Predicting Online News Popularity

Jose Padilla, Nolan Peters, Tysir Shehadey

Shiley-Marcos School of Engineering, University of San Diego

ADS 505: Applied Data Science for Business

Professor Jules Malin

October 14, 2024

Problem Statement and Justification for the Proposed Approach

Since the internet's rise in popularity, online media and news outlets have struggled to replicate the reliable revenue streams generated by print and cable media. In contrast to these traditional models, internet news media is often driven by site visits with free content for the user. Although this model has proven successful financially, there are risks to maintain journalistic integrity wherein financial stakeholders prioritize site visits, captured by sensationalized headlines, controversy, and polarization of its audience. With recognition of the state of the industry, the purpose of this project is to increase Mashable's article popularity (shares), while promoting transparency and consistency of journalistic ethics. To accomplish this, we predict shares based on article parameters and metadata using a machine learning model. Then, we identify the article traits that are manipulable by the author that have the greatest impact on popularity through feature analysis. With the support of the proposed model, Mashable will strengthen its journalistic voice, better surface important stories, and increase ad and subscription revenue.

Methodology

Exploratory Data Analysis

Our preliminary analysis of Mashable article content and metadata reveals some interesting insights and relationships between features and the target variable, shares. First, the distribution of shares is right skewed by viral articles with hundreds of thousands of shares. This is discussed further in the outliers section. Next, we analyze shares by day of the week. Figure 1, which shows the distribution of total shares by day of the week, reveals that the median share count, and overall distribution, is higher for articles that are published on either Saturday or Sunday, when compared to the weekday observations. This can be taken into consideration as a

potential focus for when to publish a given article. Next, we analyze the distribution of shares by data channel, or article topic. Figure 2 shows that articles in the *World* category perform the worst in terms of shares with the lowest median share count and with the smallest variance. Articles in the *Social Media* and *Lifestyle* tend to perform the best, but also have the highest variance in their share counts. While this may not be immediately actionable by Mashable, extracting preference amongst their readership for articles can support resource allocation and content decisions in the future.

Figure 1

Distribution of Shares by Data Channel

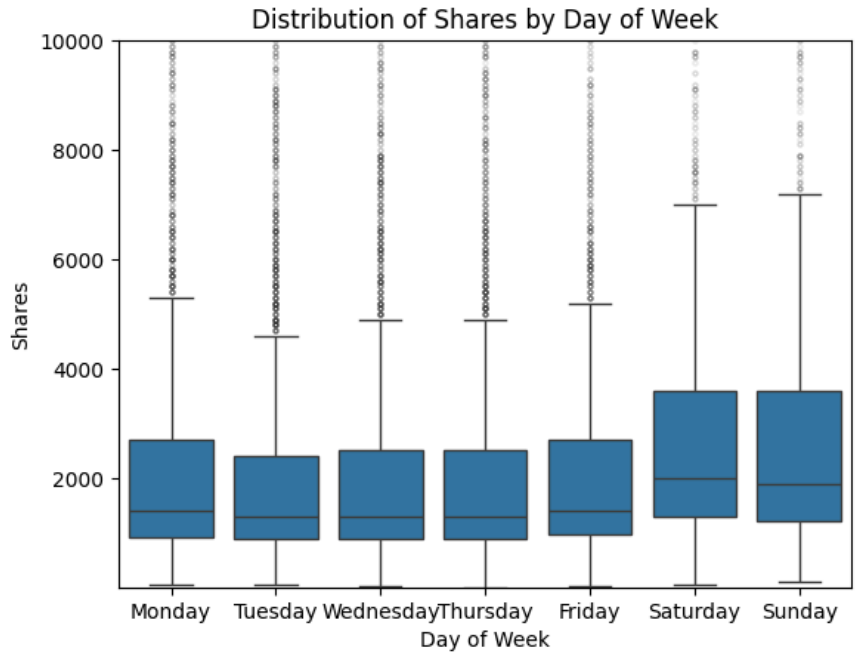
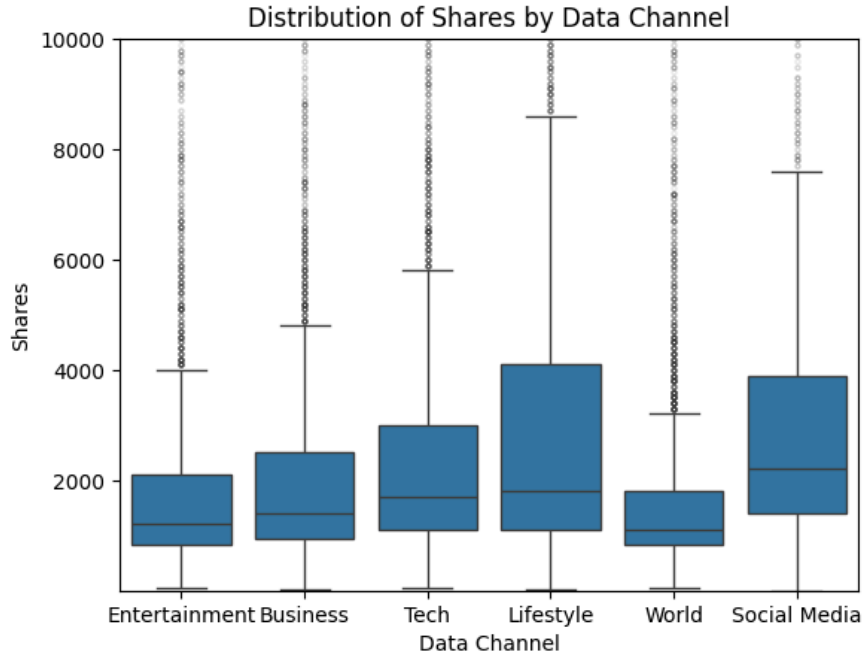


