Brunel University London

Coursework Assignment: Online News Popularity Analysis

Student ID: 2244428

DEPARTMENT OF COMUTER SCIENCE

24TH APRIL, 2023

1. Data Description and Research Question:

In this research, the given case highlights a fictional situation including a service company that deals with the news channel platform. To provide better service, they have conducted a survey based on Online News Popularity and gathered various data for their research from different resources. This survey aims to analyze numerous related data to find valuable insights from the data and create a Machine Learning model which can predict the popularity of the news channel. For that use distributed data analysis approach is used whose main purpose is to process large and complex datasets a heterogeneous dataset has been taken under consideration which is about articles published by Mashable.

*** Data Description:** The data about the articles on the Online News Popularity information is a heterogeneous set of features that was published by Mashable (www.mashable.com). This data does not contain the original content because of authorization, however, some related features are mentioned which are useful for the statistical analytical details of the research.

Resource Link: https://archive-beta.ics.uci.edu/dataset/332/online+news+popularity
The data is presented in a CSV file with '39797' instances and '61' attributes.

Attribute Information:

- Number of Attributes: 61 (58 predictive attributes, 2 non-predictive, 1 goal field).
- The data dictionary is mentioned in the appendix section.
- The dataset seems to have all continuous variables along with the target variable 'shares'; but some variables seem to be categorical, which we will confirm in the EDA section.

Research Question: The objective of this analysis is to predict popularity based on the number of shares which is the target variable.

2. Data Preparation and Cleaning:

Data cleaning and preparation are the very first and important steps in data analysis, as they help ensure that the data is accurate, complete, and in the right format for analysis. This report will provide an overview of data cleaning and preparation using some common techniques that we applied to show its importance for analysis purposes.
We used 'Google Colab' for data cleaning and preparation purpose where we imported the necessary libraries and the main dataset taken from the source website as 'news_popularity'. To understand the dataset, we checked with the first 10 rows and checked the dimension of the dataset. To check the quality of the data we went through all the variables of the dataset and we consider variable **'url'** is not required for our analysis, hence, we removed this specific variable from the dataset and now the dataset contains 60 variables. All the rest variables are in a numeric format which means they are continuous variables. But as many of these variables seem to have binary values which signifies that these are categorical variables in the form of numeric.
In the data profiling process, which involves examining the data to identify any issues or inconsistencies, such as missing values, duplicate records, or data that is out of range, we applied relevant codes to inspect and we found no such missing values in the dataset.

In data validation, while eyeballing the overall structure of the data, we checked to ensure that it meets certain criteria or standards and found 15 variables:
'num_keywords','data_channel_is_lifestyle', 'data_channel_is_entertainment','data_channel_is_bus', 'data_channel_is_socmed','data_channel_is_tech','data_channel_is_world','weekday_is_monday','weekday_is_tuesday','weekday_is_wednesday','weekday_is_thursday','weekday_is_friday','weekday_is_saturday', 'weekday_is_sunday', 'is_weekend' as a categorical variable and rest 45 are continuous variables.
In data transformation, the data needs to be converted into a certain format if they are not already. And here as we noticed 15 categorical variables; but they are already in numeric form, hence they may need not to be converted as we always require categorical data to be converted into numeric or factor for data analysis.
We mentioned some graphical representations as 'Box Plot' for the variables and some of the variables have outliers but the values are not implausible. Therefore, no need to remove them.
On the overall view, the dataset is cleaned as it does not have any missing values, the criteria match with the requirements, the data is in the right format and the outliers do not impact the data much, hence we exported the dataset as **'Online_news_popularity_cleaned.csv'** and consider the dataset for the forthcoming process.

3. Exploratory Data Analysis (EDA):

In exploratory data analysis, also known as EDA, we tried to gain a better understanding of our dataset containing information about Online News Popularity by analyzing and investigating data and summarizing their main characteristics. This analysis will emphasize recognizing patterns within the data. For the EDA process, we used the **'R'** except for the clustering adding various libraries which are useful for data analysis, statistical computing, and graphical visualization.

The EDA also includes a data description part where we checked that there are **'19644'** observations(unlike the data description provided on the source site) and **'60'** variables and all the variables are continuous.

Descriptive statistics: In summary statistics, we eyeballed the minimum, maximum, median, and quartiles of each variable along with the internal structure of the data and all seem to be fine as there is no missing value, oddity, or misleading data value.

Data Transformation: In this step, we checked that the 15 variables: **data_channel_is_lifestyle, data_channel_is_entertainment,data_channel_is_bus,data_channel_is_socmed,data_channel_is_tech,data_channel_is_world,weekday_is_Monday,weekday_is_Tuesday,weekday_is_Wednesday,weekday_is_Thursday,weekday_is_Friday,weekday_is_Saturday,weekday_is_Sunday,is_weekend** have 2 and **num_keywords** has 10 categorical values. However, these are already in numeric form, so need not to transform these as the analysis can be done on only numeric data, so we left the format as it is. Later we will use variable standardization based on different criteria.

Data Visualization: In data visualization, it was quite tedious to plot for all the variables, hence we tried to extract the useful features first, then once the feature extraction is done, we will apply the data visualization to identify their trend.

