Success of an Advertisement

Project to build simple model to predict whether the advertisements will produce a net gain or not.

Submitted By: G Dinesh Seervi

Submitted to: Mr. Arihant Jain

Acknowledgement

We extend our sincere thanks to our respected Head of the division Mr Arihant Jain for allowing us to use the facilities available We extend our sincere and heartfelt thanks to our esteemed guide, Mr Arihant Jain sir, for providing us with the right guidance and advice at the crucial junctures and for showing me the right way. We also take this opportunity to express a deep sense of gratitude to all our teachers and our friends for their cordial support, valuable suggestions and guidance.

Declaration

The project submitted herewith is a result of our own efforts in totality and in every aspects of the project works. All information that has been obtained from other sources had been fully acknowledged.

INDEX

1. Objective

In advertising, AI and ML elevate the ability to make buying decisions that emulate human decision making. However, with the right AI technologies and data insights, it's actually possible to significantly improve decision-making processes beyond human capabilities.

While many advertisers don't yet realize it, ML is the only viable solution to this growing problem. Among other things, the technology makes it possible to analyse and gain insights from vastly more data than a human ever could ever process in their lifetimes. What's more, it also automates advertising decisions based on these insights, making ads more efficient and effective, and driving business revenue in the process.

In this project we will perform a basic exploratory data analysis of the data, also we will make a simple model to predict whether the advertisements will produce a net gain or not. This is a binary classification problem where you need to predict whether an ad will lead to a net gain.

Description

The source of the dataset is from a competition in hackerearth. This dataset is also available on kaggle website. The dataset consist of three files in it i.e. train file, test file and sample submission file.

Data Description:

Train.csv: 26049 x 12 [including headers]: training data set

Test.csv: 6514 x 11 [including headers]: test data set

Sample_submission.csv: 6514 x 2 [including headers]: Sample submission file

About train file

This dataset consists of data if an advertisement will be success or not which is in the column net gain.

2. Data

Header	Description
id	-Unique id for each row
ratings	-Metric out of 1 which represents how much of the targeted demographic watched the advertisement
airlocation	-Country of origin
airtime	-Time when the advertisement was aired
average_runtime(minutes_per_week)	-Minutes per week the advertisement was aired
targeted_sex	-Sex that was mainly targeted for the advertisement
genre	-The type of advertisement
industry	-The industry to which the product belonged
economic_status	-The economic health during which the show aired
relationship_status	-The relationship status of the most responsive customers to the advertisement
expensive	-A general measure of how expensive the product or service is that the ad is discussing.
money_back_guarantee	-Whether or not the product offers a refund in the case of customer dissatisfaction.
netgain [target]	-Whether the ad will incur a gain or loss when sold

3. Approach

Step 1: The dataset is downloaded from the kaggle website.

Step 2: unzipping the dataset

Step 3: loading the necessary libraries required for the model in Jupyter Notebook

Step 4: Loading the train and test data-set using pandas.read_csv

Step 5: Check for missing Value in the dataset

Step 6: Data Imputation and Correlation Matrix

Step 7: Exploratory Data Analysis

Step 8: Data Visualization

Step 9: Encoding the categorical variable

Step 10: Build a Model

Step 11: Predict the targeted Variable

Step 12: Calculate the accuracy score on test and train set

Step 13: Compare the accuracy score of different model and select the best model

Step 14: apply the best model for test set

Step 15: submit the Sample Submission

4. Detailed Data description

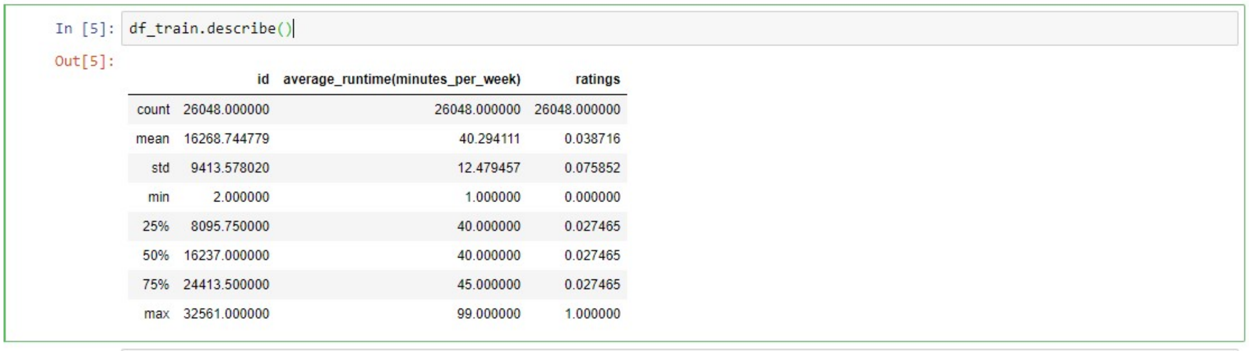
Source of the Dataset

Link: https://www.kaggle.com/rohanchreddy/advertsuccess

Percentage of missing value

0% missing value in the dataset. There are no missing values in the dataset

Mean and Standard Deviation fun () on the train set



```
In [5]: df_train.describe()
Out[5]:
```

	id	average_runtime(minutes_per_week)	ratings
count	26048.000000	26048.000000	26048.000000
mean	16268.744779	40.294111	0.038716
std	9413.578020	12.479457	0.075852
min	2.000000	1.000000	0.000000
25%	8095.750000	40.000000	0.027465
50%	16237.000000	40.000000	0.027465
75%	24413.500000	45.000000	0.027465
max	32561.000000	99.000000	1.000000

5. Correlation Matrix

When two sets of data are strongly linked together we say they have a High Correlation.

The word Correlation is made of Co- (meaning "together"), and Relation

Correlation is Positive when the values increase together, and Correlation is Negative when one value decreases as the other increases a correlation is assumed to be linear (following a line). Correlation examples Correlation can have a value:

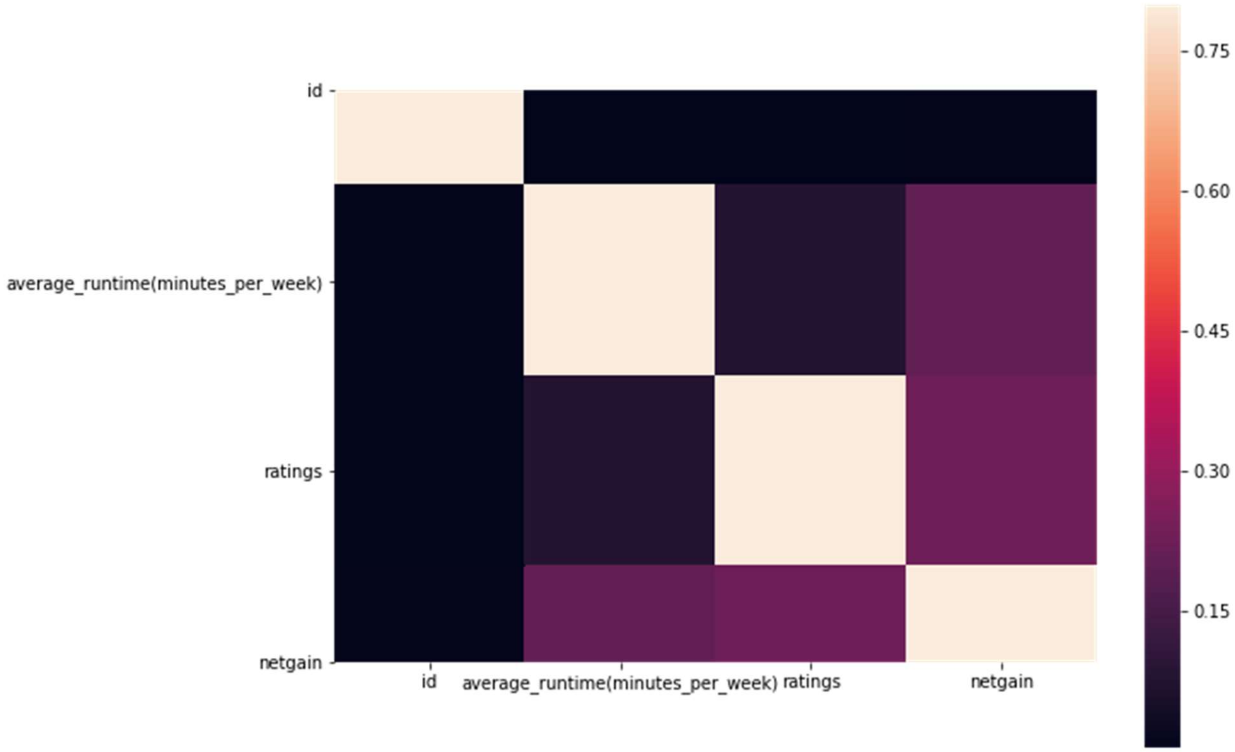
1 is a perfect positive correlation

0 is no correlation (the values don't seem linked at all)

-1 is a perfect negative correlation

The value shows how good the correlation is (not how steep the line is), and if it is positive or negative.

