y_test_pred))
```

Listing 1.7: Polynomial Regression Example (degree 2) L.Wood 2023

The results of the RMSE and R^2 for the polynomial regression models are shown in Figure 17. When comparing the ability to predict gross sales between the quadratic polynomial and cubic polynomial, there were minor changes in the performance of the two models. There was a small improvement with degree 2 and some overfitting showing in degree 3. The slight overfitting could indicate that the cubic model was starting to perform too well on the training set and not perform as well on the test set.

The RMSE numbers on the quadratic and cubic polynomials were lower than the linear and multiple regression models for the training and test sets. This would mean that both of the polynomial models had a lower spread of prediction errors. When looking at the R^2, the quadratic polynomial showed an improvement to 0.96 whereas the cubic polynomial remained the same. The training and test sets performed relatively the same in the polynomial regression models regardless of the change in the degrees. No additional polynomial models with higher degrees were performed since there was minimal improvement between degree 2 and degree 3.

Model	Training Feature	Set	RMSE	R2
Polynomial Regression, degree 2	Independent Variables	Training	81,503.77	0.95
Polynomial Regression, degree 2	Independent Variables	Test	81,724.57	0.96
Polynomial Regression, degree 3	Independent Variables	Training	78,111.96	0.96
Polynomial Regression, degree 3	Independent Variables	Test	80,014.38	0.96

Fig. 17: Polynomial Regression Results (RMSE & R^2) L.Wood 2023

5.1.4 Elastic Net Regression Analysis

The last model used to predict gross sales was an elastic net regression. An elastic net regression is a regularized model that helps prevent over-fitting. This model is a combination of a lasso regression, which reduces the coefficients of variables that are not important, and a ridge regression, which keeps the coefficients of the variables small. It primarily helps keep the turns from the higher degrees from getting big. The elastic net regression used the independent variables to predict gross. The first elastic net regression used a degree power of 3. To see if a different degree would impact how well the model predicts gross. the elastic net regression using a degree power of 8 was also run. The elastic net regression code example is shown in Listing 1.8.

```
1
2 #Elastic Net Regression using all Independent variables
3 #To run the elastic net regression, an additional module
    needed to be imported
4
5 from sklearn.linear_model import ElasticNet
6 power = 3 #degree power
7 poly_process = PolynomialFeatures(degree=power, include_bias=
    False)
8
9
10 X = train_set[['Year', 'Attendance','Capacity','
    Gross_Potential','Performances']]
11 y = train_set['Gross']
12 X_poly = poly_process.fit_transform(X)
13
14 X_test = test_set[['Year', 'Attendance','Capacity','
    Gross_Potential','Performances']]
15 y_test = test_set['Gross']
16 X_poly_test = poly_process.fit_transform(X_test)
17
18 reg_lr_model = ElasticNet(alpha=0.3, l1_ratio=0.5)
19 reg_lr_model.fit(X_poly,y)
20
21 y_pred = reg_lr_model.predict(X_poly)
22 print('Results for elastic net on training data')
23 print('Polynomial regression with degree ', power)
24 print('  Default settings')
25 print('Internal parameters:')
26 print('    Bias is ', reg_lr_model.intercept_)
27 print('    Coefficients', reg_lr_model.coef_)
28 print('    Score', reg_lr_model.score(X_poly,y))
29
30 print('MAE is ', mean_absolute_error(y, y_pred))
31 print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
32 print('MSE is ', mean_squared_error(y, y_pred))
33 print('R^2  ', r2_score(y, y_pred))
34
35 y_test_pred = reg_lr_model.predict(X_poly_test)
36 print()
37 print('Results for elastic net on test data')
38
39 print('MAE is ', mean_absolute_error(y_test, y_test_pred))
40 print('RMSE is ', np.sqrt(mean_squared_error(y_test,
    y_test_pred)))
41 print('MSE is ', mean_squared_error(y_test, y_test_pred))
42 print('R^2    ', r2_score(y_test, y_test_pred))
```

Listing 1.8: Elastic Net Regression Example (degree 3) L.Wood 2023

The results of the RMSE and R^2 for the elastic net regression models are shown in Figure 18. When comparing the performance between the elastic net regression with degree 3 to degree 8, there was some improvement, although small. The RMSE numbers for both the test and training set on both degrees were lower than the linear and multiple regression models. The RMSE numbers were higher than the quadratic polynomial model. This would put the performance of the elastic net models in the middle when it comes to the spread of the prediction errors. When looking at the R^2, the elastic net regression models did not perform as well as the quadratic polynomial, even with the improvement at degree 8. No additional elastic net models with a degree higher than 8 were performed since there was minimal improvement between degree 3 and degree 8.