Correlation Analysis: In correlation analysis, we examined the relationship between variables in the data to determine if there is a correlation or association between them by applying the **'cor'** function storing the value in **'cor_newsfame'** by rounding the resulting correlation values to two decimal places. To visualize the correlation among the variables, we first reshaped the **'cor_newsfame'** data frame and created a new data frame **'melted_cor_newsfame'** containing the correlation coefficient values between different variables, as this frame is better feasible to generate a 'Heatmap' for correlation. Fig.1 shows the heatmap for the correlation among all the variables.

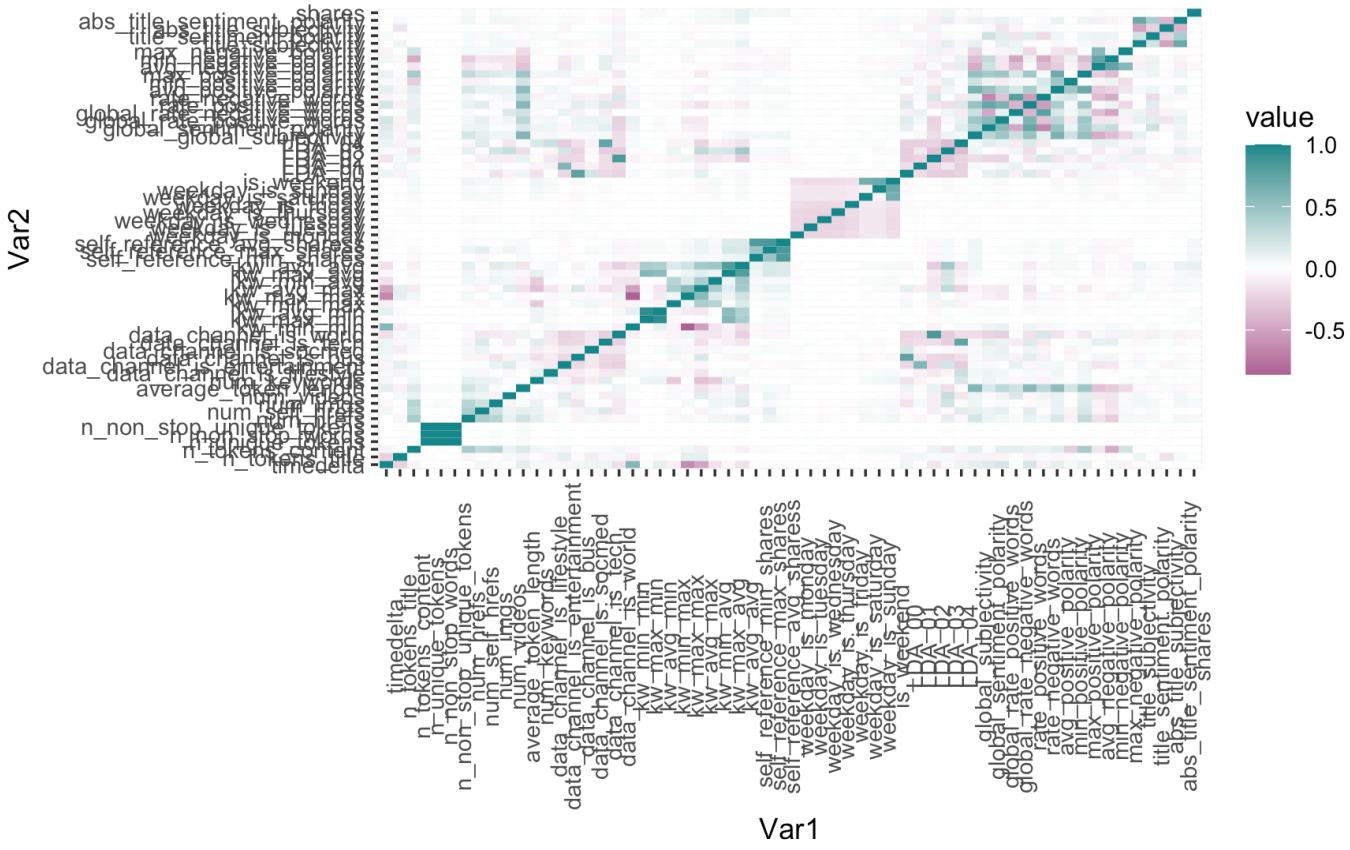
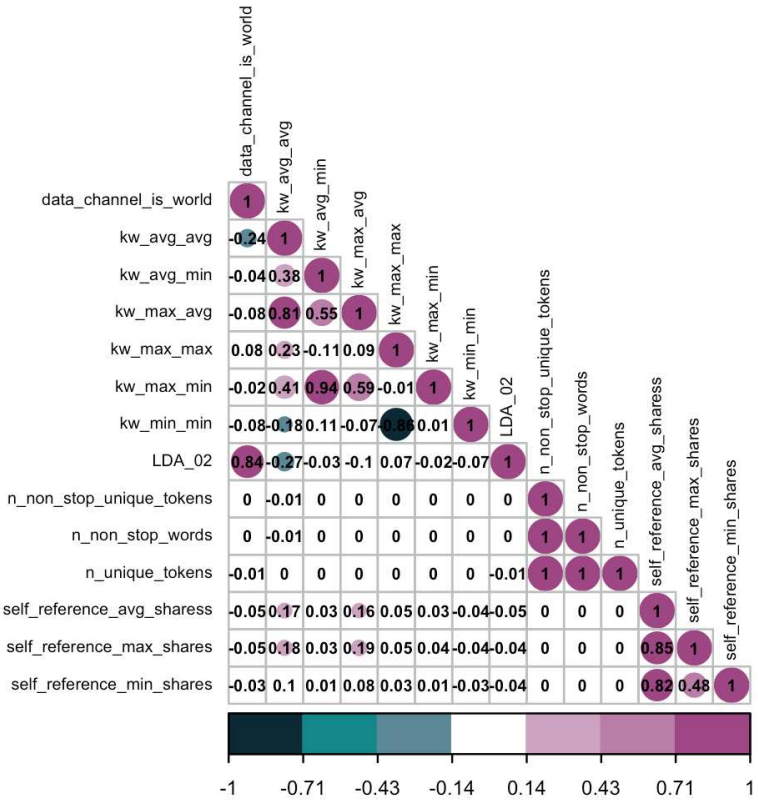