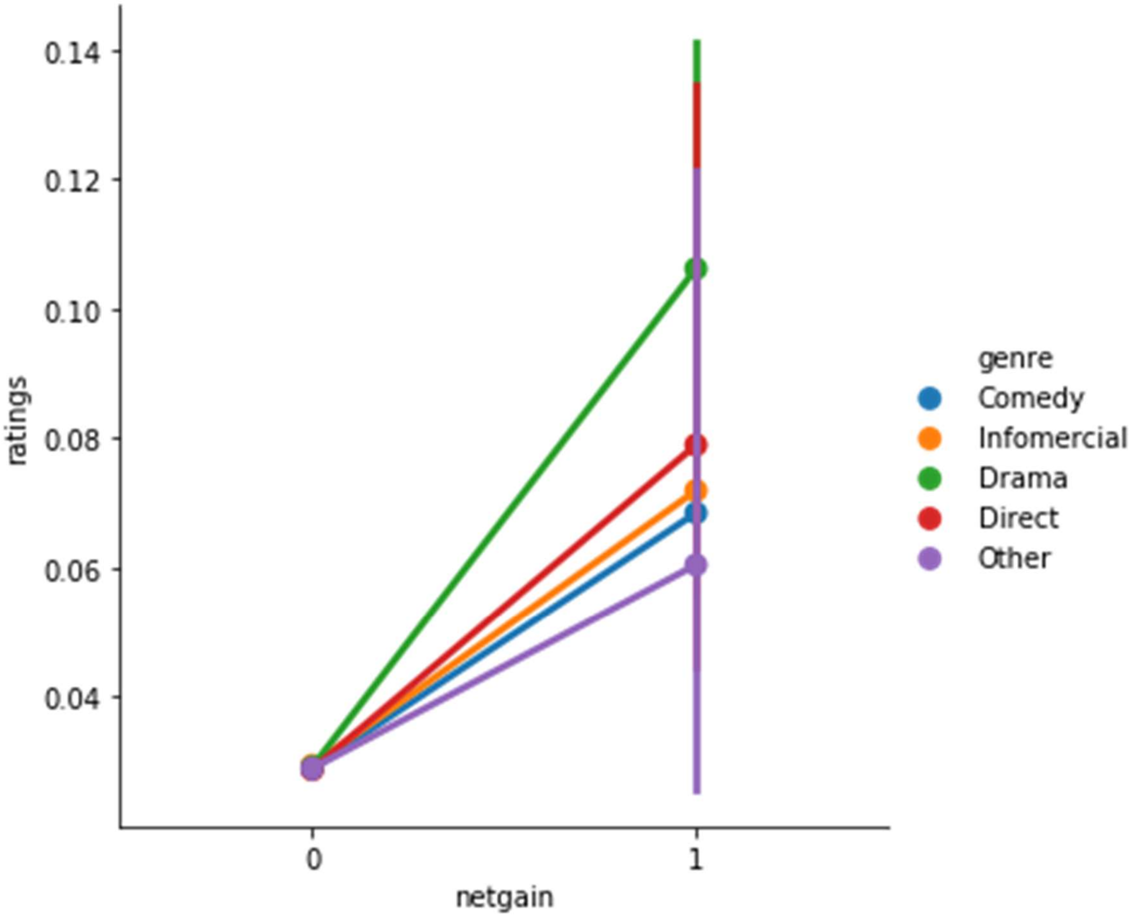
The Correlation matrix of train set:



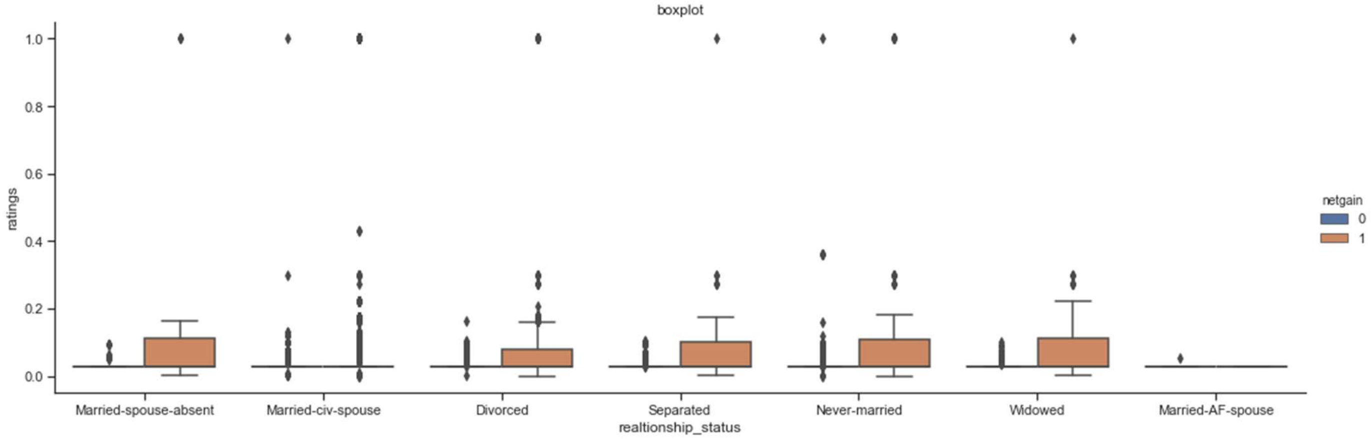
6. Data Visualization

Data visualization is the graphic representation of data. It involves producing images that communicate relationships among the represented data to viewers of the images. This communication is achieved through the use of a systematic mapping between graphic marks and data values in the creation of the visualization.

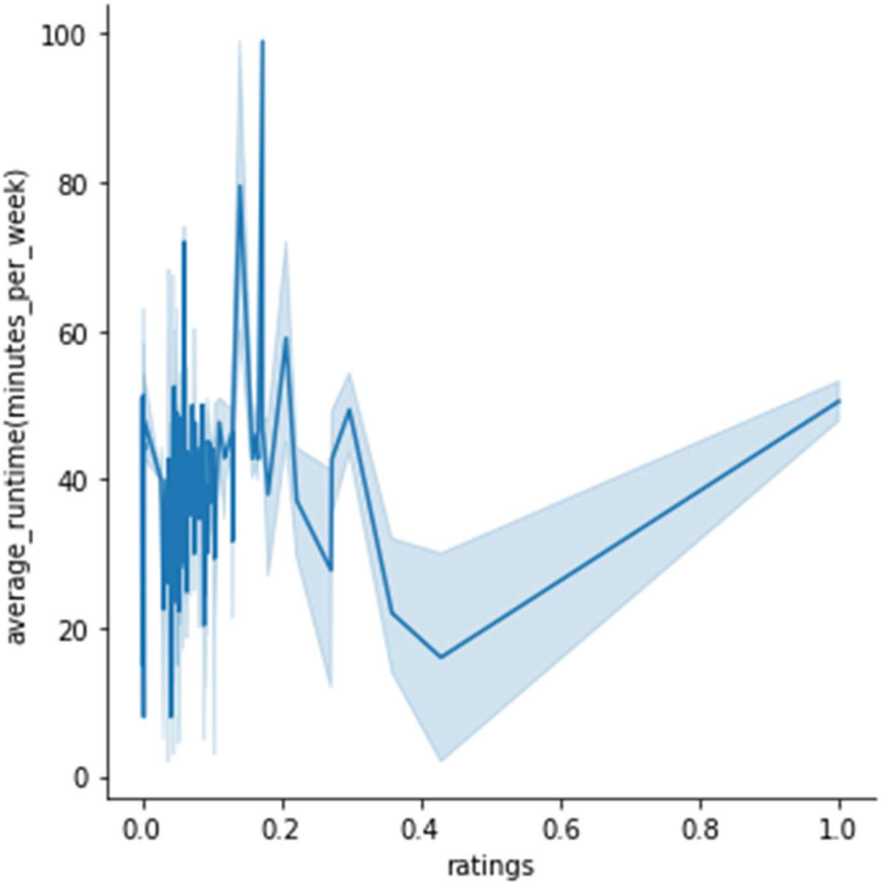
The graph representing ratings vs genre vs net gain