Figure 2

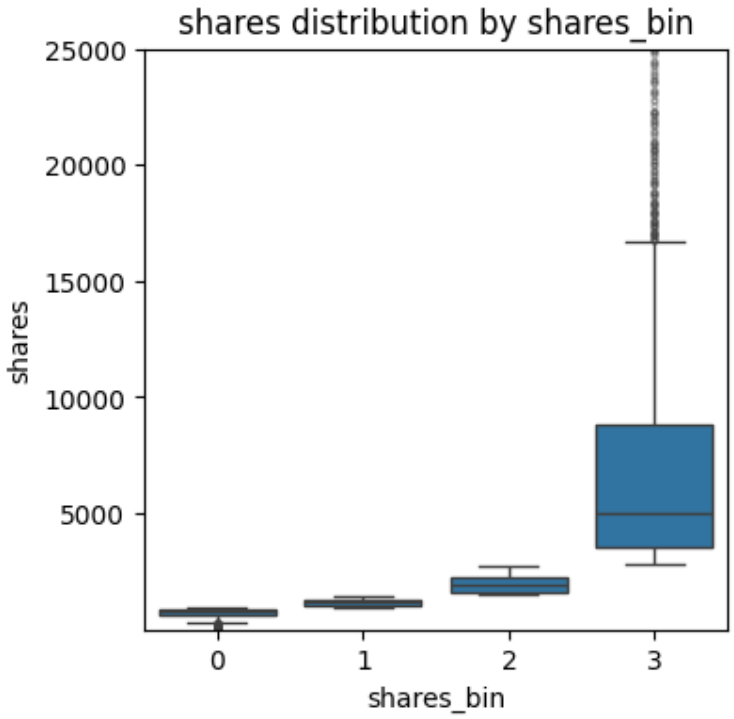
Distribution of Shares by Data Channel



Next, to further analyze the relationship between shares and other continuous variables, we binned shares based on quantiles. The shares in each group are shown in Figure 3. These 4 groups capture the virality stages or performance of an article where group 3 contains the most popular and viral articles and group 0 contains articles with shares from 5 to 942. These categories are used to further explore relationships with the metadata dataset.

Figure 3

Shares Distribution by Bin

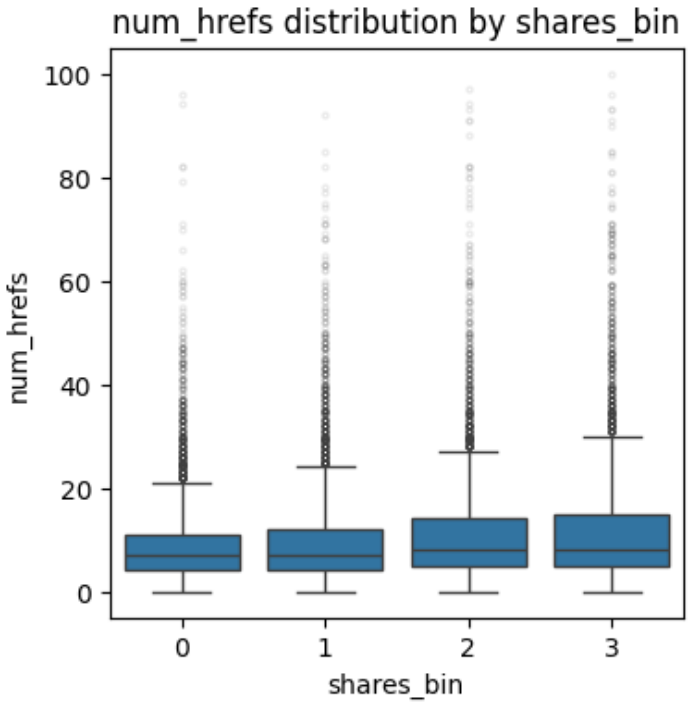


**ylim = 25,000 for improved visibility*

Continuing the analysis of the shared groupings, Figure 4 reveals that Group 2 and Group 3 skew slightly higher in terms of the number of links in the article. Mashable should monitor this trend and consider working additional links into content to improve shareability.

