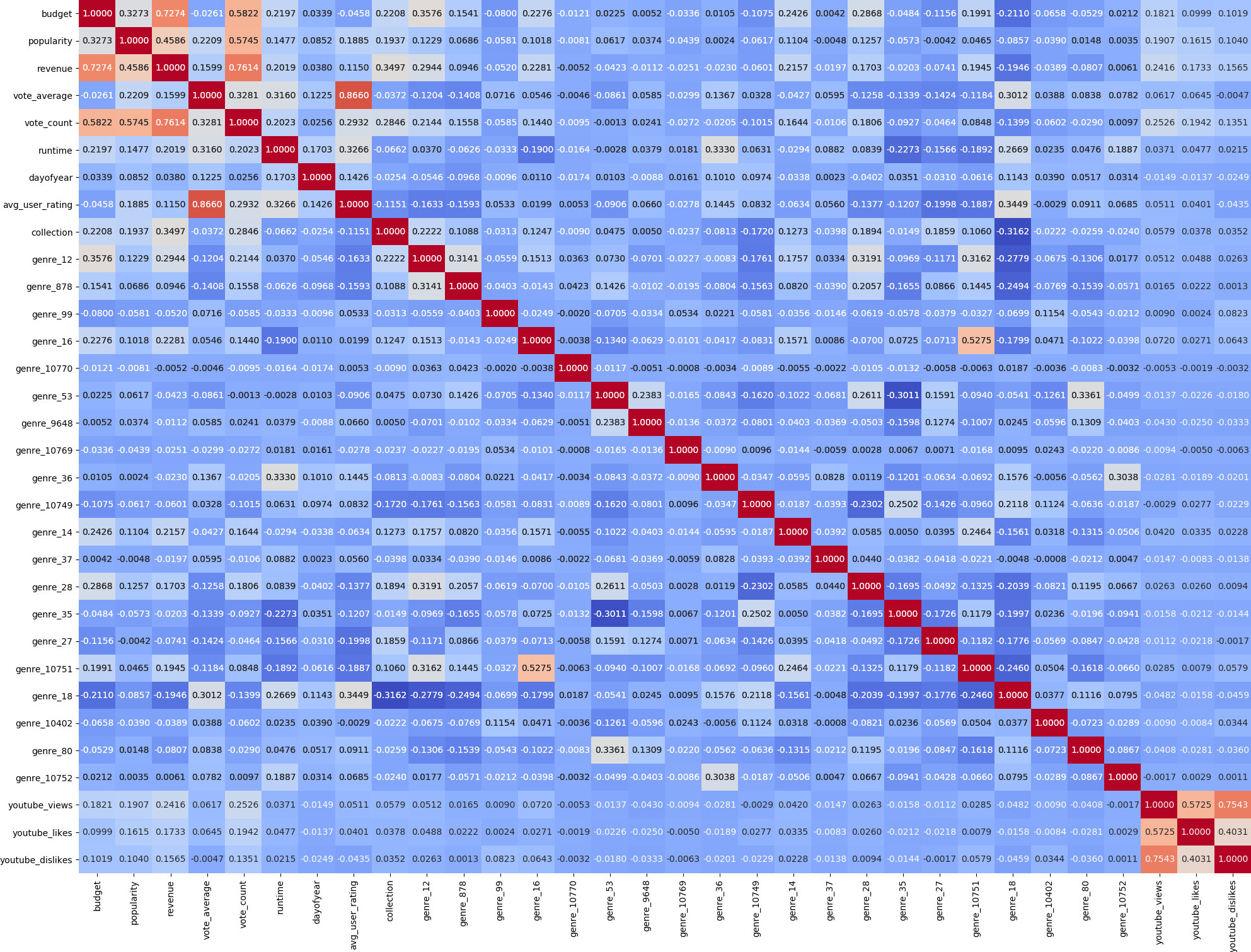
