School of Computer Science and
Computational and Applied Mathematics

COMS4048A - Data Analysis and Exploration(2022)

YouTube Analytics dataset Project

Author(s):
Phindulo Makhado - 1832463
Khanani Mathebula - 1847799
Rotondwa Mavhengani - 2114834

12th November 2022

Table of Contents

Abstract

After Google, YouTube is the second-largest search engine on the planet. Additionally, if creating videos is an important component of your marketing plan, you've undoubtedly already put some thought into creating videos for your own YouTube channel. As a platform for engaging viewers, YouTube offers YouTube Analytics, which provides a wealth of information on how to enhance the performance of videos. A YouTube channel owner can use the platform's analytics to examine information regarding the channel's overall performance, its posts, and its target audience. The channel owner can develop a content plan to promote channel growth using these engagement metrics. In this project we have decided to engineer our very own target column, in order to weigh in how good of a profit is a channel making from the dataset collected by YouTube Analytics and obtained from Kaggle. This research paper validates the weight of profit in two ways using machine learning methods including K-nearest Neighbors Classifier and Logistic Regression. Predict the value of the target variable after first determining its dependence on the independent variable. To ascertain how reliant the target variable is on the independent variable, this task uses logistic regression. The key variables that have a considerable impact on the dependent variables are chosen based on the estimated dependencies.

1 Introduction

It is possible to learn a lot from this YouTube Analytics dataset on how to engage viewers and keep them coming back for more but furthermore generating good money. Adding columns for the thumbnail link, country codes, views, and user subscriptions enhances this dataset's ability to shed light on the effectiveness of YouTube videos and the elements that affect audience engagement. A video's total interaction success can also be determined by looking at the average view percentage and watch time columns. This dataset can be used to study how viewers engage with YouTube videos. The data includes information on likes, dislikes, comments, shares, and subscribers gained or lost. This data can improve video content and keep viewers coming back for more. The main aim that we will try to achieve by using this dataset is to understand how viewers interact with YouTube videos, and then rate/predict how well YouTube channel owners can make from various interactions.

1.1 Why do we care about this dataset?

Youtube is a video-sharing platform that people worldwide engage with daily. The chosen dataset contains information about viewers' engagement. Looking at the bigger picture visualization of this dataset would not only help us but the company and creators on the platform to have a better visual understating(Which is better than staring at a table full of unexplored data) on things their doing right or things they can improve on to attract more viewers and to keep them coming back to the stream. This knowledge can help in the overall revenue of the company and its creators.

The requirement of this project was to find data less explored and with enough features worth exploring and this data meet both requirements.

2 Data

2.1 Data Description and Pre-processing

All studies for this project will make use of the YouTube Analytics dataset, which includes YouTube channels that have been active since 2010. The weight of the profit made by YouTube Channel owners is predicted using the YouTube Analytics dataset, which can be either good(1) or bad(0) weighted. There are 224 samples in the original YouTube Analytics collection. There are 19 dependent variables for each sample channel. Comments added, Shares, Dislikes, Likes, Subscribers lost, Subscribers gained, RPM (USD), CPM (USD), Average percentage viewed (%), Views, Watch time (hours), Subscribers, Your estimated revenue (USD), Impression and Impression click-through rate (%), and Impressions are all metrics that were recorded. The huge amplitude of the changing values is one of the key disadvantages. negative values for variables like "likes" and "subscribers gained." Because certain variables have a bigger impact than others, this kind of irregularity can have an impact on projections. An approach to such issues is a linear transformation. All input values can be transformed linearly by multiplying them by the maximum variable value.

2.2 Data Cleaning

An essential first step in the data analytical process is data cleaning [1]. Typically, this critical practice comes before your primary analysis and entails gathering and validating data[1]. Although this is frequently a component of data cleaning, it is not the only method. The majority of the effort is put toward identifying and fixing rogue data. Complete, accurate, irrelevant, corrupt, or improperly formatted data are examples of rogue data[1]. Also included in the process is duplication. In essence, this involves combining or eliminating similar data pieces. The way we clean our data is as follows:

