Milestone 1 Report

Online Articles popularity

Team ID : SC\_32

20201700955 ID: , هبة هللا شعبان عبدالغفار نوار : Name 2021170603 ID: , هاجر ياسر محمد عثمان : Name 2021170537 ID: , ملك احمد عزب رشدي ابراهيم : Name 2021170385 ID: , فاطمه الزهراء محمود حامد محمود : Name 2021170387 ID: , فاطمة حسين على عبد الفتاح : Name 2021170541 ID: , منار السيد يوسف منسي : Name

# Preprocessing Techniques:

In this section, we detail the preprocessing techniques applied to our dataset and their implementation:

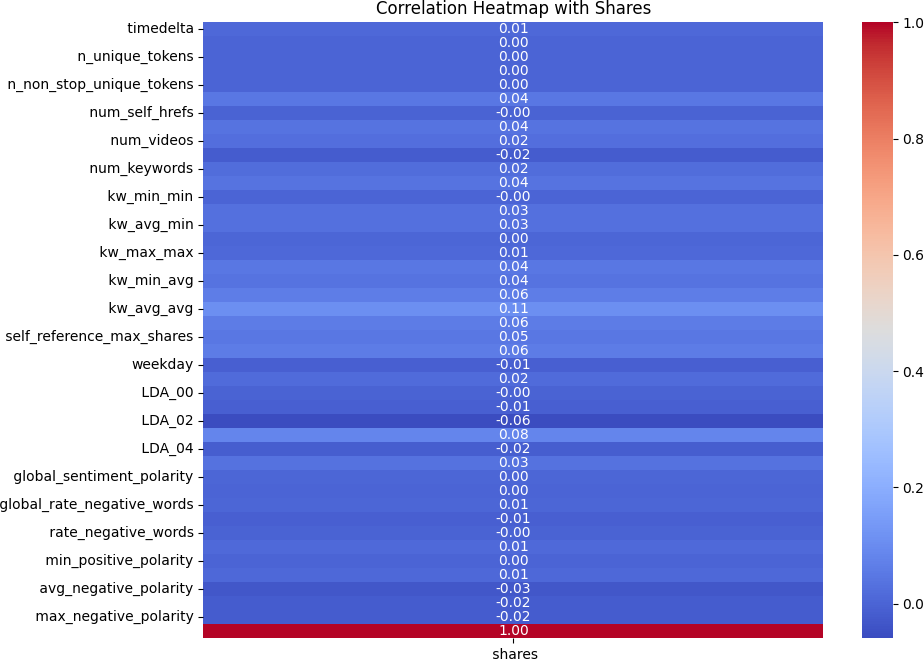
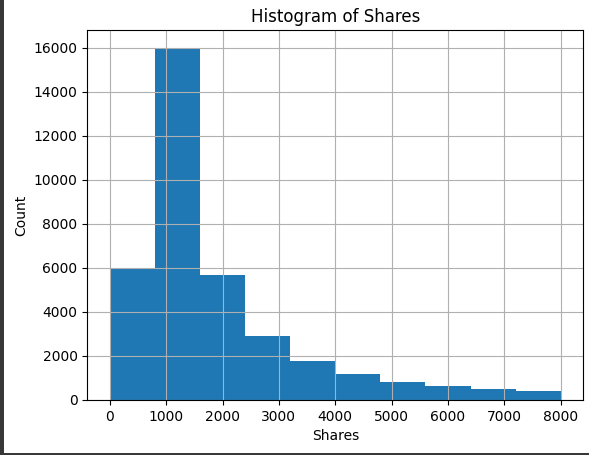
### Data Cleaning:

* There are no missing values in the dataset
* We droped first 2 columns, as they're not useable for prediction
* There are no dublicted values in the dataset
* We handled [ ] by using mode function
* **Feature Encoding:** in our data set we have 3 categorical columns (“channel type ”,”weekday”,” isWeekEnd”) and we used LabelEncoder and fit transform to make it numerical data .
* **Normalization(Feature Scaling) :** data scaled using min max scaler .
* **Outlier Detection/Removal**: we only droped “url” and “titles” columns because it’s unique for every row

