Table of Contents

Abstract

Analyzing TV commercials is crucial for understanding the factors that contribute to viewer engagement and optimizing content for effective advertising. This project focuses on a comprehensive analysis of TV commercial and non-commercial shots across different channels, utilizing a dataset from the UCI Machine Learning Repository. The data encompasses a range of audio-visual features, which we processed and loaded into Hive tables to facilitate efficient querying and analysis. Our methodology began with data preparation and storage in Hive, where we executed a series of queries to extract and examine visual and audio characteristics, such as motion distribution and spectral properties. By segmenting the analysis across multiple channels, we gained insights into variations in feature use between commercials and non-commercial content. For example, Hive queries revealed differences in motion dynamics and audio brightness between segments, highlighting the strategic elements used in advertising to capture attention. Subsequently, we combined data from all channels and employed the Random Forest model in R for feature importance analysis. This machine learning approach enabled us to identify key features that most effectively distinguish commercials from non-commercial content. The results indicated the significance of both visual and auditory components in differentiating content types, providing a data-driven basis for optimizing commercial design and classification algorithms. This project demonstrates the efficacy of using Hive for large-scale data management and R for advanced feature analysis. The insights derived are valuable for advertisers seeking to refine their strategies and for media platforms aiming to enhance content categorization and viewer experience. This study illustrates the power of integrating data processing and machine learning techniques to extract meaningful patterns from complex audio-visual datasets, contributing to the broader understanding of media content dynamics.

1. Introduction

Television commercials play a pivotal role in the advertising industry, capturing the attention of viewers and driving brand messaging. Given the intense competition for audience engagement, the design and delivery of commercials have become increasingly sophisticated, integrating compelling audio-visual elements to maximize impact. Understanding what differentiates commercials from regular programming is crucial for advertisers, broadcasters, and researchers who wish to enhance the effectiveness of advertisements and improve content organization within broadcast schedules. Consequently, analyzing the features that distinguish commercials from non-commercial content on TV is of significant interest, as it provides insights into the strategies that influence viewer retention and engagement.

1.1 The Idea

This project delves into a comprehensive analysis of TV commercial data to uncover the underlying characteristics that differentiate commercials from non-commercial programming across various channels. The dataset used for this research, sourced from the UCI Machine Learning Repository, comprises detailed information about commercial and non-commercial shots from prominent news and entertainment channels. These data entries contain a variety of audio-visual features, including motion metrics, energy distribution, and spectral properties, all of which provide a rich foundation for examining the elements that define effective advertising. The initial phase of the project will involve data pre-processing to convert the raw dataset into a suitable format for analysis. Given the high dimensionality and volume of the data, we utilize Apache Hive to structure and manage the dataset efficiently. By loading the data into Hive tables, we perform complex queries to explore patterns and correlations within the audio-visual features of both commercial and non-commercial shots. This stage of analysis would allow us to segment the data channel-wise, offering a granular view of how different networks leverage various features in their programming. Through Hive queries, we investigate visual features such as motion distribution, as well as audio features like the spectral characteristics. These analyses highlight certain trends. Analyzing these features across multiple channels provides insights into how different broadcasters employ unique strategies in their commercial content. Following the visual and audio feature analysis, we conduct a more comprehensive evaluation using machine learning techniques. Employing data analysis and Machine Learning techniques in R, we perform a feature importance analysis to identify the most significant predictors of commercial versus non-commercial content.

1.2 Significance

The results of this project have practical implications for various stakeholders in the media and advertising industry. Advertisers can leverage these insights to refine the design of commercials, emphasizing elements that have been empirically shown to capture attention. Broadcasters, on the other hand, can use this information to

optimize the scheduling and categorization of content, ensuring that commercial breaks are effectively engaging and seamlessly integrated into programming. Moreover, researchers and data scientists can draw from the methods used in this study to further explore the intersection of audio-visual features and viewer response.

Hence, This project is undertaken to understand the effectiveness of combining data management tools like Hive with machine learning techniques in R to analyze complex audio-visual datasets. By uncovering the key features that influence content classification, we contribute to a deeper understanding of the factors that make TV commercials impactful.

2. Methodology

In this study, we began by utilizing a comprehensive dataset of TV commercials, focusing on extracting meaningful insights from its rich audio-visual features. We processed and loaded the data into Hive, enabling efficient storage and structured querying. Through a series of analytical Hive queries, we explored various characteristics of the commercials, facilitating a deeper understanding of the underlying trends. The data is analyzed using Hive and R. And to interpret and present the findings visually, select aspects of the data were visualized using R.

2.1 Dataset Description

The TV Commercials data set consists of standard audio-visual features of video shots extracted from 150 hours of TV news broadcast of 3 Indian and 2 international news channels (CNNIBN, NDTV 24X7, TIMESNOW, BBC and CNN, 30 Hours each) with a total of 1,29,685 instances. For each channel, the Feature File (.txt) is represented in Libsvm data format and dimension index for different Features are as follows:

Label: +1/-1 (Commercials/Non Commercials)

Shot Length (1): The duration of a single video shot, indicating how long a shot lasts before a cut or transition.

Motion Distribution(Mean and Variance) (2, 3): Describes the spread and magnitude of motion within a video shot, derived from optical flow analysis.

Frame Difference Distribution (Mean and Variance) (4, 5): Captures sudden intensity changes between consecutive frames, highlighting transitions not covered by motion analysis.

Short time energy (Mean and Variance) (6, 7): Measures the energy or loudness of an audio signal over short time frames, capturing variations in sound intensity.

ZCR(Mean and Variance) (8, 9): Counts how frequently the audio waveform crosses the zero amplitude line, indicating signal noisiness or sharpness.

Spectral Centroid (Mean and Variance) (10, 11): Indicates the "center of mass" of the audio spectrum, reflecting the perceived brightness of the sound.

Spectral Roll off (Mean and Variance) (12, 13): Defines the frequency below which a specified percentage (usually 85%) of the total spectral energy lies, distinguishing between harmonic and noisy sounds

Spectral Flux (Mean and Variance) (14, 15): Measures the rate of change in the spectrum, capturing audio signal dynamics and transitions.

Fundamental Frequency (Mean and Variance) (16, 17): Represents the lowest frequency of a periodic audio signal, corresponding to the perceived pitch.

Motion Distribution (40 bins) (18 - 58): Describes the spread and magnitude of motion within a video shot, derived from optical flow analysis.

Frame Difference Distribution (32 bins) (59 - 91): Captures sudden intensity changes between consecutive frames, highlighting transitions not covered by motion analysis.

Text area distribution (15 bins Mean and 15 bins for variance) (92 - 122): Quantifies the placement and coverage of text overlays in frames, distributed across a grid layout.

Bag of Audio Words (4000 bins) (123 - 4123): Represent audio characteristics as a histogram-based vocabulary.

Edge change Ratio (Mean and Variance) (4124, 4125): Measures changes in edge locations between frames, used to detect movement of edges and transitions in visuals.

2.2 Dataset Processing

For simplification of our analysis, the features with bins are removed and remaining audio and visual features are kept. We then converted the data to csv format for loading the files into Hive tables. Now, we have a "TV_com" folder with the files: BBC.csv, CNN.csv, CNNIBN.csv, NDTV.csv, TIMESNOW.csv files. The folder is uploaded to HDFS.