- **Step 1: Fix structural errors:** Structural errors usually emerge as a result of poor data housekeeping. They include things like typos and inconsistent capitalization, which often occur during manual data entry
- **Step 2: Remove duplicates:** Removing duplicate values from your data set plays an important role in the cleansing process[2]. Duplicate data takes up unnecessary storage space and slows down calculations to a minimum. At worst, duplicate data can skew analysis results and threaten the integrity of the data set[2]. In our dataset, there were no duplicates.
- **Step 3: Remove unwanted outliers:** Outliers are data points that dramatically differ from others in the set. They can cause a problem with certain types of data models and analyses [3]. In this project, we remove outliers via Z-score. A Z-score is a numerical measurement that describes a value's relationship to the mean of a group of values. Z-score is measured in terms of standard deviation from the mean [3]. If a Z-score is 0, it indicates that the data point's score is identical to the mean score. A Z-score of 1.0 would indicate a value that is one standard deviation from the mean. Z-scores may be positive or negative, with a positive value indicating the score is above the mean and a negative score indicating it is below the mean. **This is how we remove outliers:**
  - **(a)** The very first step will be setting the upper and lower limit. This range stimulates that every data point will be regarded as an outlier out of this range. The formulae for both upper and lower limits:
    - Upper: Mean + 3 * standard deviation
    - Lower: Mean – 3 * standard deviation
  - **(b)** the second step is to detect how many outliers are there in the dataset based on the upper and lower limit that we set up
  - **(d)** Start with the third and the last step, where we will finally remove the detected outliers in step 2 using two techniques; we can either go for Trimming and we will discuss both techniques more closely separately. The trimming technique work with any data regardless of what kind of data distribution you are working with, trimming is an applicable and proven technique for most data types. We pluck out all the outliers using the filter condition in this technique.

2.3 Normal Distribution

It is common to see the normal distribution, commonly referred to as the Gaussian distribution, for many naturally occurring measurements, including height and birth weight [4]. It is symmetric and bell-shaped. It can range from negative infinity to positive infinity and most of its value clusters around the mean[5]. The portion of the curve below the mean will be a mirror image of the portion of the curve above the mean which makes the mean equal to the median[4][5].

Outlier influence on a data distribution's variance and standard deviation. Extreme outliers in a data distribution skew the distribution in their direction, making it challenging to understand the data. Therefore, it is crucial to eliminate the outliers. Before training the models, a number of steps were taken to clean the data and explore it[4]. These steps included data cleaning by replacing null values and correlation testing using heatmaps. We also took care of outliers since they have an impact on the variance and standard deviation of our data distribution and could potentially make data analysis challenging due to skewness. It was also possible to handle outliers and correct erroneous skewness using the Z-Score approach[5]. Normal Gaussian Distribution Testing was also done to ensure that our data was as normally distributed as possible. In this situation, Yeo-Johnson Transformation proved to be helpful, as our un-normalized data was reasonably normal after the transformation.

In parametric testing, we are well aware of the parameters or assumptions that are made, and the population distribution is always known. Here probability distribution is a normal distribution and in case it fails, we use the central limit theorem to approximate the distribution to normal distribution. In this project to show the effects of normal distribution, skewness was used. When the portion of the curve below the mean is not a mirror image of the portion of the curve above the mean, we say the distribution is skewed. Depending on the side where there is the long tail, we say the distribution is skewed towards that side. A "skewed right" distribution is one in which the tail is on the right side. A "skewed left" distribution is one in which the tail is on the left side. The skewness measures the extent to which the data values are not symmetric around the mean.