Model	Training Feature	Set	RMSE	R2
Elastic Net degree 3	Independent Variables	Training	113,947.97	0.91
Elastic Net degree 3	Independent Variables	Test	114,102.60	0.91
Elastic Net degree 8	Independent Variables	Training	101,433.45	0.93
Elastic Net degree 8	Independent Variables	Test	102,996.31	0.93

Fig. 18: Elastic Net Results (RMSE & R^2) L.Wood 2023

5.1.5 Model Adjustments

When looking at the linear relationship to gross, attendance had the strongest impact to predict gross. Since attendance is a variable that is not known before a show opening, each of the models was repeated using the independent variables but without attendance. The models could use an estimated or predicted attendance, but attendance, in general, is not known in advance. The results of the model adjustments without attendance are shown in Figure 19. The entire code for the model adjustments can be found at GitHub under Capstone_lwood.ipynb. The link to the GitHub repository is https://github.com/lwood7983/L.Wood_Final_Capstone_Project.git.

Model	Training Feature	Set	RMSE	R2
Multiple Regression	No Attendance	Training	207,191.89	0.70
Multiple Regression	No Attendance	Test	209,507.29	0.71
Polynomial Regression, degree 2	No Attendance	Training	186,117.02	0.76
Polynomial Regression, degree 2	No Attendance	Test	187,424.37	0.77
Polynomial Regression, degree 3	No Attendance	Training	180,862.43	0.77
Polynomial Regression, degree 3	No Attendance	Test	187,954.33	0.77
Elastic Net degree 3	No Attendance	Training	194,734.28	0.74
Elastic Net degree 3	No Attendance	Test	195,971.85	0.74
Elastic Net degree 8	No Attendance	Training	189,585.94	0.75
Elastic Net degree 8	No Attendance	Test	191,117.50	0.76

Fig. 19: Regression Models without Attendance Variable (RMSE & R^2) L.Wood 2023

The results of the model adjustments without attendance did not perform as well as when attendance was still a variable. This was no surprise since attendance had the strongest independent linear regression. Despite the slightly higher RMSE, spread in prediction errors, and lower R^2 with attendance removed, the models still performed fairly well.

5.1.6 Python Modeling Summary

When utilizing the independent variables in the machine learning models, the quadratic polynomial performed the best at predicting gross sales. The R^2 improved from 0.95 to 0.96 between the training and test set. An R^2 of 0.96 indicated that the relationship between gross and the independent variables explains 96% of the variance in the dataset. The closer the R^2 is to 1 the better the model fit and how well it predicted gross sales. The R^2 for the quadratic polynomial was significantly higher than the multiple regression model, which was the worst-performing model at 0.71 on the test set. The elastic net models performed better than the multiple regression, but not as well as the quadratic polynomial. The elastic net regression with degree 8 only performed slightly better than with degree 3. While the cubic polynomial performed very well, degree 2 was minimally better due to the slight amount of over-fitting. A chart of the comparison can be found in Figure 20.