A "tv_commercials" database is created in Hive using the following command.

```
hive> CREATE DATABASE tv_commercials;
OK
Time taken: 18.957 seconds
```

In the tv_commercials database, bbc_commercials, cnn_commercials, cnnibn_commercials, ndtv_commercials, timesnow_commercials tables are created and the corresponding data is loaded into the tables from the corresponding csv files.

```
hive> USE tv_commercials;
OK
Time taken: 5.754 seconds
hive> REATE TABLE bbc_commercials (label INT, Shot_Length INT, Motion_Distribution_Mean DOUBLE, Motion_Distribution_Variance DOUBLE,
    >
    > Frame_Diff_Distribution_Mean DOUBLE, Frame_Diff_Distribution_Variance DOUBLE,
    >
    > Short Time Energy Mean DOUBLE, Short Time Energy Variance DOUBLE,
    >
    > ZCR_Mean DOUBLE, ZCR_Variance DOUBLE, Spectral_Centroid_Mean DOUBLE, Spectral_Centroid_Variance DOUBLE,
    >
    > Spectral_Roll_Off_Mean DOUBLE, Spectral_Roll_Off_Variance DOUBLE, Spectral_Flux_Mean DOUBLE, Spectral_Flux_Variance DOUBLE,
    >
    > Fundamental_Frequency_Mean DOUBLE, Fundamental_Frequency_Variance DOUBLE, Edge_Change_Ratio_Mean DOUBLE, Edge_Change_Ratio_Variance DOUBLE
    >
    > ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 3.758 seconds
hive> LOAD DATA INPATH '/TV_com/BBC.csv' INTO TABLE bbc_commercials;
Loading data to table tv_commercials.bbc_commercials
Table tv_commercials.bbc_commercials stats: [numFiles=1, numRows=0, totalSize=3482331, rawDataSize=0]
OK
Time taken: 4.929 seconds
```

```
hive> CREATE TABLE cnn_commercials (label INT, Shot_Length INT, Motion_Distribution_Mean DOUBLE, Motion_Distribution_Variance DOUBLE,
    >
    > Frame Diff Distribution Mean DOUBLE, Frame Diff Distribution Variance DOUBLE,
    >
    > Short_Time_Energy_Mean DOUBLE, Short_Time_Energy_Variance DOUBLE,
    >
    > ZCR_Mean DOUBLE, ZCR_Variance DOUBLE, Spectral_Centroid_Mean DOUBLE, Spectral_Centroid_Variance DOUBLE,
    >
    > Spectral Roll Off Mean DOUBLE, Spectral Roll Off Variance DOUBLE, Spectral Flux Mean DOUBLE, Spectral Flux Variance DOUBLE,
    >
    > Fundamental_Frequency_Mean DOUBLE, Fundamental_Frequency_Variance DOUBLE, Edge_Change_Ratio_Mean DOUBLE, Edge_Change_Ratio_Variance DOUBLE)
    >
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 2.376 seconds
hive> LOAD DATA INPATH '/TV com/CNN.csv' INTO TABLE cnn commercials;
Loading data to table tv_commercials.cnn_commercials
Table tv_commercials.cnn_commercials stats: [numFiles=1, numRows=0, totalSize=4533769, rawDataSize=0]
OK
Time taken: 2.549 seconds
hive> CREATE TABLE cnnibn_commercials (label INT, Shot_Length INT, Motion_Distribution_Mean DOUBLE, Motion_Distribution_Variance DOUBLE,
    > Frame Diff Distribution Mean DOUBLE, Frame Diff Distribution Variance DOUBLE,
    > Short_Time_Energy_Mean DOUBLE, Short_Time_Energy_Variance DOUBLE,
    > ZCR_Mean DOUBLE, ZCR_Variance DOUBLE, Spectral_Centroid_Mean DOUBLE, Spectral_Centroid_Variance DOUBLE,
    > Spectral_Roll_Off_Mean DOUBLE, Spectral_Roll_Off_Variance DOUBLE, Spectral_Flux_Mean DOUBLE, Spectral_Flux_Variance DOUBLE,
    > Fundamental_Frequency_Mean DOUBLE, Fundamental_Frequency_Variance DOUBLE, Edge_Change_Ratio_Mean DOUBLE, Edge_Change_Ratio_Variance DOUBLE)
OK  > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
Time taken: 1.618 seconds
hive> LOAD DATA INPATH '/TV com/CNNIBN.csv' INTO TABLE cnnibn commercials;
Loading data to table tv_commercials.cnnibn_commercials
Table tv_commercials.cnnibn_commercials stats: [numFiles=1, numRows=0, totalSize=6660622, rawDataSize=0]
OK
Time taken: 2.199 seconds

hive> CREATE TABLE ndtv_commercials (label INT, Shot_Length INT, Motion_Distribution_Mean DOUBLE, Motion_Distribution_Variance DOUBLE,
    >
    > Frame_Diff_Distribution_Mean DOUBLE, Frame_Diff_Distribution_Variance DOUBLE,
    >
    > Short_Time_Energy_Mean DOUBLE, Short_Time_Energy_Variance DOUBLE,
    >
    > ZCR_Mean DOUBLE, ZCR_Variance DOUBLE, Spectral_Centroid_Mean DOUBLE, Spectral_Centroid_Variance DOUBLE,
    >
    > Spectral_Roll_Off_Mean DOUBLE, Spectral_Roll_Off_Variance DOUBLE, Spectral_Flux_Mean DOUBLE, Spectral_Flux_Variance DOUBLE,
    >
    > Fundamental_Frequency_Mean DOUBLE, Fundamental_Frequency_Variance DOUBLE, Edge_Change_Ratio_Mean DOUBLE, Edge_Change_Ratio_Variance DOUBLE)
    >
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 2.283 seconds
hive> LOAD DATA INPATH '/TV_com/NDTV.csv' INTO TABLE ndtv_commercials;
Loading data to table tv_commercials.ndtv_commercials
Table tv_commercials.ndtv_commercials stats: [numFiles=1, numRows=0, totalSize=3421955, rawDataSize=0]
OK
Time taken: 5.618 seconds
hive> CREATE TABLE timesnow commercials (label INT, Shot Length INT, Motion Distribution Mean DOUBLE, Motion Distribution Variance DOUBLE,
    >
    > Frame_Diff_Distribution_Mean DOUBLE, Frame_Diff_Distribution_Variance DOUBLE,
    >
    > Short_Time_Energy_Mean DOUBLE, Short_Time_Energy_Variance DOUBLE,
    >
    > ZCR_Mean DOUBLE, ZCR_Variance DOUBLE, Spectral_Centroid_Mean DOUBLE, Spectral_Centroid_Variance DOUBLE,
    >
    > Spectral_Roll_Off_Mean DOUBLE, Spectral_Roll_Off_Variance DOUBLE, Spectral_Flux_Mean DOUBLE, Spectral_Flux_Variance DOUBLE,
    >
    > Fundamental_Frequency_Mean DOUBLE, Fundamental_Frequency_Variance DOUBLE, Edge_Change_Ratio_Mean DOUBLE, Edge_Change_Ratio_Variance DOUBLE)
    >
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 1.3 seconds
hive> LOAD DATA INPATH '/TV_com/TIMESNOW.csv' INTO TABLE timesnow_commercials;
Loading data to table tv_commercials.timesnow_commercials
Table tv_commercials.timesnow_commercials stats: [numFiles=1, numRows=0, totalSize=7891302, rawDataSize=0]
OK
Time taken: 3.279 seconds
```

To ensure that the data is properly loaded into the tables, the following describe query is executed to check the bbc_commercials table

```
hive> DESCRIBE bbc_commercials;
OK
label                   int
shot_length             int
motion distribution mean        double
motion_distribution_variance    double
frame_diff_distribution_mean    double
frame_diff_distribution_variance        double
short_time_energy_mean  double
short_time_energy_variance      double
zcr_mean                double
zcr variance            double
spectral_centroid_mean  double
spectral_centroid_variance      double
spectral_roll_off_mean  double
spectral_roll_off_variance      double
spectral_flux_mean      double
spectral flux variance  double
fundamental_frequency_mean      double
fundamental_frequency_variance  double
edge_change_ratio_mean  double
edge_change_ratio_variance      double
Time taken: 2.627 seconds, Fetched: 20 row(s)
hive>
```

2.3 Analysis

The visual, audio and other aspects of the data are queried and analyzed using HiveQL. Feature importance analysis is performed using hive and R.