2.4 Correlation testing

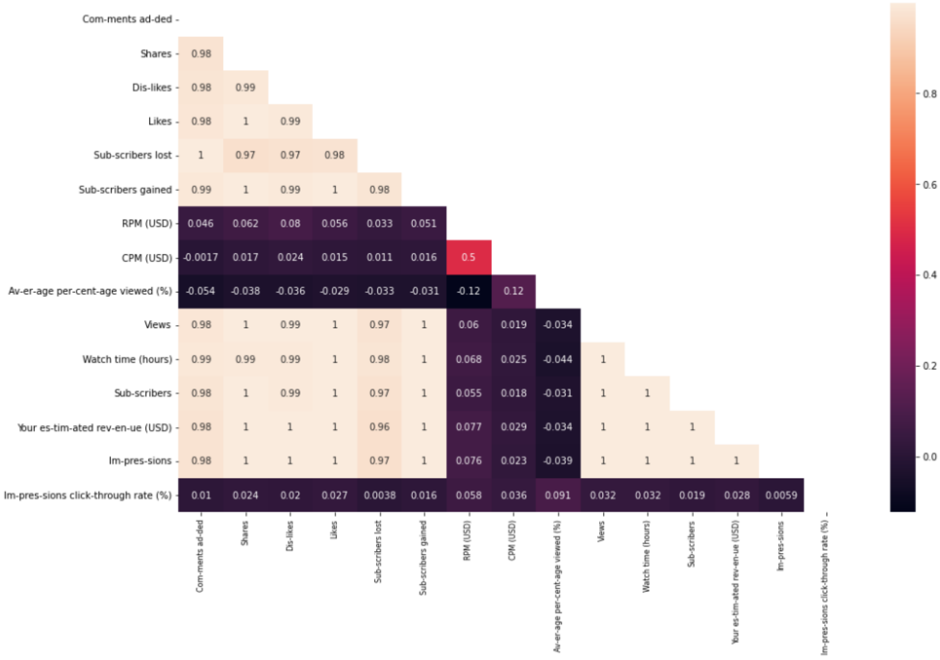
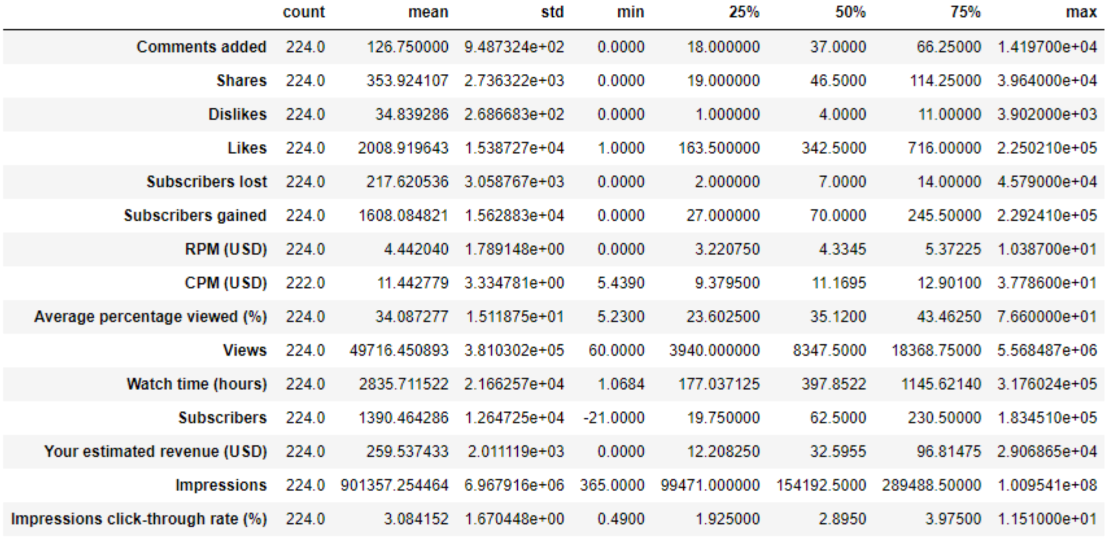


Figure 1: Correlation test matrix

Source: Generated by our python code

As shown in the above figure, correlation analysis is a powerful statistical tool used for the analysis of many different data across many different fields of study. Correlation matrices can help identify relationships among a great number of variables in a way that can be interpreted easily either numerically or visually. Correlation is a relationship of dependency between variables where a change in the observed value of one variable is reflected by a unit change in another. Correlations are used to develop statistical models in machine learning as well as more traditional methods such as multiple and simple linear regression. Correlation matrices are tables was used to detail the correlation coefficients between different variables in an easily visualized manner. A correlation matrix provides a sort of index where one can see how strong the correlation is among different variables. Correlation matrix to quantify and summarize linear relationships between variables. A correlation matrix is closely related to the covariance matrix. We can interpret the correlation matrix as being a re-scaled version of the covariance matrix. In fact, the correlation matrix is identical to a covariance matrix computed from standardized features. Correlation in this project was represented using a correlation matrix heat map. The correlation matrix provides us with another useful summary graphic that can help us to select features based on their respective linear correlations.