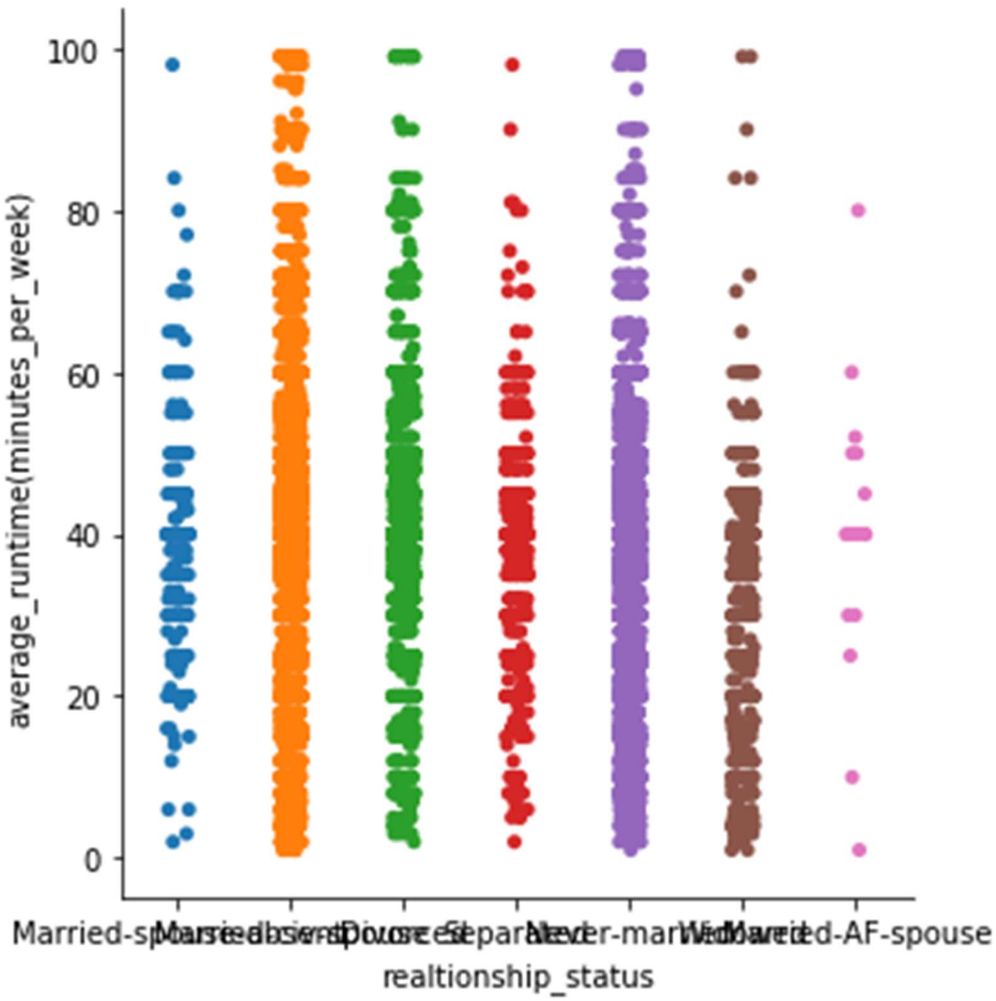
The graph representing relationship status vs ratings vs net gain



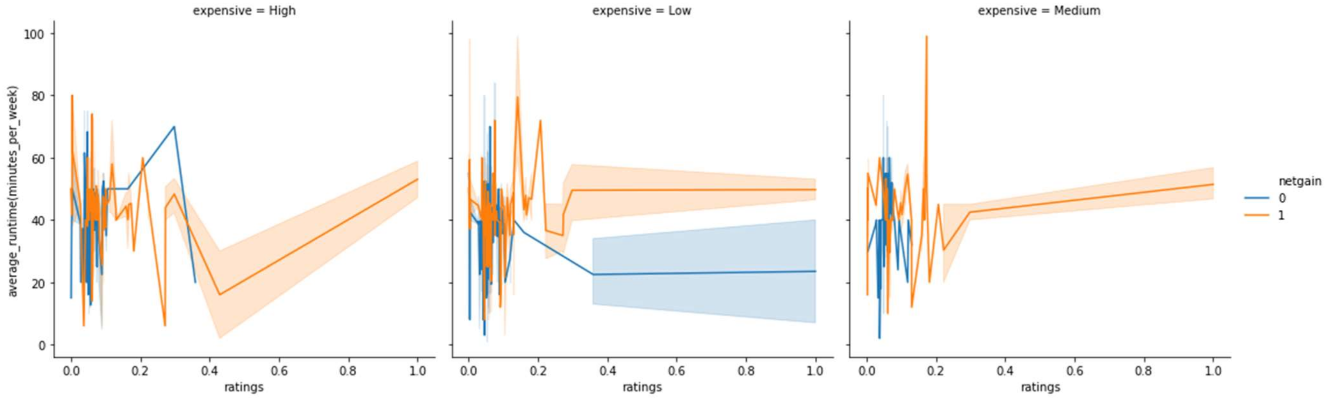
Graph representing ratings vs average runtime



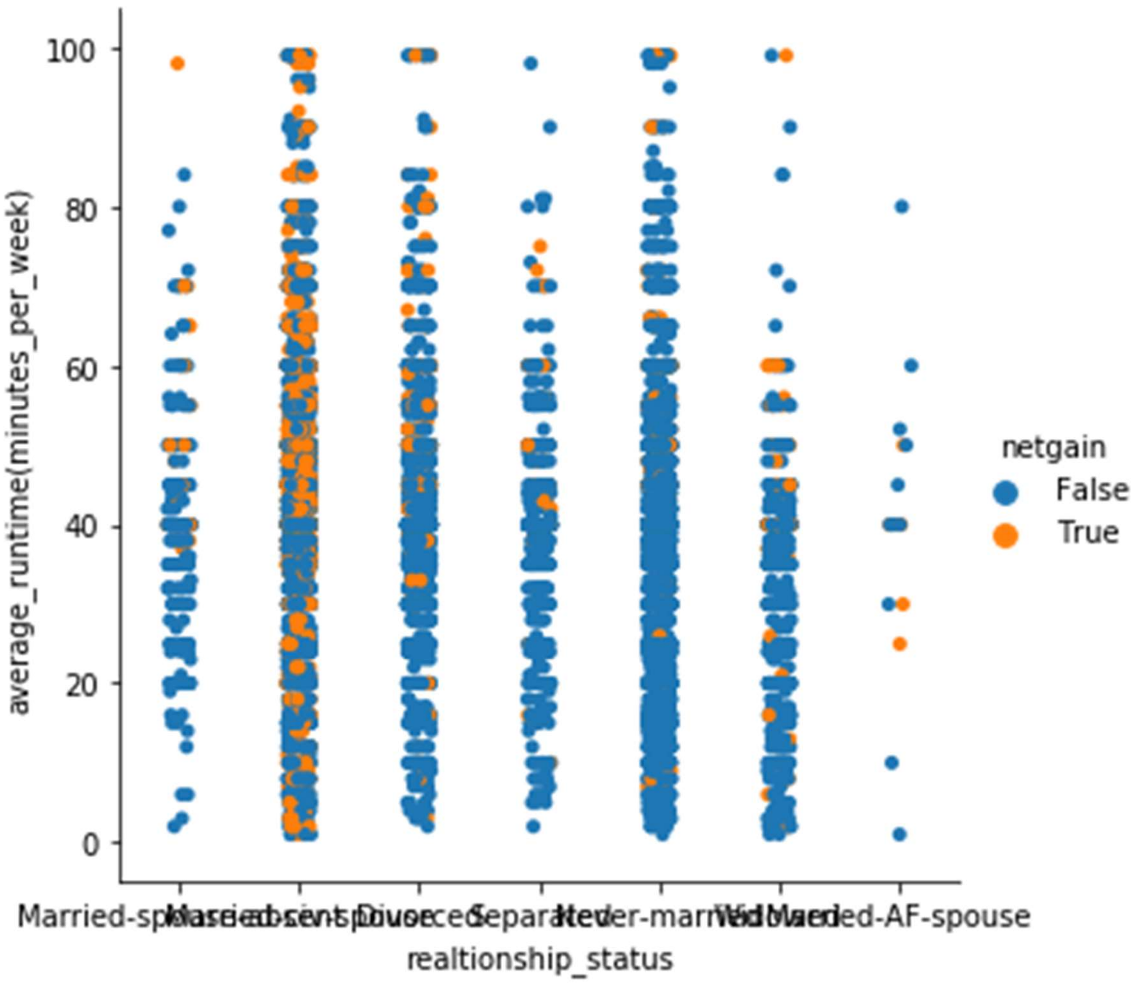
Graph representing relationship status vs runtime