Figure 4

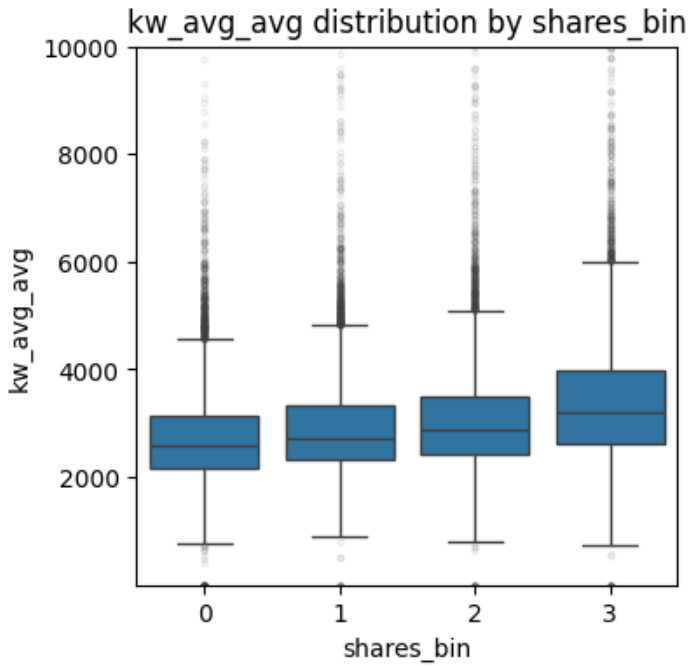
Number of Links Distribution by Bin



Next, we look at the distribution of average keyword shares in Figure 5. This reveals that the highest bin skews higher in average keyword share counts. This relationship will be explored further in the correlation analysis section, but worth noting the impact and relationship that average keywords have on shares.

Figure 5

Number of Shares per Average Keyword Distribution by Bin



Finally, we analyze the correlations between the continuous variables in the dataset. Figure 6, shown below, is a heatmap representation of these correlations (all relevant variables are included in a single visual for efficiency), highlighting the most prominent relationships. There are a few expected correlations between related variables (e.g. num_hrefs and num_self_hrefs), these are ignored in this portion of the analysis, but taken into consideration when selecting variables for modeling. The best keyword average shares (kw_avg_avg) variable has the strongest correlation with shares. This is expected given this variable is inherently linked with shares, however, we decided to leave it in the analysis given its prevalence in article distribution. We will monitor this throughout the remainder of the analysis.

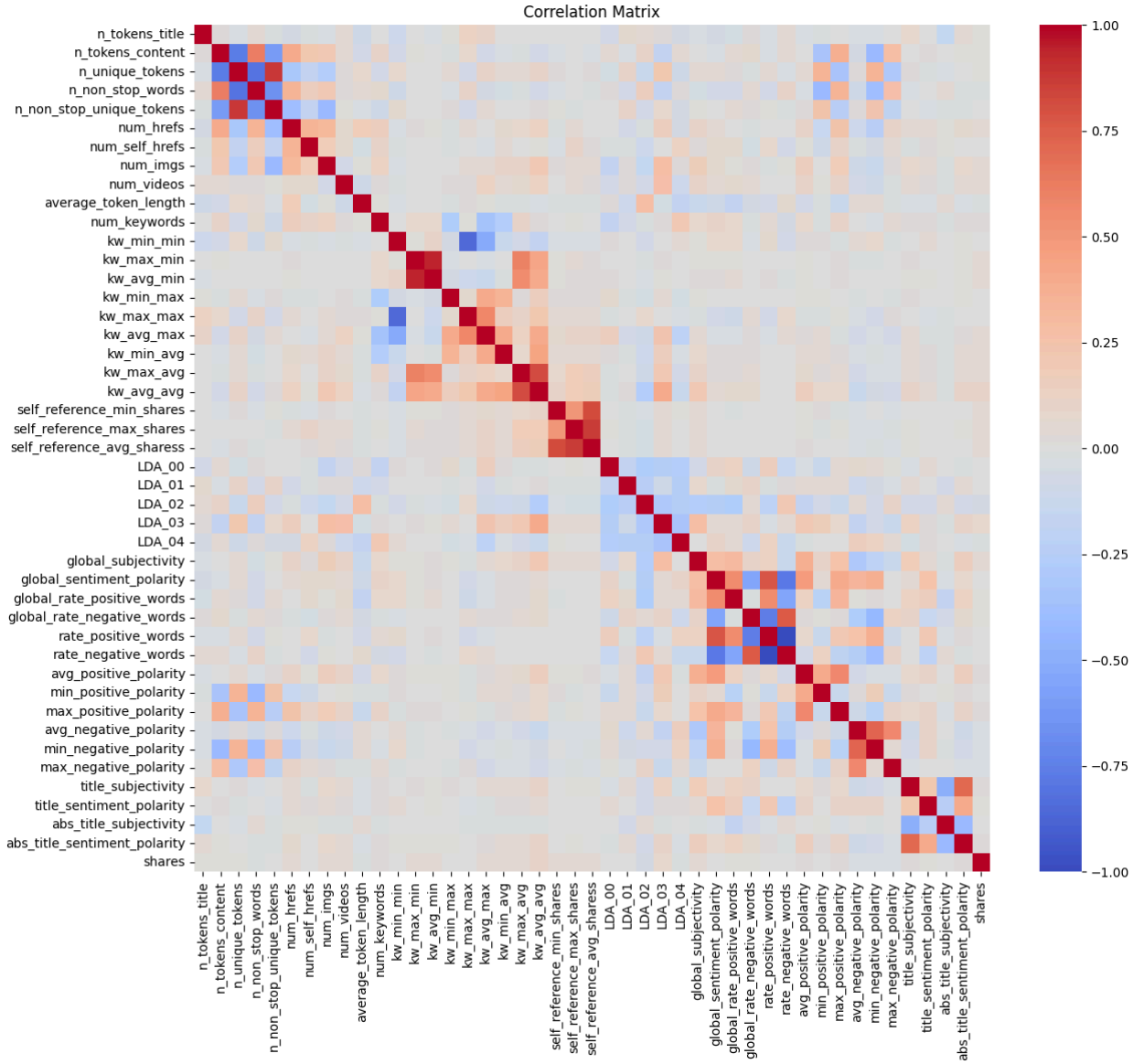
As expected, the length of the article (n_tokens_content) has a positive relationship with the number of links and images in the article. Interestingly, there is a positive relationship with the max positive polarity of the positive words. The average polarity has a slight positive

correlation as well. Polarity, per the correlation analysis, does not directly impact shares, but this reveals a potential relationship between the *type* of content developed where short articles are more negative and longer articles are more positive. Additionally articles with longer words on average are deemed more subjective but more positive in their sentiment. This may be due to the nuance that is able to be captured in more eloquent language and length of the copy as compared to the more brief articles.