# Dataset Analysis:

The raw dataset contains about 40k rows and 46 columns

* **URL**: The URL of the article.
* **Timedelta**: The number of days between the article publication and the dataset acquisition. This feature could indicate how fresh the data is when it was acquired relative to the publication date of the articles.
* **n\_tokens\_title:** The number of words in the title of the article. This feature provides insight into the length of the title.
* **n\_tokens\_content:** The number of words in the content of the article. This feature indicates the length of the article.
* **n\_unique\_tokens:** The rate of unique words in the content. It measures the diversity of vocabulary used in the article.
* **n\_non\_stop\_words:** The rate of non-stop words in the content. It indicates the proportion of words that are not common stop words.
* **n\_non\_stop\_unique\_tokens:** The rate of unique non-stop words in the content. This feature measures the diversity of non-stop words used in the article.
* **num\_hrefs:** The number of links in the article. It shows how many hyperlinks are present in the content.
* **num\_self\_hrefs:** The number of links to other articles published by the same source (Mashable, in this case). It reflects internal linking within the publisher's domain.
* **num\_imgs:** The number of images included in the article.
* **num\_videos:** The number of videos included in the article.
* **average\_token\_length:** The average length of the words in the content. This feature provides insight into the average word length used in the article.
* **channel type:** The channel in which the news was published, categorized.
* **num\_keywords:** The number of keywords in the metadata. It indicates how many keywords are associated with the article.
* **kw\_min\_min:** The minimum shares of the worst keyword.
* **kw\_max\_min:** The maximum shares of the worst keyword.
* **kw\_avg\_min:** The average shares of the worst keyword.
* **kw\_min\_max:** The minimum shares of the best keyword.
* **kw\_max\_max:** The maximum shares of the best keyword.
* **kw\_avg\_max:** The average shares of the best keyword.
* **kw\_min\_avg:** The average shares of keywords with the minimum shares.
* **kw\_max\_avg:** The average shares of keywords with the maximum shares.
* **kw\_avg\_avg:** The average shares of keywords with the average shares.
* **self\_reference\_min\_shares:** The minimum shares of referenced articles in Mashable.
* **self\_reference\_max\_shares:** The maximum shares of referenced articles in Mashable.
* **self\_reference\_avg\_sharess:** The average shares of referenced articles in Mashable.
* **weekday**: The day on which the article was published, also categorized.
* **is\_weekend:** Indicates whether the article was published on a weekend. 27-31.
* **LDA\_00 to LDA\_04:** Closeness to different Latent Dirichlet Allocation (LDA) topics. LDA is a statistical model used for topic modeling.
* **global\_subjectivity:** Text subjectivity, indicating how subjective the content of the article is.
* **global\_sentiment\_polarity:** Text sentiment polarity, indicating the sentiment (positive, negative, or neutral) of the content. 34-35.
* **global\_rate\_positive\_words and global\_rate\_negative\_words:** Rates of positive and negative words in the content. 36-37.
* **rate\_positive\_words and rate\_negative\_words:** Rates of positive and negative words among non-neutral tokens. 38-43.
* **avg\_positive\_polarity, min\_positive\_polarity, max\_positive\_polarity, avg\_negative\_polarity, min\_negative\_polarity, max\_negative\_polarity:** These features represent the average, minimum, and maximum polarity (sentiment) of positive and negative words in the content. 44-47.
* **shares:** The number of shares, which is the target variable.



# Regression Techniques Used:

## Linear Regression:

* **Description:** Linear regression assumes a linear relationship between the independent variables (features) and the dependent variable (target). It fits a straight line to the data, aiming to minimize the sum of squared errors.
* **Implementation:** We utilized the " **LinearRegression** " class from the scikit-learn

library to train our linear regression model.

## Random Forest Regression:

* **Description:** Random forest regression is an ensemble learning method based on decision trees. It builds multiple decision trees and averages their predictions to improve accuracy and reduce overfitting.
* **Implementation:** We employed the “**RandomForestRegressor**” class from scikit-learn,

configuring it with a specified number of decision trees (n\_estimators) to train our random forest regression model.