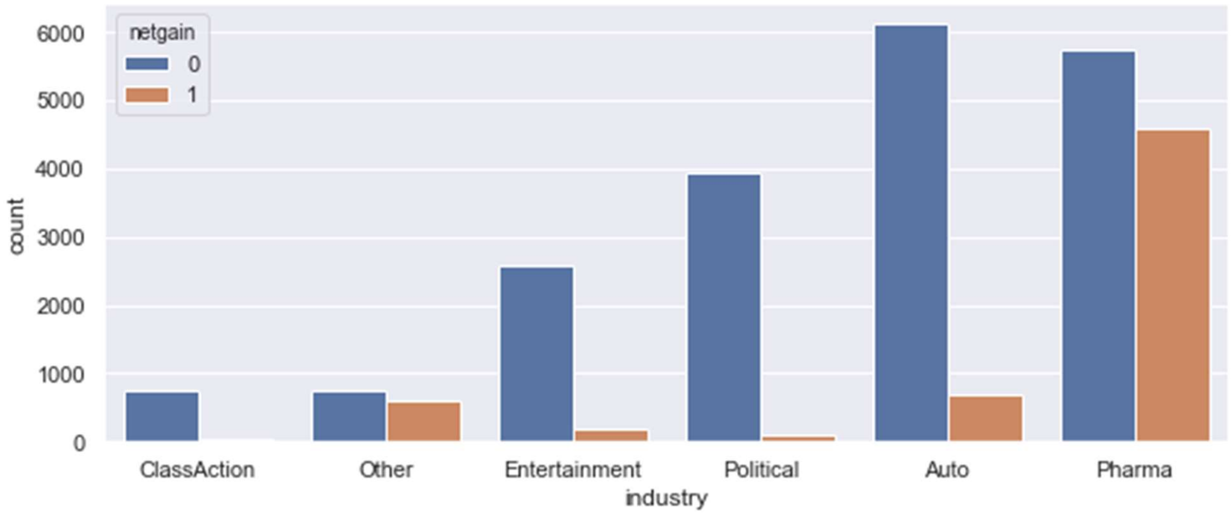
Graph representing ratings vs average runtime vs net gain



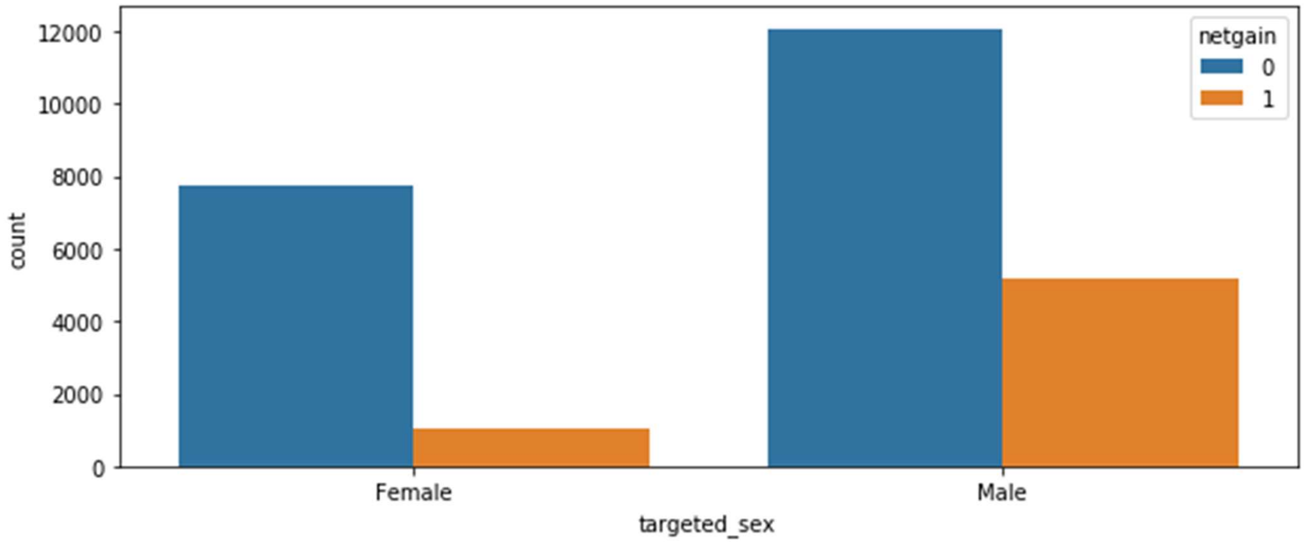
Graph representing relationship status vs runtime vs netgain



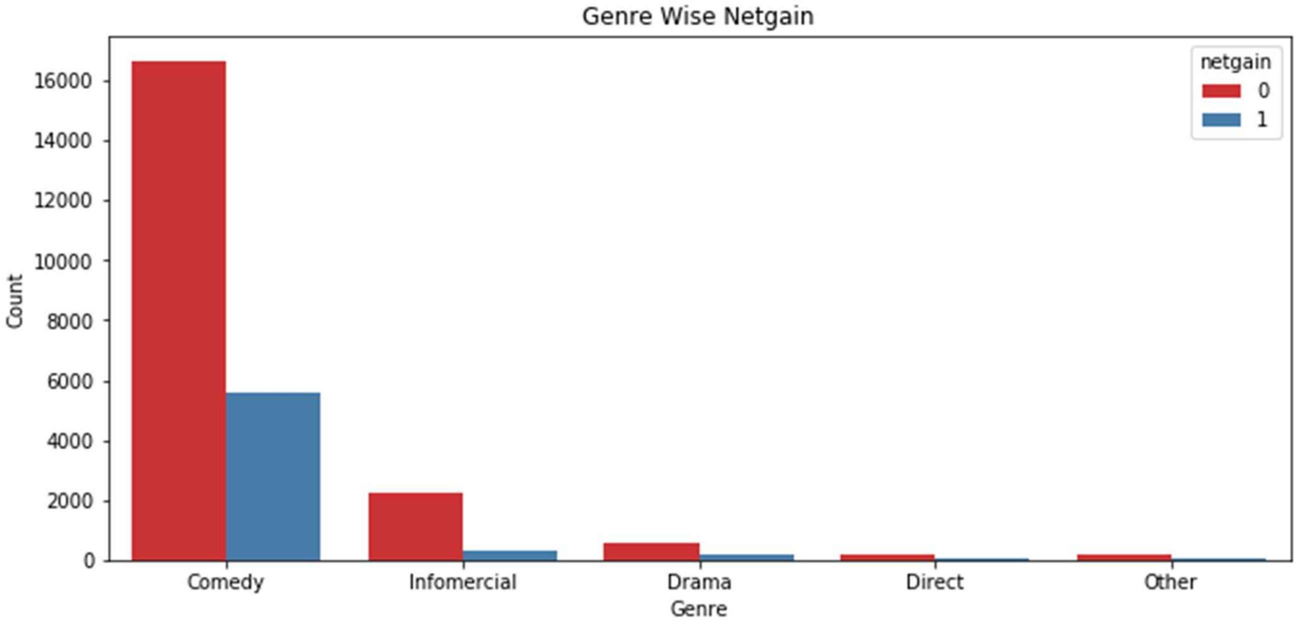
Graph representing industry vs net gain



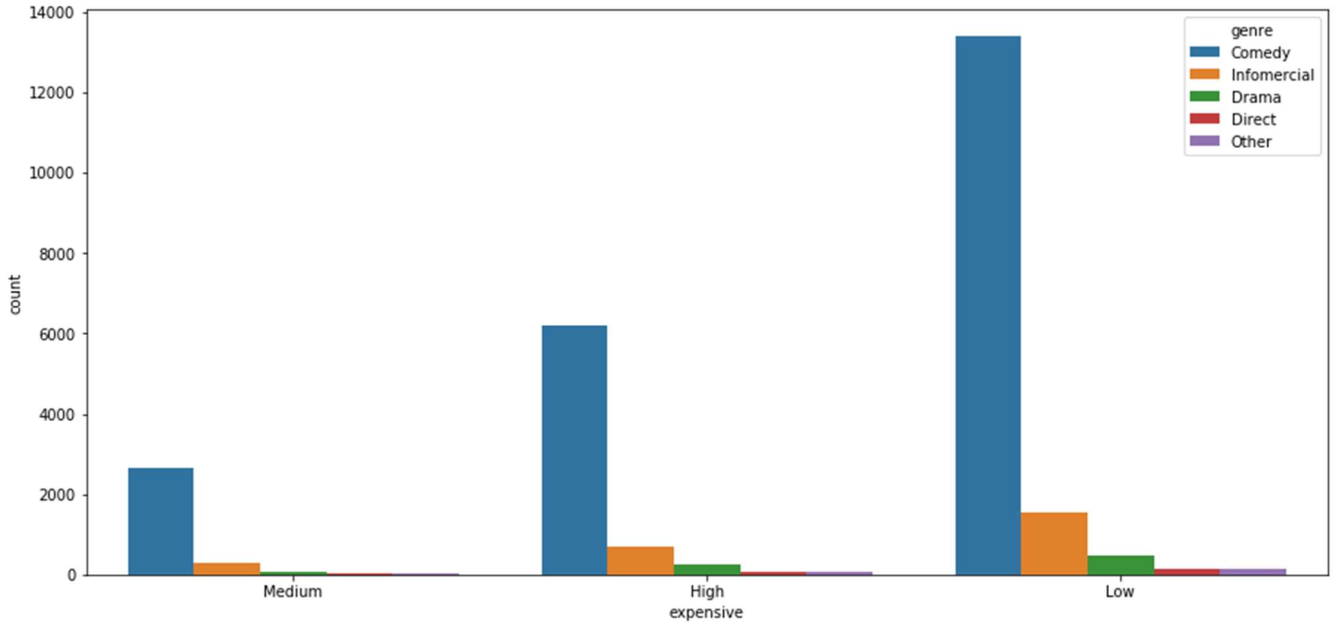
Graph representing targeted sex vs net gain