2.5 Feature Scaling



	count	mean	std	min	25%	50%	75%	max
Comments added	224.0	126.750000	9.487324e+02	0.0000	18.000000	37.0000	66.25000	1.419700e+04
Shares	224.0	353.924107	2.736322e+03	0.0000	19.000000	46.5000	114.25000	3.964000e+04
Dislikes	224.0	34.839286	2.686683e+02	0.0000	1.000000	4.0000	11.00000	3.902000e+03
Likes	224.0	2008.919643	1.538727e+04	1.0000	163.500000	342.5000	716.00000	2.250210e+05
Subscribers lost	224.0	217.620536	3.058767e+03	0.0000	2.000000	7.0000	14.00000	4.579000e+04
Subscribers gained	224.0	1608.084821	1.562883e+04	0.0000	27.000000	70.0000	245.50000	2.292410e+05
RPM (USD)	224.0	4.442040	1.789148e+00	0.0000	3.220750	4.3345	5.37225	1.038700e+01
CPM (USD)	222.0	11.442779	3.334781e+00	5.4390	9.379500	11.1695	12.90100	3.778600e+01
Average percentage viewed (%)	224.0	34.087277	1.511875e+01	5.2300	23.602500	35.1200	43.46250	7.660000e+01
Views	224.0	49716.450893	3.810302e+05	60.0000	3940.000000	8347.5000	18368.75000	5.568487e+06
Watch time (hours)	224.0	2835.711522	2.166257e+04	1.0684	177.037125	397.8522	1145.62140	3.176024e+05
Subscribers	224.0	1390.464286	1.264725e+04	-21.0000	19.750000	62.5000	230.50000	1.834510e+05
Your estimated revenue (USD)	224.0	259.537433	2.011119e+03	0.0000	12.208250	32.5955	96.81475	2.906865e+04
Impressions	224.0	901357.254464	6.967916e+06	365.0000	99471.000000	154192.5000	289488.50000	1.009541e+08
Impressions click-through rate (%)	224.0	3.084152	1.670448e+00	0.4900	1.925000	2.8950	3.97500	1.151000e+01

Figure 2: features statistical description

Source: Generated by our python code

The variables are dispersed widely, as seen in the figure that was just presented. As an illustration, the total Views numbers are enormous when compared to the RPM(USD) values. In machine learning models, a single variable with such a high value may dominate other values during training. For instance, while performing K-nearest neighbor KNN, the performance of the KNN model will be dominated by the data points with high distances if the nonuniform data is not standardized. Thus, before training any machine learning model, feature scaling is a crucial step that must be taken care of. There are numerous approaches to feature scaling. Standardization and normalizing have been the most widely utilized and well-liked methods in the machine-learning community. The best way hasn't been shown in practice, at least not theoretically. The dataset's features have been scaled via standardization.

3 Objective/Aim

Our primary goal was to evaluate the effectiveness of several machine learning algorithms for forecasting profit/revenue generated weight in a YouTube Analytics dataset extracted from the well-known Kaggle dataset collection of datasets. We investigated this using Logistic Regression and K-Nearest Neighbors as our machine-learning algorithms. In order to generate a binary target column that categorizes the weight of the estimated revenue as good(1) or bad(0), we used feature engineering on this dataset. To do this, we assigned 1 to any value bigger than or equal to 0.725 in the target column and 0 to every other value. This gave us a more appropriate and adaptable target column to use for categorizing. The Root Mean Squared Error (RMSE) and the Confusion Matrix were used to evaluate the performance of the entire model on the training and test data sets. The standard deviation of the residuals is known as RMSE (prediction errors). The RMSE is a measure of how to spread out the residuals, which are a measure of how distant the data points are from the regression line.

4 Data Partitioning

Avoiding overfitting is the basic goal of splitting/partitioning the data. The machine learning algorithm may perform well in the training dataset if overfitting occurs, but poorly in the testing dataset. For training and testing purposes of our model, we should have our data broken down into three distinct dataset splits:

- **Training Set(80% of dataset):** It is the set of data that is used to train and make the model learn the hidden features/patterns in the data. n each epoch, the same training data is fed to the neural network architecture repeatedly, and the model continues to learn the features of the data. The training set should have a diversified set of inputs so that the model is trained in all scenarios and can predict any unseen data sample that may appear in the future.
- **Testing Set(20% of dataset):** The test set is a separate set of data used to test the model after completing the training. It provides an unbiased final model performance metric in terms of accuracy, precision, etc. To put it simply, it answers the question of "How well does the model perform?".
- **Validation Set(25% of dataset):** The validation set is a set of data, separate from the training set, that is used to validate our model performance during training. This validation process gives information that helps us tune the model's hyperparameters and configurations accordingly. t is like a critic telling us whether the training is moving in the right direction or not. The model is trained on the training set, and, simultaneously, the model evaluation is performed on the validation set after every epoch. The main idea of splitting the dataset into a validation set is to prevent our model from over fitting i.e., the model becomes good at classifying the samples in the training set but cannot generalize and make accurate classifications on the data it has not seen before.

5 Feature Selection Techniques Implemented

Choosing a subset of the dataset's most important traits to describe the target variable is the process of feature selection. In terms of computation time, generalization performance, and interpretational issues, it improves the efficiency of machine learning challenges [6]. The categories utilized to classify them are filter-based, wrapper-based, and embedding forms of feature selection techniques. Based on preset requirements, filter-based techniques eliminate features. Wrapper-based techniques use a modeling procedure that is viewed as a "black box" to evaluate and rank features [6]. The embedded approaches use feature selection methods including least absolute shrinkage and selection operator (LASSO) and random forest (RF) feature selection. In this project we utilized the following techniques in the following way:

- **Sequential Forward Feature Selection:** Shrinkage of an initial d-dimensional feature space into a k-dimensional feature subspace is accomplished using the family of greedy search algorithms, which includes sequential feature selection techniques. The objective is to select the subset of features that are most relevant to the job at hand, maximizing computing efficiency and minimizing generalization error by removing unnecessary information (that acts as noise). The sequential forward selection technique entails performing the following procedures to identify the N features that will fit in the K-features subset and are the most suitable among all of them.
  - The best feature among all the features is first and foremost chosen.
  - Then, by pairing this top feature with one of the other traits, the ideal pair of features is determined.
  - The two top features are then combined with one of the remaining features to create the optimal feature triplet.
  - Up until K features—the predetermined number of features—are selected, this process is repeated.
- **Overall Regularized Trees with Overall Feature Importance:** The term "feature significance" describes methods that rate input features according to how well they are able to predict a given target variable. In a predictive modeling project, feature importance scores are crucial because they offer insight into the data, insight into the model, and the foundation for dimensionality reduction and feature selection, which can increase a predictive model's efficiency and effectiveness in solving the problem.

6 Machine Learning Algorithms Implementation and Methodology

In this project, machine learning approaches are employed to forecast estimated revenue weight as well as to establish how dependent estimated revenue weight is on other variables. This section provides explanations of the suggested methodology. The dependence of the predicted revenue weight on the other 19 independent variables is also determined using logistic regression (predictors). The selection of significant predictors is then made based on the dependence of the predicted revenue weight on the independent variables. Finally, expected revenue weight is forecasted using KNN and logistic regression, taking into account both all and chosen predictors. The following subsections outline how the various machine-learning approaches function.

6.1 logistic regression Implementation

Logistic Regression is a supervised learning algorithm that is used when the target variable is categorical. Hypothetical function h(x) of linear regression predicts unbounded values. But in the case of Logistic Regression, where the target variable is categorical we have to strict the range of predicted values. Consider a classification problem, where we need to classify whether an email is spam or not. So, the hypothetical function of linear regression could not be used here to predict as it predicts unbound values, but we have to predict either 0 or 1. We implemented logistic regression in the following way:

- Separate the data into training and test sets.
- Sklearn is used to fit a logistic regression model.
- Make a prediction using the model and the test data.

- Utilize the confusion matrix to assess the model's accuracy.