1. Commercial count and percentage

a. Hive query to observe the count of commercials in each channel:

```
hive> SELECT 'BBC' AS channel, COUNT(*) AS commercials_count FROM bbc_commercials WHERE label = 1
    >
    > UNION ALL
    >
    > SELECT 'CNN', COUNT(*) FROM cnn commercials WHERE label = 1
    >
    > UNION ALL
    >
    > SELECT 'CNNIBN', COUNT(*) FROM cnnibn_commercials WHERE label = 1
    >
    > UNION ALL
    >
    > SELECT 'NDTV', COUNT(*) FROM ndtv_commercials WHERE label = 1
    >
    > UNION ALL
    >
    > SELECT 'TIMESNOW', COUNT(*) FROM timesnow_commercials WHERE label = 1;
Query ID = hadoopuser 20241029132414 6369d016-bbb3-4f3a-89f8-fd5591c60074
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0011)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 10 .........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
Map 8 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 11 .....   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 5 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 7 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 9 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 10/10  [==========================>>] 100%  ELAPSED TIME: 296.35 s
--------------------------------------------------------------------------------
OK
```

Output:

```
BBC     8416
CNN     14411
CNNIBN  21693
NDTV    12564
TIMESNOW        25147
Time taken: 311.959 seconds, Fetched: 5 row(s)
```

b. Query to count the total number of instances (shots) per channel and the percentage of commercials by channel:

```
hive> SELECT channel,
    >
    > COUNT(CASE WHEN label = 1 THEN 1 END) AS commercials_count, COUNT(*) AS total_shots,
    > (COUNT(CASE WHEN label = 1 THEN 1 END) / COUNT(*)) * 100 AS commercials_percentage
    >
    > FROM (
    >
    >    SELECT 'BBC' AS channel, label FROM bbc_commercials
    >
    >    UNION ALL
    >
    >    SELECT 'CNN', label FROM cnn_commercials
    >
    >    UNION ALL
    >
    >    SELECT 'CNNIBN', label FROM cnnibn_commercials
    >
    >    UNION ALL
    >
    >    SELECT 'NDTV', label FROM ndtv commercials
    >
    >    UNION ALL
    >
    >    SELECT 'TIMESNOW', label FROM timesnow_commercials
    >
    > ) AS channel data
    >
    > GROUP BY channel;
Query ID = hadoopuser_20241029133425_a00270e4-8dc4-4734-bdca-9c453b009796
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application 1729485659280 0011)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 5 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
Map 7 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 179.98 s
--------------------------------------------------------------------------------
OK
```

Output:

```
BBC     8416    17721   47.491676541955876
CNN     14411   22546   63.91821165616961
CNNIBN  21693   33118   65.50214384926626
NDTV    12564   17052   73.68050668543279
TIMESNOW        25147   39253   64.06389320561485
Time taken: 185.362 seconds, Fetched: 5 row(s)
```

2. Average shot length of Commercials vs Non-Commercials

Query to compare average shot length of commercials and non commercials by channel:

```
hive> SELECT 'BBC' AS channel, label, AVG(Shot_Length) AS avg_shot_length
    >
    > FROM bbc_commercials GROUP BY label
    >
    > UNION ALL
    >
    > SELECT 'CNN', label, AVG(Shot_Length)
    >
    > FROM cnn_commercials GROUP BY label
    >
    > UNION ALL
    >
    > SELECT 'CNNIBN', label, AVG(Shot_Length)
    >
    > FROM cnnibn_commercials GROUP BY label
    >
    > UNION ALL
    >
    > SELECT 'NDTV', label, AVG(Shot Length)
    >
    > FROM ndtv_commercials GROUP BY label
    >
    > UNION ALL
    >
    > SELECT 'TIMESNOW', label, AVG(Shot Length)
    >
    > FROM timesnow_commercials GROUP BY label;
Query ID = hadoopuser_20241029134314_37ddb84d-53b2-4983-af49-7e0370919814
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0011)
--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 10 .........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
Map 8 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 11 .....   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 5 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 7 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 9 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 10/10  [==========================>>] 100%  ELAPSED TIME: 48.89 s
--------------------------------------------------------------------------------
```

Output:

```
BBC     NULL    NULL
BBC     -1      178.60715821152192
BBC     1       75.58471958174906
CNN     NULL    NULL
CNN     -1      226.57892795672487
CNN     1       63.84761640413573
CNNIBN  NULL    NULL
CNNIBN  -1      133.69704131652662
CNNIBN  1       62.73604388512423
NDTV    NULL    NULL
NDTV    -1      215.42455983953644
NDTV    1       60.80945558739255
TIMESNOW        NULL    NULL
TIMESNOW        -1      182.25962424672102
TIMESNOW        1       61.55092058694874
```

3. Visual Features Analysis

a. Query to observe average motion distribution in commercials by channel:

```
hive> SELECT 'BBC' AS Channel, AVG(Motion_Distribution_Mean) AS Avg_Motion_Distribution_Mean
    >
    > FROM bbc_commercials
    >
    > WHERE label = 1
    >
    > UNION ALL
    >
    > SELECT 'CNN' AS Channel, AVG(Motion_Distribution_Mean)
    >
    > FROM cnn_commercials
    >
    > WHERE label = 1
    >
    > UNION ALL
    >
    > SELECT 'CNNIBN' AS Channel, AVG(Motion_Distribution_Mean)
    >
    > FROM cnnibn_commercials
    >
    > WHERE label = 1
    >
    > UNION ALL
    >
    > SELECT 'NDTV' AS Channel, AVG(Motion_Distribution_Mean)
    >
    > FROM ndtv commercials
    >
    > WHERE label = 1
    >
    > UNION ALL
    > SELECT 'TIMESNOW' AS Channel, AVG(Motion_Distribution_Mean)
    >
    > FROM timesnow commercials
    >
    > WHERE label = 1;
Query ID = hadoopuser_20241029141604_b65f9027-5270-4948-9b31-41add67128e9
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1729485659280_0014)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 10 .........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
Map 8 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 11 .....   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 5 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 7 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 9 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 10/10  [==========================>>] 100%  ELAPSED TIME: 57.85 s
--------------------------------------------------------------------------------
OK
BBC     2.283927227067495
CNN     2.68072266678233
CNNIBN  2.6741484217950515
NDTV    2.5821041368990794
TIMESNOW        2.6132909043225805
Time taken: 91.713 seconds, Fetched: 5 row(s)
```