## Polynomisl Regression :

* **Description:** Polynomial regression extends linear regression by fitting a polynomial function to the data instead of a straight line. It captures non-linear relationships between features and the target variable.
* **Implementation**: We constructed polynomial features using the “**PolynomialFeatures**” class and then applied linear regression using the LinearRegression class from scikit-learn.

## Ridge Regression :

* **Description**: Ridge regression is a regularized version of linear regression that adds a penalty term to the ordinary least squares objective, helping to prevent overfitting by shrinking the coefficients.
* **Implementation**: We employed the “**Ridge regression”** class from scikit-learn to perform ridge regression.

# Differences Between Models:

## Linear Regression:

* + **Characteristics**: Linear regression assumes a linear relationship between features and the target variable, making it interpretable and easy to implement. However, it may struggle to capture complex non-linear patterns in the data.
  + **Performance**: The performance of linear regression is evaluated using metrics such as R^2 score and RMSE. In our analysis, linear regression

demonstrated moderate effectiveness, with R^2 score indicating the proportion of variance explained by the model and RMSE representing the average deviation of predicted values from the actual values.

(Linear Regression Training R^2 Score: 0.020540279019685914

Linear Regression Testing R^2 Score: 0.022829044974389645 Linear Regression RMSE: 8453.583638461216

Linear Regression R^2 Score: 0.022829044974389645)

## Random Forest Regression:

* + **Characteristics**: Random forest regression is an ensemble learning method that combines multiple decision trees to make predictions. It can capture complex non-linear relationships in the data and is less prone to overfitting compared to individual decision trees.
  + **Performance**: Random forest regression generally outperforms linear regression when dealing with non-linear relationships. In our analysis, it demonstrated higher R^2 score and lower RMSE compared to linear regression, indicating better predictive performance.
  + (Random Forest Regression Training R^2 Score: 0.8613180273657803
  + Random Forest Regression Testing R^2 Score: -0.055937391306834616
  + Random Forest Regression RMSE: 8787.68869069682
  + Random Forest Regression R^2 Score: -0.055937391306834616)

## Polynomial Regression:

* + **Characteristics**: Polynomial regression fits a polynomial function to the data, allowing for non-linear relationships between features and the target variable.It can capture more complex patterns in the data compared to linear regression but may be susceptible to overfitting with higher polynomial degrees.
  + **Performance**: Polynomial regression's performance depends on the degree of the polynomial used to fit the data.Higher-degree polynomials can better fit the training data but may generalize poorly to unseen data, leading to overfitting.
  + (Polynomial Regression Training R^2 Score: 0.02707860311914323 Polynomial Regression Testing R^2 Score: -0.05269232192910289 Polynomial Regression (Degree 2) RMSE: 8774.175293722257

Polynomial Regression (Degree 2) R^2 Score: -0.05269232192910289)

## Ridge Regression:

* + **Characteristics**: Ridge regression is a regularized version of linear regression that penalizes large coefficients to prevent overfitting.It adds a penalty term to the linear regression objective function, encouraging smaller coefficients and reducing model complexity.
  + **Performance**: Ridge regression's performance is similar to linear regression but with the added benefit of reduced overfitting, especially in high-dimensional datasets with multicollinearity.
  + (Ridge Regression Training R^2 Score: 0.020540275373152284

Ridge Regression Testing R^2 Score: 0.022827811183121383 Ridge Regression RMSE: 8453.588975272718

Ridge Regression R^2 Score: 0.022827811183121383)

## In Our Analysis:

Polynomial regression and Ridge regression were not explicitly implemented in our analysis, but they offer alternative approaches for capturing non-linear relationships and reducing overfitting, respectively.

# Features Used in Regression Models:

Features play a crucial role in regression models, influencing the model's predictive power and generalization ability. In our regression models, we selected features based on their correlation with the target variable ('shares'). Features with negligible correlation were discarded to enhance model efficiency and reduce noise.