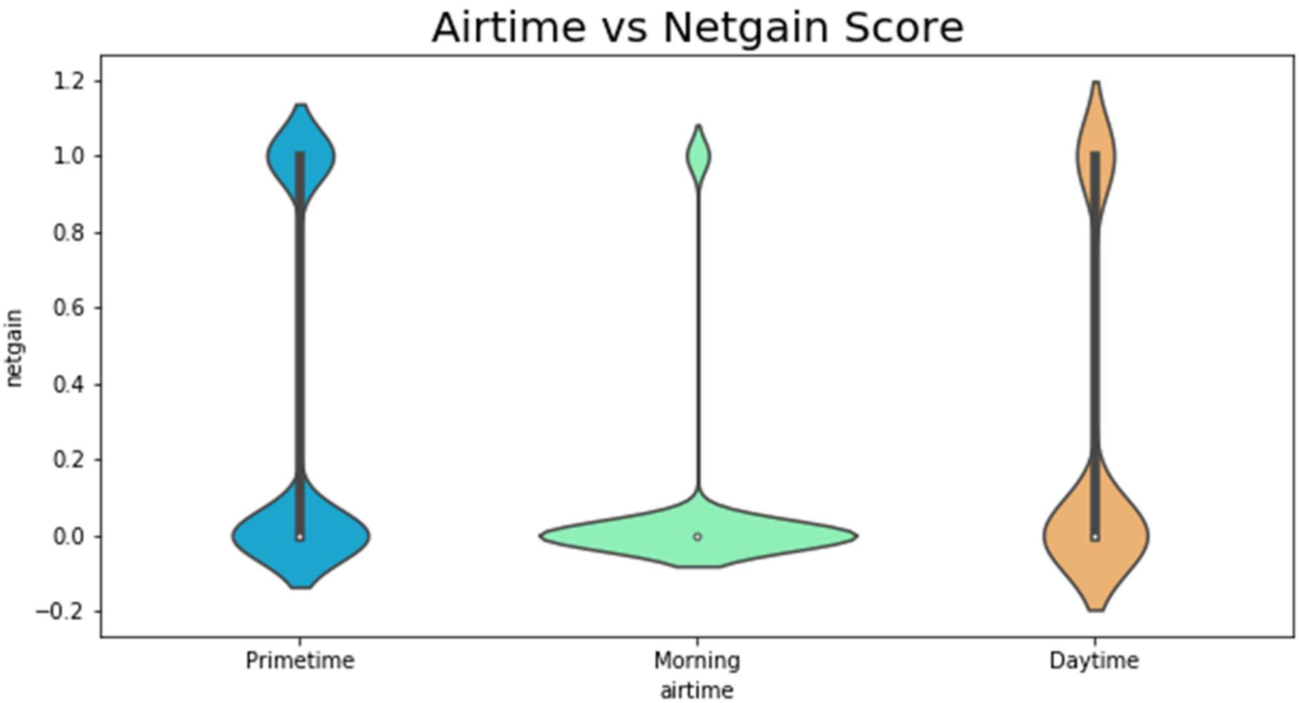
Graph representing genre vs net gain



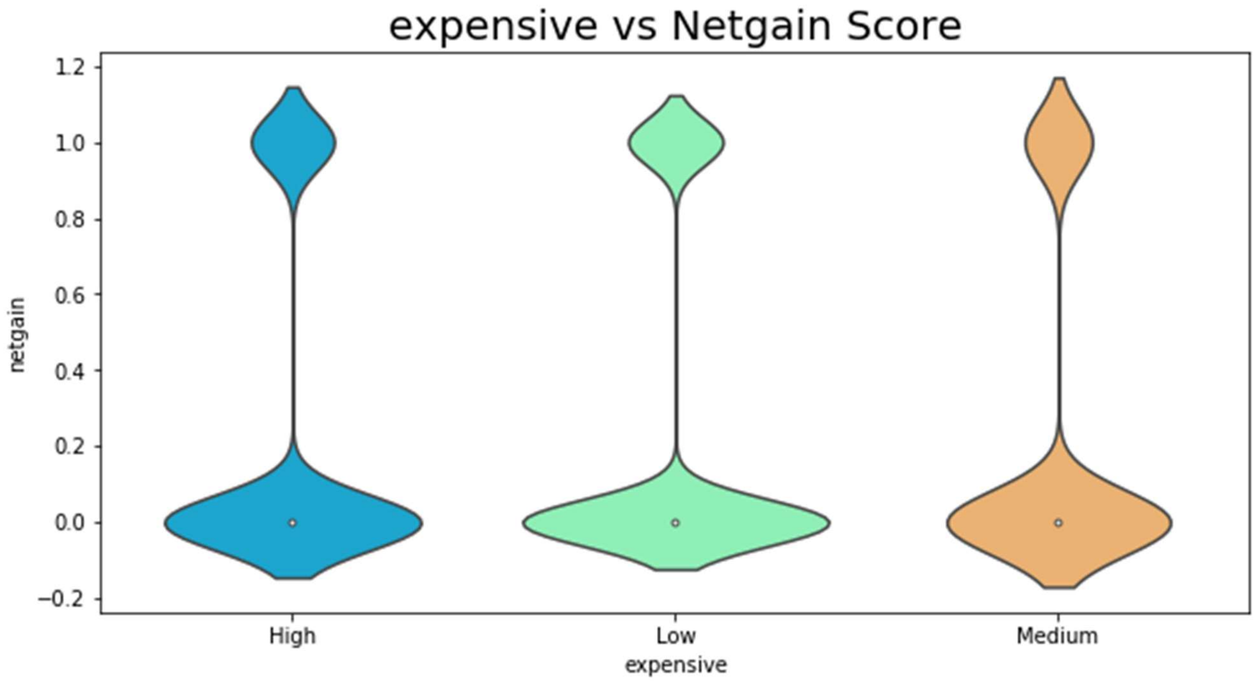
Graph representing expensive vs genre



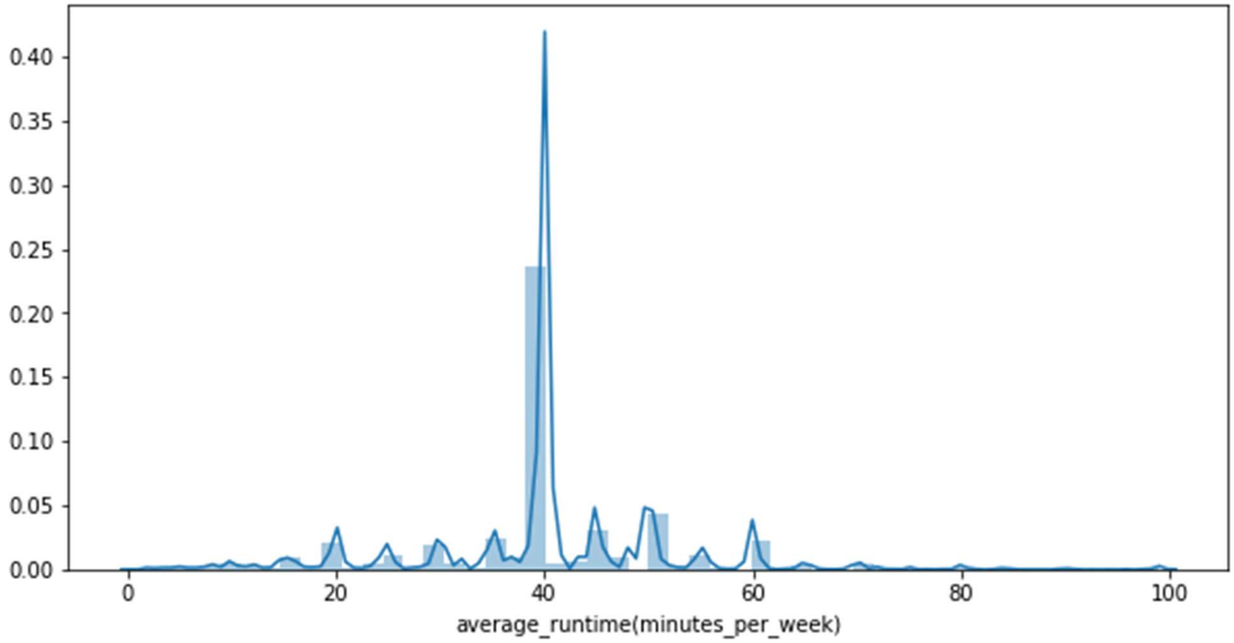