b. Query to observe average motion distribution in non commercials by channel:

```
hive> SELECT 'BBC' AS Channel, AVG(Motion_Distribution_Mean) AS Avg_Motion_Distribution_Mean
    >
    > FROM bbc_commercials
    >
    > WHERE label = -1
    >
    > UNION ALL
    >
    > SELECT 'CNN' AS Channel, AVG(Motion_Distribution_Mean)
    >
    > FROM cnn_commercials
    >
    > WHERE label = -1
    >
    > UNION ALL
    >
    > SELECT 'CNNIBN' AS Channel, AVG(Motion_Distribution_Mean)
    >
    > FROM cnnibn commercials
    >
    > WHERE label = -1
    >
    > UNION ALL
    >
    > SELECT 'NDTV' AS Channel, AVG(Motion Distribution Mean)
    >
    > FROM ndtv_commercials
    >
    > WHERE label = -1
    >
    > UNION ALL
    >
    > SELECT 'TIMESNOW' AS Channel, AVG(Motion_Distribution_Mean)
    >
    > FROM timesnow_commercials
    >
    > WHERE label = -1:
Query ID = hadoopuser 20241029141856 25fce70b-deb7-4b46-874c-4bd7c7f59416
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0014)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 10 .........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
Map 8 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 11 .....   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 5 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 7 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 9 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 10/10  [==========================>>] 100%  ELAPSED TIME: 45.54 s
--------------------------------------------------------------------------------
OK
BBC     1.8871365943680098
CNN     2.442601206663389
CNNIBN  3.113957629289213
NDTV    2.6414119768219333
TIMESNOW        2.5963781785182554
Time taken: 56.641 seconds, Fetched: 5 row(s)
```

c. Queries to export the motion distribution data to visualize in R.

```
hive> INSERT OVERWRITE DIRECTORY '/Project/bbc_motion_dist.txt'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > SELECT Motion_Distribution_Mean
    > FROM bbc_commercials
    > WHERE label = 1;
Query ID = hadoopuser_20241029144321_5fe76f5c-3e26-4878-9574-6c2182ef31cd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0014)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 12.02 s
--------------------------------------------------------------------------------
Moving data to directory /Project/bbc motion dist.txt
OK
Time taken: 15.919 seconds
hive> INSERT OVERWRITE DIRECTORY '/Project/bbc_motion_dist_nc'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > SELECT Motion Distribution Mean
    > FROM bbc_commercials
    > WHERE label = -1;
Query ID = hadoopuser_20241029144557_56abc834-f94e-41d7-9eea-4cebbffd4677
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application 1729485659280 0014)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 18.17 s
--------------------------------------------------------------------------------
Moving data to directory /Project/bbc_motion_dist_nc
OK
Time taken: 21.814 seconds
```

To visualize in R:

```
> commercials_bbc <- read.csv("C:/Users/HP/Downloads/bbc_com_motion_dist.csv", header = FALSE)
> non_commercials_bbc <- read.csv("C:/Users/HP/Downloads/bbc_ncom_motion_dist.csv", header = FALSE)
> colnames(commercials_bbc) <- c("Motion_Distribution_Mean")
> colnames(non_commercials_bbc) <- c("Motion_Distribution_Mean")
```

```
> par(mfrow = c(1, 2))
> hist(
+     commercials_bbc$Motion_Distribution_Mean,
+     main = "BBC Commercials",
+     xlab = "Motion Distribution Mean",
+     col = "blue",
+     border = "black",
+     breaks = 30
+ )
> hist(
+     non_commercials_bbc$Motion_Distribution_Mean,
+     main = "BBC Non-Commercials",
+     xlab = "Motion Distribution Mean",
+     col = "red",
+     border = "black",
+     breaks = 30
+ )
```

d. Queries to get average motion distribution by commercial/non commercial for some channels separately:

```
hive> SELECT label, AVG(Motion Distribution Mean) AS avg motion, AVG(Motion Distribution Variance) AS var motion
    > FROM timesnow_commercials
    > GROUP BY label;
Query ID = hadoopuser_20241029152554_b173edbc-8abd-4c62-a049-3c86f5e61669
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application 1729485659280 0015)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 13.81 s
--------------------------------------------------------------------------------
OK
NULL    NULL    NULL
-1      2.5963781785182554      1.4753203247075413
1       2.6132909043225805      1.7064962696544357
Time taken: 16.964 seconds, Fetched: 3 row(s)
```

```
hive> SELECT label, AVG(Motion_Distribution_Mean) AS avg_motion, AVG(Motion_Distribution_Variance) AS var_motion
    > FROM cnn_commercials
    > GROUP BY label;
Query ID = hadoopuser_20241029152916_f2c78e20-7c42-495f-8b58-a28ffffca025
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0015)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.43 s
--------------------------------------------------------------------------------
OK
NULL    NULL    NULL
-1      2.442601206663389       1.470050779690187
1       2.68072266678233        1.7636357805842922
Time taken: 14.022 seconds, Fetched: 3 row(s)
```

e. Query to analyze overall Edge Change Ratio of commercials and non commercials:

```
hive> SELECT
    >
    >     CASE
    >
    >         WHEN label = 1 THEN 'Commercials'
    >
    >         ELSE 'Non-Commercials'
    >
    >     END AS Category,
    >
    >     AVG(Edge_Change_Ratio_Mean) AS Avg_Edge_Change_Ratio,
    >
    >     COUNT(*) AS Count
    >
    > FROM (
    >
    >     SELECT label, Edge_Change_Ratio_Mean FROM bbc_commercials
    >
    >     UNION ALL
    >
    >     SELECT label, Edge_Change_Ratio_Mean FROM cnn_commercials
    >
    >     UNION ALL
```

```
    >     SELECT label, Edge_Change_Ratio_Mean FROM cnnibn_commercials
    >
    >     UNION ALL
    >
    >     SELECT label, Edge Change Ratio Mean FROM ndtv commercials
    >
    >     UNION ALL
    >
    >     SELECT label, Edge_Change_Ratio_Mean FROM timesnow_commercials
    >
    > ) AS combined data
    >
    > GROUP BY label;
Query ID = hadoopuser_20241029153514_6a5e6782-4236-4cd8-a556-e24df4eecfb6
Total jobs = 1
Query ID = hadoopuser_20241029153514_6a5e6782-4236-4cd8-a556-e24df4eecfb6
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0015)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 5 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
Map 7 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 3 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 41.39 s
--------------------------------------------------------------------------------
OK
Non-Commercials NULL    5
Non-Commercials 0.4993988751468954      47454
Commercials     0.501368805652172       82231
Time taken: 45.543 seconds, Fetched: 3 row(s)
```

