Sintia Stabel
sstabel@syr.edu
IST – 736 Text Mining Final Project
Sklearn TF-IDF vectorizer
Sklearn Vader Sentiment Intensity Analyzer
Sklearn Random Forest
Sklearn Support Vector Machines
Latent Dirichlet Allocation (LDA) – Topic Modeling

September 9th, 2023
Alternate topic:

"Unlocking the NBA Draft Code: A Data-Driven Approach to Predicting Rookie Success"

Introduction:

The NBA draft stands as a pivotal juncture in the realm of professional basketball, shaping the destinies of young, aspiring players. It represents a fusion of talent, strategy, and high-stakes decision-making, where the future of both players and teams' teeters on the edge of uncertainty.

In recent years, the landscape of NBA draft analysis has undergone a profound transformation, driven by advanced data analytics and cutting-edge technologies. Amid this transformative wave, two formidable tools have risen to prominence: sentiment analysis and text mining. These tools have become invaluable companions to teams, scouts, analysts, and enthusiasts alike.

Consider the power of unraveling the opinions and underlying narratives that envelop each draft prospect. Sentiment analysis grants us this remarkable ability. By scrutinizing the nuances of tonality, emotion, and context woven into diverse texts—ranging from scouting reports and interviews to media coverage—we gain access to profound insights into the strengths and vulnerabilities of these budding athletes.

This paper embarks on an odyssey to explore the realm of sentiment analysis and the application of LDA (Latent Dirichlet Allocation) techniques within the domain of the NBA Draft Analysis dataset. Our pursuit is rooted in the profound question of whether these text mining methodologies harbor the potential to unlock the secrets of a player's trajectory: whether they are destined for the limelight of stardom or face the daunting specter of being branded a "bust."

Throughout this expedition, we will navigate the labyrinthine intricacies of the NBA draft, uncover its historical tapestry, and dissect the pivotal elements that steer draft decisions. Our journey will traverse the domain of sentiment analysis, where the sentiments articulated by analysts, scouts, and experts converge to shape the narrative cocooning each prospective talent.

Analysis

Data Preparation and Collection

The dataset utilized in this project was compiled through a combination of web scraping and data integration techniques. The primary objective was to gather comprehensive pre-draft analysis information for

NBA players, including their strengths, weaknesses, outlook, and other related attributes. The dataset was obtained from the website nbadraft.net, which provides in-depth analysis and profiles of NBA draft prospects.

To collect this data, a Python script was developed to automate the process of scraping player-specific analysis pages on the website. The following key steps were involved in data collection:

Draft History Data: The script accessed the NBA draft history through the nba_api, fetching information about drafted players, including their names and draft years. This served as the foundation for selecting players for analysis.

Filtering the Draft History: To ensure a relevant dataset, the draft history data was filtered to include only records from the year 2000 onwards, as this was deemed the most pertinent period for analysis.

Web Scraping Analysis Pages: The script iterated through the filtered draft history dataset, constructing URLs for each player's analysis page on nbadraft.net. These URLs were used to access the pre-draft analysis content for each player.

Beautiful Soup Parsing: Beautiful Soup, a Python library for web scraping, was employed to parse the HTML content of each player's analysis page. The script specifically targeted sections of interest, including NBA comparisons, strengths, weaknesses, outlook, and notes.

Data Storage and Update: The collected analysis data for each player was stored in a dictionary and then written to a text file named player_dict.txt. Furthermore, the original draft history dataframe was updated to include the collected analysis data for each player.

Sleep Timer: To ensure ethical and considerate web scraping practices, a sleep timer was incorporated into the script. This timer imposed a delay of 60 seconds between each web request to avoid overwhelming the website's server.

Data Cleanup and Additional Scraping: The script performed various data cleanup and additional scraping tasks to address unique issues associated with different player analysis pages, such as handling variations in text formats and extracting an "overall" category when present.

NBA Player Statistics: Towards the end of the script, an attempt was made to collect NBA player statistics, including Player Efficiency Rating (PER), Win Shares per 48 minutes (WS/48), and Value Over Replacement Player (VORP). These statistics were extracted by constructing specific URLs based on player names.

Data Storage: The collected player statistics were stored in a dataframe named player_stats_DF and exported to a CSV file for further analysis and research.