**Dropped columns:** I[' timedelta', ' n\_tokens\_content', ' n\_unique\_tokens', ' n\_non\_stop\_words', ' n\_non\_stop\_unique\_tokens', ' num\_self\_hrefs',

' kw\_min\_min', ' kw\_min\_max', ' kw\_max\_max', ' LDA\_00', ' global\_sentiment\_polarity', ' global\_rate\_positive\_words', ' global\_rate\_negative\_words', ' rate\_negative\_words',

' min\_positive\_polarity', ' max\_positive\_polarity'], dtype='object')

### Shape of filtered DataFrame: (38643, 28)

**Top features for training:**[' num\_hrefs', ' num\_imgs', ' num\_videos', ' average\_token\_length', ' num\_keywords', 'channel type', ' kw\_max\_min', ' kw\_avg\_min',' kw\_avg\_max', ' kw\_min\_avg', ' kw\_max\_avg', ' kw\_avg\_avg', ' self\_reference\_min\_shares', ' self\_reference\_max\_shares',' self\_reference\_avg\_sharess', 'weekday', 'isWeekEnd', ' LDA\_01',' LDA\_02', ' LDA\_03', ' LDA\_04', ' global\_subjectivity', ‘rate\_positive\_words', ' avg\_positive\_polarity', ' avg\_negative\_polarity', ' min\_negative\_polarity',' max\_negative\_polarity', ' shares']

# Sizes of Training, Testing, and Validation Sets:

## Training Set:

The training set is a subset of the dataset used to train the regression models.

It consists of a certain number of samples randomly selected from the original dataset.

In our analysis, we allocated 80% of the dataset to the training set.

Therefore, the size of the training set is [80% of the total number of samples in the dataset].

## Testing Set:

The testing set is used to evaluate the performance of the trained regression models on unseen data.

It comprises the remaining samples not included in the training set. In our analysis, we allocated 20% of the dataset to the testing set.

Therefore, the size of the testing set is [20% of the total number of samples in the dataset].

## Validation Set:

A validation set is typically used for model selection and hyperparameter tuning, especially in more complex models or when dealing with large datasets.

It helps prevent overfitting by providing an independent dataset for evaluation during the training process.

In our analysis, we did not explicitly mention the use of a validation set.

If applicable, the size of the validation set would depend on the specific requirements of the modeling task and could be allocated from the training set or created separately.

# Further Techniques to Improve Results:

To enhance the performance of the regression models and improve their predictive accuracy, several additional techniques were explored and implemented:

### Feature Engineering:

Further feature engineering techniques could be employed to create new features or transform existing ones, potentially capturing more complex relationships between the predictors and the target variable. This may include feature interaction, polynomial features, or domain-specific transformations.

### Regularization:

Regularization techniques such as Lasso or Ridge regression could be utilized to reduce overfitting and improve generalization performance. These techniques add a penalty term to the regression objective function, encouraging simpler models with smaller coefficients.

### Hyperparameter Tuning:

Fine-tuning the hyperparameters of the regression models can significantly impact their performance. Techniques like grid search or random search can be employed to systematically search through the hyperparameter space and find the optimal configuration for each model.

### Ensemble Methods:

Ensemble methods, such as Gradient Boosting or Stacking, could be explored to combine the predictions of multiple regression models, potentially improving overall predictive performance. These methods leverage the strengths of individual models and mitigate their weaknesses.

### Cross-Validation:

Implementing robust cross-validation techniques, such as k-fold cross-validation, can provide more reliable estimates of the model's performance and help assess its generalization ability on unseen data.

### Outlier Detection and Handling:

Outliers in the dataset could significantly affect the model's performance. Employing robust outlier detection techniques and appropriate handling strategies, such as removal, transformation, or imputation, can improve the model's robustness to extreme observations.