Fig. 1: Correlation Heatmap for all the Variables

We again tried to find out the variables which are strongly correlated with each other having more than 0.8 correlation coefficient with the code 'high_cor_newsfame <- which(abs(cor_newsfame) > 0.8 & upper.tri(cor_newsfame), arr.ind = TRUE)'. Here we considered the **'0.8'** correlation coefficient as the benchmark for strong correlation. And referring to all the above approaches, we found 14 variables **"n_unique_tokens", "n_non_stop_words","n_non_stop_unique_tokens", "kw_max_min","kw_avg_min","kw_min_min","kw_max_max","kw_max_avg","kw_avg_avg","self_reference_min_shares","self_reference_avg_sharess","self_reference_max_shares","data_channel_is_world","LDA_02"** that are strongly correlated with each other. To confirm the same we again plotted the correlation heatmap for these specific variables which is shown in Fig.2.
Hence we decided to discard **'data_channel_is_world,n_non_stop_words, kw_avg_min,kw_min_min, n_non_stop_unique_tokens, kw_avg_avg,self_reference_avg_sharess'** variables based on each pair with high correlation coefficient. Now we created a new data frame as **'newsfame_nonco'** having **53** variables to consider for further analysis, but before that, we did the final test for the correlation for the rest 53 variables with the same condition for **>0.8**, and we could finally confirm with the 53 variables.



Heavily Correlated Variables

n_unique_tokens,
n_non_stop_words
n_non_stop_unique_tokens
kw_max_min
kw_avg_min
kw_min_min
kw_max_max
kw_max_avg
kw_avg_avg
self_reference_min_shares
self_reference_avg_sharess
self_reference_max_shares
data_channel_is_world
LDA_02"

Fig. 2: Correlation Heatmap for Specified Variables

Dimension Reduction: Here our dataset is considered to be high dimensional having 53 features, hence, we need to inspect which variables explain the data more, and based on that we could reduce the dimension by removing the unnecessary features. There are various methods used for dimension reduction, and we used **'Principal Component Analysis'** also called **'PCA'** which is one of the popular methods.

We performed PCA on **'newsfame_nonco'** and created a new variable **'pc_newsfame'** where the PCA function's results are assigned. We validated the summary and structure of the PCA function to access further.

We plotted a **Scree plot** for **pc_newsfame** that uses PCA to visualize the amount of variance explained by each principal component (PC) in the dataset.