The resulting dataset combines web scraping techniques with meticulous data manipulation to provide comprehensive pre-draft analysis information for NBA players. It contains 1425 rows and 24 columns. The columns include descriptive values such as player names, team, and season as well as the analytical commentary to be used in this study in columns: “Strengths”, Weaknesses” and “Outlook”.

Figures 1 and 2 show the first 5 rows of the resulting dataset prior to transformation and cleaning techniques were employed.

Figure 1 – NBA Draft Post 2000 Data Unprocessed

	PERSON_ID	PLAYER_NAME	SEASON	ROUND_NUMBER	ROUND_PICK	\
0	1641705	Victor Wembanyama	2023	1	1	
1	1641706	Brandon Miller	2023	1	2	
2	1630703	Scoot Henderson	2023	1	3	
3	1641708	Amen Thompson	2023	1	4	
4	1641709	Ausar Thompson	2023	1	5	

	OVERALL_PICK	DRAFT_TYPE	TEAM_ID	TEAM_CITY	TEAM_NAME	...	\
0	1	Draft	1610612759	San Antonio	Spurs	...	
1	2	Draft	1610612766	Charlotte	Hornets	...	
2	3	Draft	1610612757	Portland	Trail Blazers	...	
3	4	Draft	1610612745	Houston	Rockets	...	
4	5	Draft	1610612765	Detroit	Pistons	...	

	First Name	Last Name	Suffix1	Suffix2	Overall	NBA Comparison	\
0	Victor	Wembanyama	NaN	NaN	101.0	Ralph Sampson	
1	Brandon	Miller	NaN	NaN	95.0	Paul George	
2	Scoot	Henderson	NaN	NaN	97.0	Derrick Rose/Ja Morant	
3	Amen	Thompson	NaN	NaN	93.0	Latrell Sprewell	
4	Ausar	Thompson	NaN	NaN	93.0	Trevor Ariza	

Figure 2 – NBA Draft Post 2000 Data Unprocessed - Continued

	Strengths	\
0	A generational talent … The sky is the limit a...	
1	An extremely talented 6’9 SF/PF with a lanky f...	
2	Henderson is a 6’2 190 guard with outstanding ...	
3	Thompson is a 6’7 210 perimeter player who is ...	
4	Thompson is a 6’7 215 wing with the physical t...	

	Weaknesses	\
0	He has bulked up considerably, but he still ne...	
1	Though he has solid athleticism, he lacks grea...	
2	3-point shot needs improvement (27 3FG%); adde...	
3	Jump shot is a big concern currently (25 3FG%)...	
4	Can be too reliant on his athleticism; doesn’t...	

	Outlook	\
0	Victor Wembanyama is probably the most hyped t...	
1	Incoming Alabama freshman … 2022 McDonald’s Al...	
2	NaN	
3	NaN	
4	NaN	

	Notes
0	NaN
1	NaN
2	NaN
3	Measured 6’ 5.75” barefoot, 8’ 7.50” standing ...
4	Measured 6’ 5.75” barefoot, 8’ 8.00” standing ...

[5 rows x 24 columns]

To analyze the text data, computerized processing methods were used to convert text data into vectorized data, from which computer algorithms can extract word frequencies and conduct further analysis. Prior to vectorization, however, the contents from the variables “Strengths”, Weaknesses” and “Outlook”, were preprocessed with the following actions to produce more accurate results during modeling:

- Special Characters and Punctuation: regular expressions (regex) were used to remove any characters that are not alphanumeric or spaces from the text, including numerical digits.
- Convert Text to Lowercase: all text was converted to lowercase to ensure uniformity in text representation. This step was done to avoid any issues related to case sensitivity during analysis.
- Tokenization: The text was tokenized using NLTK's word_tokenize function. Tokenization breaks the text into individual words, which are referred to as tokens.
- Stopwords Removal: Stopwords are common words that do not add much meaning to the text (e.g., 'the', 'and', 'is'). The function removes stopwords using NLTK's set of English stopwords, retaining only meaningful words in the text.
- Removal of Nan Values: Missing values can hinder a model's ability to learn from the data and make accurate predictions. The rows containing nan values were dropped as in this case there was only one occurrence in the entire dataset.

