Sports Articles Analysis

Abstract:

This research transitions from a supervised learning paradigm to an unsupervised learning methodology in order to scrutinize sports-related media content from December to March. By utilizing metadata encompassing article titles, publishing platforms, posting timestamps, content classifications, and the identification of athlete names, we developed a comprehensive array of features that encapsulate both textual and temporal aspects of audience engagement. The dataset underwent preprocessing and enhancement through customized algorithms designed to identify stylistic indicators (e.g., inquiries, quotations, numerical data) and to correlate athlete references across multiple postings. Subsequently, KMeans clustering was employed to categorize the articles into distinctive content archetypes. Through exploratory visualizations and cluster analyses, we elucidate patterns that indicate the platforms, types of articles, and timing intervals that correlate with elevated engagement metrics. The outcomes yield practical insights for media strategists, underscoring the significance of content style, temporal considerations, and athlete relevance in enhancing outreach and influence.

Data Summary:

The dataset contains sports-related articles posted between December and March. Each record includes the article title, number of views, category, platform source, and the post's date and time. Another dataset provides athlete names to tag player mentions.

Feature Engineering:

Feature engineering played a central role in transforming raw sports media metadata into a structured format suitable for clustering analysis. This phase involved extracting and crafting meaningful variables from existing data columns, enhancing both the expressiveness and predictive capacity of the dataset. Below is a detailed overview of the steps taken:

---

1. Temporal Feature Extraction

From the post_date and post_time columns, several derived time-based features were created to capture behavioral patterns related to content timing:

- weekday: Extracted the day of the week (e.g., Monday, Tuesday) to detect performance differences by day.
- time_bin: Discretized the posting time into 2-hour intervals, effectively creating categorical segments such as 08:00-10:00, 14:00-16:00, etc. This allowed us to observe which time windows garnered more engagement.

These features were useful in identifying patterns of optimal posting times across platforms.

---

2. Text-Based Feature Engineering

The Title column, rich in editorial cues and engagement triggers, was a primary source for custom text features. Using rule-based pattern detection, the following attributes were extracted:

- title_length: Character count of the title, often linked with click-through rates.
- title_word_count: Number of words in the title.
- has_quote: Boolean indicating the presence of quotation marks (") – often used in player or coach quotes.
- has_question: Boolean capturing the presence of a question mark (?), commonly used in opinion pieces or speculative articles.
- has_number: Identifies whether the title contains numeric values – useful for rankings, stats, or listicles.

These engineered features help in quantifying the editorial style and intent behind the titles, which are known to impact viewer attention and engagement.

---

3. Athlete Name Tagging (Entity Presence Feature)

Using the list of athletes provided in players.xlsx, a matching process was performed:

- Player: If a player's name was mentioned in the Title, the player's name was tagged in a new column.
- This binary tagging strategy created a proxy for personalization, where the impact of name mentions on views could be studied.
- Though currently limited to single matches, it forms a foundation for more advanced NLP-based Named Entity Recognition (NER) in future work.

4. Categorical Encoding

To prepare the dataset for machine learning models:

- Categorical variables such as Source, category, article_type, weekday, time_bin, and Player were transformed using One-Hot Encoding via a ColumnTransformer.
- This created sparse binary features for each unique category, allowing KMeans clustering to work effectively in high-dimensional space.

---

5. Final Feature Set

The final dataset consisted of both numerical and encoded categorical features, offering a balanced representation of content characteristics:

- Numerical: views, title_length, title_word_count
- Boolean/Textual: has_quote, has_question, has_number
- Categorical (Encoded): Source, category, article_type, weekday, time_bin, Player

This feature matrix served as the input to the clustering model, enabling the discovery of distinct groupings within the media content landscape.

Analysis Breakthrough:

The analysis started with a traditional regression analysis trying to predict number of views based on meta data mentioned above. The analysis started with trying to fit in Title as variable to gauge clickbait effect or title attraction.

First method of catching 'Title' as a variable was using simple feature extraction using TfidfVectorizer after preprocessing it for NLP.