4. Audio Features Analysis

```
hive> SELECT 'BBC' AS channel, AVG(Spectral_Centroid_Mean) AS avg_spectral FROM bbc_commercials WHERE label = 1
    > UNION ALL
    > SELECT 'CNN', AVG(Spectral_Centroid_Mean) FROM cnn_commercials WHERE label = 1
    > UNION ALL
    > SELECT 'CNNIBN', AVG(Spectral_Centroid_Mean) FROM cnnibn_commercials WHERE label = 1
    > UNION ALL
    > SELECT 'NDTV', AVG(Spectral_Centroid_Mean) FROM ndtv_commercials WHERE label = 1
    > UNION ALL
    > SELECT 'TIMESNOW', AVG(Spectral_Centroid_Mean) FROM timesnow_commercials WHERE label = 1;
Query ID = hadoopuser_20241029151425_6795dd30-0e22-4a7b-b355-67382638e7f4
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0015)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 10 .........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
Map 8 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 11 .....   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 5 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 7 ......   SUCCEEDED      1          1        0        0       0       0
Reducer 9 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 10/10  [==========================>>] 100%  ELAPSED TIME: 45.31 s
--------------------------------------------------------------------------------
OK
BBC     3007.973439206745
CNN     3559.590556083694
CNNIBN  3552.936531467902
NDTV    3547.455031438463
TIMESNOW        3550.7527164312337
Time taken: 51.707 seconds, Fetched: 5 row(s)
```

Queries to get Audio features by commercial/non-commercial:

```
hive> SELECT label, AVG(Spectral_Centroid_Mean) AS avg_spectral FROM bbc_commercials GROUP BY label;
Query ID = hadoopuser_20241029151956_5a702f76-5d82-437b-919a-5ab5bbd1245e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0015)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 12.58 s
--------------------------------------------------------------------------------
OK
NULL    NULL
-1      2566.099829233018
1       3007.973439206745
Time taken: 15.658 seconds, Fetched: 3 row(s)


hive> SELECT label, AVG(Spectral Roll Off Mean),AVG(Spectral Flux Mean) FROM bbc commercials GROUP BY label;
Query ID = hadoopuser_20241029152306_95cec6b1-bd60-4592-96bc-e0943fe482d4
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1729485659280_0015)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 17.70 s
--------------------------------------------------------------------------------
OK
NULL    NULL    NULL
-1      5066.249234672636       970.3155372892317
1       6002.718592541346       986.5677229711276
Time taken: 24.464 seconds, Fetched: 3 row(s)
```

5. Feature Importance analysis

After the analysis using Hive, we combined the data from all the channels and moved it into a csv file in HDFS. This file is brought to our computer storage for analysis using R. Query:

```
hive> CREATE TABLE tv_commercials_combined AS
    >
    > SELECT * FROM bbc_commercials
    >
    > UNION ALL
    >
    > SELECT * FROM cnn_commercials
    >
    > UNION ALL
    >
    > SELECT * FROM cnnibn_commercials
    >
    > UNION ALL
    >
    > SELECT * FROM ndtv_commercials
    >
    > UNION ALL
    >
    > SELECT * FROM timesnow_commercials;
Query ID = hadoopuser_20241029165343_219d4a17-387a-4487-a883-9f6c3e9fa7be
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1729485659280_0017)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 3 ..........   SUCCEEDED      1          1        0        0       0       0
Map 4 ..........   SUCCEEDED      1          1        0        0       0       0
Map 5 ..........   SUCCEEDED      1          1        0        0       0       0
Map 6 ..........   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 29.79 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/tv_commercials.db/tv_commercials_combined
Table tv_commercials.tv_commercials_combined stats: [numFiles=5, numRows=129690, totalSize=25267326, rawDataSize=25137636]
OK
Time taken: 45.275 seconds
```

The dataset is loaded into tv_commercials dataframe in the R environment. Then it is checked for duplicates as there might be repeated commercial shots in multiple channels which might affect the analysis by creating a bias. 18,063 duplicated rows (shots) are found and are removed.

```
> library(data.table)
> duplicates <- tv_commercials[duplicated(tv_commercials), ]
> print(nrow(duplicates))
[1] 18063
> tv_commercials <- unique(tv_commercials)
```

Then the data is checked for outliers by plotting boxplots of the features. These outliers could possibly deviate us from the generalized pattern that is to be observed and analyzed. The outliers can be analyzed in a separate analysis, but, here the focus is on finding the most important features that affect the majority of shots being classified as commercials or not. So, we proceed with analyzing the remaining data (inliers).

```
> remove_outliers <- function(x) {
+     Q1 <- quantile(x, 0.25)
+     Q3 <- quantile(x, 0.75)
+     IQR <- Q3 - Q1
+     x[x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)] <- NA
+     return(x)
+ }
> tv_commercials_clean <- tv_commercials %>% mutate(across(where(is.numeric),
remove_outliers)) %>% na.omit()
```

After removing the duplicates and outlier analysis, about 80,000 rows (shots) are present in the dataframe. With this dataframe correlation analysis is performed. The data is checked for highly correlated features. Features with high correlation (generally > 0.8) to other features are retrieved as shown below and are removed by keeping one feature from the highly correlated pair to retain the information.

```
> cor_matrix <- cor(tv_commercials[, -1], use = "complete.obs")
> corrplot(cor_matrix, method = "circle", tl.cex = 0.7)
> threshold <- 0.8
> high_corr_indices <- which(abs(cor_matrix) > threshold, arr.ind = TRUE)
> high_corr_indices <- high_corr_indices[high_corr_indices[, 1] != high_corr_indices[, 2], ]
> high_corr_features <- data.frame(Feature1 = rownames(cor_matrix)[high_corr_indices[, 1]],
+                                  Feature2 = colnames(cor_matrix)[high_corr_indices[, 2]],
+                                  Correlation = cor_matrix[high_corr_indices])
> print(high_corr_features)
                    Feature1                   Feature2 Correlation
1          Spectral_Flux_Mean     Short_Time_Energy_Mean   0.8434697
2      Spectral_Flux_Variance Short_Time_Energy_Variance   0.8740536
3      Spectral_Roll_Off_Mean     Spectral_Centroid_Mean   0.9721369
4  Fundamental_Frequency_Mean     Spectral_Centroid_Mean   0.8004824
5  Spectral_Roll_Off_Variance Spectral_Centroid_Variance   0.9864762
6      Spectral_Centroid_Mean     Spectral_Roll_Off_Mean   0.9721369
7  Spectral_Centroid_Variance Spectral_Roll_Off_Variance   0.9864762
8      Short_Time_Energy_Mean         Spectral_Flux_Mean   0.8434697
9      Spectral_Flux_Variance         Spectral_Flux_Mean   0.8166740
10 Short_Time_Energy_Variance     Spectral_Flux_Variance   0.8740536
11         Spectral_Flux_Mean     Spectral_Flux_Variance   0.8166740
12     Spectral_Centroid_Mean Fundamental_Frequency_Mean   0.8004824
```

```
> features_to_remove <- c("Spectral_Flux_Variance", "Short_Time_Energy_Mean", "Spectral_Roll_Off_Mea
n", "Fundamental_Frequency_Mean", "Spectral_Centroid_Variance")
> tv_commercials_clean <- tv_commercials_clean %>% select(-all_of(features_to_remove))
```

After correlation analysis the data frame contains 15 features with the dependent variable being 'label' and remaining variables, which should now be analyzed for their importance, being independent variables. After the preprocessing of data, to analyze the importance of features in classifying the shot as commercial or non commercial, a random forest model is built on the data and based on the random forest model, the feature importance is extracted.

```
> set.seed(123)
> rf_model <- randomForest(label ~ ., data = tv_commercials_clean, importance = TRUE)
> importance <- importance(rf_model)
> varImpPlot(rf_model)
```