Latent Dirichlet Allocation (LDA) is provided by our data engineers to reveal hidden topics in the article. From the correlation analysis, LDA topic 3 is positively correlated with shares and LDA topic 2 is negatively correlated. The same relationship is revealed with best keyword average shares (kw_avg_avg) as with shares, the LDA model may have recognized a density of keywords. LDA topic 2 also has a negative relationship with the polarity of the article. In future analysis we will provide more in depth analysis of these topics.

Figure 6

Correlation Heat Map Across Continuous Variables



Data Wrangling and Pre-processing

Before performing any data manipulation steps, it is important to ensure the column names are properly formatted. Spaces or hidden characters in column names can and did lead to issues during data manipulation. The column names were stripped of any leading or trailing

spaces using the .str.strip() function for both the features and target variables. This ensures all columns can be properly referenced during data pre-processing and analysis.

The next step taken in pre-processing was checking for missing values. This dataset did not include any missing values in either the features or the target variable. This was confirmed using the .isnull().sum() function. It was concluded that no further imputation or deletion was necessary for handling missing data.

The presence of outliers and how to handle them was an important part of pre-processing as the target variable had extreme outliers. Ultimately, it was decided to not remove outliers from the target variable as they represent real-world data where certain articles had gone viral, receiving a disproportionately high number of shares. These outlier data points are not inaccurate but show the variance of shares an online article can achieve. Removing these outliers would create a model that does not accurately predict article popularity, as they have the chance to go extremely viral.

Through the analysis of metadata distributions, some outlier records were removed: n_unique_tokens and n_non_stop_words are rate variables and therefore must be between 0 and 1, num_hrefs is typically below 100 for a standard article page, link counts above this are found on home pages which are not relevant to this study. The variable num_video being above 15, given its rare that video counts above this would be present in a standard article that does not solely focus on video aggregation. Finally, num_img we expect to fall up to 20 images for visual-heavy content, and likely skews more for longer content, but more than this is unreasonable and will load slowly for most users.

Data splitting

The dataset was split into training and testing sets, with 20% of the data allocated for testing and a fixed random state to ensure reproducibility. Splitting the data allows us to evaluate the model's performance on unseen data to ensure that it can generalize well. The features are then standardized using a standard scaler to make sure each feature has a mean of zero and a standard deviation of one. Standardization is crucial because it ensures that features with larger scales do not dominate the model's learning process. Scaling ensures that all features contribute equally to the learning process.

Model Strategies

In this project, we aimed to predict article popularity (measured in online shares) based on article parameters and metadata using various statistical machine learning algorithms. Using the selected algorithms we used we wanted to discover and provide insights into which article traits have the greatest impact on the popularity, and how can these insights be used to improve both the reach of articles and maintain journalistic integrity.

We employed three models, a support vector regression model, a Random Forest model, and an XGboost model. Each model was selected because of the unique strengths it can offer and the interpretability each model offers depending on the problem being solved.

The first model used is a support vector regression (SVR). This model was chosen for its ability to model complex relationships in the data by maximizing the margin between actual and predicted values. In this case, a lot of the features were small and large thus leading to complex relationships. Although outliers were accounted for, it was decided that it was appropriate to keep these as they are valid points captured. The second model utilized is a Random Forest regression model. This particular ensemble model is highly robust to overfitting and capturing

non-linear relationships among the data. Since the data contained a high number of features, there are likely non-linear relationships present. This model was also used to complement one of the final models chosen; XGboost.

The third and final model used is XGboost. This model is known for its ability when it comes to efficiency and performance when dealing with tabular data. The dataset is a relatively high dimensional dataset which involves complex relationships and XGboost does a great job capturing these relationships. In pair with the tree based model, we leveraged the SHAP (SHapley Additive exPlanations) library allowing interpretation of predictions and understanding of the significance of each feature into a prediction. While feature importance charts can show us which features are being used the most by a model to generate a prediction, feature importance does not provide the error each feature is generating towards a prediction. SHAP allowed us to see the error each feature provides towards a prediction.

Validation and Testing

Support Vector Regressor

The SelectKBest method was used to select the top 20 features based on their correlation with the target variable. The number of features was reduced with the goal of preventing overfitting as we did not want the model to be too complex or capture noise in the data. With less features, the model can focus on the most important variables which should lead to better generalization on unseen data. Another goal of reducing the number of features was to speed up training and prediction times. This is particularly beneficial for the Support Vector Regressor (SVR) model as it requires significant computational resources.