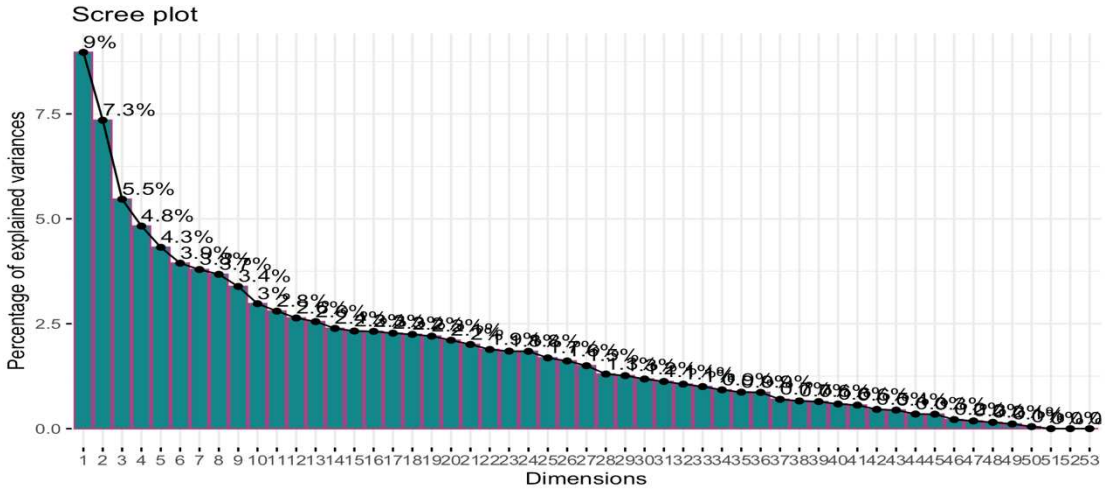


Fig. 3: Scree Plot for PC

Fig.3 shows the Scree plot for PC that displays the amount of variance explained by each principal component on the y-axis, and the number of principal components on the x-axis. "The **'elbow'** method is used to interpret the scree plot where looking for an 'elbow' shape on the curve and retaining all components before the point where the curve flattens out"(Statistics Globe). Here it appears to occur at the 35th principal component. However, it is still unclear, so we went for a different approach.

We calculated the **'Proportion of Variance Explained'**, also called PEV, and saves the values in the variable **'pc_newsfame_pev'** that describes the amount of variability in the data that is explained by each principal component. We used this PEV as **'pc_newsfame_pev'** to present the PEV plot that presents the proportion of variance explained by each principal component in a PCA model graphically where the x-axis indicates the principal component count and the y-axis indicates the Cumulative PEV.

As a starting value, initially, we set 80% for the variance explanation, and this PEV plot shows that 29 components result in a variance close to 80%, which means **29** components explain the data upto 80%. However, we did not want to consider that amount of components, so we plotted another PEV plot with 70% for the variance explanation, where the plot shows that **20** components result in a variance close to 70%. Again we plotted another PEV plot with 65% for the variance explanation, and the plot shows that **17** components result in a variance close to 65%. Hence, we considered these **17 components** that explain the data upto 65% for our depth analysis and model creation.
Fig.4 shows the PEV plot with 65% for variance explanation.

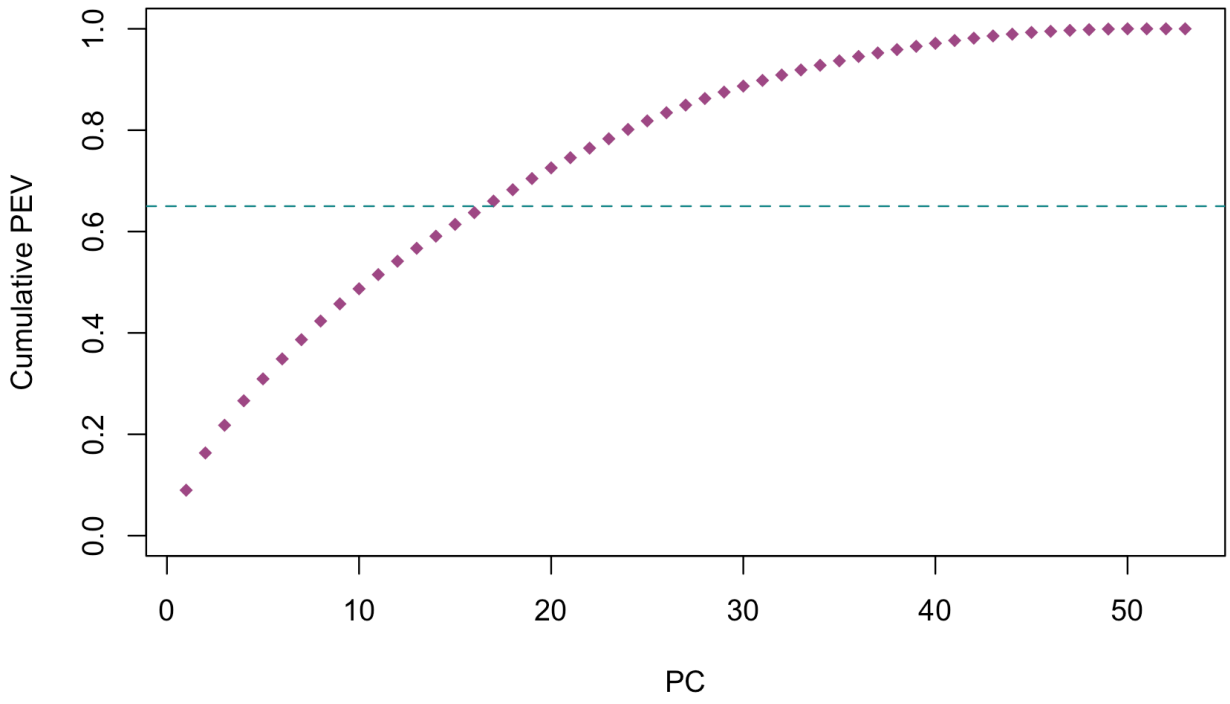


Fig. 4: PEV Plot`

As PCA is a feature extraction method that does not select the features, instead it transforms the features and then selects some of all the variables of the data set, we need to select the variables based on the rotation value. So, we created a variable **'pc_newsfame_loadings'** to assign the 17 principal components rotation values. The variable holding the maximum rotation value is considered to contribute the most to that specific PC and we found that the variable **'LDA_00'** has the maximum rotation value for both components PC4 and PC6, hence, we took **'LDA_00'** once. Now we have 16 variables to be observed. We presented the bar plot and Variable correlation plot to visualize the contribution of variables to the first 16 PC counts. Fig.5 shows the variable correlation plot which also signifies the variable contribution in PC count.