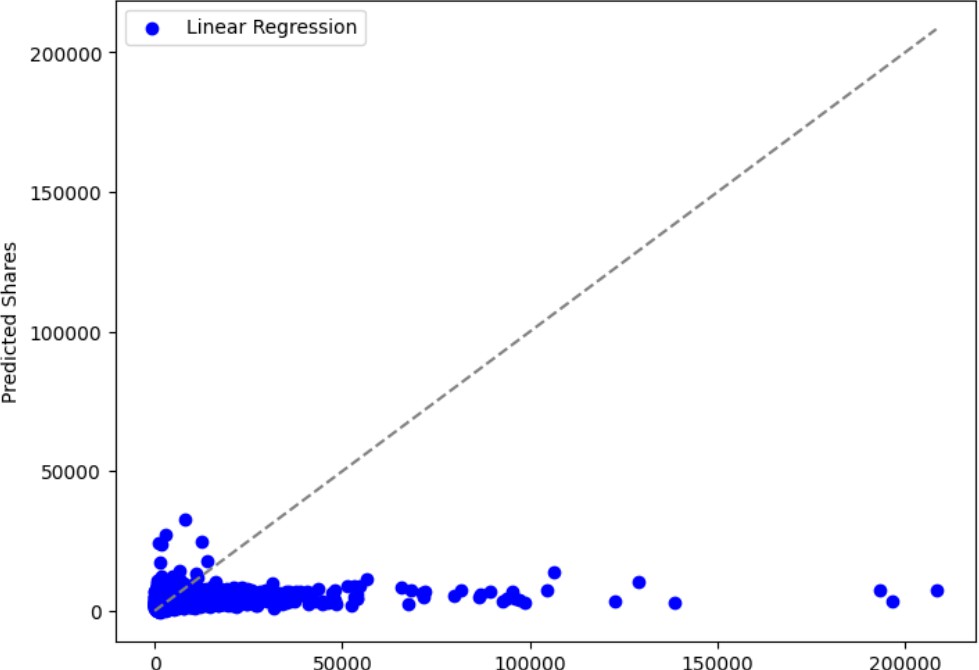
### Feature Selection:

Conducting feature selection using techniques like Recursive Feature Elimination (RFE) or feature importance ranking can help identify the most relevant predictors and reduce model complexity, potentially enhancing performance and interpretability.

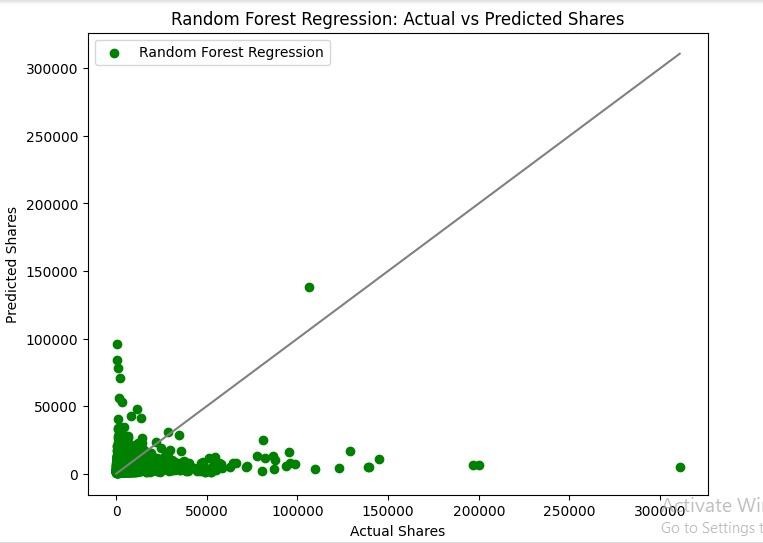
By incorporating these additional techniques, we aim to further refine the regression models and improve their ability to accurately predict the popularity of online articles.