Figure 3 – Strengths, Weaknesses and Outlook After Preprocessing

	Strengths_Preprocessed	Weaknesses_Preprocessed	Outlook_Preprocessed
0	unique player tremendous frame ability face ba...	lack great length four 61075 wingspan solid el...	projected possible midsecond round pick 2021 n...
1	high basketball iq greatest strength strong pe...	average athlete lacking great speed leaping ab...	NaN
2	crafty scorer find hole defense passing skill ...	small even point guard ability get inside larg...	NaN
3	physically gasol great upper body strength lik...	unlike pau marc average athlete heavy legged p...	NaN
4	NaN	NaN	NaN

Next – Beau's part of the data transformation and explanation of calculations for busts should go here I think..

Sentiment Analysis and Score Assignment:

To conduct a comprehensive Sentiment Analysis, sentiment scores were systematically assigned to specific columns within the dataset, namely, "Strengths_Preprocessed," "Weakness_Preprocessed," and "Outlook_Preprocessed." This intricate process was facilitated through the utilization of Python's Vader Sentiment library, which encompasses the Sentiment Intensity Analyzer.

Before embarking on the Sentiment Analysis journey, it was imperative to address missing or NaN (Not-a-Number) values within the dataset to ensure the integrity of the analysis. This endeavor primarily involved the "Strengths" and "Weaknesses" columns, whereas the "Outlook" column harbored an extensive number of missing values, surpassing the 1000 mark. The strategy employed for handling missing values was to systematically drop rows that contained NaN values in either the "Strengths" or "Weaknesses" columns.

Following the removal of all rows featuring NaN values in the "Strengths" and "Weaknesses" columns, the dataset remained robust, comprising a total of 1042 data points. Each of these data points corresponded to a distinct draft player, thereby offering a comprehensive foundation for the subsequent Sentiment Analysis.

Data Exploration

Having completed the data cleaning and preprocessing, the NBA draft dataset was now ready for exploratory analysis. Two parallel studies took place while exploring the dataset for capturing the positive and negative sentiments of the NBA Drafts analysis reports. In the first exploratory analysis, the columns

Strengths_Preprocessed and Weaknesses_Preprocessed were combined into a single string. In the second version, the column Weaknesses_Preprocessed was evaluated by itself.

While analyzing the sentiment scores assigned to the "Strengths_Weaknesses_Combined," it became evident that the dataset exhibited a substantial skew toward positive scores. This trend is clearly illustrated in the bar chart presented in Figure 4, where the majority of scores fall within the range of 0.76 to 1.00. Conversely, when examining the sentiment scores solely for "Weaknesses_Preprocessed," a more balanced distribution is observed, albeit with a slight skew toward the positive end, as depicted in Figure 5.

In addition to these visualizations, Figure 6 introduces WordClouds that showcase the most frequently occurring words based on their associated sentiment scores. These WordClouds offer a visual representation of the prominent terms within the dataset, shedding light on the prevalent themes and sentiments.