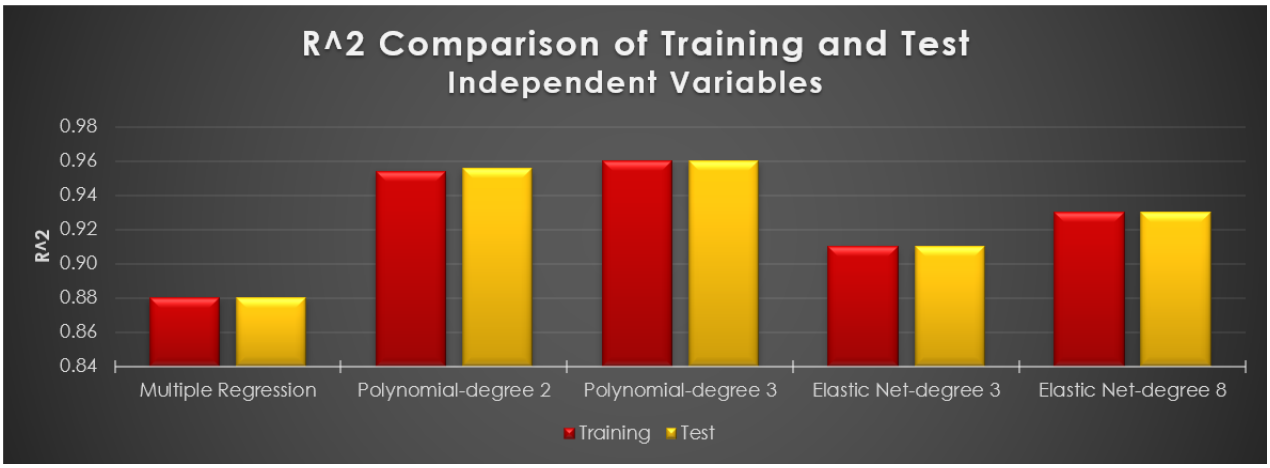


Fig. 20: R^2 Comparison (Independent Variables). L.Wood 2023

When looking at the RMSE with the independent variables, the quadratic polynomial performed the best at predicting gross. There was a very slight increase in the quadratic polynomial test set and the spread of the actual values compared to the predictions, but when looking side by side with the other models, the increase was insignificant. Multiple regression had the highest RMSE with the highest spread in prediction errors. The elastic net regression with degrees 3 and 8 was around the middle. The quadratic polynomial model performed similarly to the cubic polynomial but had a minimally higher RMSE. A chart of the comparison can be found in Figure 21.

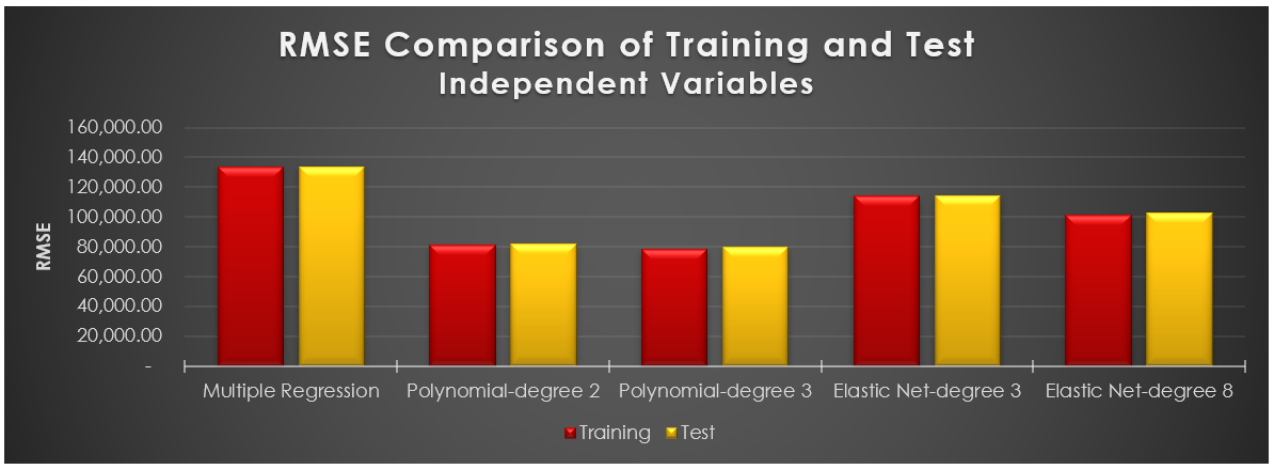


Fig. 21: RMSE Comparison (Independent Variables) L.Wood 2023

A chart comparing the R^2 on the models where attendance was removed as an independent variable can be found in Figure 22. The polynomial regression with degree 2, or quadratic polynomial, performed the best at predicting gross. The R^2 of 0.77 indicated that the relationship between gross and the independent variables, less attendance, explains 77% of the variance in the dataset. The closer the R^2 is to 1 the better the model fit and how well it predicted gross sales. The multiple regression model had the lowest R^2 and was the worst model fit. The elastic net regression with degrees 3 and 8 was in the middle. The polynomial regression with degree 3, or cubic polynomial, performed comparably to the quadratic polynomial. The quadratic polynomial might show a tiny amount of underfitting indicating that the training set might have a few more data instances where the model did not fit well.