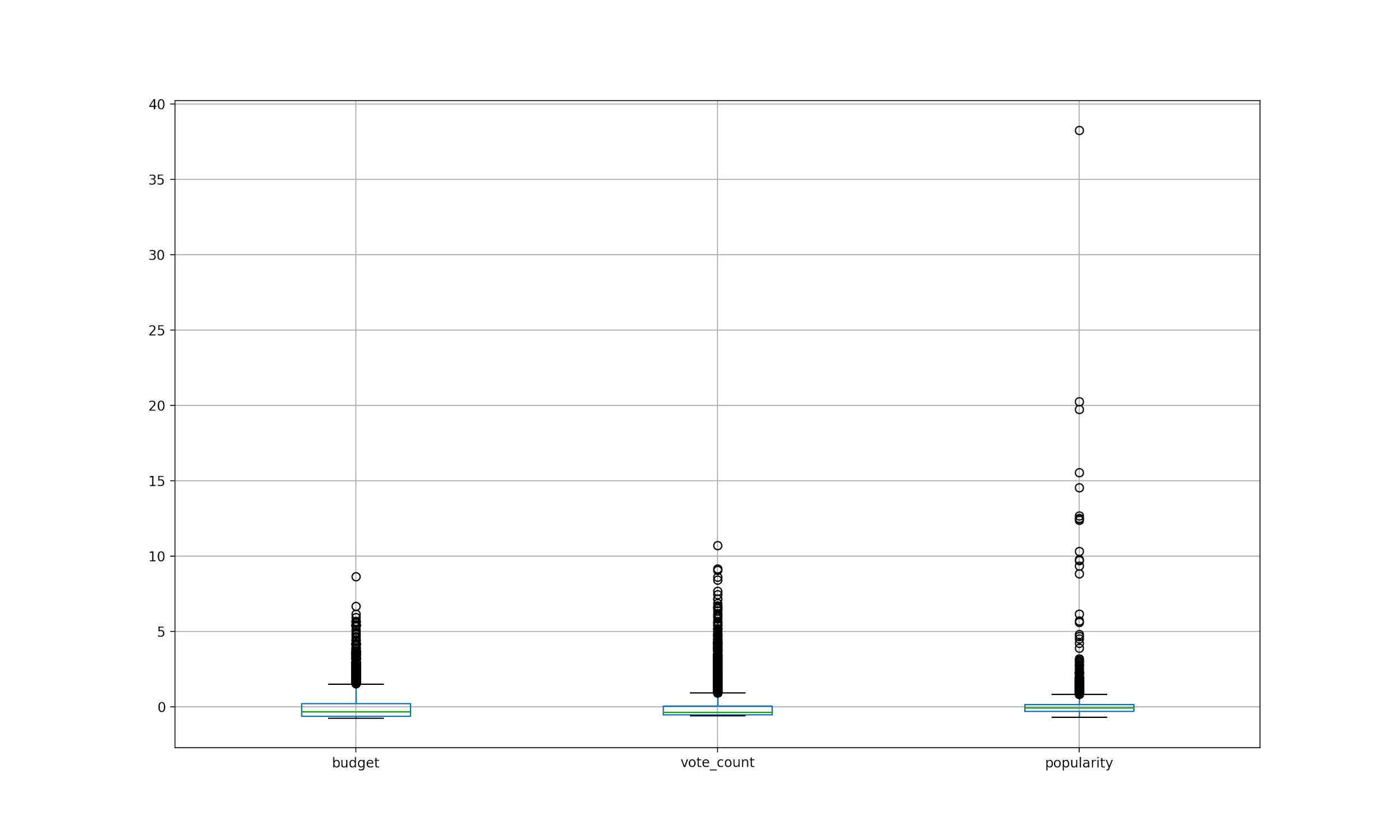
Feature Selection:

1. Correlation Matrix:



A heatmap of correlation matrix of the input features is formed. From the graph we can observe that revenue has a high correlation with budget, popularity, vote\_count and collection. There is some correlation between revenue and youtube data as well as some genres but we will be removing those features to reduce dimensionality of input space.

1. Outlier Detection and Removal:



Budget, vote\_count and popularity were on different scales so we used StandardScaler to normalize data and bring each column on the same scale. Box Plots are drawn to check for outliers. We can observe that there are some outliers in data which need to be removed before feeding into the machine learning pipeline.   
We used Interquartile Range to remove outliers.

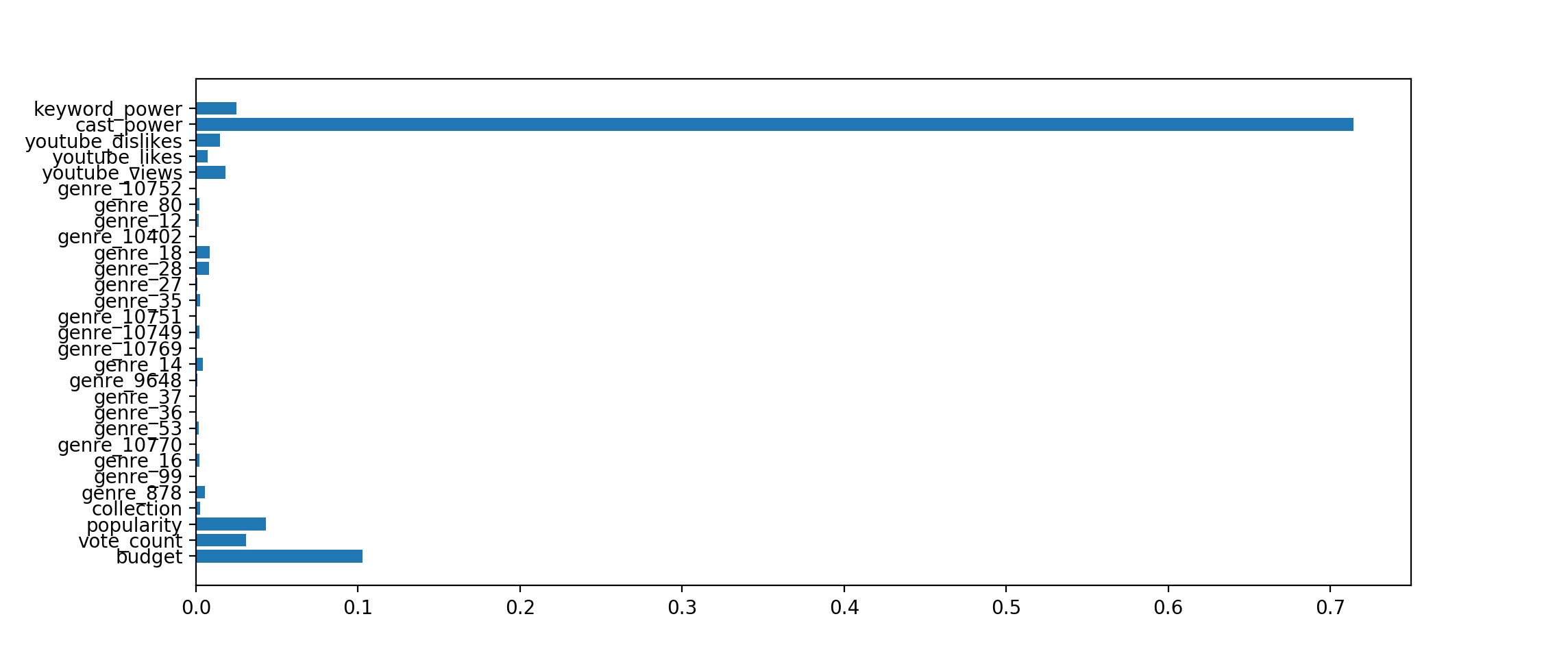
Q1 = 25th Percentile of Data

Q3 = 75th Percentile of Data

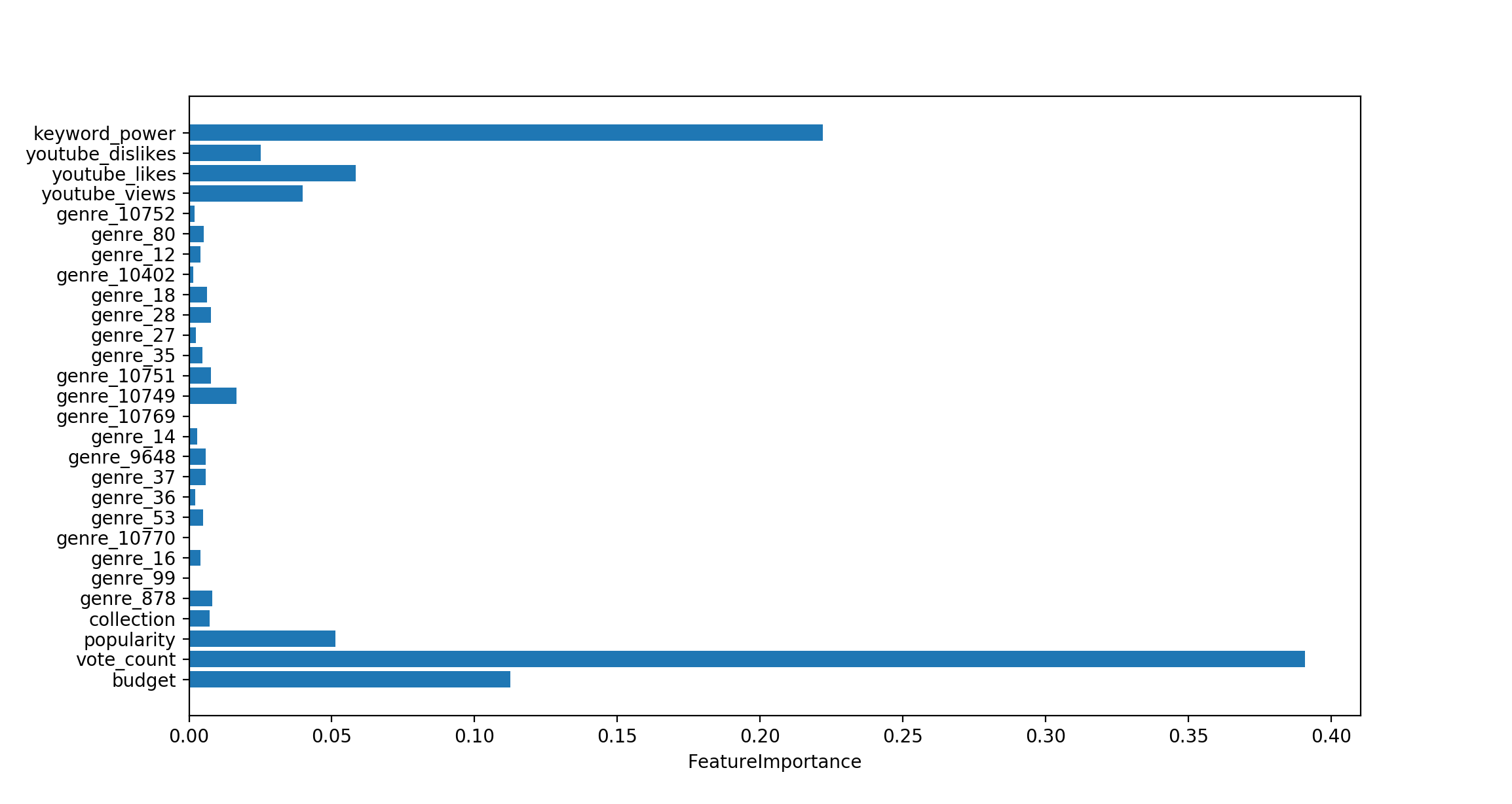
IQR = Q3 - Q1

We filtered data between Q1 - 1.5\*IQR and Q3 + 1.5\*IQR, for budget, vote\_count and popularity.

1. Feature Engineering:



Making a chart for featureImportance in GBT regression, we can see that keyword\_power, cast\_power, budget, popularity, vote\_count, youtube\_view are most important. But because a lot of input rows do not have cast\_power, we decided to exclude it.

Removing cast power and making a horizontal bar graph we get the above figure. So we selected keyword\_power, youtube\_views, youtube\_likes, popularity, vote\_count and budget to reduce dimensionality and overfitting. We used SQLTransformer to select required features, VectorAssembler to assemble features, StandardScaler to make all features on the same scale.

1. Modelling:

After feature engineering, the next step is to model. We used multiple algorithms available in Spark ML library like Linear Regressor, GBTRegressor, RandomForest Regressor and Decision Tree. To hypertune parameters for each, we used TrainValidationSplit along with ParamGridBuilder in which different hyperparameters were defined to be tuned. Through use of these, we reduced overfitting and identified best tuned models.

In order to get the best model, we compared different best tuned models on r2 score. Best model identified is saved for prediciton.

1. Evaluation:

Final model is evaluated on R2 score and below R2 scores are calculated.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train | Validation | Test |
| R2 | 0.991673 | 0.76530 | 0.97831 |

We observed that overfitting is a problem which needs to be addressed. So next plan is to use regularization to reduce overfitting.