Figure 4 – Distribution of Sentiment Scores - Strengths_Weaknesses_Combined

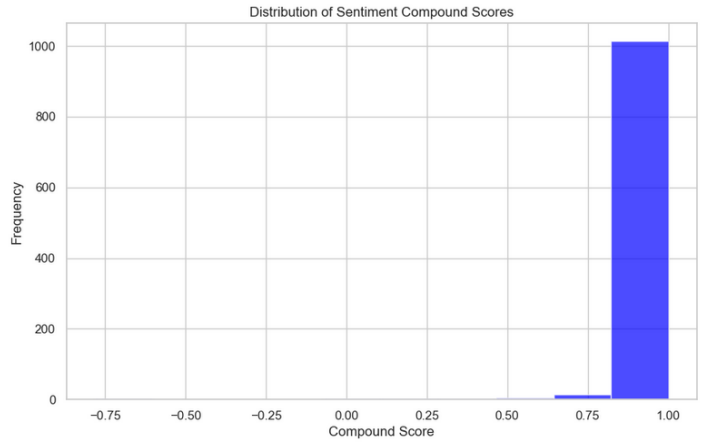


Figure 5 – Distribution of Sentiment Scores - Weaknesses Preprocessed

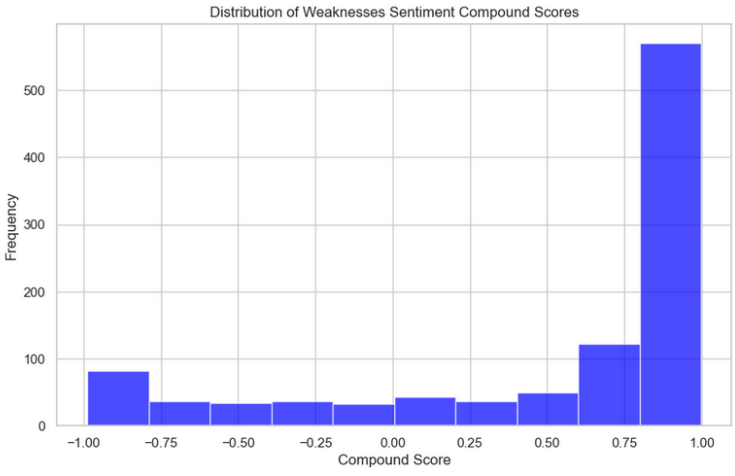


Figure 6 – WordClouds based on Positive and Negative Sentiment Scores using Weaknesses Preprocessed

The objective of the sentiment analysis is to determine if negative reviews vs positive reviews by the analysts can be used as predictors to determine if a player will be successful or a "bust". Figure 6 shows that with the Weaknesses only scores, most of the players which can be considered "busts" based on their career statistics, have sentiment scores that are positive and above 0.75. Figure 7 shows detailed scores distributions for each scenario, where "bust" is a yes, no, or to be determined.

Figure 6 – Sentiment Scores Distributions for Bust2 Criteria – Weakenesses_Compound Scores

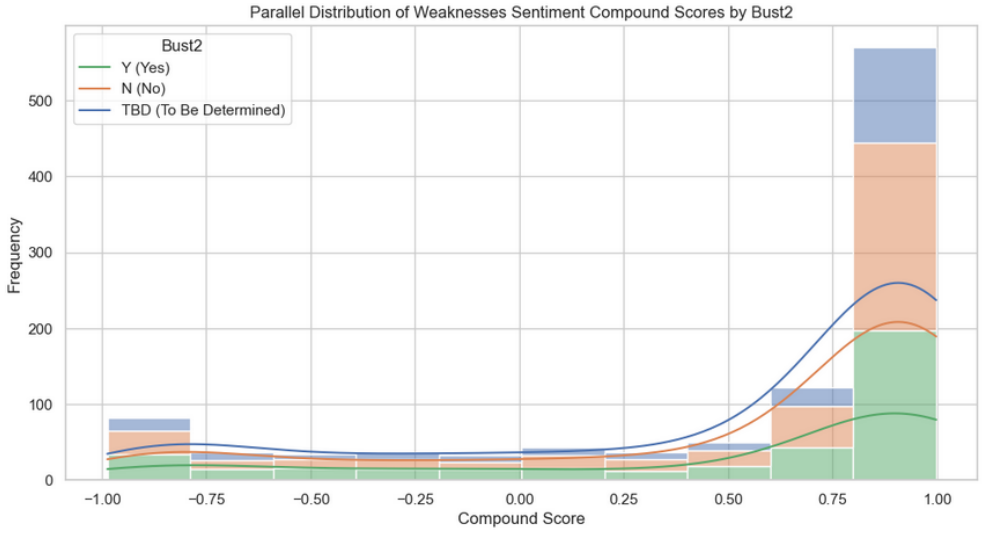
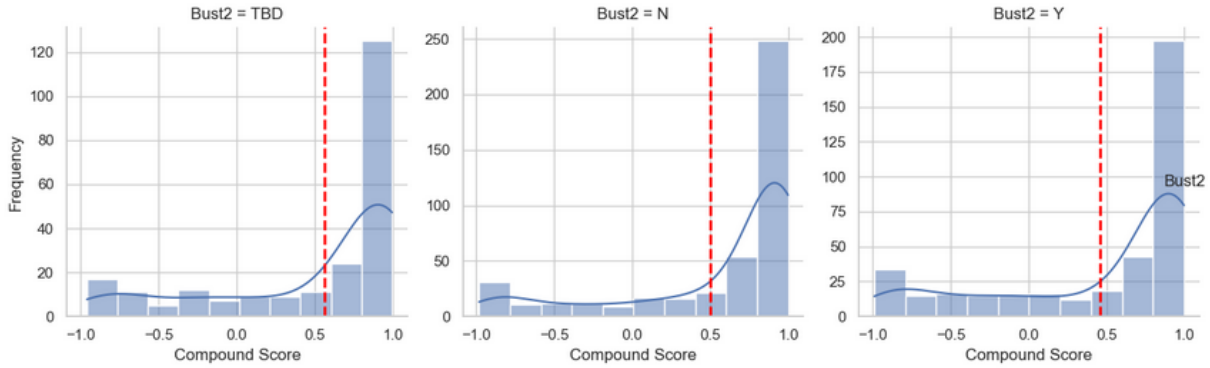