Graph representing airtime vs expensive



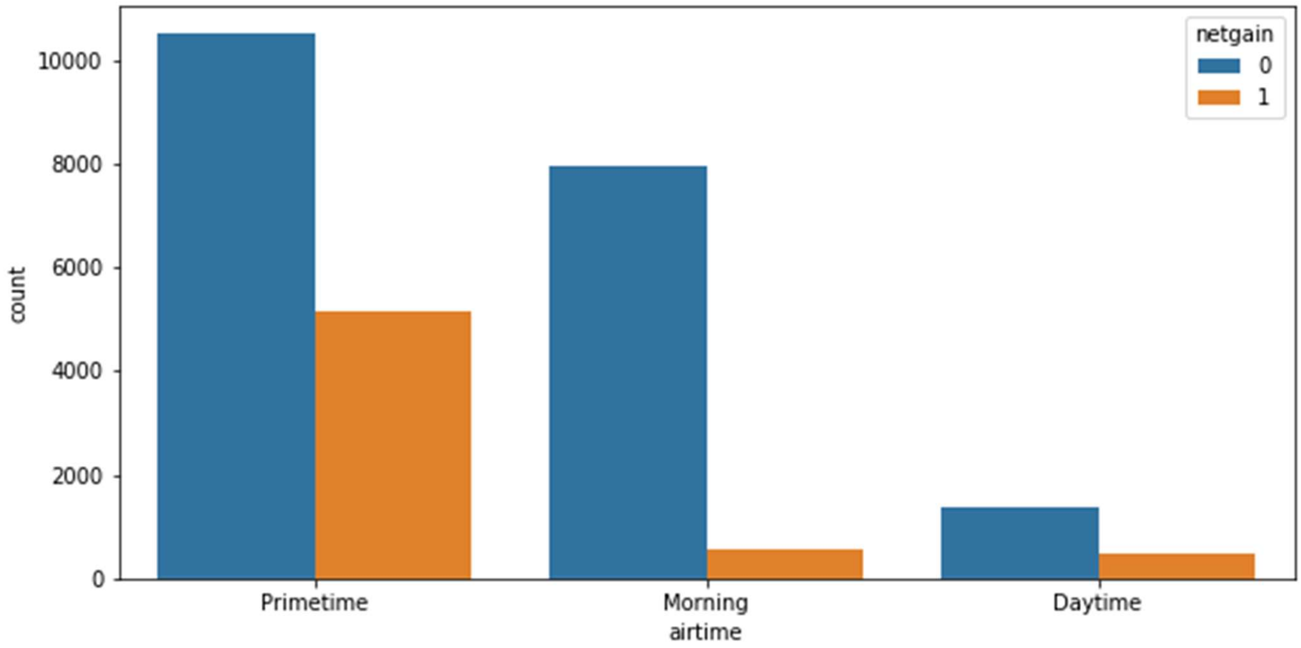
Graph representing expensive vs net gain



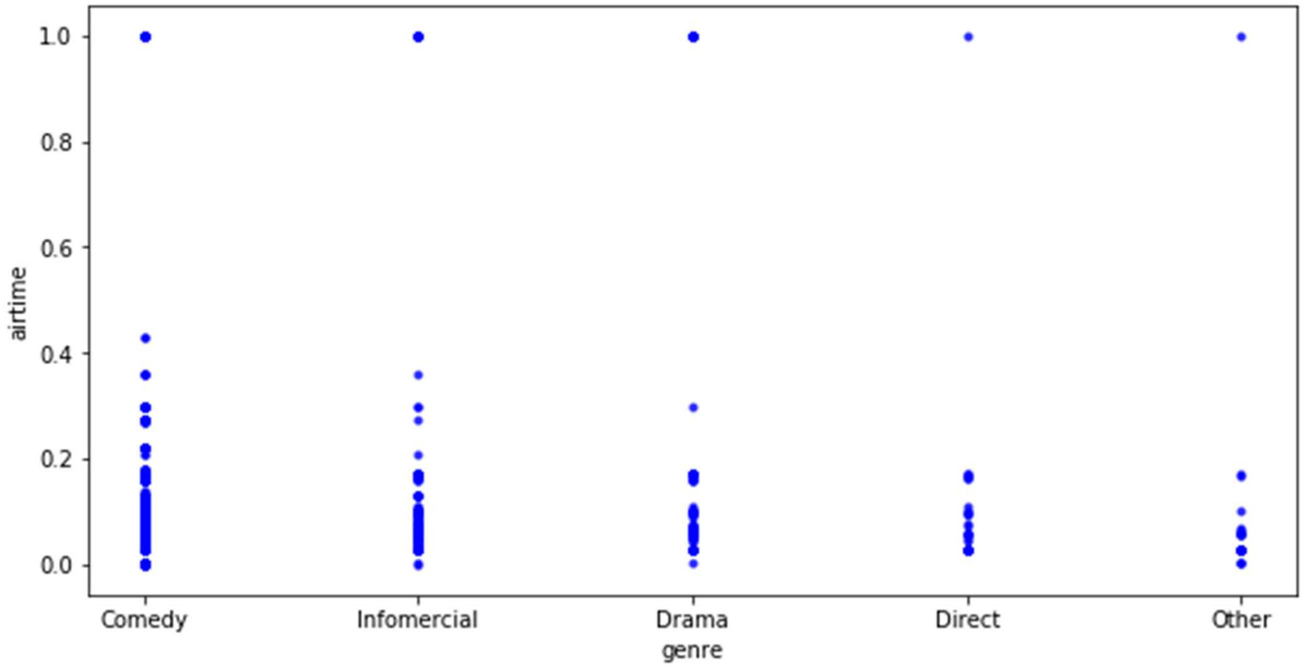
Graph representing the flow of average runtime