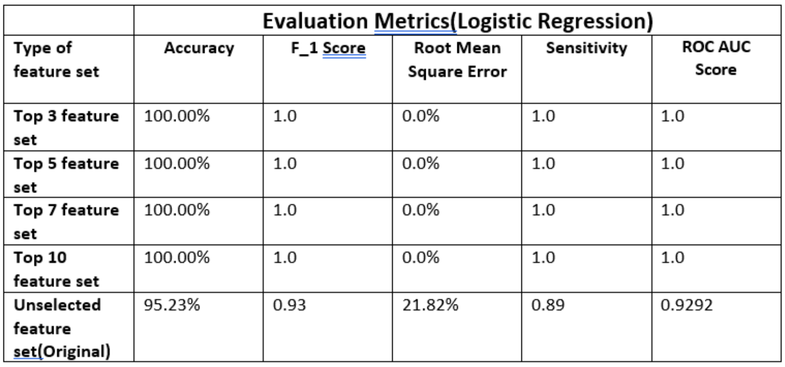
To reduce the mistake, we use a cost function. Every time the learning rate step is taken in the for loop, the error will decrease. and the weight and bias will likewise change with each iteration. This fit function will assist in determining the data's ideal weight and bias for our model. In this regard, we discovered that 0.03, or when we have the most ideal weights, is the best learning rate.

6.2 K-Nearest Neighbours Implementation

A straightforward technique called K-nearest neighbors categorizes new cases based on how similar to existing cases they are. A case is allocated to the class that is most prevalent among its K closest neighbors as determined by a distance function by a majority vote of those neighbors. The distance function can be either hamming, standard, Euclidean, Manhattan, or Minkowski. The value of K, the sole hyperparameter for this algorithm, defines the number of samples that will be considered the nearest neighbors. Finding a plot between the error on the test set and K indicating values is one method of determining the best value of K. We then select the K value that has the lowest error rate. We discovered that k=2 gave us the lowest error rate in this area.

7 Experimental results and further discussion

There are a total of 19 variables. The target variable weight rating is considered the dependent variable and the other 18 variables are assumed as predictors or independent variables in this work. one type of analysis approach is followed in this report: The value of the estimated revenue's weight is predicted using predictors.

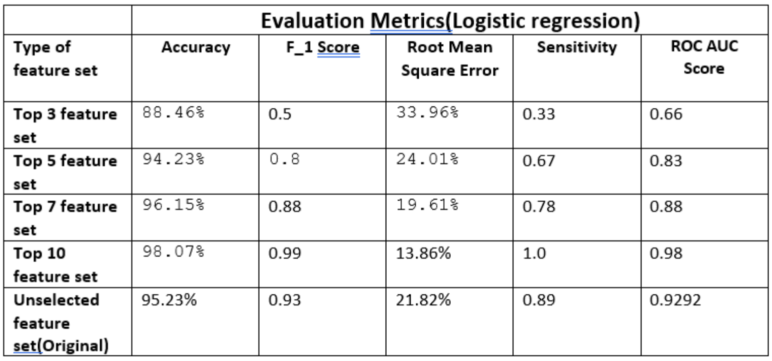


	Evaluation Metrics(Logistic Regression)				
Type of feature set	Accuracy	F_1 Score	Root Mean Square Error	Sensitivity	ROC AUC Score
Top 3 feature set	100.00%	1.0	0.0%	1.0	1.0
Top 5 feature set	100.00%	1.0	0.0%	1.0	1.0
Top 7 feature set	100.00%	1.0	0.0%	1.0	1.0
Top 10 feature set	100.00%	1.0	0.0%	1.0	1.0
Unselected feature set(Original)	95.23%	0.93	21.82%	0.89	0.9292

Figure 3: Results obtained from feature set selected by forward selection and the original set, on Logistic Regression

Source: Generated by our python code and compiled

As shown in the above figure, we obtained unrealistic results when training the model on selected feature sets.