Figure 7 – Weakness Compound Scores Distribution of Bust = Yes, Bust2 = N, and Bust2 = Y



Figures 8 and 9 underscore the severity of the skewness in the data when combining the Strenghts and Weakenesses columns.

Figure 8 - Sentiment Scores Distributions for Bust2 Criteria – Strengths_Weakenesses_Compound Scores

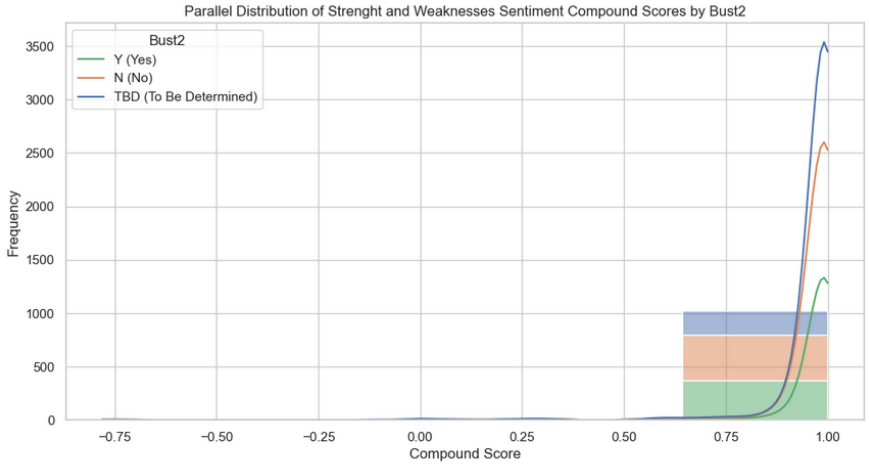
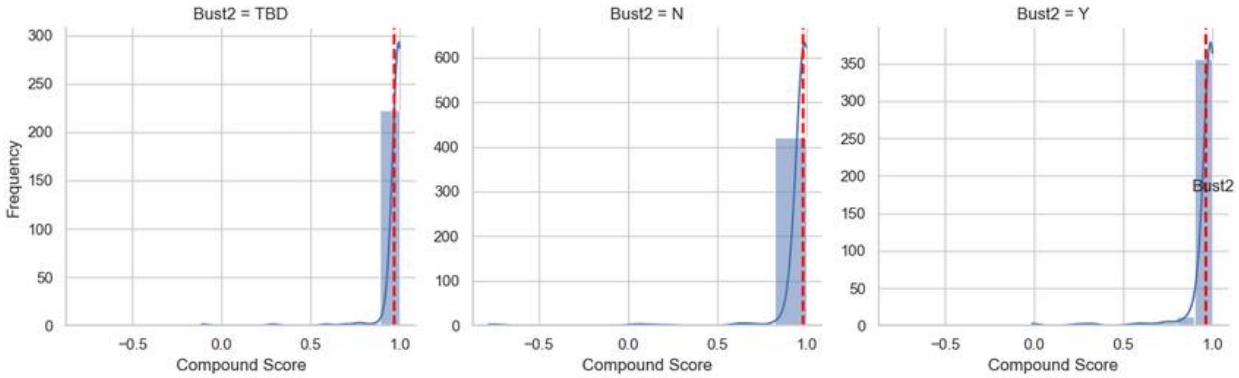


Figure 9 – Sentiment Scores Distributions for Bust2 Criteria – Strengths_Weakenesses_Compound Scores



With the sentiment scores at hand the top 10 most positive and top 10 most negative player reviews can be obtained from the dataset. Figures 10 and 11 show them with along with their respective NBA players and bust or no bust labels.