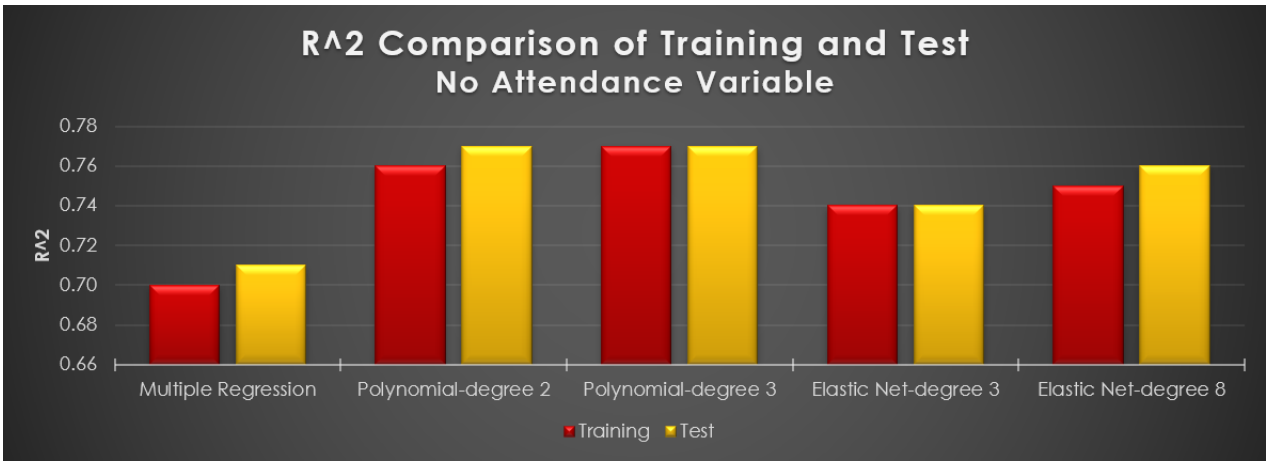


Fig. 22: R^2 Comparison (No Attendance Variable) L.Wood 2023

A chart comparing the RMSE results where the attendance variable was removed can be found in Figure 23. The quadratic and cubic polynomial models performed similarly when predicting gross sales on the test set. However, when it came to training, it appeared that the cubic polynomial model indicated a slight amount of over-fitting. This means that in the test set's spread of the prediction errors (actual values vs predicted values) was higher. This can cause the model to lose its ability to generalize to new data and its ability to predict gross. The multiple regression model had the highest RMSE and performed the worst in predicting gross. The elastic net regression with degrees 3 and 8 performed better than the multiple regression but not as well as the quadratic or cubic polynomials. For this comparison, the quadratic polynomial performed the best.

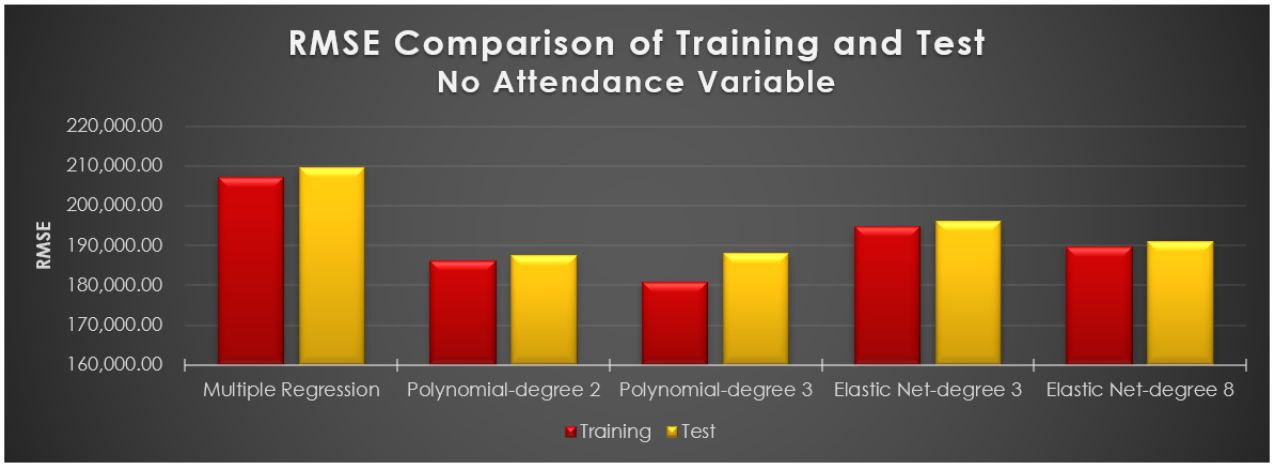


Fig. 23: RMSE Comparison (No Attendance Variable) L.Wood 2023

5.2 Modeling in Excel

The data cleaning that was previously described in section 3.4, and the EDA process, section 4.3, addressed non-numeric variables. The modeling used in Excel was a regression. This model was performed to look at the relationship comparing gross and year. When looking at regression between gross and year, the R^2 value was 0.91. This means that 91% of the variance in the dependent variable (Gross) is explained by the independent variable(Year). The results of the regression analysis can be seen in Figure 24. A scatter plot in Figure 25 with a regression line was also created to better display the relationship.

