Ranking Social Media News Feed Posts

The dataset I was working with that included tweets from Twitter users, along with information about how users have interacted with those tweets. The dataset contained 14 following columns: 'Keywords_relevance', 'Hashtags_relevance', 'Mentions_relevance', 'Interaction_rate', 'Mention_count', 'Followers_Followings','Seniority', 'Listed_count', 'Length', 'Hashtags', 'URL', 'Multimedia', 'Popularity', 'Relevance'. The goal of the project was to predict whether a user will find a particular tweet relevant in the future.

EDA.

The shape of the dataset is (26180 rows, 14 columns). No missing values were observed. All data types were float64. A statistical summary of the numerical features also was performed. Features' visualizations start with distribution of Relevance that has the focus on how the classes in the target variable are distributed, which is crucial for understanding the balance or imbalance in the dataset. Next are the boxplots of all features except 'Relevance'. They are useful for visualizing the distribution of data and detecting outliers, shown by points beyond the whiskers of the boxplot.

Histograms for each feature segmented by the 'Relevance' category are given. This allows for a comparison of how the distribution of each feature varies between relevant and not relevant instances. The histograms are overlaid to facilitate direct comparison, and a legend is included to differentiate between the two categories. Other types of Histograms used were histograms with a log scale on the y-axis that made it easier to see the distribution of data across orders of magnitude; and overlapping histograms with log scale that highlighted the differences in the distribution of feature values between the two categories of 'Relevance'.

The heatmap reveals the degree to which each pair of features is correlated. Features with high positive or negative correlation may provide redundant information to a model. If the heatmap shows many features with high correlation, this could imply that dimensionality reduction techniques like Principal Component Analysis might be beneficial. However, as was seen from mid-term report, PCA analysis was useless. So, this version of project doesn't contain it anymore.

The pair plot results show the relationships between the selected features and how they vary with the target variable. If there were clear patterns or separations between relevant and not relevant points in any of the scatter plots, those feature pairs might have been good predictors for classifying relevance.

Final piece of the visualizations were the outliers. Boxplots provided a visual summary of the distribution of the data including median, quartiles, and outliers for the features with outliers.

Feature analysis.

For the features where the ANOVA test was successful, the p-values are all 0.0 (or very close to 0, which might be rounded). This indicates that there are statistically significant differences in the means of these features between different 'Relevance' groups.

Based on mutual information scores, "**Listed_count**" has the highest mutual information score, suggesting it is the most informative feature regarding the target variable.

"**Interaction_rate"** and "**Followers_Followings"** also have relatively high scores, indicating a strong dependency with the target variable. Some features like "**Hashtags"** have a score of 0, which suggests they do not provide any information about the target variable in the context of mutual information.

The boxplots for **Listed_count** show that tweets deemed relevant tend to have a higher number of lists they are a part of, indicating that users who are listed more often may produce more relevant tweets. The **Interaction_rate** boxplot shows a more pronounced difference between relevant and not relevant tweets, with relevant tweets having a higher median interaction rate. This suggests that tweets with higher engagement are more likely to be relevant.

The statistics for **Listed_count** indicate that while the mean is higher for non-relevant tweets, the median is higher for relevant tweets, which suggests that the mean might be influenced by outliers. On the other hand, the statistics for **Interaction_rate** suggest a clearer trend where relevant tweets have higher interaction rates on average and a higher median.

Polynomial features for **Interaction_rate** and an interaction feature between **Listed_count** and **Interaction_rate** were created. All engineered features were normalized using StandardScaler to ensure they're on the same scale, which is particularly important for models that are sensitive to the scale of the input features, such as SVM or k-NN.

Recursive Feature Elimination (RFE) method with a random forest classifier was used to select the most important features. All features received a ranking of 1 in the RFECV process. It suggests that the model performance was best when all these features were included. This means each feature adds value to the model, and removing any of them would degrade the model's performance according to the metric used (accuracy).

Model Building.