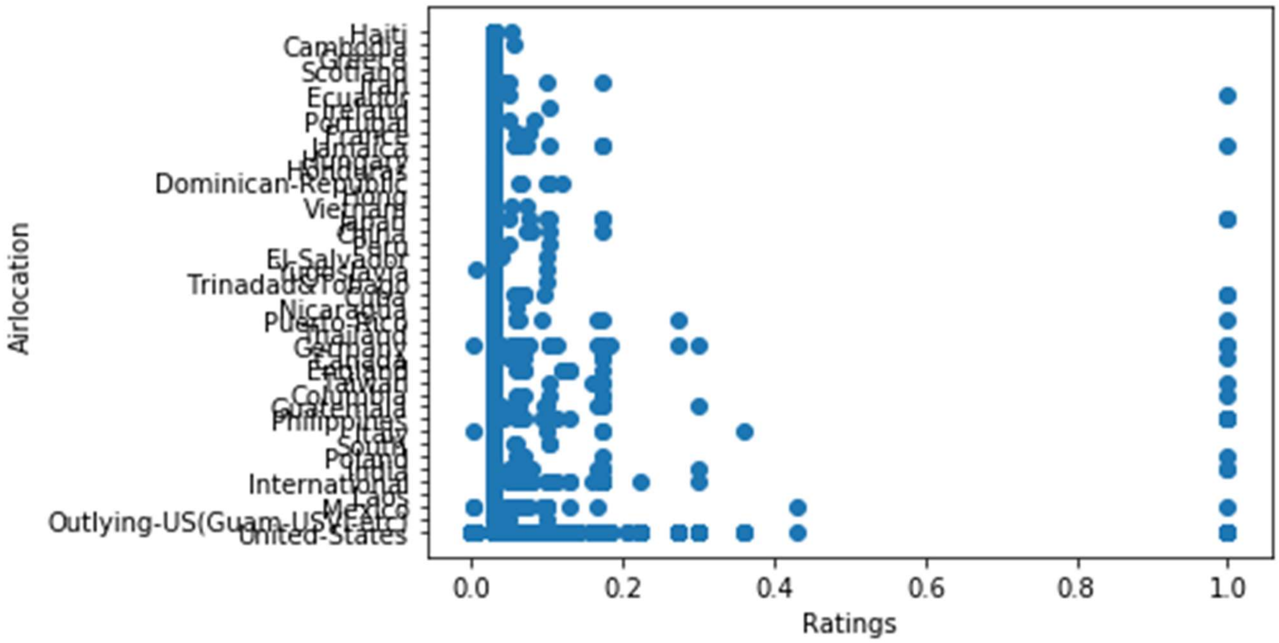
Graph representing airtime vs net gain



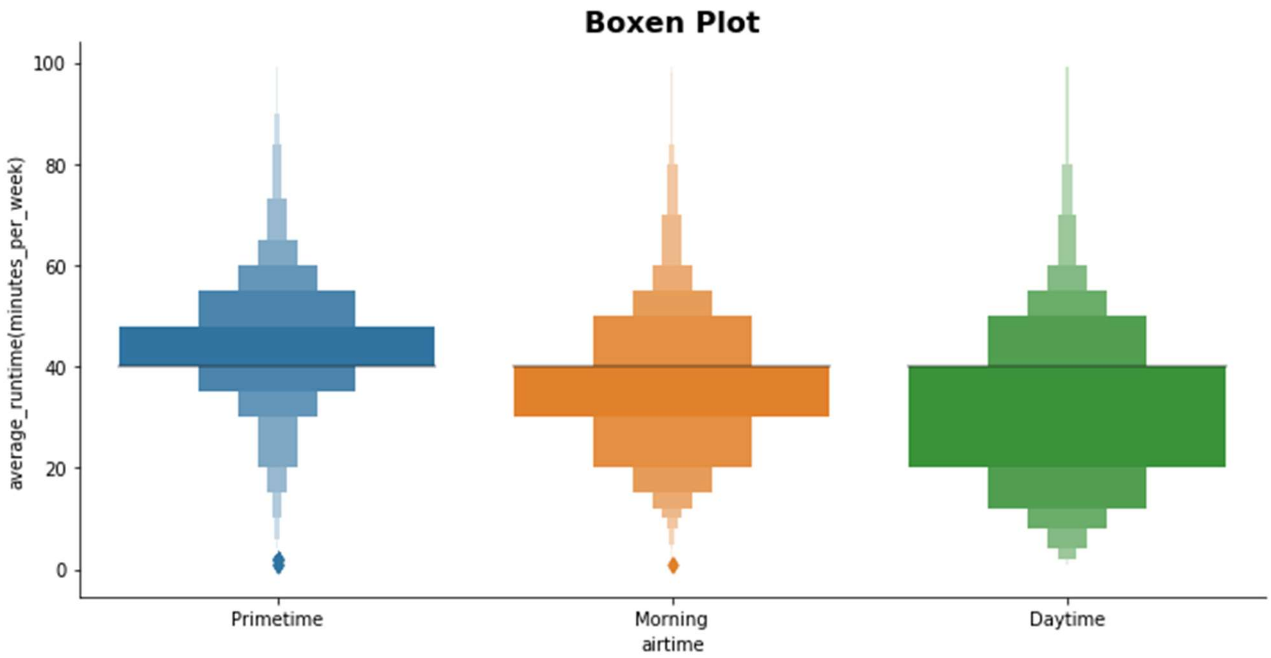
Graph representing genre vs airtime



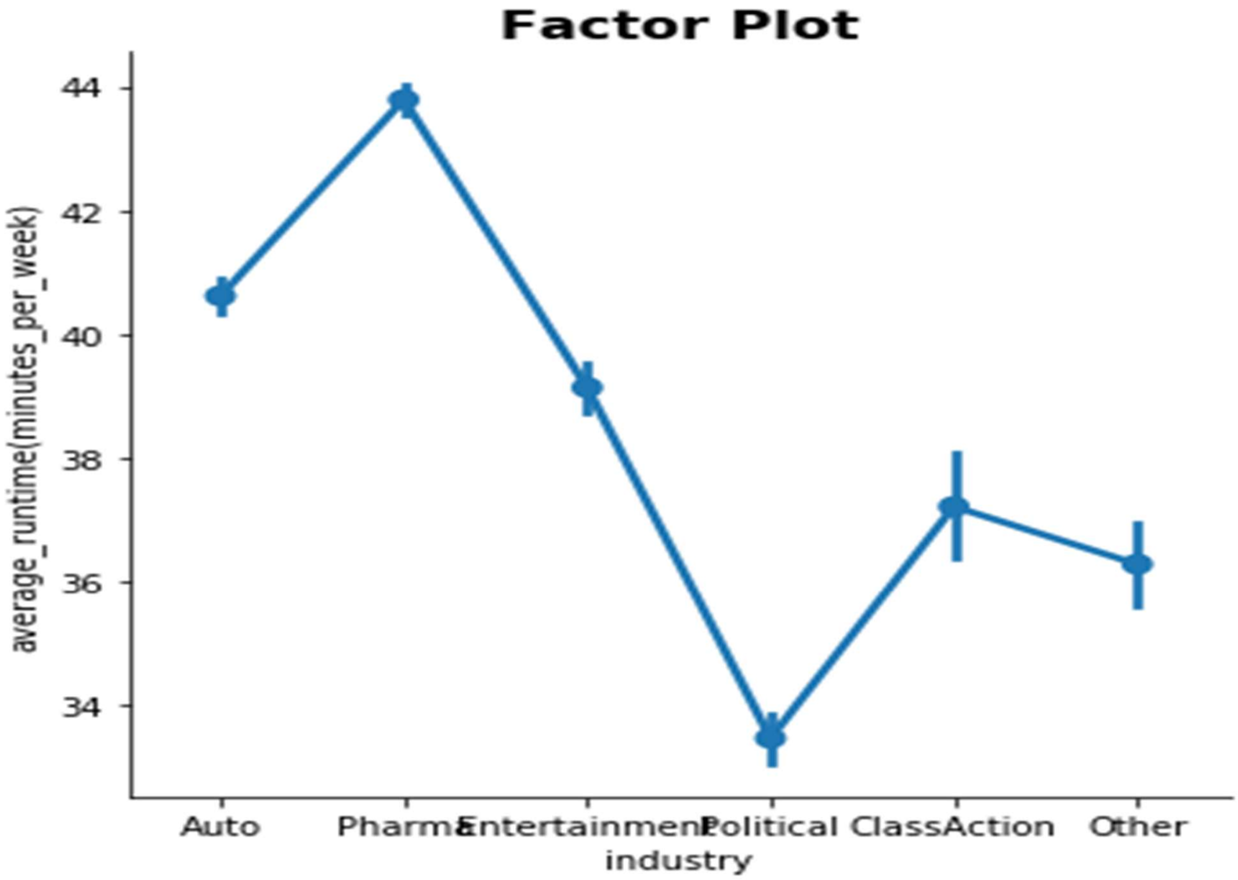
Graph representing air location vs ratings



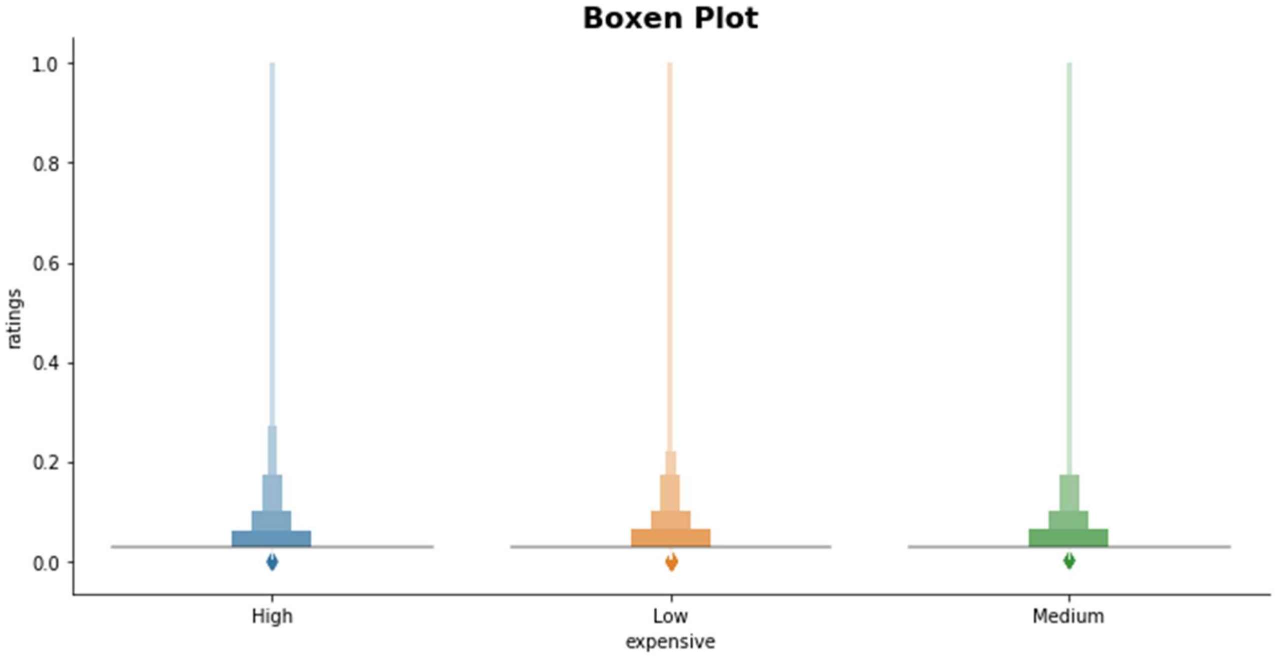
Graph representing airtime vs average runtime