SUMMARY OUTPUT

Regression Analysis: Year and Gross

Regression Statistics	
Multiple	0.95551
R Square	0.913
Adjusted	0.90952
Standard	1.26E+08
Observat	27

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regressic	1	4.14E+18	4.14039E+18	262.3569268	9.14E-15
Residual	25	3.95E+17	1.57815E+16		
Total	26	4.53E+18			

	Coefficients	*andard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	-1E+11	6.22E+09	-16.08618629	1.06997E-14	-1.1E+11	-8.7E+10	-1.1E+11	-8.7E+10
Year	50276342	3103969	16.19743581	9.13801E-15	43883598	56669086	43883598	56669086

Fig. 24: Regression Analysis: Year and Gross. The R^2 of 0.91 indicates 91% of the gross sales are explained by the year. L.Wood 2023

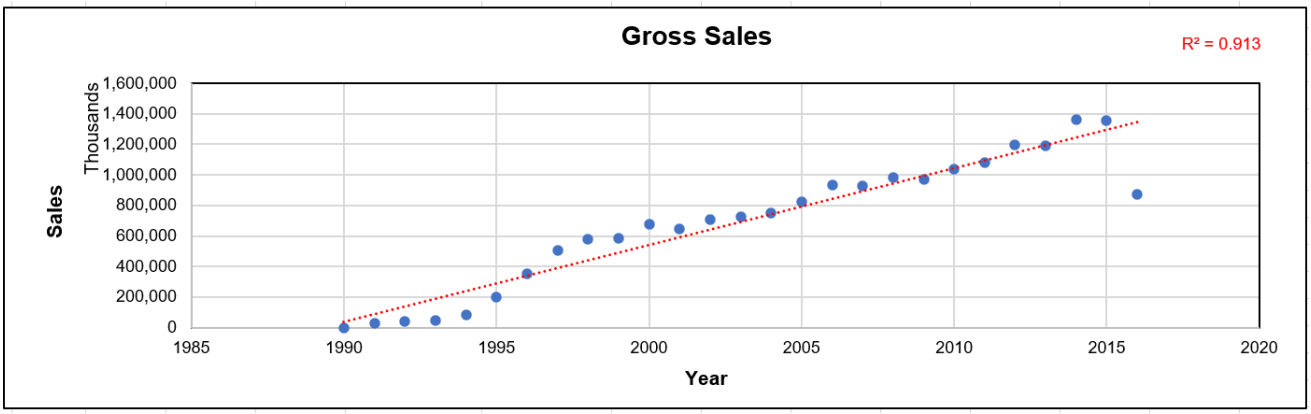


Fig. 25: Scatter Plot with Regression Line: Year and Gross. The chart an upward trend in shows gross sales over the years. There was a jump in gross sales around 1996-1997. L.Wood 2023

The complete visualizations and results can be found on GitHub under broadway_clean.xlxs, The link to the GitHub repository is https://github.com/lwood7983/L.Wood_Final_Capstone_Project.git.