The features are retrieved and listed below in order of their importance in the classification of the shots as commercials or non commercials

```
 [1] "Short_Time_Energy_Variance"
 [2] "Frame_Diff_Distribution_Mean"
 [3] "Motion_Distribution_Variance"
 [4] "Frame_Diff_Distribution_Variance"
 [5] "Motion_Distribution_Mean"
 [6] "ZCR_Mean"
 [7] "Spectral_Roll_Off_Variance"
 [8] "Spectral_Flux_Mean"
 [9] "ZCR_Variance"
[10] "Fundamental_Frequency_Variance"
[11] "Spectral_Centroid_Mean"
[12] "Shot_Length"
[13] "Edge_Change_Ratio_Mean"
[14] "Edge_Change_Ratio_Variance"
```

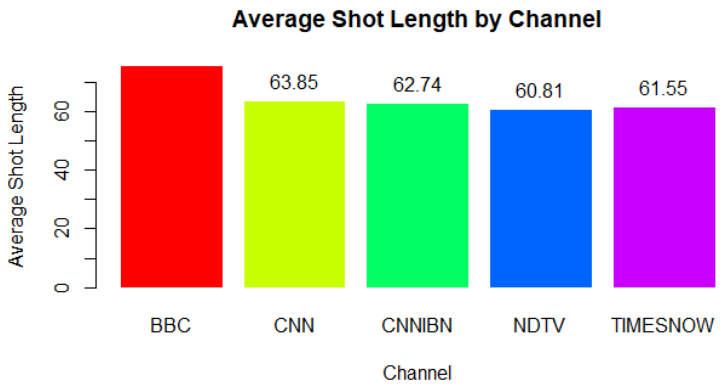
3. Discussion

The study started with processing the data and loading it into Hive tables in the Hadoop ecosystem. It proceeded towards analyzing the visual and audio features of the commercials and non commercials across the channels. The feature importance analysis is also performed.

3.1 Initial Insights

The analysis of commercials across various news channels reveals the variations in the proportion of commercial shots relative to the total broadcast content. BBC features commercials in 47.5% of its shots, suggesting a relatively lower commercial density compared to other networks. Conversely, CNN and TIMESNOW have similar commercial proportions, both registering around 64%, indicating a high degree of commercial presence. The data for CNNIBN and NDTV further emphasize this trend, with CNNIBN showing a commercial shot percentage of 65.5%, while NDTV leads with the highest commercial saturation at 73.6%. These insights reflect how different broadcasting strategies affect the airtime devoted to commercials, with Indian channels like NDTV showing a pronounced reliance on commercial content. This contrast highlights the distinct commercial programming patterns between international and Indian news channels.

The comparative analysis of shot lengths between commercial and non-commercial content across various news channels uncovers a clear pattern. For all channels examined, commercial shots are significantly shorter than non-commercial ones, reflecting the fast-paced, attention-grabbing nature of advertisements.



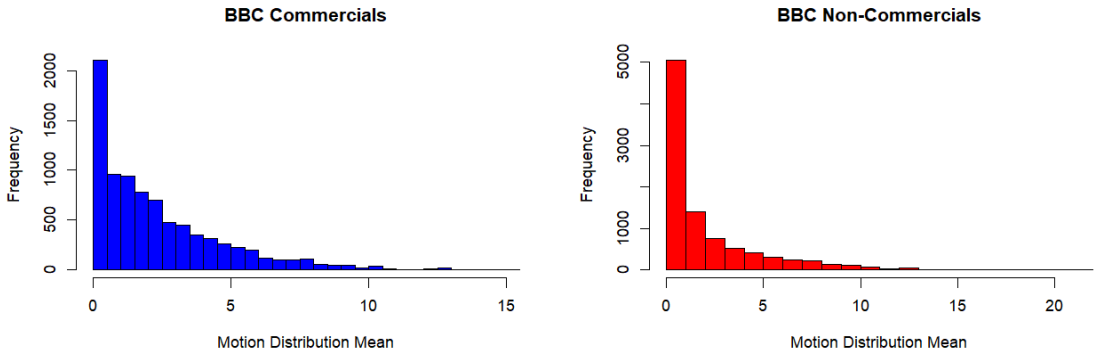
BBC's commercials have an average shot length of 75.6 frames, a notable contrast to its longer non-commercial shots averaging 178.6 frames. CNN exhibits an even greater disparity, with non-commercial shots averaging 226.6 frames compared to commercial shots at 63.8 frames. Similarly, CNNIBN, NDTV, and TIMESNOW show consistent trends, with commercial shots averaging around 60 to 63 frames, while non-commercial shots are markedly longer, ranging from 133.7 to 215.4 frames. These differences highlight the distinct pacing and structure between editorial content and advertisements, with commercials designed for quick visual impact.

3.2 Visual and Audio Features analysis

Various visual and audio features of the commercial and non commercial shots are analyzed and compared across different channels.

3.2.1 Visual

The analysis of motion distribution between commercials and non-commercial content reveals notable trends across the examined channels.



The histograms for BBC, depicted in the visualizations, illustrate that commercials exhibit a slightly higher average motion distribution mean (2.28) compared to non-commercial content (1.88). This indicates that commercials are generally characterized by more dynamic and visually engaging sequences, which likely aim to capture viewers' attention. Similar patterns are observed across other channels: CNN shows an increase from 2.44 in non-commercials to 2.68 in commercials, while NDTV and TIMESNOW demonstrate modest changes from 2.64 to 2.58 and 2.5 to 2.61, respectively. Interestingly, CNNIBN deviates slightly, with non-commercial content averaging a higher motion distribution (3.11) than commercials (2.67). These variations highlight that, while commercials typically feature more motion to maintain viewer engagement, some channels may adopt distinctive content structuring strategies, reflecting differences in editorial and advertising practices.

Further, the analysis of Edge Change Ratio (ECR) across all channels highlights key differences in the visual dynamics between commercials and non-commercial segments. The overall ECR for commercials is higher, compared to non-commercial content. This indicates that commercials are visually more dynamic, featuring frequent and pronounced changes in edges, likely to keep viewers engaged and enhance the impact of the advertising content.

These findings underscore how commercials leverage rapid motion an d visual transitions to draw attention, contrasting with the relatively more stable and less visually disruptive nature of non-commercial content, such as news reporting or editorial segments. This pattern reflects a strategic emphasis in advertising on visually stimulating techniques designed to captivate audiences quickly.