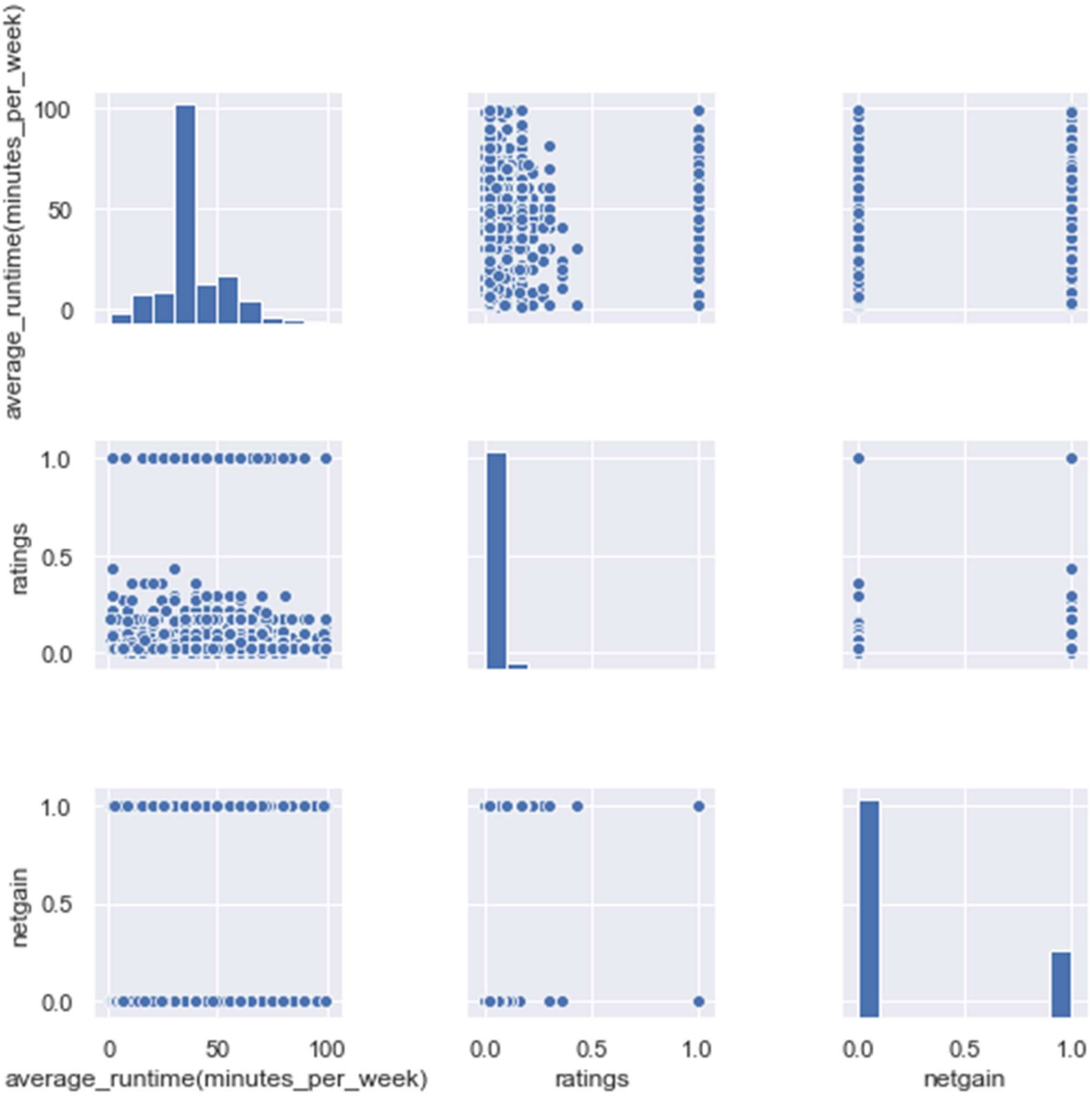
Graph representing industry vs average runtime



Graph representing expensive vs ratings



Pair graph on dataset



7. One Hot Encoding

Library: `sklearn.preprocessing`.OneHotEncoder

Description: A one hot encoding is a representation of categorical variables as binary vectors. This first requires that the categorical values be mapped to integer values. Then, each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1.

Uses: One of the major problems with Machine Learning is the fact that you cannot work directly with categorical data. Machine Learning is, after all, a bunch of mathematical operations translated to a computer via low-level programming languages.

Computers are brilliant when dealing with numbers. So, we must somehow convert our input data (in whichever sequential format it be) to numbers. Then only our innocent mathematically gifted GPUs and CPUs will be able to process the data. Once we are done with the processing on the numbers, we need another mechanism to somehow revert the output to the same format as that of the input data. This is where One-Hot Encoding is used.

8. Label Encoding

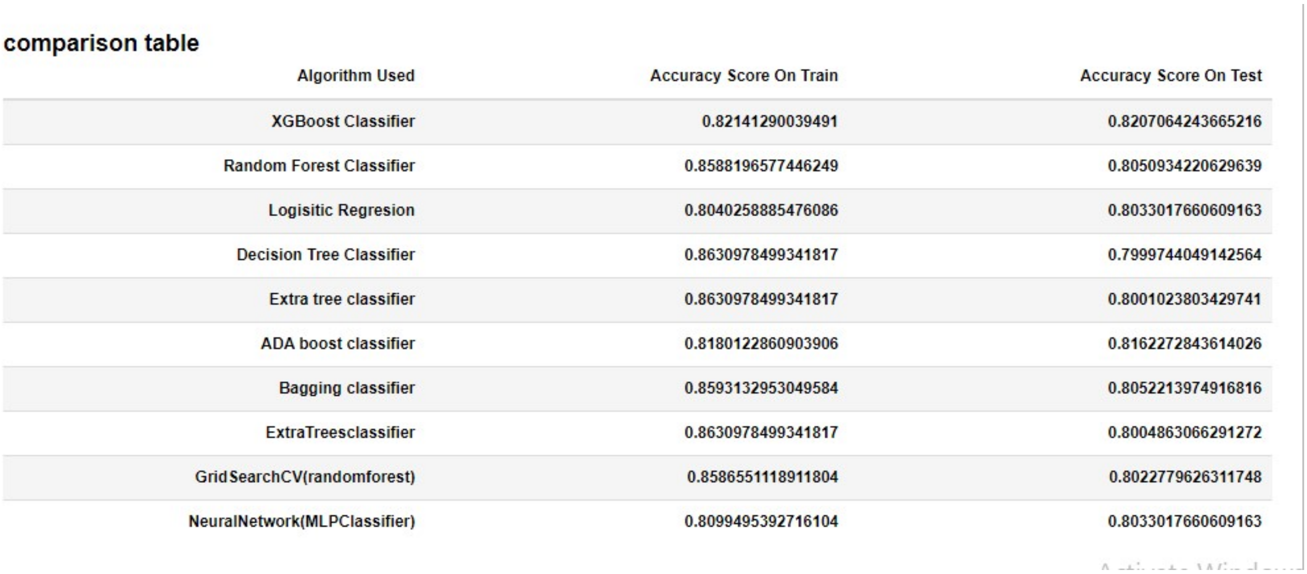
Library: `sklearn.preprocessing.LabelEncoder()`

Description: Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Uses: Encoding categorical variables is an important step in the data science process. Because there are multiple approaches to encoding variables, it is important to understand the various options and how to implement them on your own data sets. The python data science ecosystem has many helpful approaches to handling these problems.one of the best approach is label encoding.

9. Selection of Best model

After data imputations and encoding of the data various model are build. The Metris “**Accuracy_score**” is used to compare the models. Comparison table is drawn which consist of algorithm used and accuracy score on train and test dataset.



comparison table

Algorithm Used	Accuracy Score On Train	Accuracy Score On Test
XGBoost Classifier	0.82141290039491	0.8207064243665216
Random Forest Classifier	0.8588196577446249	0.8050934220629639
Logisitic Regresion	0.8040258885476086	0.8033017660609163
Decision Tree Classifier	0.8630978499341817	0.7999744049142564
Extra tree classifier	0.8630978499341817	0.8001023803429741
ADA boost classifier	0.8180122860903906	0.8162272843614026
Bagging classifier	0.8593132953049584	0.8052213974916816
ExtraTreesclassifier	0.8630978499341817	0.8004863066291272
Grid SearchCV(randomforest)	0.8586551118911804	0.8022779626311748
NeuralNetwork(MLPClassifier)	0.8099495392716104	0.8033017660609163

10. Conclusion

From the above comparison table it is found that XGBoost algorithm has the best accuracy score both on train and test dataset. XGBoost algorithm gives 0.82 accuracy on both test and train set and also is dis-satisfy the condition of overfitting. Hence XGBoost model is chosen over other models for the further predictions. Sample submission file is created using XGBoost model.

11. Future Enhancement

- **Building Better Audience Profiles**

  Among other things, ML has the capacity to draw on data from a wide variety of sources, including search engines, social media, and other third-party platforms. It can then draw connections and make sense of it in a way that a normal human brain isn’t equipped to do.

- **Discovering New Audiences**

  The same processes that make it possible to build better audience profiles can also help advertisers discover new audiences that they never before thought to target. ML can help uncover unlikely but valuable correlations between consumer demographics, interests, and online behaviour that reveal new potential target audiences.

- **ML Saves Time, Creates Efficiencies**

  Historically, most advertisers have relied on spreadsheets and basic analytics software to track their data and generate insights to optimize their campaigns. All too often, this becomes a tedious, error-prone and laborious strategy that generates only marginally valuable insights. ML is a worthwhile investment because it vastly broadens your dataset and analysis capabilities beyond that of advertisers