5.2.1 Additional Excel Analysis

A deeper analysis was done to determine what factors contributed to the rising gross sales. One potential factor could be inflation and another could be attendance. To determine if inflation has any impact on the increase in gross, a consumer price index (CPI) per year was located at and used from https://www.minneapolisfed.org/about-us/monetary-policy/inflation-calculator/consumer-price-index-1913 [5]. The data was originally pulled from the U.S. Bureau of Labor Statistics [8].

The first analysis regarding the increase in gross sales was done around inflation. An analysis was completed comparing the average ticket price per year vs the adjusted inflated average ticket price per year. This was calculated by (gross/attendance)/CPI*100. For example, for the year 1991, the calculation would look like this: (30,665,673/ 559,179)/136.2*100). A regression analysis was executed to compare the inflation-adjusted average ticket price to the year. The results of the regression can be found in Figure 26. Inflation does account for a significant portion of the increase in gross sales over the years, based on the slope (adjusted inflation). However, it does not account for all of the increase. If the slope was 0, then inflation could account for almost all of the increase in gross sales over the years. A scatter plot was also created to visualize the average ticket price vs the average ticket price when inflation was removed (adjusted inflation). The scatter plot can be shown in Figure 27. When inflation was removed, the average ticket price has a minimal incline, especially when compared to the average ticket price.

SUMMARY OUTPUT

Regression Analysis: Average Ticket Price (Adjusted Inflation) and Year

Regression Statistics	
Multiple R	0.784276895
R Square	0.615090249
Adjusted R Square	0.599693859
Standard Error	5.021880813
Observations	27

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	1007.5178	1007.52	39.9503	1.3E-06
Residual	25	630.48217	25.2193		
Total	26	1638			

	Coefficients	*tandard Erro*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	1963.929573	6.2565157	313.901	1.8E-46	1951.04	1976.82	1951.04	1976.82
Adjusted Inflation	1.100819308	0.1741631	6.32062	1.3E-06	0.74212	1.45951	0.74212	1.45951

Fig. 26: Regression Analysis: Adjusted Inflation Average Ticket Price & Year. L.Wood 2023

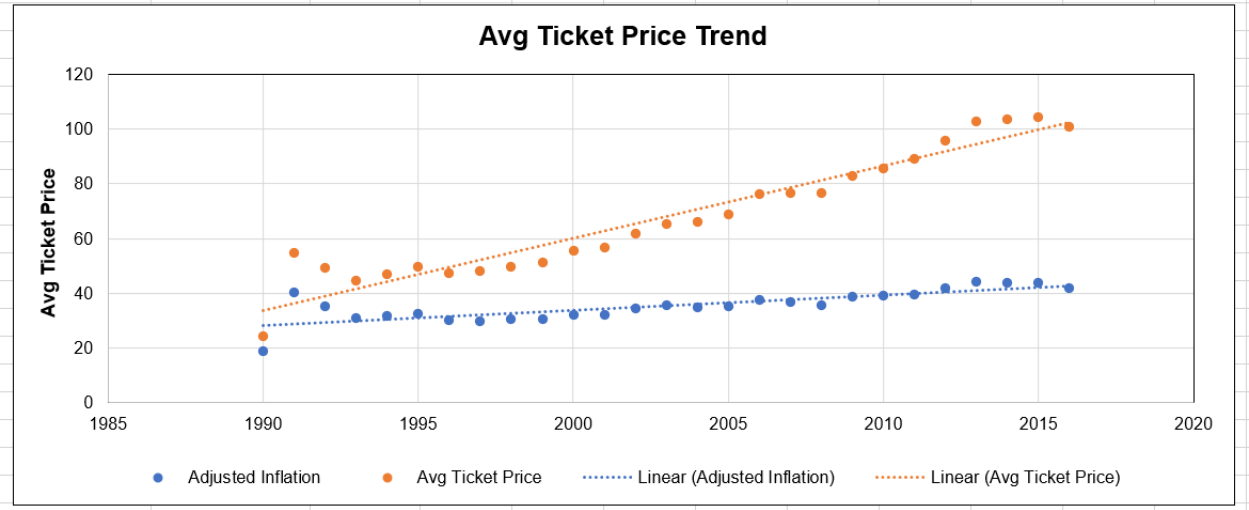


Fig. 27: Scatter Plot: Average Ticket Price & Average Ticket Price (Adjusted Inflation). The chart shows the impact of inflation over the years. The average ticket price without inflation (adjusted inflation) shows a minimal incline whereas the average ticket price with inflation shows a higher average ticket price as well as a rapid increase in price. L.Wood 2023

The second analysis was to determine what impact attendance possibly had on the rising gross sales per year. A scatter plot was created to compare gross and attendance and is shown in Figure 28. The scatter plot shows that both attendance and gross have an upward trend. There was a huge jump in attendance between 1996 - 1997 which could be related to the clean-up as described in "The Unexpected Lessons of Times Square's Comeback"[9]. Around 2000, however, it appears that while attendance started to decline, gross sales continued to rise. This would indicate that non-inflation gross is not attendance driven.