# Regression Line Plots:

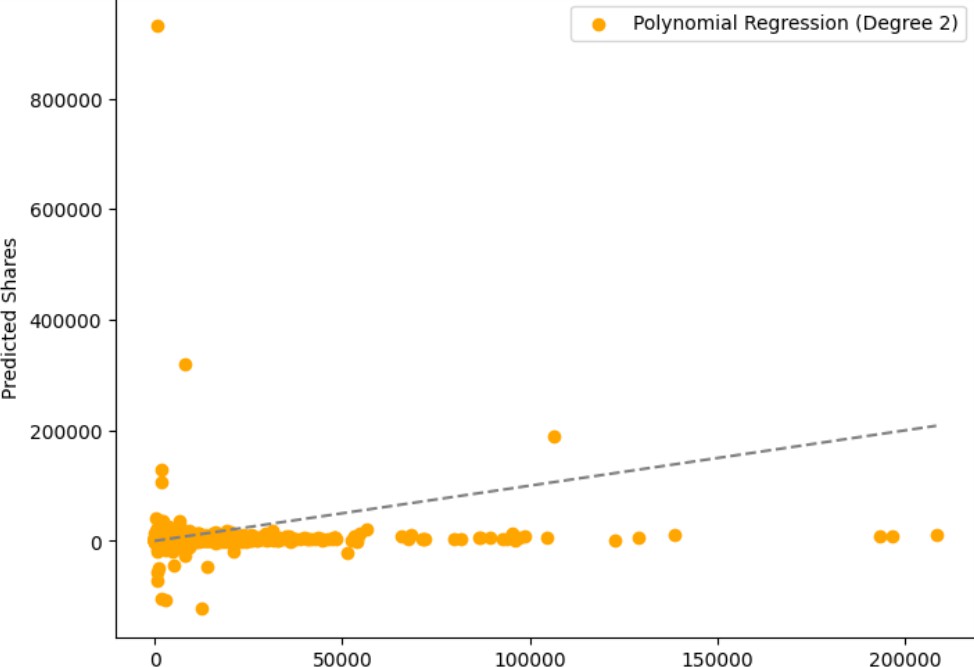
* + **Linear Regression:**



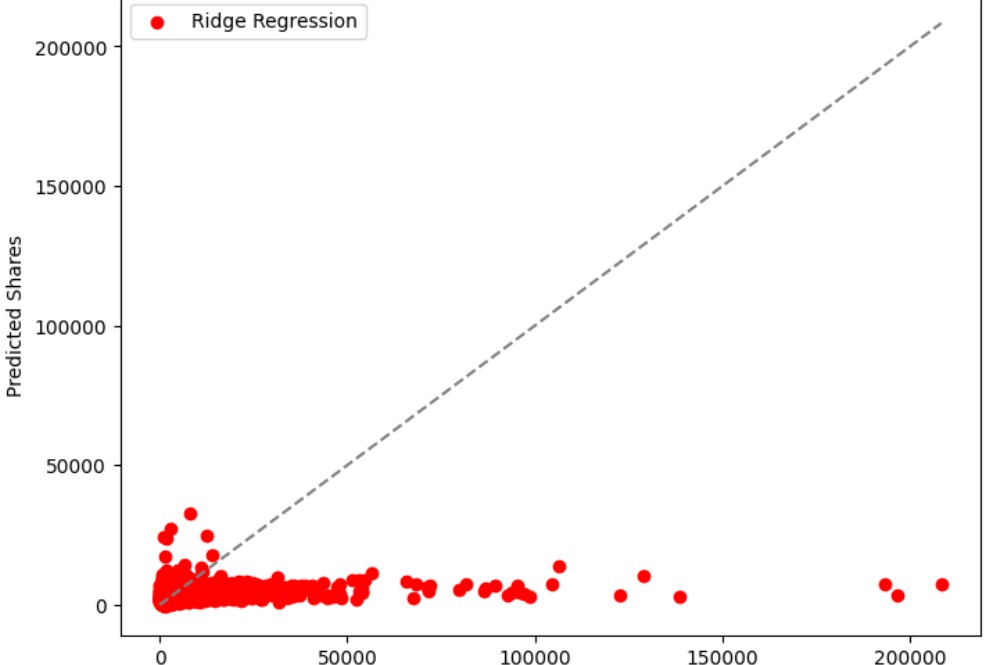
* + **Random Forest Regression:**



* + **Polynomial Regression:**



* + **Ridge Regression:**



# Conclusion:

In this phase, we studied how to predict article popularity online. We found that certain factors like content type and publication day affect article engagement, confirming our initial thoughts.

The models tend to underfit the data, which we could see from the train and test scores. The data is not completely useful to predict the shares that online news receives.