Given the nature of the problem (classification), I considered the following algorithms for candidate models:

- Logistic Regression: A simple yet effective baseline for binary classification problems.
- Random Forest: An ensemble method that can capture complex relationships in the data.
- Gradient Boosting: Another powerful ensemble method known for its good performance on a variety of tasks.
- Support Vector Machine (SVM): Effective in high-dimensional spaces, even in cases where the number of dimensions exceeds the number of samples.

Results from the initial round of model training and evaluation:

Logistic Regression:
Accuracy: 76.59%
Precision (Relevant): 72%
Recall (Relevant): 60%
F1-Score (Relevant): 65%

Random Forest:
Accuracy: 78.74%
Precision (Relevant): 73%
Recall (Relevant): 67%
F1-Score (Relevant): 70%

Gradient Boosting:
Accuracy: 78.97%
Precision (Relevant): 73%
Recall (Relevant): 68%
F1-Score (Relevant): 70%
SVM:
Accuracy: 76.74%
Precision (Relevant): 71%
Recall (Relevant): 62%
F1-Score (Relevant): 66%

The Gradient Boosting model has the highest accuracy among the models tested, making it the best performer in this set. It's followed closely by the Random Forest model. Both of these models are ensemble methods, which typically perform well on a variety of classification tasks because they can capture complex relationships in the data and are less likely to overfit compared to individual decision trees. Among other three models trained (XGBoost, NB, KNN) the XGBoost showed the accuracy of approximately 78.84%. This is a competitive result, slightly below the accuracy of the Gradient Boosting model but on par with the Random Forest model.

Model Comparison.

Next step of the model building was the hyperparameter tuning using the **GridSearchCV** which performs an exhaustive search over the specified parameter grids.
GB = ({'learning_rate': 0.05, 'max_depth': 6, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 300, 'subsample': 0.9}, 0.7988924132071155).
XGB = ({'colsample_bytree': 1, 'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 200}, 0.7964572935503772).
RF = {'bootstrap': True, 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 200}, 0.7967437991346237).
The GB model has achieved slightly higher accuracy than RF in the tuning process, making it the leading model according to these results. The XGB parameters are also promising.

With these optimized hyperparameters, retraining of each model on the full training dataset was performed and their performance was evaluated on the test set to get a clear understanding of how well they might perform in practice. The comparison is fair since they've all been tuned.

Gradient Boosting Classification Report:

- Accuracy: 79.01%
- Precision: 82.15% (Not Relevant), 73.01% (Relevant)
- Recall: 85.35% (Not Relevant), 68.12% (Relevant)
- F1-Score: 83.72% (Not Relevant), 70.48% (Relevant)

XGBoost Classification Report:

- Accuracy: 78.91%
- Precision: 81.95% (Not Relevant), 73.04% (Relevant)
- Recall: 85.47% (Not Relevant), 67.65% (Relevant)
- F1-Score: 83.67% (Not Relevant), 70.24% (Relevant)

Random Forest Classification Report:

- Accuracy: 79.22%
- Precision: 81.91% (Not Relevant), 73.89% (Relevant)
- Recall: 86.16% (Not Relevant), 67.29% (Relevant)
- F1-Score: 83.98% (Not Relevant), 70.43% (Relevant)

Looking at the accuracy, all models are performing similarly, with Random Forest marginally outperforming the others post hyperparameter tuning. However, it's essential to look beyond accuracy, especially as I have an imbalanced dataset.