The Regression results for the same were:

The regression model comparison chart shows that **Lasso Regression, Elastic Net, Dummy Regressor, and Lasso Least Angle Regression** performed identically with the lowest MAE (1.2737), MSE (2.5867), RMSE (1.5976), and RMSLE (0.2327), but with a poor R^2 score of -0.0307, indicating they didn't explain variance better than the mean predictor. **Random Forest Regressor** had the lowest MAPE (0.2389) but performed worse in terms of RMSE and R^2. **Extra Trees Regressor** had the worst performance among reasonable models with high RMSE (1.9999) and R^2 (-0.6461), while **Least Angle Regression (lar)** produced nonsensical, extreme values, indicating a model failure. Overall, none of the models provided strong predictive power (all R^2 scores are negative), but simpler models performed comparably to more complex ones.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lasso	Lasso Regression	1.2737	2.5867	1.5976	-0.0307	0.2327	0.2392	0.3240
en	Elastic Net	1.2737	2.5867	1.5976	-0.0307	0.2327	0.2392	0.0990
dummy	Dummy Regressor	1.2737	2.5867	1.5976	-0.0307	0.2327	0.2392	0.0980
llar	Lasso Least Angle Regression	1.2737	2.5867	1.5976	-0.0307	0.2327	0.2392	0.0950
br	Bayesian Ridge	1.2758	2.5930	1.5997	-0.0336	0.2330	0.2396	0.1610
gbr	Gradient Boosting Regressor	1.2792	2.7203	1.6425	-0.0981	0.2382	0.2392	0.4830
rf	Random Forest Regressor	1.3095	3.0074	1.7266	-0.2123	0.2518	0.2389	2.5300
lightgbm	Light Gradient Boosting Machine	1.3640	3.0569	1.7378	-0.2343	0.2562	0.2563	0.2500
knn	K Neighbors Regressor	1.3833	3.1712	1.7739	-0.2866	0.2608	0.2594	0.1180
ridge	Ridge Regression	1.4142	3.2059	1.7847	-0.3099	0.2657	0.2665	0.1310
ada	AdaBoost Regressor	1.5764	3.1877	1.7823	-0.3139	0.2726	0.3309	0.5840
et	Extra Trees Regressor	1.4902	4.0300	1.9999	-0.6461	0.2875	0.2778	5.5890
omp	Orthogonal Matching Pursuit	1.5409	4.0478	2.0085	-0.6865	0.3027	0.2878	0.0990
huber	Huber Regressor	1.7263	4.8082	2.1883	-0.9988	0.3455	0.3303	0.4000
dt	Decision Tree Regressor	1.6380	4.7539	2.1765	-1.0024	0.3132	0.3063	0.1460
par	Passive Aggressive Regressor	1.8863	5.7722	2.3964	-1.3743	0.3890	0.3634	0.1480
lr	Linear Regression	2.0025	6.6816	2.5777	-1.7484	0.4289	0.3873	0.6950
lar	Least Angle Regression	209924766.3619	9267845585009149952.0000	963067083.0260	-3728803005081905664.0000	4.9498	39059522.7557	0.1930

Since this performed poorly, another attempt was taken and this time with using more robust NLP technique of Sentence Transformer and the results for that was also as follows:

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
par	Passive Aggressive Regressor	1882.0337	227774738.9067	9173.0888	-0.0422	1.9003	2.2046	0.1630
huber	Huber Regressor	2133.5494	227914596.0810	9233.4650	-0.0962	2.0701	5.7841	0.1820
dummy	Dummy Regressor	2896.7517	226210636.2500	9185.6567	-0.1137	2.7464	18.6859	0.0480
en	Elastic Net	2882.4048	225656834.4500	9179.8970	-0.1300	2.6810	17.8336	0.0480
br	Bayesian Ridge	2978.1270	227586662.3000	9345.6261	-0.2843	2.6131	17.9037	0.1430
ada	AdaBoost Regressor	3484.8369	227100237.8582	9465.9736	-0.3722	2.9431	25.6494	0.3630
lightgbm	Light Gradient Boosting Machine	4449.7353	245240444.9030	10678.6379	-1.3937	3.0279	32.4470	0.8430
et	Extra Trees Regressor	3038.4865	243191518.4897	10849.9188	-3.0585	2.4515	18.2229	2.3590
knn	K Neighbors Regressor	3024.9797	264681900.0000	11313.2330	-4.7923	2.2604	16.9265	0.0590
ridge	Ridge Regression	6353.0453	280740340.4000	13116.4303	-5.0316	3.3241	52.1466	0.0490
rf	Random Forest Regressor	3795.7609	302582622.0809	13159.6432	-6.5698	2.5564	25.6322	13.8760
omp	Orthogonal Matching Pursuit	6270.1269	304300022.4000	14071.0823	-7.6067	3.3063	51.5286	0.0520
lasso	Lasso Regression	15889.3371	622679084.8000	23895.7398	-36.1405	4.3295	154.9000	0.0790
gbr	Gradient Boosting Regressor	4398.1488	686914040.0409	20021.6677	-42.5158	2.3649	29.6489	1.2800
dt	Decision Tree Regressor	5508.7006	901035968.2743	24118.2951	-86.6815	2.5309	42.6662	0.4960
llar	Lasso Least Angle Regression	887029.2954	506368679321200.0000	7133335.2227	-4769470.1797	3.9926	8778.8214	0.1310
lr	Linear Regression	29017929.7689	65857912053305032.0000	178424691.9160	-4530928030.8789	5.5421	323013.2922	0.0540