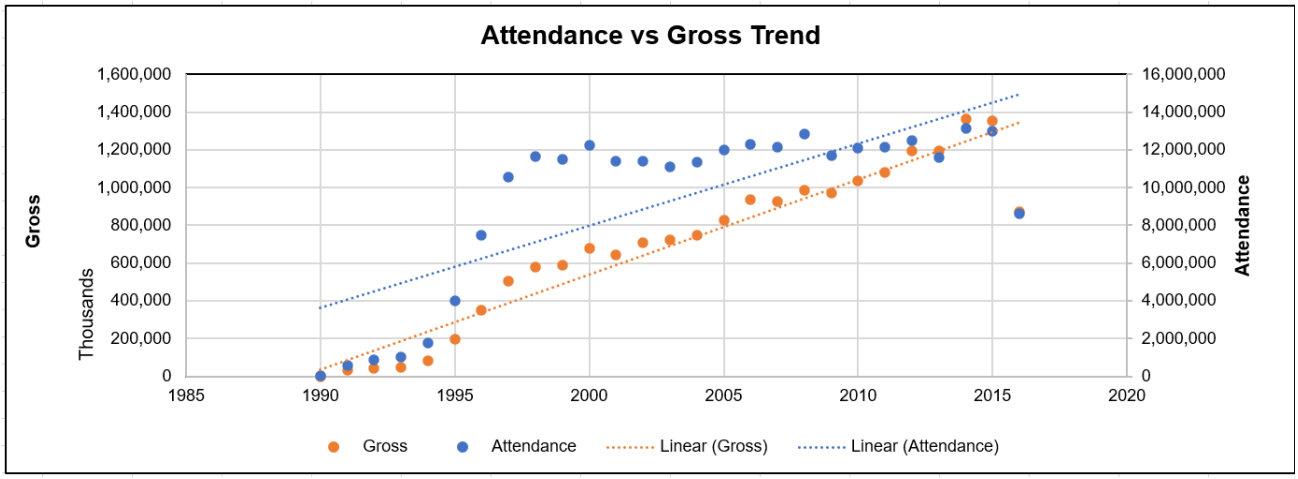


Fig. 28: Scatter Plot: Gross & Attendance. The chart shows that there was a massive jump in attendance between 1996-2000. After 2000 attendance remained steady and then started to decline. However, gross sales continued to rise. L.Wood 2023

The complete visualizations and results can be found on GitHub under broadway_clean_inflation.xlxs. The link to the GitHub repository is https://github.com/lwood7983/L.Wood_Final_Capstone_Project.git.

6 Conclusion

"Broadway is a boulevard full of famous theatres where visitors from all over the world gather to watch the top musicals of the season and other great plays." [3] For this project the goal was to determine which factors make a successful show and use gross sales as the dependent variable to predict.

When looking at all of the independent variables, the quadratic polynomial performed the best. The R^2 improved to 0.96 in the test set which means the relationship between gross and the independent variables explains 96% of the variance in the dataset. The closer the R^2 is to 1 the better the model fit and how well it predicts gross sales. The RMSE remained relatively the same between the training and test sets, meaning that the standard deviation, or spread of the prediction error, had minimal change. The worst-performing model was the multiple regression due to the lowest R^2 and the highest spread of prediction errors. The conclusion results can be found in Figure 29.