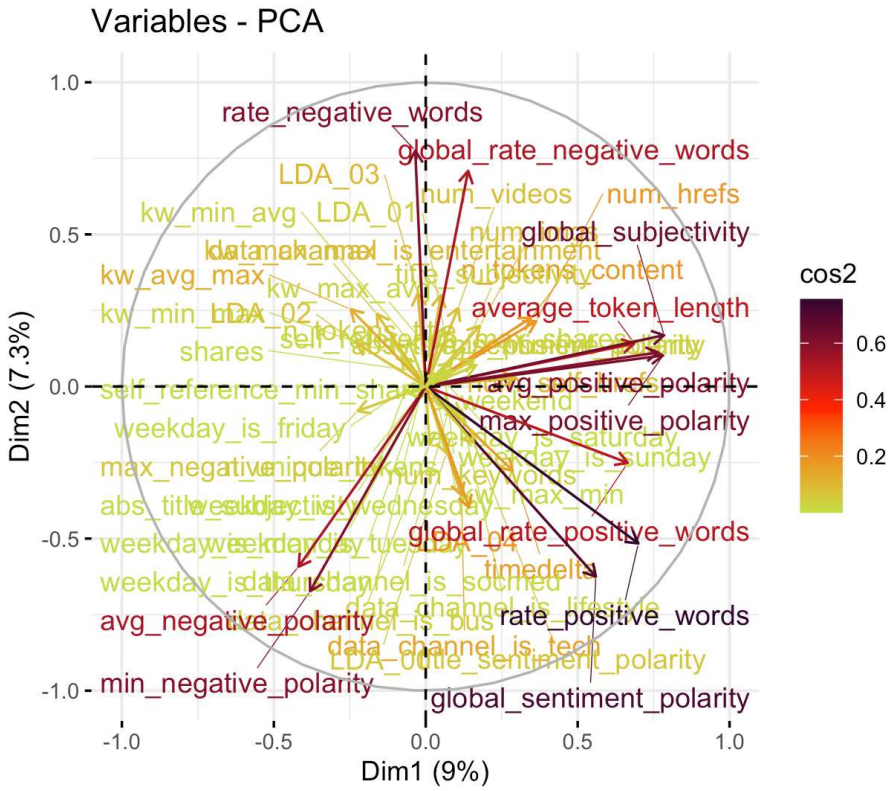


Fig. 5: Variation Correlation Plot

Based on rotation values and graphical representation of contribution, we determined the 16 variables which have the most contribution in PC respectively: **'global_subjectivity, rate_negative_words,kw_avg_max, n_tokens_content,LDA_00,is_weekend, kw_max_avg, LDA_02,LDA_01, kw_max_min, data_channel_is_entertainment,self_reference_min_shares, max_negative_polarity,weekday_is_wednesday,weekday_is_tuesday,weekday_is_monday'**. We used **`Histogram plot'** to see how each of the non-categorical variables of the data set is distributed as `Histogram` is a good way to check the data distribution and normality.

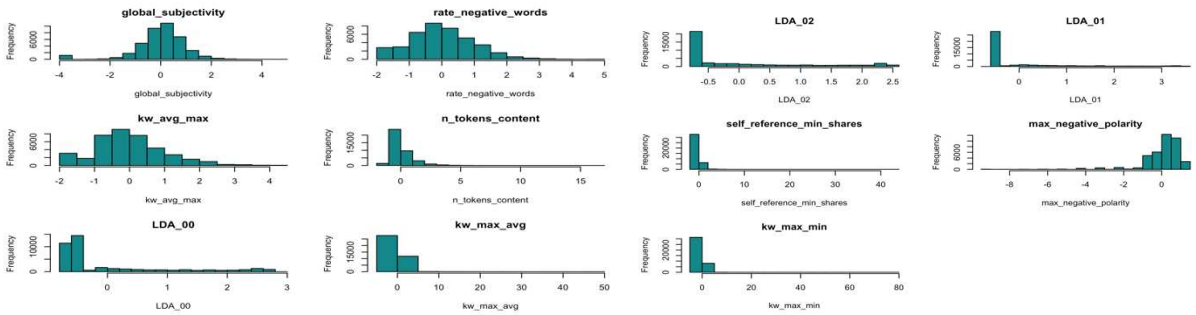


Fig. 6: Histogram of each Non-Categorical Variable

Fig.6 shows the histogram for each non-categorical variable where '**global_subjectivity',' rate_negative_words',** and **'kw_max_min'** seem to be normally distributed where the rest of the variables seem to be left or right skewed. So if required we could use transformation for modeling.

Finally, we created a new dataset as **'newsfame_final'** containing all the above-mentioned variables along with the target variable **'shares'**. This is our final dataset which is considered for the modeling purpose.