Figure 10 – Top 10 Most Positive Scores and Analysis Reviews

```
Top 10 Most Positive Scores:
         PLAYER_NAME Bust2                          Weaknesses_Preprocessed  \
0        Jalen Green   TBD  green good shooter yet sniper beyond arc still...
1         RJ Barrett     N  may peaked degree high school clearly top kid ...
2  Marvin Bagley III     Y  lack wingspan big men class continue fill fram...
3      James Wiseman   TBD  obvious elephant room go draft played minute c...
4         Kyle Kuzma     N  kuzma show great deal talent skill still need ...
5     Chase Budinger     N  strong oneonone player area game focus end flo...
6     Charles Bassey   TBD  need work shooting range consistency hesitates...
7      Victor Claver     Y  need add strength although he tough doesnt see...
8       Johnny Davis   TBD  asked goto scorer last season necessity aggres...
9       Tyrese Maxey   TBD  struggled efficiencyconsistency freshman kentu...

   Weaknesses_Preprossed_Compound
0                          0.9979
1                          0.9977
2                          0.9973
3                          0.9967
4                          0.9964
5                          0.9963
6                          0.9962
7                          0.9959
8                          0.9953
9                          0.9948
```

Figure 11 – Top 10 Most Negative Scores and Analysis Reviews

```
Top 10 Most Negative Scores:
           PLAYER_NAME Bust2  \
1041     Willie Warren     Y
1040         AJ Price     N
1039      Kemba Walker     N
1038     Chris Richard     Y
1037  DeMarcus Cousins     N
1036    Luke Harangody     N
1035    Isaiah Jackson   TBD
1034  Danilo Gallinari     N
1033     Grant Jerrett     Y
1032    Dakari Johnson     Y

                              Weaknesses_Preprocessed  \
1041  warren suffers poor shot selection asked numbe...
1040  high level nba athlete quickness fine price do...
1039  measurement 61 shoe win remains undersized com...
1038  may struggle keep speed nba game doesnt great ...
1037  cousin lack maturity mental focus evident nega...
1036  luke biggest weakness toughest obstacle overco...
1035  point viewed project next level weighs 200 lb ...
1034  high level european player always equal contri...
1033  jerrett failed prove mediocre player college l...
1032  johnson ceiling relatively low essentially saw...

      Weaknesses_Prepossed_Compound
1041                        -0.9866
1040                        -0.9846
1039                        -0.9743
1038                        -0.9739
1037                        -0.9729
1036                        -0.9720
1035                        -0.9709
1034                        -0.9699
1033                        -0.9685
1032                        -0.9628
```

Models and Methods

TF-IDF Vectorizer

For the analysis of the NBA Drafts dataset, the TF-IDF vectorization method was employed. TF-IDF, which stands for "Term Frequency-Inverse Document Frequency," is a technique used to transform textual data into numerical features suitable for machine learning. This approach plays a crucial role in converting text data into a format that can be utilized by machine learning algorithms. It works by calculating a score for each term in the text, indicating its significance in a specific document relative to the entire dataset.

The term frequency (TF) component of TF-IDF measures how frequently a term appears in a document, giving insight into the importance of that term within that document. Conversely, the inverse document frequency (IDF) component quantifies the rarity of a term across all documents. Terms that are rare and appear in only a few documents are assigned higher IDF scores, highlighting their unique nature and potential importance. By combining the TF and IDF scores, the TF-IDF vectorizer generates a numerical representation for each document, where term importance is weighted based on its frequency within the document and rarity across the dataset.

Text Preprocessing: Lemmatization and Stemming

In addition to employing TF-IDF vectorization, the text data underwent preprocessing steps known as lemmatization and stemming. These techniques were applied to enhance the quality and consistency of the textual content before feeding it into the machine learning models.