The SVR model is created using the RBF kernel, which is useful for non-linear regression tasks. The model is trained on the selected training features and target variable to find

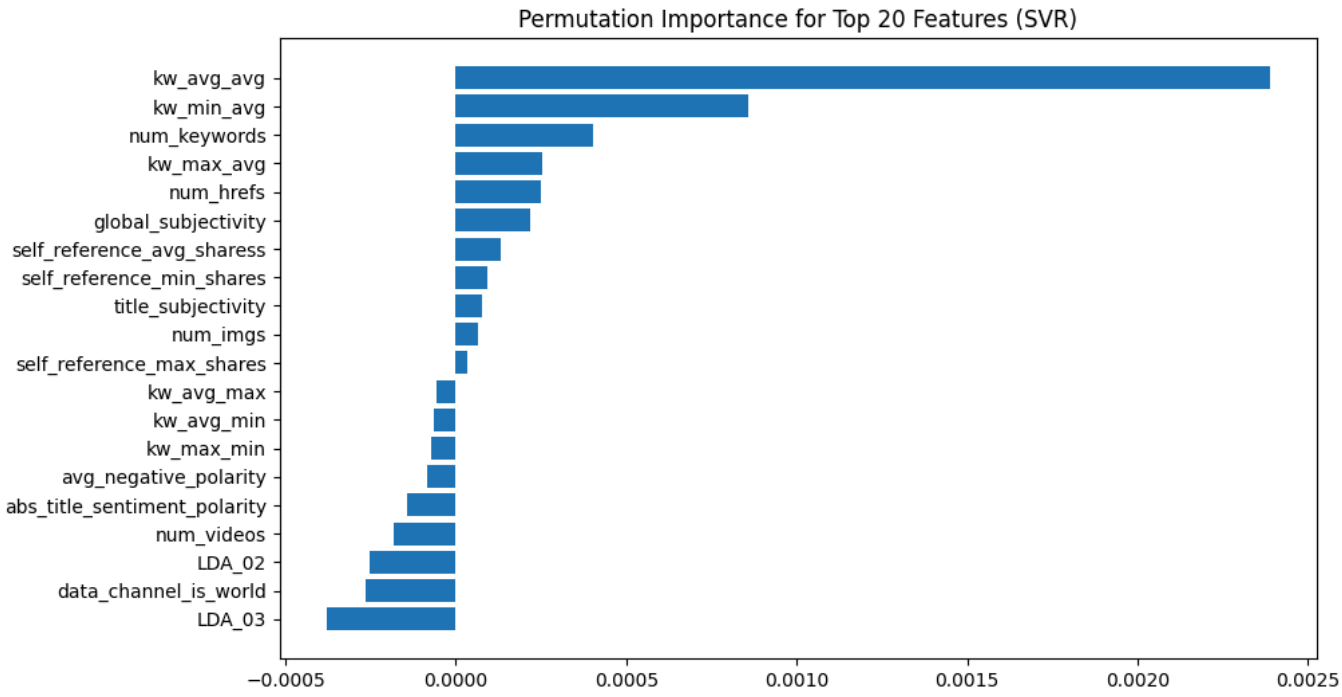
patterns and relationships in the data. The model was then evaluated by predicting the test set values.

The mean absolute error and root mean squared error are calculated to assess the model's performance. The MAE produced a value of 2,360.6, meaning that the model's predictions differ from the actual values by 2360 shares. The RMSE produced a value of 11,125.14, indicating that the model's larger prediction errors are substantial and have a strong influence on the overall error, suggesting the presence of some significant differences in predictions.

A sample of 1,000 test records was used to speed up the process of calculating the permutation importance. This method helps to identify the importance of each feature by evaluating how performance changes when a feature's values are randomly shuffled. By repeating the process multiple times, the permutation importance is calculated, allowing an understanding of which features contribute most to the model's performance.

Figure 7

Permutation Importance for Support Vector Regressor



This bar chart displays the permutation importance of the top 20 features in the SVR model. The most influential feature is the average of keyword averages, followed by the minimum average of keywords and the number of keywords as a whole. This shows that keyword-related features play a significant role in the model's predictions. Other notable variables include the number of links (h_refs) and global subjectivity. The chart helps focus attention on the key features driving the model's performance.

XGBoost

XGBoost, an extreme gradient boosting algorithm, was used as one of the main models along with the SVR and Random Forest. While our SVR and Random Forest models performed well, we ran into issues with the SVR, where it took a while to run on the whole large dataset. XGBoost is fantastic in this sense because it is optimized for parallel computation training and