[Additional Work: Here the target variable **'shares'** is a continuous variable. As we proceed with the individual analysis, I try to create different machine learning models as an experiment. Hence, I decided to create another dataset having the target variable 'shares' as a categorical variable. Therefore, I created a new variable as **'Target_category'** converting it to a categorical variable by creating categories based on median value i.e 1400 of the main target variable **'shares'** to consider this as a new categorical target variable and saved the dataset as **'newsfame_final_categorical'**.

I exported the 2 datasets in CSV format for forthcoming analysis as follows:

1. OnlineNewsPopularity_Final_Contineous.csv: for **'newsfame_final'** having target variable **'shares'** as a continuous variable.
2. OnlineNewsPopularity_Final_Categorical.csv: for **'newsfame_final_categorical'** having target variable **'Target_category'** as a categorical variable.]

Cluster Analysis: After finalizing our dataset, we wanted to do additional analysis before creating the machine learning model. Cluster analysis discovers the structure of the data by splitting the groups of data into clusters having similarities. We used **'K-Means'** cluster analysis in **'Python'**. We imported the dataset **'OnlineNewsPopularity_Final_Contineous'**.
As clustering is an unsupervised machine learning technique, we did not consider the target variable and dropped the variable taking the rest of the variables for this analysis. By using the 'Elbow method' for the graph shown in Fig.7, we determined the optimal number of the cluster for the data is **'4'**.

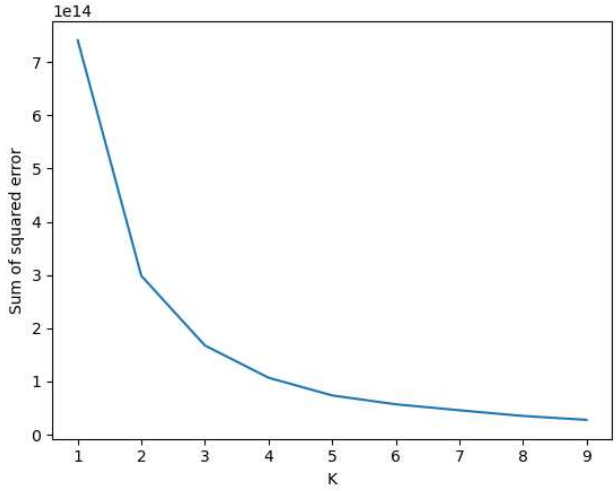


Fig. 7: Elbow for K-Means Clustering

In this graph, the curve seems to be flattened out at 4 where we got the 'elbow point', hence, the number of clusters should be 4.

So we created 4 clusters as 0,1,2,3 and saved the values with a new variable as 'cluster' added in the data frame. Cluster 3 has the maximum data i.e. '17418', and cluster 2 has the minimum data i.e. '4153' having similarities. As the cluster variable is added to the dataset, we can consider which data belongs to which cluster as mentioned in Fig.8.

LDA_02	LDA_01	self_reference_min_shares	max_negative_polarity	kw_max_min	weekday_is_wednesday	weekday_is_tuesday	weekday_is_monday	cluster
0.040005	0.378279	496.0	-0.200000	0.0	0	0	1	1
0.050096	0.050047	0.0	-0.100000	0.0	0	0	1	1
0.033351	0.033334	918.0	-0.133333	0.0	0	0	1	1
0.494651	0.419300	0.0	-0.166667	0.0	0	0	1	1
0.028575	0.028794	545.0	-0.050000	0.0	0	0	1	1

```
#gives predicted class labels (cluster) for each data point
km.labels_
```

```
array([1, 1, 1, ..., 0, 3, 0], dtype=int32)
```

Check the total count of observation belongs to specific cluster

```
Counter(km.labels_)
```

```
Counter({1: 6015, 3: 17418, 2: 4153, 0: 12058})
```

Fig. 8: Cluster Analysis

We finally plotted the graphical representation of these 4 clusters to show how these are distributed in Fig.9.

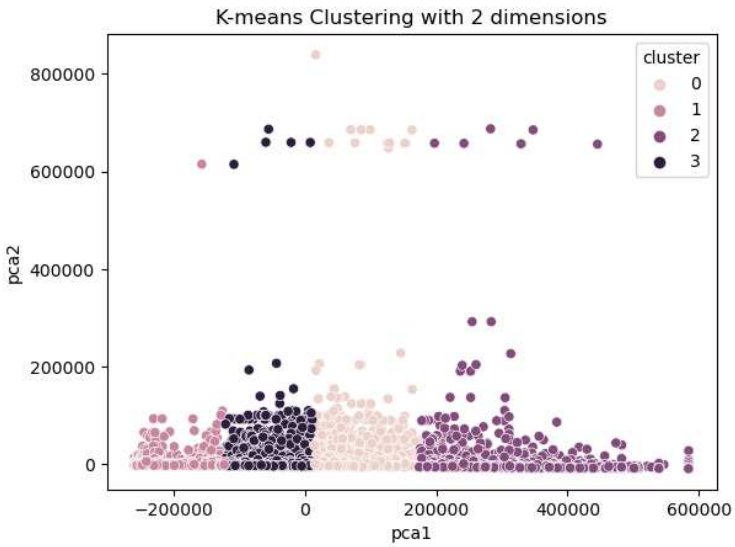


Fig. 9. Cluster 0,1,2,3

4. Machine Learning Prediction:

For Machine Learning predictions, we have created three different models individually using both the target variable **'shares'** and **'Target_categorical'** as continuous and categorical variables, and the models are built using Python.