Lemmatization: Lemmatization involves reducing words to their base or dictionary forms to normalize variations of words. For instance, "running" and "ran" would be lemmatized to "run." This process helps ensure that different inflections of a word are treated as the same term, leading to better feature representation and improved model performance.

Stemming: Stemming, on the other hand, involves removing prefixes and suffixes from words to reduce them to their root forms. For example, "running" and "runner" would both be stemmed to "run." While stemming is more aggressive than lemmatization, it can sometimes result in words that are not actual words, but the technique is useful in simplifying words to their core meanings.

Vader Sentiment Intensity Analyzer

The VADER (Valence Aware Dictionary and sEntiment Reasoner) Sentiment Intensity Analyzer is a powerful tool for assessing the sentiment or emotional tone of a piece of text. It operates on the principle of lexicon-based sentiment analysis, which means it relies on pre-compiled dictionaries of words and their associated sentiment scores. These lexicons contain thousands of words, each tagged with a polarity score, indicating whether the word is positive, negative, or neutral, as well as an intensity score, which quantifies the strength of the sentiment.

VADER's magic lies in its ability to handle complex sentiment expressions, including negations and context-based sentiment shifts. It assigns a sentiment polarity score to each word in a text, considering both the individual word's sentiment and the context in which it appears. This is crucial for understanding phrases like "not good," where the negation changes the overall sentiment. VADER also considers capitalization and punctuation for added context. After evaluating all the words and phrases in a text, VADER calculates an overall compound sentiment score, which represents the text's overall sentiment intensity. This compound score can range from -1 (extremely negative) to +1 (extremely positive), with 0 indicating a neutral sentiment. Researchers and analysts often use this compound score to determine the sentiment of a text document accurately, making VADER a valuable tool for sentiment analysis in various applications, from social media monitoring to customer feedback analysis.

Model 1: Random Forest Ensemble (3 Cross Validation)

Random Forest combined with TF-IDF vectorization was used to predict the positive, negative, or neutral sentiments for the player's "Weaknesses_Preprocessed" analysis. Random Forest is an ensemble learning algorithm that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) of the individual trees. It is particularly effective in handling complex relationships and avoiding overfitting.

At the core of the Random Forest algorithm are individual decision trees. Decision trees are hierarchical structures that break down a dataset into smaller and more manageable subsets by asking a series of binary questions based on feature attributes. These questions are designed to effectively split the data into different classes, ultimately leading to a classification decision at the tree's leaves.

The decision tree works by recursively splitting the data into subsets, with each split being determined by a feature and a threshold. The algorithm aims to find the best feature and threshold that results in the purest subsets, where samples within each subset predominantly belong to a single class (label). The process continues until a predefined stopping criterion is met, such as a maximum depth or a minimum number of samples in a leaf node.

While decision trees have the potential to model complex relationships in the data, they are prone to overfitting, capturing noise in the data and leading to poor generalization to unseen data. Random Forest builds on the strength of decision trees while addressing their limitations. Instead of relying on a single decision tree, Random Forest creates an ensemble of multiple decision trees, each trained on a different subset of the data and with some randomness introduced.

Model 2: Supervised Vector Machines (SVM + 3 Cross Validations)

In Model 2, Support Vector Machines (SVM) with linear kernels were employed for sentiment analysis, and the model's performance was assessed using 3-fold cross-validation. SVM is a powerful supervised learning algorithm used for classification tasks. It works by finding the optimal hyperplane that best separates data points belonging to different classes. In this context, SVM was utilized to classify sentiments in the "Weaknesses_Preprocessed" text data. By optimizing the hyperplane's parameters, SVM aims to maximize the margin between different sentiment classes, enhancing its classification accuracy.

Model 3: Supervised Vector Machines (SVM + Lemmatization + 3 Cross Validations)

Building upon Model 2, Model 3 incorporated lemmatization as a preprocessing step before employing SVM with linear kernels. Lemmatization is a text normalization technique that reduces words to their base or dictionary forms. By lemmatizing the text data, common variations of words are transformed into their root form, reducing dimensionality, and potentially improving the model's performance. Like Model 2, Model 3 underwent 3-fold cross-validation to evaluate its effectiveness in sentiment classification.