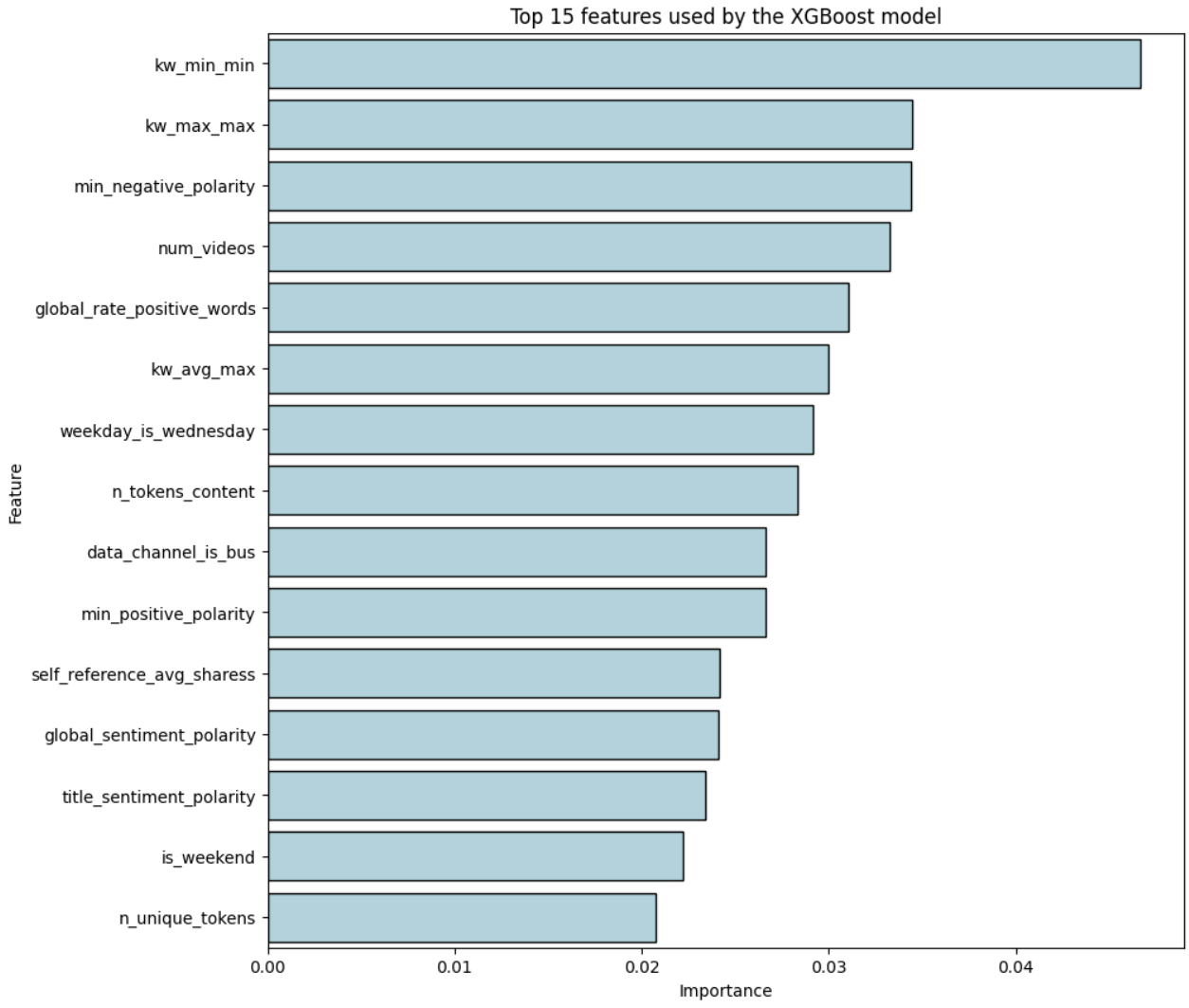
includes regularization which is great when training on redundant data. Another great feature of using tree-based models is they include feature importance which provides you the F score of all the features. One downside of XGboost is that it is a black-box model which makes it difficult to explain on a feature-level basis. However, we were able to use the SHAP library which gives us insights into what the model is doing with each feature to make the final prediction.

After successfully running the XGBoost model on our training and testing data sets, we were able to achieve a mean absolute error of 3,011, a mean squared error of 117,703,955, and a root mean squared error of 10,849. With these metrics, we know that our model had an average difference of 10,849 shares per prediction from the actual value. A downside of Xgboost is that if not tuned properly, it can overfit or perform poorly on the data. These values are with hyperparameter tuning which tells us we should further explore tuning our model.

The bar plot below shows the top 15 features the XGboost model utilized to generate a prediction. If we compare this bar chart to the other two, we can see that there are overlaps, especially with the SVR model.