- Decision Tree Model:

In machine learning, decision trees are a type of algorithm used for both classification and regression tasks. And here we have built the model based on the classification which means we considered the target variable as categorical that is **'Target_categorical'**.
We imported the dataset as **'OnlineNewsPopularity_Final_Categorical'** which contains the target variable in the categorical format as **'Low & High'**. Hence, we encoded it to numerical values before creating the model. We split the entire dataset randomly into training and testing datasets as **X_train, X_test, y_train,** and **y_test** based on **80%** and **20%** respectively so that I could build the model on the training datasets to train the model where **X_train** and **y_train** are the feature and target variables of the training set, respectively and applied that model to make the prediction on the testing dataset to test the model where **X_test** and **y_test** are the feature and target variables of the testing set respectively.

We created a **DecisionTreeClassifier** instance with a maximum depth of 3, and fit it to the training data. The accuracy **'0.6263084878294867'** of the model on the testing data is then printed to the console.

- Random Forest Model:

Then we created another machine learning model known as the Random Forest model, a popular machine learning algorithm that can be used for both classification and regression tasks. Here we have built the model based on the classification by taking the target variable as categorical that is **'Target_categorical'**.
We loaded a dataset as **'OnlineNewsPopularity_Final_Categorical'** and here the target variable has to be encoded into numeric values as it contains categorical values **'Low & High'**. We split the entire dataset randomly into training and testing datasets as **X_train, X_test, y_train,** and **y_test** based on **80%** and **20%** respectively so that I could build the model on the training datasets to train the model where **X_train** and **y_train** are the feature and target variables of the training set, respectively and applied that model to make the prediction on the testing dataset to test the model where **X_test** and **y_test** are the feature and target variables of the testing set respectively.
We created a **RandomForestClassifier** instance with the number of trees 10 and fit it to the training data. The accuracy **'0.6263084878294867'** of the model on the testing data is then printed to the console. The output of this code will be the accuracy of the Random Forest model on the testing data is **0.6119308866187413**.

- Linear Regression Model:

We created the final machine learning model known as the Linear Regression model, another machine learning algorithm that can be used for regression tasks. Here we have built the model based on the regression by taking the target variable as continuous numeric that is **'shares'**.

We loaded a dataset as **'OnlineNewsPopularity_Final_Contineous'** and here the target variable is 'share'. We split the main dataset into 2 parts as taring and test datasets as we did in the above classification models
We split the entire dataset randomly into training and testing datasets as **X_train, X_test, y_train,** and **y_test** based on **80%** and **20%** respectively so that I could build the model on the training datasets to train the model where **X_train** and **y_train** are the feature and target variables of the training set, respectively and applied that model to make the prediction on the testing dataset to test the model where **X_test** and **y_test** are the feature and target variables of the testing set respectively. Created an instance of the **LinearRegression** class, which represents the linear regression model, and calculated the mean squared error (MSE) as **219007767.5820227**, root mean squared error (RMSE) as **'14798.911026897307'**, and the R-squared score as **'0.011224508436615821'**. Here we are concerned about the R-squared score which is around **'0.01'**.

5. High Performance Computational Implementation:

For high-performance computational, we have created the same models for classification and regression using PySpark which is installed in Google Colab.

- HPC Implementation for Classification model using Decision And Random Forest Model on the dataset where the target variable 'Target_categorical' is a categorical variable.

We imported the dataset **'OnlineNewsPopularity_Final_Categorical.csv'** to the PySpark environment and checked the basic structure of the dataset. We created both the scatter plot and correlation plot to check the correlation among the numeric variables. We imported the important libraries to create the machine-learning models and used **OneHotEncoder** to convert the 'Target_categorical' variable into numeric form and created a vector for the rest numeric variables. We divided the main dataset into 2 datasets as Training and Testing datasets with 80% and 20% of the main dataset. Now we built a decision tree model to make predictions by configuring and training the Decision Tree classifier using the training data and testing the performance of the Decision Tree classifier using the testing data. Similarly, we have built the Random Forest model for the same dataset.
We got the same accuracy for both the decision tree and random forest as **1.000** which means **100%** of accuracy.

- HPC Implementation for Regression model using Linear Regression on the dataset where the target variable 'shares' is a continuous variable.

We imported the dataset **'OnlineNewsPopularity_Final_Contineous.csv'** to the PySpark environment and checked the basic structure of the dataset. We imported important libraries to create the machine-learning models. We divided the main dataset into 2 datasets as Training and Testing datasets with 80% and 20% of the main dataset. Now we initialized the linear regression model to make predictions and extract the prediction as the mean squared error (MAE): **0.03223372725540053,** root mean squared error (RMSE): **0.11742594936943072**, and R2: **2.0872192862952943e-14** which is a very smaller value almost near to **'0'**.