the **Passive Aggressive Regressor** outperformed all others with the lowest MAE (1882.03), MSE (2.28e+08), RMSE (9173.09), RMSLE (1.90), and MAPE (2.20), albeit with a still slightly negative R^2 (-0.0422), indicating limited predictive power. The **Huber Regressor** followed closely, showing moderate performance with better robustness (MAE: 2133.55, R^2: -0.0962). In contrast, models like **Linear Regression**, **Lasso**, and **Lasso Least Angle Regression** performed extremely poorly, with astronomical error values and highly negative R^2 scores, suggesting model instability or divergence. Complex models like **Gradient Boosting**, **Random Forest**, and **Extra Trees** did not outperform simpler approaches in this case, with poor R^2 values and high error metrics, likely due to overfitting or data issues. Overall, simpler linear models like Passive Aggressive and Huber proved more effective and stable for this particular dataset.

Note : for the most promising model's hyper-parameter tuning was also tried to capture nitty-gritties and reach the best model but the evidence clearly pointed towards low predictive power of all regression models.

To capture non-linear relationships, Neural Network Model was also tested which again performed badly with an $R^2 = 0.00033668668865730511.$

The constant Failure Regression for the dataset led to the motivation of carrying out clustering to identify patterns in data and hence communicate a story.

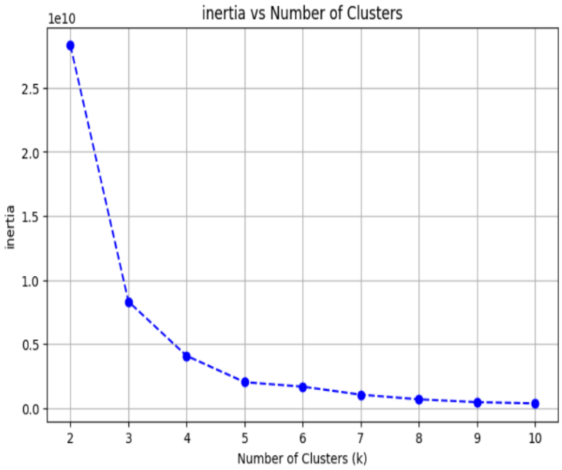
Clustering:

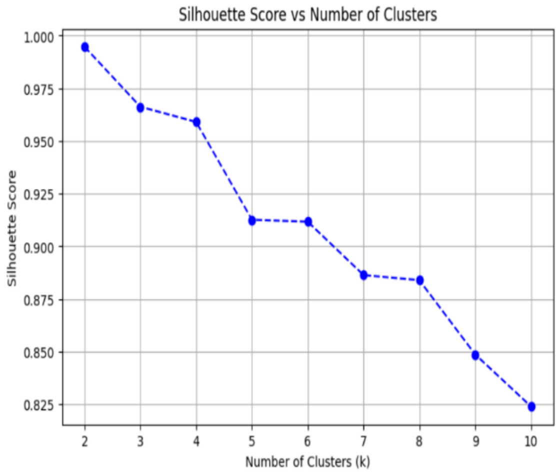
The primary algorithm used was, KMeans Clustering

- A classic partitioning method that forms k clusters by minimizing within-cluster variance.