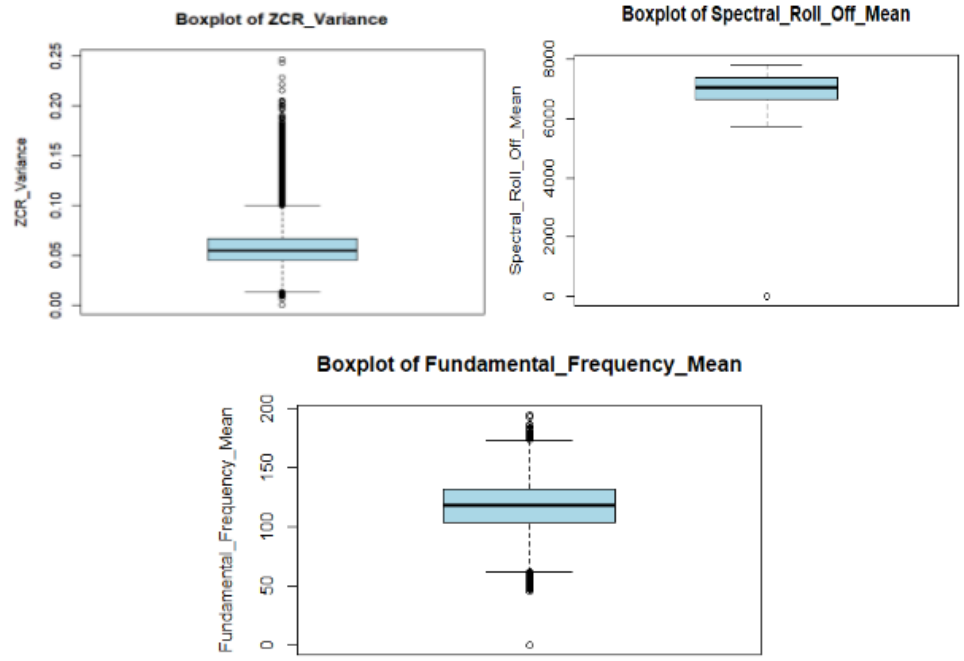
3.2.2 Audio

The analysis of audio features reveals significant distinctions between commercials and non-commercial segments, particularly in terms of spectral properties. The spectral centroid mean, which reflects the "brightness" of audio, is consistently higher for commercials across all channels, indicating a trend toward using more sonically engaging and attention-grabbing audio in advertisements. For example, BBC commercials have a spectral centroid mean of 3008, compared to lower means for non-commercial content, which stand at 2566. Additionally, other audio characteristics, such as spectral roll-off and spectral flux, show notable differences on BBC. Commercials exhibit a higher spectral roll-off value of 6002.7 compared to 5066 for non-commercials, suggesting that commercials feature a broader range of higher frequencies. Similarly, spectral flux, which measures the rate of change in the audio spectrum, is slightly higher for commercials (986.5) than non-commercials (970), indicating more frequent and dynamic audio variations.

These observations suggest that commercials are crafted to be audibly distinct, utilizing brighter, more dynamic sound profiles to captivate the audience, unlike the more stable and less varied audio patterns of non-commercial content.

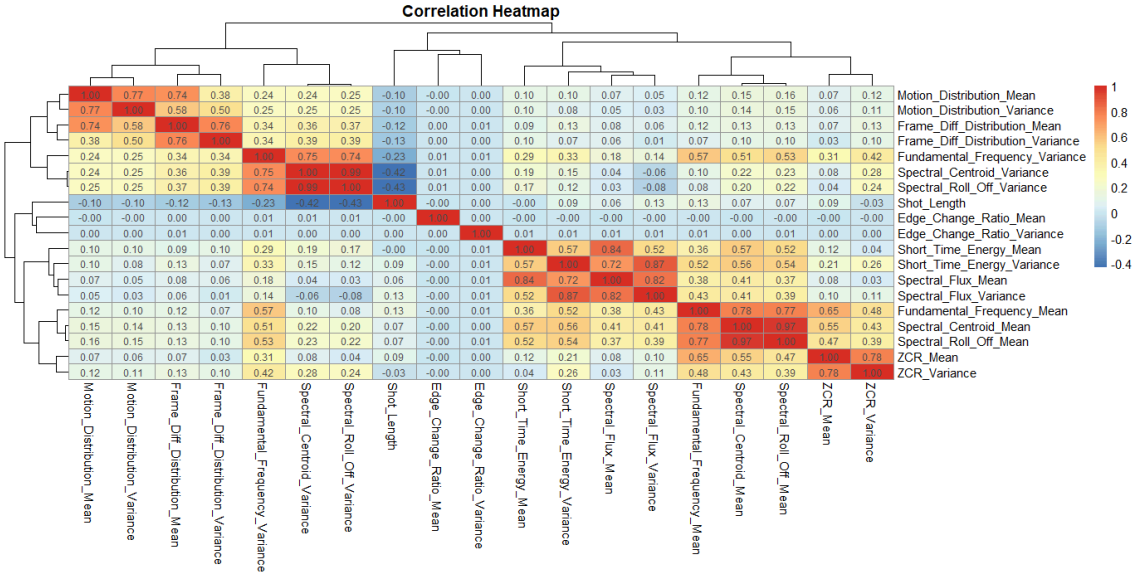
3.3 Feature importance Analysis

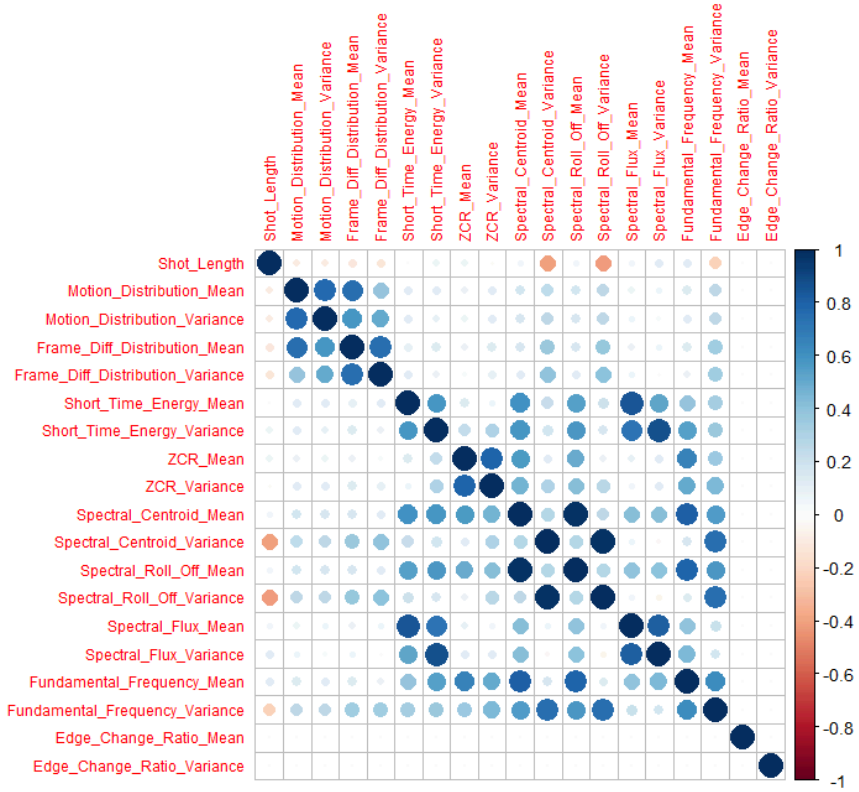
For understanding the important features that distinguish between commercials and non commercials, we combined the data of all channels using Hive and downloaded the data in csv format to the local computer system to perform analysis using R. Duplicated rows (shots) from the data are removed as there can be multiple commercial shots across multiple channels that can be the same. Then boxplots for the columns are plotted to observe the outliers. Some boxplots are shown below.



Outliers are observed in most of the columns. These might deviate our general analysis of the features' importance on the commercial classification. As we have a significant amount of data and these outliers can be handled separately in another analysis and are hence removed to proceed with this analysis.