Figure 8

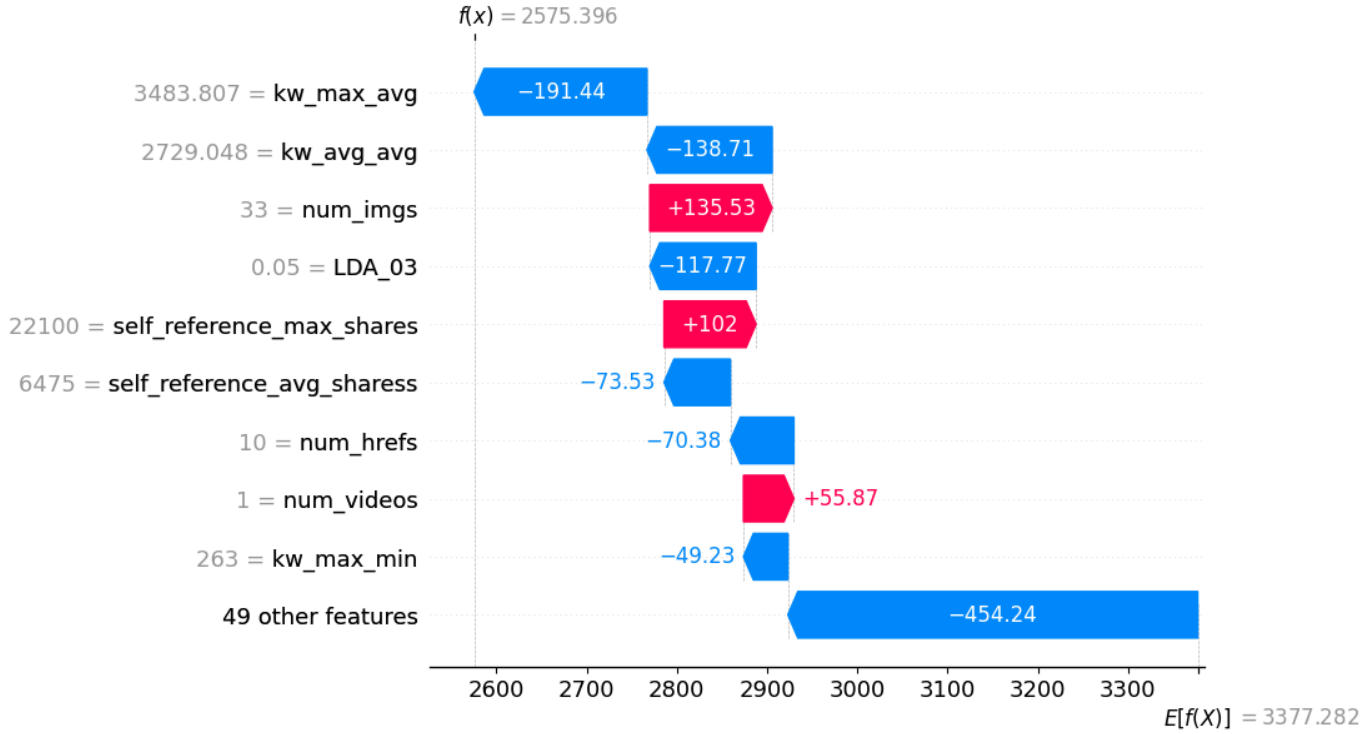
Feature importance for XGboost model



The bar plot below was generated by leveraging the SHAP library with the XGboost model. This particular plot provides what value goes into the model generating a prediction. The *f(x) = 2575.396* is a given prediction the model has created. The values below show what went into generating the final prediction. We can translate these values as being similar to the coefficients a simple linear regression model generates to achieve a prediction. The *E[f(x)]=3377.282* is the average prediction the XGBoost model generated.