To select the ideal k, the **Silhouette Score and Inertia** waere calculated for different values of k (typically ranging from 2 to 10):

The optimal K chosen primarily was 5, but similarity in clusters in terms of views and low frequency suggested us to move towards clubbing cluster 1,2 and 3 and we obtain three clusters say A,B,C





Cluster Characterization

After clustering, articles were assigned a cluster label. Each cluster was then analyzed for distinguishing features:

Examples of cluster-based insights:

- Cluster A: High-view articles, frequent use of numbers in titles, posted during weekday mornings, often tagged with high-profile athletes.
- Cluster B: Short titles, question-heavy articles, low views, primarily from Twitter.
- Cluster C: YouTube podcast content, longer titles, posted late evenings, associated with categories like NBA or NFL.

This segmentation provided editorial personas or content archetypes, revealing what styles and timing combinations are most effective.

Each cluster can be interpreted as a **"profile" of content strategy":**

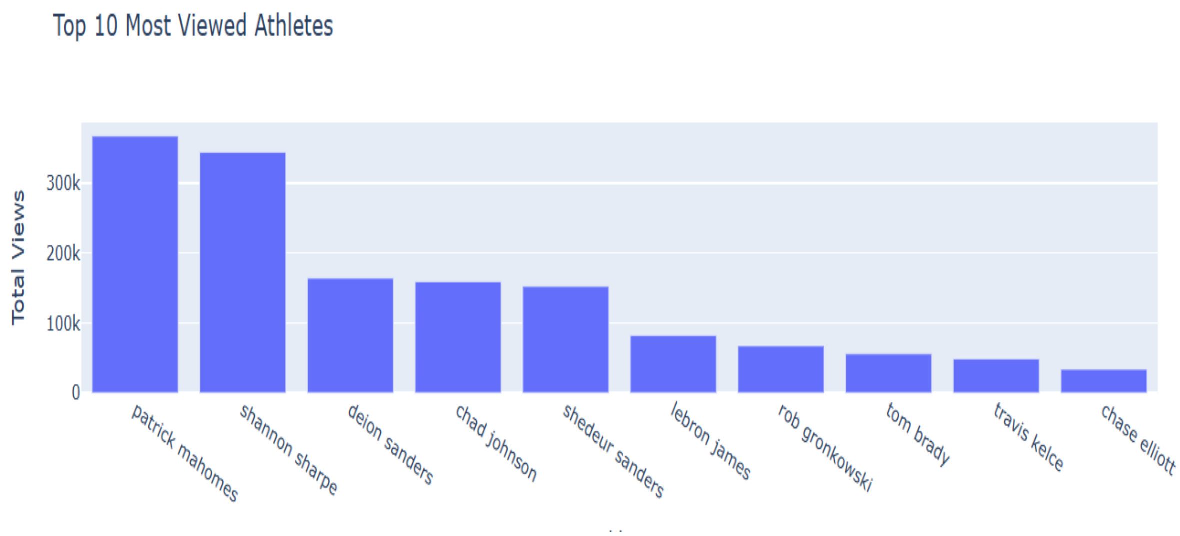
- Cluster A has the fewest entries (25) but the highest average views (~46,530), indicating it includes highly viral content. These titles are longer, more descriptive (highest title_word_count and title_length), and post during peak times (Friday evenings, 20–22). The content typically lacks questions and quotes, but includes some numbers and modal or most frequent athlete mentions is *Shannon Sharpe*.
- Cluster B is the largest group (1116 entries) but has the lowest average views (~300), likely representing routine content. It tends to be slightly shorter in length and word count, with moderate use of questions, quotes, and numbers.
- Cluster C (66 entries) has higher average views than Cluster B (~7701) and leans toward quote-heavy, number-including content. It posts mostly on Monday early