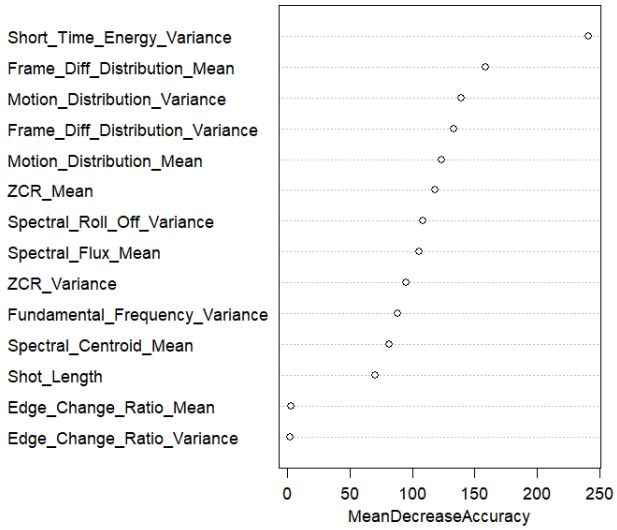
Further, correlation analysis is performed by plotting the correlation heatmaps as shown below.





From observing these heatmaps and analyzing the correlated features, an observation is made and the "Spectral_Flux_Variance", "Short_Time_Energy_Mean", "Spectral_Roll_Off_Mean","Fundamental_Frequency_Mean","Spectral_Centroid_variance" features are removed as they are highly correlated and provide the same information to the model as their correlated counterparts. Hence, the data frame is left with 15 features and around 80,000 rows.

With the preprocessed data, a random forest model is built. The Random Forest model was chosen for this analysis due to its strong ability to handle complex, high-dimensional data, making it highly effective for the diverse audio-visual features present in the TV commercials dataset. Given the large number of attributes extracted from the commercials and non-commercials, Random Forest can efficiently manage interactions among these features without needing extensive pre-processing or selection. Its ensemble approach, which aggregates results from multiple decision trees, enhances predictive performance while minimizing the risk of overfitting, providing more reliable generalization on unseen data. Additionally, Random Forest offers the advantage of built-in feature importance assessment, allowing us to pinpoint which features significantly influence the classification between commercial and non-commercial content. This aspect is crucial for understanding which visual or auditory attributes contribute most to distinguishing between these segments. The model's ability to capture non-linear relationships further ensures that any complex patterns within the data are effectively represented, enhancing the overall analytical capability of the study. Using the random forest model, the importance of features is visualized as shown below.



From the observations made using the random forest model, it can be concluded that the top most contributors that distinguish between commercial and non commercial shots are the Short Time Energy, Frame Difference Distribution, Motion Distribution while the edge change ratio is insignificant. High Short Time Energy values, for instance, imply that commercials tend to use more intense audio bursts to grab

viewer attention, while Frame Difference Distribution and Motion Distribution suggest that commercials generally have more rapid and dynamic visual changes compared to the steadier content of non-commercial programming. In contrast, the insignificance of the Edge Change Ratio suggests that abrupt changes in edges or contours are less influential in distinguishing between these content types.

These insights are highly valuable for media analysts, advertisers, and broadcasters. Advertisers can leverage this information to design commercials that are more engaging by understanding the visual and auditory elements that are most impactful. Media companies and broadcasters can utilize this knowledge to optimize content segmentation and ensure that advertising slots maintain high viewer engagement. Additionally, developers of video analytics software can use these feature importance insights to improve automated content classification systems, making them more efficient in separating and categorizing different types of broadcast material.

4. Concluding Remarks

This project "TV Commercial Data Analysis" successfully demonstrated the effective integration of Hive for data processing and R for data visualization and advanced analytical modeling. By utilizing the TV commercials dataset from the UCI Machine Learning Repository, we started a comprehensive exploration of commercial and non-commercial shots from various channels. The data processing phase involved cleaning and formatting the dataset to ensure compatibility with Hive, followed by the creation of structured tables that facilitated efficient querying and analysis.

Through the application of Hive queries, we gained valuable insights into the visual and audio features inherent in the commercials and non-commercials. This analysis not only revealed patterns across different channels but also highlighted the distinguishing characteristics that separate commercial content from non-commercial content. By aggregating the data from all channels, we developed an overall view of the features that are most influential in classification tasks. Our work also involved the implementation of a Random Forest model in R, which served as a powerful tool for feature importance analysis. This step was critical in identifying which attributes had the most significant impact on differentiating commercials from non-commercials. The results underscored certain features such as duration, audio tone, and visual elements that emerged as pivotal in shaping viewer perceptions and engagement with commercial content.

In conclusion, our project not only showcased the capabilities of Hive and R in handling and analyzing large datasets but also contributed meaningful insights into the television advertising landscape. The findings have potential implications for advertisers and content creators, suggesting that an emphasis on specific visual and audio characteristics could enhance the effectiveness of commercials.

5. Future Work

The insights gained from our analysis of TV commercials provide a valuable foundation, yet there remains significant potential for expanding this research. Future work could focus on several key areas to enhance our understanding of the dynamics that play within television advertising.

First and foremost, a more comprehensive approach to data collection would greatly benefit the analysis. By incorporating a wider array of datasets, including additional channels, genres, and time periods, we can enrich our analysis. This could involve aggregating data from various sources beyond the UCI ML Repository, such as scraping data from broadcasting networks or utilizing APIs that track advertisements. Such extensive data collection would not only improve the robustness of our findings but also allow for a more comprehensive analysis of trends and patterns in advertising strategies. Another area for exploration is the clustering of commercials into different types based on their features and objectives. By applying clustering algorithms, we can categorize commercials into segments such as emotional appeal, humor, informational, and celebrity endorsement. This segmentation would provide a clearer picture of how different styles resonate with audiences and how they impact engagement and conversion rates. Understanding these clusters could lead to targeted advertising strategies that align with specific viewer preferences.

Moreover, the integration of additional analytical techniques could further deepen our insights. Future research could explore advanced machine learning models beyond random forests, such as gradient boosting machines or neural networks, which may uncover more intricate relationships within the data. There is an opportunity to explore related works that examine the impact of cultural, social, and economic factors on advertising effectiveness. Investigating how different demographics respond to various commercial types could yield valuable information for advertisers aiming to tailor their strategies for specific audience segments.

References

1. https://archive.ics.uci.edu/dataset/326/tv+news+channel+commercial+detection+dataset
2. https://www.javatpoint.com/r-tutorial
3. https://r4ds.had.co.nz/data-visualisation.html
4. Programming, Hive, O'Reily publications
5. Cohen, J., & Golden, L. (2016) "Visual Features in Television Commercials and Their Impact on Audience Engagement." Television & New Media, 17(5), 444-461. doi:10.1177/1527476415581007.
6. Huang, J., & Sarigöllü, E. (2014). "The Effect of Television Advertising on Consumer Purchase: Evidence from the Food and Beverage Sector." Journal of Marketing Management, 30(1-2), 35-54. doi:10.1080/0267257X.2013.837029.
7. Kumar, A., & Gupta, A. (2016). "Analyzing the Impact of Commercials on Buying Behavior through Machine Learning Techniques." International Journal of Marketing Studies, 8(4), 63-75. doi:10.5539/ijms.v8n4p63.
8. T. Can and P. Duygulu, "Detection and Tracking of TV Commercials," 2007 IEEE 15th Signal Processing and Communications Applications, Eskisehir, Turkey, 2007, pp. 1-4, doi: 10.1109/SIU.2007.4298593
9. S. -H. Yang, C. -W. Fan and Y. -C. Chen, "An improved automatic commercial detection system," 2011 Visual Communications and Image Processing (VCIP), Tainan, Taiwan, 2011, pp. 1-4, doi: 10.1109/VCIP.2011.6115993.