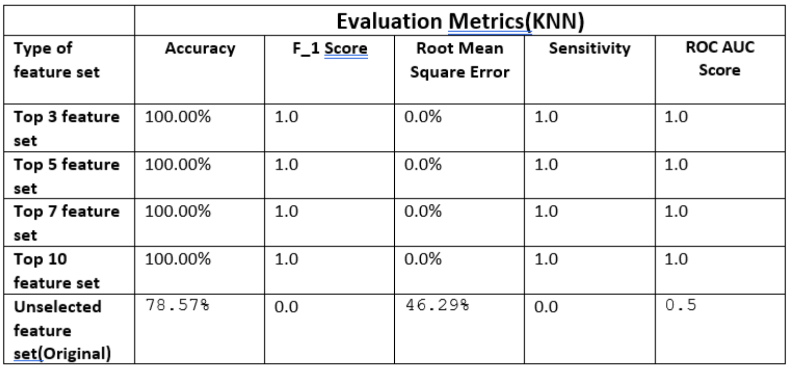


	Evaluation Metrics(Logistic regression)				
Type of feature set	Accuracy	F_1 Score	Root Mean Square Error	Sensitivity	ROC AUC Score
Top 3 feature set	88.46%	0.5	33.96%	0.33	0.66
Top 5 feature set	94.23%	0.8	24.01%	0.67	0.83
Top 7 feature set	96.15%	0.88	19.61%	0.78	0.88
Top 10 feature set	98.07%	0.99	13.86%	1.0	0.98
Unselected feature set(Original)	95.23%	0.93	21.82%	0.89	0.9292

Figure 4: Results obtained from the feature set selected by Overall Regularized Trees with Overall Feature Importance and the original set, on Logistic Regression

Source: Generated by our python code and compiled

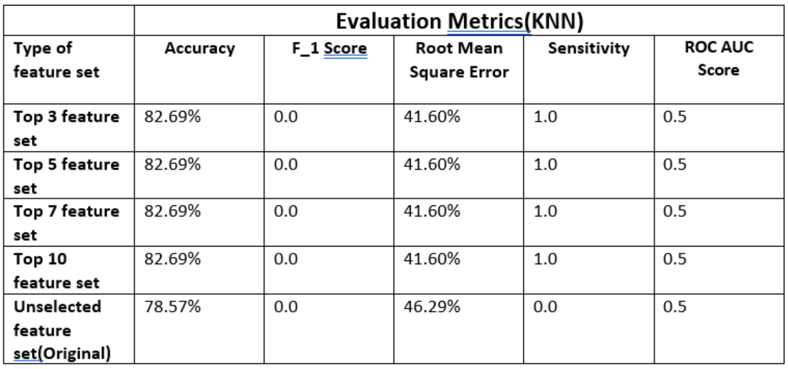
With results obtained from training the logistic regression model on features selected by the Overall Regularized Trees with Overall Feature Importance, they were very realistic, and in fact they did show that the more features that are selected, the more the accuracy of the model will increase and become better.



	Evaluation Metrics(KNN)				
Type of feature set	Accuracy	F_1 Score	Root Mean Square Error	Sensitivity	ROC AUC Score
Top 3 feature set	100.00%	1.0	0.0%	1.0	1.0
Top 5 feature set	100.00%	1.0	0.0%	1.0	1.0
Top 7 feature set	100.00%	1.0	0.0%	1.0	1.0
Top 10 feature set	100.00%	1.0	0.0%	1.0	1.0
Unselected feature set(Original)	78.57%	0.0	46.29%	0.0	0.5

Figure 5: Results obtained from feature set selected by forward selection and the original set, on KNN

Source: Generated by our python code and compiled



	Evaluation Metrics(KNN)				
Type of feature set	Accuracy	F_1 Score	Root Mean Square Error	Sensitivity	ROC AUC Score
Top 3 feature set	82.69%	0.0	41.60%	1.0	0.5
Top 5 feature set	82.69%	0.0	41.60%	1.0	0.5
Top 7 feature set	82.69%	0.0	41.60%	1.0	0.5
Top 10 feature set	82.69%	0.0	41.60%	1.0	0.5
Unselected feature set(Original)	78.57%	0.0	46.29%	0.0	0.5

Figure 6: Results obtained from the feature set selected by Overall Regularized Trees with Overall Feature Importance and the original set, on KNN

Source: Generated by our python code and compiled

The Logistic Regression Algorithm was used for both test modes, and it produced the best classification results of the estimated revenue samples. With this algorithm, the accuracy of the cross-validation and percentage split modes is 97.11% and 95.66%, respectively. The aforementioned graphs unmistakably demonstrate that the Logistic Regression algorithm outperforms the other algorithms. The accuracy percentage of successfully classifying the instances into each class according to weight characteristics serves as the benchmark for the classification experiment for the weight of the predicted revenue samples. In our investigation, the predicted revenue samples were divided into categories using the two classification algorithms. K-cross validation and an 80% percentage split are additional methods used to assess the classification outcomes produced by the two classification algorithms. Additionally, a few of the common performance metrics (statistics) are computed to assess how well the algorithms work. Recall, accuracy, the F1-measure, and ROC values are the common performance metrics. The tables above show the occurrences that were appropriately categorized as a consequence of classifying the estimated revenue samples.