Figure 9

SHAP coefficient values for given prediction



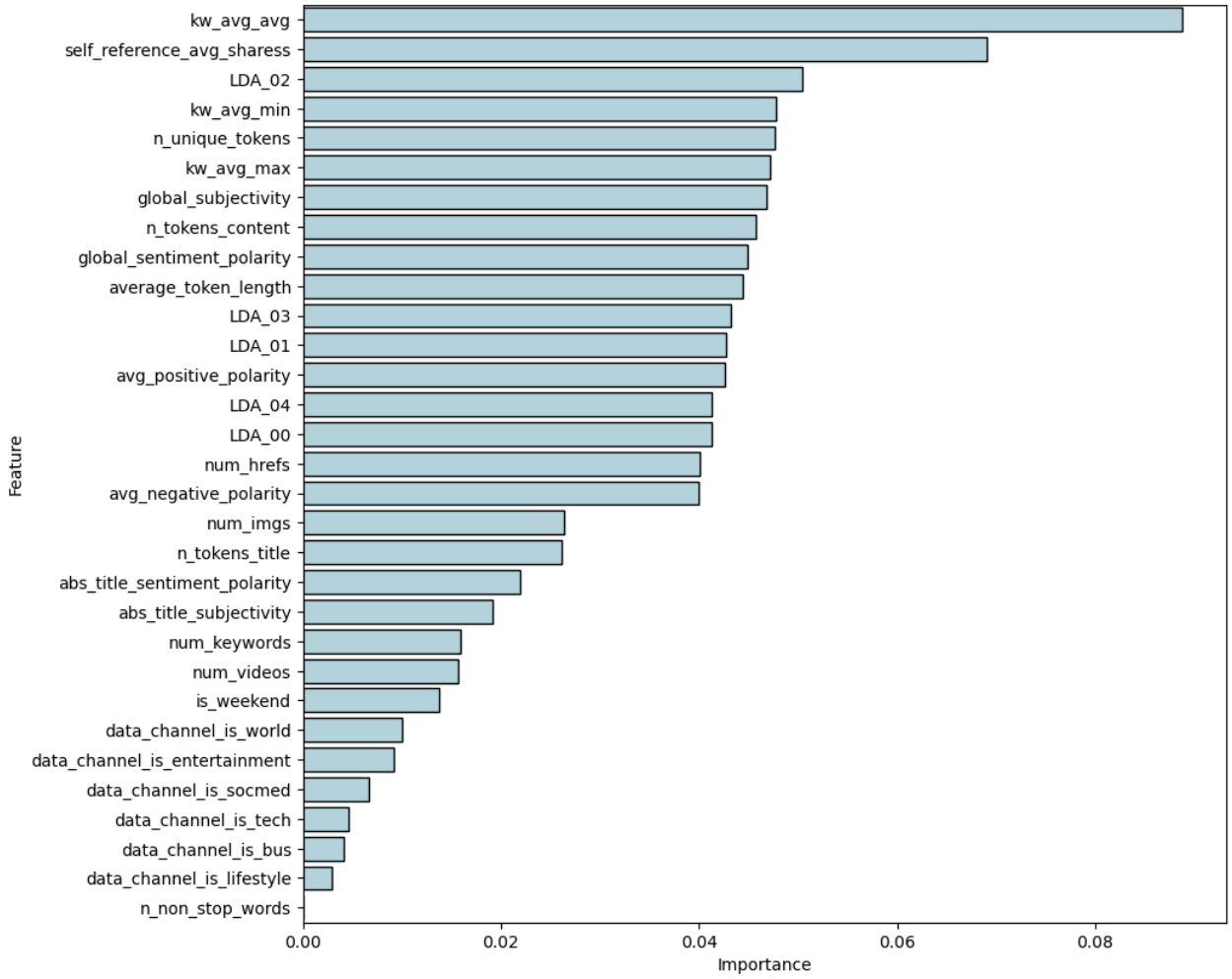
Random Forest

A Random Forest model is trained using a subset of variables (removing related variables and keeping average outputs or base level of engineered variables) with random search cross-validation to find optimal parameters. Random Forest is tested for a number of reasons including its handling of non-linear relationships, which are often present in internet media. Random Forest also provides direct insight into feature importance which is pivotal in relying key information back to Mashable and its editors. Through cross validation, the optimal model was found to include 200 trees, with max features being the square root of the total and the max depth of the trees set to 20. On the test set, this resulted in the Mean Absolute Error (MAE) of 2,458.07 and Root Mean Squared Error (RMSE) of 14,179.79.

Analysis of feature importance, as shown in Figure 8, reveals that average shares of average key words (kw_avg_avg) is the most important predictor. This is consistent with other models, but not terribly significant given keywords are likely optimized for already by editors and SEO professionals. Although, it is interesting that the average keywords are more important than the worst or best in terms of driving shares (these are also top 6 features). The average shares of referenced articles is the second most important feature and tells us that linking to articles that have demonstrated strong performance metrics will improve the shareability of a given article. This may seem obvious from an SEO perspective, but still worth noting given this may be a rather easy implementation from a content perspective. The third most important feature is the closeness to LDA topic 2. From earlier analysis, this stems from an inverse relationship. As such, editors could potentially steer away from producing content that aligns with LDA topic 2. Further analysis from engineering will be needed to decipher LDA topics. One final note is article length and subjectivity does seem to influence shares. From earlier analysis, highest binned group shares skewed slightly lower in total token count and slightly higher in subjectivity. As we continue to evaluate developing journalistic, shareable content at Mashable, this model suggests that short form opinion pieces with strong key word consideration and outlinking drive the most shares for Mashable articles.

Figure 10

Random Forest Feature Importance



Results and Final Model Selection

Based on the performance of our models, it is recommended to use the Random Forest and XGBoost models due to their strong predictive accuracy and versatility. Both models include their strengths and weaknesses. For example, while Random Forest is great and simple to use, it is more difficult to interpret. XGBoost is similar in this sense but when paired with SHAP, it can become a strong algorithm and is still simple to interpret. Random Forest is relatively easy to use, requires minimal tuning, and provides built-in feature importance metrics, making it a solid

choice for quick and reliable insights. Its interpretability is straightforward compared to more complex models, allowing for clear identification of key features influencing article popularity.

XGBoost, while more complex to configure, offers greater flexibility and superior performance on larger datasets with intricate feature interactions. Its integration with SHAP (SHapley Additive exPlanations) further enhances interpretability, as SHAP values allow us to explain individual predictions and understand the contribution of each feature in a more granular and intuitive way. This pairing provides actionable insights into which features drive the most impact on article shares, enabling more informed editorial strategies.

Table 1

Model Performance

Model	**MAE**	**RMSE**
SVR	2,360	11,125
Random Forest	2,458	14,175
XGBoost	3,007	10,846

Table 1 shows the metrics used to compare the different models. The main two metrics chosen are the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Both of these are great metrics to use but also include their flaws. MAE is the average of the absolute difference between the predicted and actual values. MAE treats all errors, including outliers, equally and a lower value is usually preferred. Root Mean Squared Error is the squared average difference between predicted and actual values. RMSE gives more weight to larger errors

making it more sensitive to outliers. In regards to the MAE, the SVR model performed the best. When it came to the RMSE, the XGBoost model performed better. It is concluded that the SVR model performs better on the whole dataset overall. The XGBoost model performs better when it comes to outliers, which the dataset had a large quantity of.

Discussion and Conclusions

The purpose of this project is to increase Mashable's article popularity (shares), while promoting transparency and consistency. Through the implementation of the Random Forest or XGBoost models, editors at Mashable can predict shares for their articles within roughly 2,500 shares and make editorial changes to the content based on this predicted share count. More importantly, this model can support these changes through the development of actionable and transparent feature importance output. This analysis can influence change at the article level as well as determine general content strategies to support the effort to strengthen Mashable's journalistic voice, better surface important stories, and increase ad and subscription revenue through increased article shares. In future cases, if supported by the journalists, this model could be deployed directly into the editorial pipeline with suggestions for improvements. Further, this analysis could be run quarterly to analyze changes in trends and adapt content accordingly without straying far from Mashable's core offerings.

GitHub Link:

https://github.com/tshehadey/Projects/blob/main/ADS505%20Final%20Project.ipynb

References

Dykes, B. (2020). Effective data storytelling: How to drive change with data, narrative, and visuals. Wiley.

Fernandes, K., Vinagre, P., Cortez, P., & Sernadela, P. (2015). Online News Popularity [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5NS3V.

Appendix