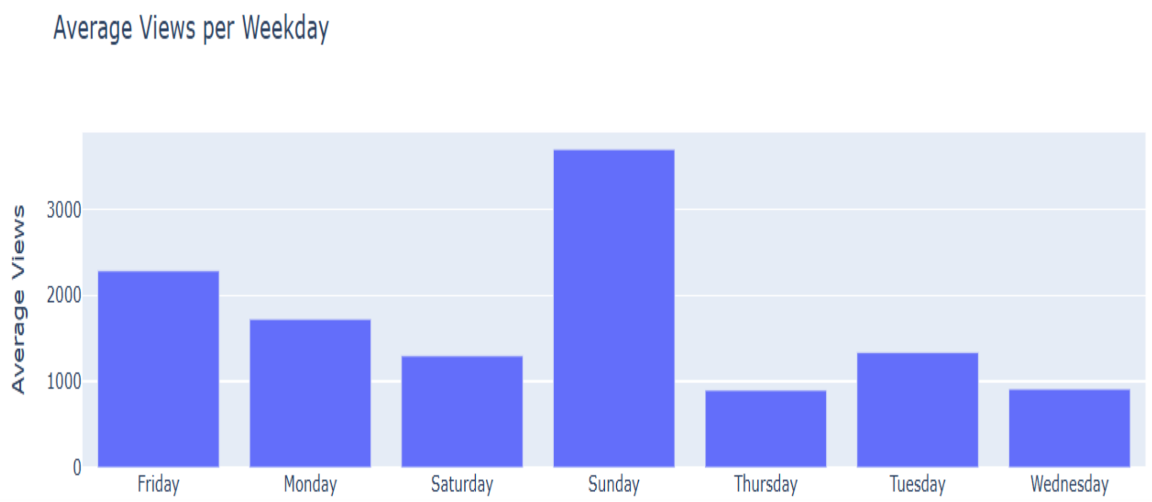
mornings, and its titles suggest dramatic or standout commentary (e.g., “We Could See That From a Mile Away”).

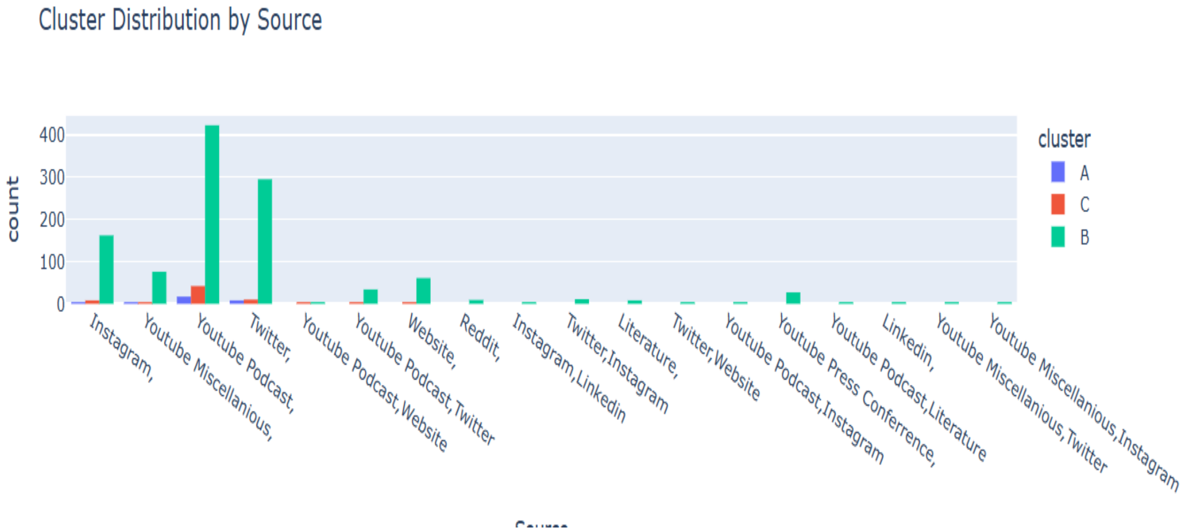
All clusters mode category is NFL, mention Shannon Sharpe, and are sourced from YouTube Podcasts, focused on Core Sport/On Court topics