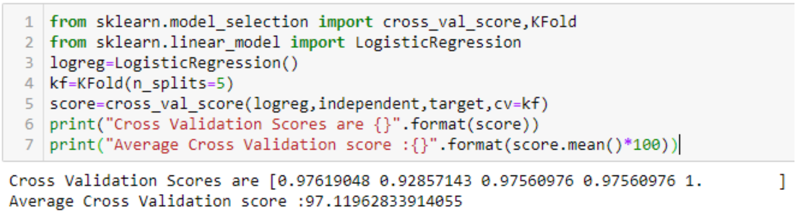
7.1 Cross-Validation Technique Implemeted

A crucial tool in the Data Scientist's toolbox is cross-validation. It enables us to make better use of our data. We frequently divide our data into training and validation/test sets while developing a machine learning model utilizing certain data. The validation/test set is used to validate the model with data that it has never seen before, whereas the training set is used to train the model. The traditional method is a straightforward 80%–20% split, occasionally with other values like 70%–30% or 90%–10%. We perform many splits during cross-validation. We can split into 3, 5, 10, or any other K number. These splits are known as folds, and there are numerous ways we can use to make them. With that being said, in this project, we will utilize the following way of performing cross-validation in our data set.

7.1.1 K-Fold Cross-Validation

The entire dataset is divided into K equal-sized pieces using the K-Fold cross-validation procedure. Each division is referred to as a "Fold." We refer to it as K-Folds because there are K pieces. The remaining K-1 folds are utilized as the training set, while One Fold is used as a validation set.

Until each fold is employed as a validation set and the remaining folds are the training set, the procedure is repeated K times. The mean accuracy of the k-models validation data is used to calculate the model's final accuracy. We used the whole dataset as both a training set and a validation set. We obtained the following results from the cross-validation we performed in our dataset:



```python
from sklearn.model_selection import cross_val_score,KFold
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
kf=KFold(n_splits=5)
score=cross_val_score(logreg,independent,target,cv=kf)
print("Cross Validation Scores are {}".format(score))
print("Average Cross Validation score :{}".format(score.mean()*100))
```

```
Cross Validation Scores are [0.97619048 0.92857143 0.97560976 0.97560976 1.        ]
Average Cross Validation score :97.11962833914055
```

Figure 7: Snippet of our K-Cross Validation code when k=5 folds

Source: Generated by our python code

The KFold() sklearn class was used. The k-fold cross-validation implementation in sklearn is provided in the above snippet. The validation was done in the overall data set. We obtained an average cross-validation score of 97.11%, which is a good evaluator to the models that we used our splitter, selected, and unselected data to train. The models we train achieved a reasonably good accuracy, with the logistic regression model achieving 95.23% on the unselected(original) split dataset.

8 Conclusion

We examined how the results vary for each type of model when the check mode is altered. The evaluation of classifiers on each set of estimated revenue figures is the observation. Following the application of the k-cross-validation, the results are defined in terms of the percent of effectively labeled instances, precision, recall, F measure, and ROC. On the same datasets, various classifiers including Logistic Regression and K-Nearest Neighbor are assessed. The trials' findings support our conclusion that the Logistic Regression Algorithm performs better in type projects when compared to the assistance of the K-Nearest Neighbor Algorithm. Additionally, we used cross-Validation, a highly potent tool, to help us make better use of our data and to provide us with a lot more details on the performance of our model.

References

[1] Dasu, T. and Johnson, T., 2003. Exploratory data mining and data cleaning. John Wiley Sons.

[2] Gudivada, V., Apon, A. and Ding, J., 2017. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. International Journal on Advances in Software, 10(1), pp.1-20.

[3] Bakker, M. and Wicherts, J.M., 2014. Outlier removal, sum scores, and the inflation of the Type I error rate in independent samples t tests: the power of alternatives and recommendations. Psychological methods, 19(3), p.409.

[4] Seltman, H.J., 2012. Experimental design and analysis.

[5] Stahl, S., 2006. The evolution of the normal distribution. Mathematics magazine, 79(2), pp.96-113.

[6] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J. and Liu, H., 2017. Feature selection: A data perspective. ACM computing surveys (CSUR), 50(6), pp.1-45.