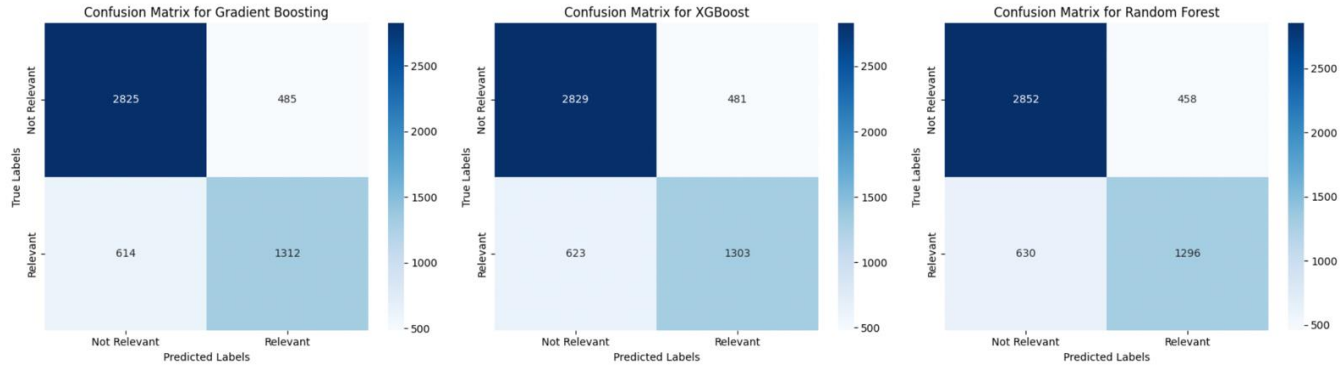
The F1-score is a more informative metric as it balances precision and recall, and it's particularly useful if the cost of false positives and false negatives is high. Random Forest has the highest F1-score for the 'Not Relevant' class, while Gradient Boosting leads slightly in the 'Relevant' class.

In conclusion, while Random Forest has a slight edge in overall accuracy and F1-score for the 'Not Relevant' class, Gradient Boosting has the best performance for the 'Relevant' class.

Feature importance.

Feature importances derived from the Random Forest model showed that most three important features are **Interaction_rate** (31.47%), **Popularity** (18.04%), **Listed_count** (10.46%). **Interaction_rate**, **Popularity,** and **Listed_count** are the most important features when it comes to predicting the relevance of a tweet according to the Random Forest model. These three features alone account for almost 60% of the importance in the model's decisions.

Model Performance (Confusion Matrix Analysis).



Model Validation.

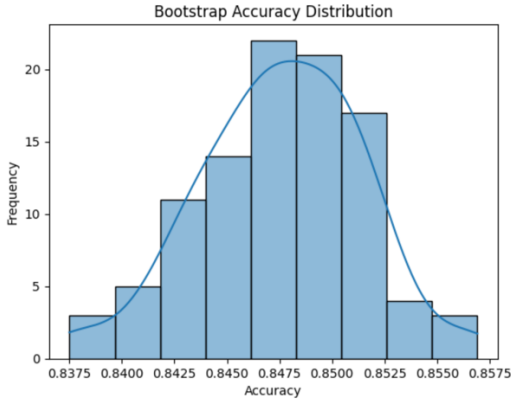
The cross-validation results for the Gradient Boosting, XGBoost, and Random Forest models across 5 different folds of the data are as follows:

- **Gradient Boosting** has cross-validated accuracy scores ranging from approximately 75.90% to 78.13%.
- **XGBoost** shows similar variation with scores from about 77.04% to 78.76%.
- **Random Forest** has scores ranging from roughly 76.28% to 78.80%.

These results suggest that the models are relatively stable across different subsamples of the data, as indicated by the consistency of the accuracy scores across the folds. Random Forest seems to show a slightly higher variance in its scores.

With stratified cross-validation, we ensure that the proportion of the classes is consistent across all folds, which is particularly important for datasets with imbalanced classes. The results indicate that all three models have a similar level of robustness, with none significantly outperforming the others across the cross-validation folds.

The bootstrap validation results indicating accuracy ranging from approximately 83.9% to 85.50% are quite promising. This suggests that the Random Forest model is fairly stable and performs consistently across different samples of the dataset.



95.0% confidence interval 0.839 and 0.855

The accuracy of the stacking model is approximately 79.26%. When compared to the bootstrap validation results for the Random Forest model, which showed an accuracy from about 83.90% to 85.50%, the stacking model doesn't seem to outperform the Random Forest.