6. Performance Evaluation and Comparison of Methods:

As we have created three different machine-learning models, two classification models, and one regression model using Python and PySpark as High-Performance Computational implementation, the results are given as per the models.

Machine Learning Models created by Python:

- For the Decision Tree model, the accuracy is: **'0.626'**
- For the Random Forest model, the accuracy is: **'0.611'**
- For the Linear Regression model, the R-square value is: **'0.01'**

Machine Learning Models created using PySpark:

- For the Decision Tree model, the accuracy is: **'1.0'**
- For the Random Forest model, the accuracy is: **'1.0'**
- For the Linear Regression model, the R-square value is: **'0'**

From the above results using Python, we found that the accuracy difference between the decision tree, and the random forest model is around **0.01** i.e. **1%**, which is minimal, hence, we could consider that there is no such difference in accuracy for both models and consider the accuracy is **'0.6'**, where, in the linear regression model, the R-square value is '0.01' and differs from the above classification models' result.

From the results using PySpark, we found that the accuracy for both the decision tree model and random forest model is **'1.0'**, hence, these two models are no different from each other, where, in the linear regression model, the R-square value is almost **'0'** and that again differs from the above classification models' result.

However, the accuracy for the classification models created in Python and PySpark has differed where models designed by Python give an accuracy of **'0.6'** i.e. **60%** of accuracy, and models designed by PySpark give an accuracy of **'1.0'** which is **100%** of accuracy. But there is no such difference between the regression models created by Python and PySpark as in both cases the models give the R-square value almost **'0'.**

7. Discussion of the Findings:

Here for the linear regression model, the **R-squared** value is near 0, which means that the independent variables in the regression model are not good predictors of the dependent variable, and the model does not explain much of the variance in the dependent variable. In this case, the regression model is generally considered to be a **poor fit** for the data, and it is not able to provide a good explanation of the relationship between explanatory and response variables.

However, the classification models generated by Python and PySpark give **60%** and **100%** of accuracy for this model prediction respectively.

"Achieving **100%** machine learning model accuracy is typically a sign of some error, such as overfitting; that is, the model learns the characteristics of the training set so specifically that it

cannot generalize to unseen data in the validation and evaluation sets" (Iguazio). Where in some cases, **60%** accuracy might be considered good if the consequences of misclassification are not too severe in this context, we tried to predict the popularity of news based on the number of shares. Hence,

Final Result: Overall, we consider that the linear regression machine learning model taking the **continuous target variable** for prediction is not feasible to predict the popularity of the news based on the number of shares, where we can consider the **60%** accuracy of the model and confirm any of the classification models in between the **Decision Tree** or **Random Forest** model taking the **categorical target variable** created in **Python** for the prediction model.
NOTE: For this context, transforming the target variable **'shares'** from continuous to categorical as **'Taregt_categorical'** seemed to be effective and enhanced the model prediction possibility.

8. Data Management Plan and Author Contribution Statement:

The complete analysis of **'Online News Popularity'** has been done by team effort where each member has contributed and represented their tasks accordingly to achieve the overall goal. The parts covered as a team:

- Data Description & Research Question
- Data Preparation and Cleaning
- Exploratory Data Analysis

Afterward, the depth analysis has been done individually where each team member represented their discrete analysis over the machine learning models and tried to achieve the objectives. The sections covered as an individual:

- Machine Learning Prediction
- High-Performance Computational Implementation

Due to critical analysis and experiment, the below two sections of this analysis have been done individually.

- Performance evaluation and comparison of methods
- Discussion of the findings

The final Data Management Form will be attached in the appendix section to get more details about the overall plan.

REFERENCES:

1. (Dhruvil Karani,2022): https://towardsdatascience.com/a-practical-guide-on-k-means-clustering-ca3bef3c853d
2. Clustering as a part of EDA (Andres Rojas,2018): https://www.neuroelectrics.com/blog/2018/07/06/clustering-methods-in-exploratory-analysis/
3. Iguazio: https://www.iguazio.com/glossary/model-accuracy-in-ml/
4. The optimal value of correlation coefficient for strong correlation < 0.7: https://www.westga.edu/academics/research/vrc/assets/docs/scatterplots_and_correlation_notes.pdf
5. SCREE Plot: http://www.sthda.com/english/wiki/eigenvalues-quick-data-visualization-with-factoextra-r-software-and-data-mining
6. Statistics Globe: https://statisticsglobe.com/scree-plot-pca
7. Dataset Source: https://archive-beta.ics.uci.edu/dataset/332/online+news+popularity

Authorization Contribution Statement

In this distributed data analysis project, Amit Kedia, Sooraj Krishnakumar, Payal Parida, and Payalben Patel jointly discovered the data, formed the research question, performed data cleaning, and implemented Linear Regression, Support Vector Regressor, and Random Forest Regressor (500 trees) models. Sooraj Krishnakumar handled data preparation and applied Random Forest Regressor (100 trees) and Decision Tree models. Payal Parida and Payalben Patel jointly conducted the Exploratory Data Analysis and implemented PCA, K-means, and visualizations.

As per the experiment purpose, the machine learning and HPC techniques are done individually by me based on confirmation.