	Training		Test	
Model (Independent Variables)	**RMSE**	**R2**	**RMSE**	**R2**
Multiple Regression	133,737.51	0.88	133,737.10	0.88
Polynomial Regression, degree 2	81,503.77	0.95	81,724.57	0.96
Polynomial Regression, degree 3	78,111.96	0.96	80,014.38	0.96
Elastic Net degree 3	113,947.97	0.91	114,102.60	0.91
Elastic Net degree 8	101,433.45	0.93	102,996.31	0.93

Fig. 29: Training & Test Results (Independent Variables). L.Wood 2023

Additional modeling was performed with attendance removed as an independent variable since attendance most likely is unknown in advance. Again, the quadratic polynomial performed the best. The R^2 improved from 0.76 to 0.77, meaning the relationship between the gross and independent variables (less attendance) explains 77% of the variance in the dataset. The RMSE between the training and test remained relatively the same which shows a minimal change in the spread of the prediction errors. The multiple regression model performed the worst with the lowest R^2 and the highest spread of prediction errors. The conclusion results can be found in Figure 30.

	Training		Test	
Model (No Attendance)	**RMSE**	**R2**	**RMSE**	**R2**
Multiple Regression	207,191.89	0.70	209,507.29	0.71
Polynomial Regression, degree 2	186,117.02	0.76	187,424.37	0.77
Polynomial Regression, degree 3	180,862.43	0.77	187,954.33	0.77
Elastic Net degree 3	194,734.28	0.74	195,971.85	0.74
Elastic Net degree 8	189,585.94	0.75	191,117.50	0.76

Fig. 30: Training & Test Results (No Attendance Variable) L.Wood 2023

In Excel, ANOVA testing was performed comparing Type vs Gross, Theatre vs Gross, and Quarter vs Gross. Based on those comparisons, the Type being Musicals and Quarter 2 showed the highest average gross sales. Theatre, while significant to gross, had a minimal chance that the average mean would be the same across theatres.

A linear regression was performed comparing year and gross sales. The R^2 value was 0.91. This means that 91% of the variance is explained by the independent variable (Year). Charts of the results can be found in Figure 24 and 25.

The results also showed an increase in gross sales over the years. A further look into the climb in gross sales determined that while inflation is a major driver, there is also a non-inflation factor such as attendance. The results suggest the best way to increase gross sales would be to find reasons and methods to increase the average ticket price.

6.1 Limitations

There were a couple of limitations to this project, both around the dataset. The first is that the dataset was limited to only August 1990 to August 2016. Another limitation is that the dataset did not include any years during and post Covid.

6.2 Future Work

For future work, there would be several items to look further into. The first would be to see what the impact on the variables is after Covid. The variables would include additional attributes such as price and attendance after upgrades were done to theatres. [6] Another future work item would be to see if reviews on shows have an impact on their success. The reviews could include sentiment analysis by looking at social media or newspaper reviews. A third item would be to see if Tony Awards have any impact on the success of the show or even if a successful show earns a Tony Award. This could even go further into understanding who the leads are in shows and if that impacts the success.

References

1. https://www.broadwayleague.com/research/statistics-broadway-nyc/
2. https://www.broadwayleague.com/
3. Broadway musicals, https://www.introducingnewyork.com/broadway#:~:text=Broadway%20is%20a%20boulevard%20full,Manhattan%20at%20an%20oblique%20angle.
4. BWW, T.: A brief history of broadway, https://www.broadwayworld.com/article/A-Brief-History-of-Broadway-20211219
5. MINNEAPOLIS, F.R.B.O.: Consumer price index, 1913-, https://www.minneapolisfed.org/about-us/monetary-policy/inflation-calculator/consumer-price-index-1913-
6. Paulson, M.: Curtains up! how broadway is coming back from its longest shutdown., https://www.nytimes.com/2021/09/13/theater/broadway-reopening.html
7. Sand, R.: Broadway: The engine that helps fuel new york city's economy, https://www.forbes.com/sites/rogersands/2023/01/20/broadway-the-engine-that-helps-fuel-new-york-citys-economy/?sh=6bd122bc53cc
8. STATISTICS, U.B.O.L.: Consumer price index, https://www.bls.gov/cpi/
9. Stern, W.J.: The unexpected lessons of times square's comeback, https://www.city-journal.org/html/unexpected-lessons-times-square%E2%80%99s-comeback-12235.html