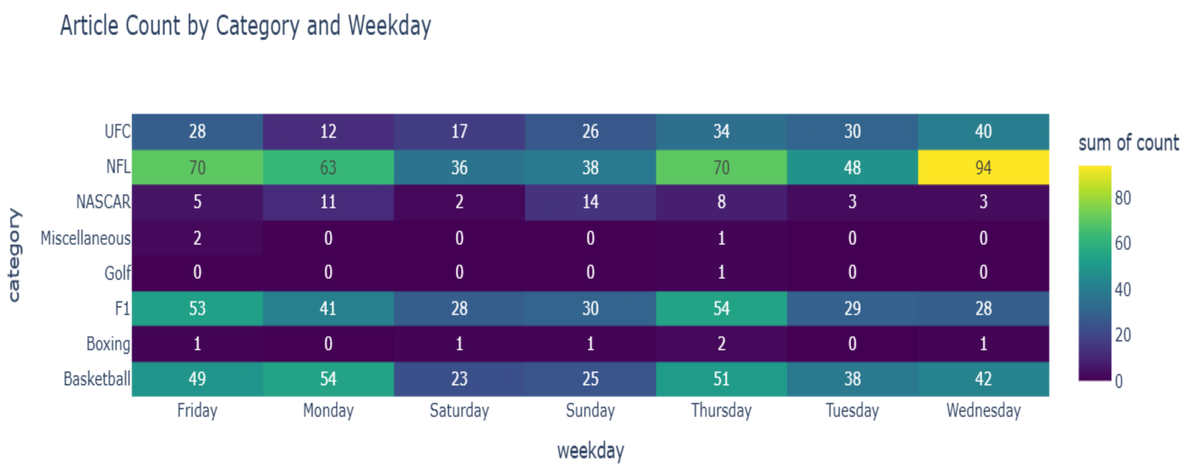
cluster	Count	Views	title_length	has_quote	has_question	has_number	title_word_count	post_time	category	Source	article_type	weekday	time_bin	Title	athlete_mentions
A	25	46530.280000	114.400000	0.120000	0.000000	0.320000	19.600000	11.440000	NFL	Youtube Podcast,	Core Sport/On Court	Friday	20-22	Shannon Sharpe Claims LeBron James' Total Poin...	shannon sharpe
B	1116	300.439964	105.198925	0.106631	0.060036	0.286738	17.019713	11.805556	NFL	Youtube Podcast,	Core Sport/On Court	Thursday	00-02	Despite Lakers’ Defensive Struggles, Shannon S...	lewis hamilton
C	66	7701.287879	114.696970	0.196970	0.000000	0.454545	18.863636	10.530303	NFL	Youtube Podcast,	Core Sport/On Court	Monday	00-02	“We Could See That From a Mile Away”: Chad Joh...	shannon sharpe

Visualizations:









The visual analyses reveal key patterns in article engagement and content distribution. NFL, Basketball, and F1 dominate article coverage across weekdays, with NFL showing particularly high volume on Wednesdays and Fridays. Cluster B is the most prevalent across content sources, especially on platforms like Twitter, Instagram, and YouTube Podcasts, while Clusters A and C are significantly less represented. In terms of audience engagement, Sundays garner the highest average views, followed by Fridays, indicating optimal publishing days. Furthermore, articles featuring athletes such as Patrick Mahomes, Shannon Sharpe, and Deion Sanders attract the most attention, highlighting their strong influence on readership metrics.

Limitations & Future Scope:

- *Player Matching*: Limited to exact string matches; NLP-based entity recognition can improve accuracy.
- *Context:* Actual article content (body text) not analyzed—future work can include semantic embeddings.
- *Platform-Specific Clustering*: Could cluster separately for each platform to find optimized strategies per channel.

Conclusion:

This study illustrates the challenges and opportunities in analyzing sports media content using machine learning techniques. While initial attempts at view prediction through regression—despite applying both traditional and advanced NLP embeddings—proved largely ineffective due to weak signal strength and low variance explanation, the pivot to clustering provided more actionable insights. The unsupervised approach, leveraging KMeans and engineered features, successfully segmented articles into meaningful archetypes. These clusters unveiled distinct editorial patterns tied to engagement metrics, highlighting the importance of timing, stylistic choices, and athlete mentions. Cluster A, though rare, represented high-performing content, while Cluster B captured the routine media churn, and Cluster C offered a middle ground with thematic depth and moderate traction.

These insights offer valuable guidance for media strategists seeking to optimize content delivery. The findings advocate for a data-informed approach to content planning, one that factors in not just audience preferences but also platform dynamics and timing. Future enhancements could involve integrating full-text analysis, deeper entity recognition, and platform-specific models to further refine content strategy and reader impact.

Appendix:

Attaching my code files and dataset for reproducibility:
https://github.com/NamanDudhoria/Sports-Article-Analysis