Model 4: Supervised Vector Machines (SVM + Stemming + 3 Cross Validations + Bigrams)

Model 4 followed a similar approach to Model 3 but utilized stemming instead of lemmatization as a preprocessing step. Stemming involves removing suffixes or prefixes from words to obtain their root forms, which can help reduce the vocabulary size and improve the SVM model's performance. Stemming may be more aggressive than lemmatization, potentially leading to different results. As with the previous models, Model 4 underwent 3-fold cross-validation to assess its ability to classify sentiments effectively.

Model 5: Latent Dirichlet Allocation (LDA) – Topic Modeling

In Model 5, a different and insightful approach to text analysis was employed, focusing on topic modeling using Latent Dirichlet Allocation (LDA). Unlike the previous models, which were designed for sentiment prediction, LDA is an unsupervised machine learning technique with broader applications. It's primarily used to reveal hidden topics within a collection of text documents, making it particularly relevant for uncovering the underlying themes in the context of NBA draft player analysis.

LDA operates on the principle that each document in the dataset is a mixture of various topics, and each topic is, in turn, a mixture of specific words or terms. By examining the patterns of word co-occurrence across documents, LDA effectively identifies these latent topics and quantifies their prevalence within the dataset. This approach goes beyond sentiment analysis and enables a deeper exploration of the content contained within the NBA Drafts Analysis text data.

So, how can LDA help predict whether NBA draft players will be successful ("non-busts") or underperform ("busts")? While LDA itself doesn't provide direct predictions about players' outcomes, it offers valuable insights into the characteristics and themes associated with player weaknesses. By identifying latent topics within the strengths and weaknesses' commentaries, analysts can gain a nuanced understanding of the

types of weaknesses that are commonly mentioned and their relative importance. This information can then be used as a feature in predictive models or as part of a broader analysis to assess whether certain weakness themes are indicative of future success or challenges in the NBA. In essence, LDA serves as a powerful tool for uncovering hidden patterns and trends within textual data, which can inform more informed decision-making in player evaluation and selection processes.

Results

TF-IDF Vectorizer Results

Figure 10 portrays the initial 30 features or words from the processed vocabulary, derived through the TF-IDF vectorizer applied to the reviews data is presented. It also showcases the first 10 sample TF-IDF scores associated with each feature or word, after the meticulous cleansing and pre-processing steps.

Figure 10 – First 30 Vocabulary Words/Features and First 10 Vectorization Score Columns

```
First 30 Vocabulary Words:
['aac' 'aahs' 'aaron' 'aau' 'abandoned' 'abilities' 'abilitiesplays'
 'ability' 'abilitydoesnt' 'abilties' 'able' 'abnormal' 'abort'
 'aboveaverage' 'abovetherim' 'absent' 'absolute' 'absolutely' 'absorb'
 'absorbing' 'abundance' 'abuse' 'abused' 'abysmal' 'academic'
 'academically' 'academics' 'academy' 'acb' 'acc']

First 10 Sample Vectorization Score Columns:
[[0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.06024411 0.         0.        ]
 [0.         0.         0.         ... 0.0635197  0.         0.        ]
 ...
 [0.         0.         0.         ... 0.02719611 0.         0.        ]
 [0.         0.         0.         ... 0.03215775 0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]]
```

Model 1: Random Forest Ensemble (3 Cross Validation) Results:

While predicting sentiment for the NBA Drafts data, several different scenarios were tested. Given that the sentiment scores were less skewed for the “Weaknesses” evaluations performed by the scouts and analysts, all models were applied the “Weaknesses_Preprocessed” variable. In scenario 1, the entire dataset containing 1042 rows was used. In scenario 2 a composite of records containing 220 top positive scores, 220 top negative scores, and 130 (all that was available) neutral scores were used. Scenario 3 was made of only the top 200 most negative scores and top 200 most positive scores. For scenarios 1 and 2, three categorical labels were used to classify the sentiment scores. For neutral scores, the label 0 was assigned. For negative and positive scores, the labels 1 and 2 were assigned respectively. Scenario 3 only had positive and negative labels which were classified as 0 for negative and 1 for positive sentiments.

Model 1.1 Random Forest 3CV Scenario 1 Results:

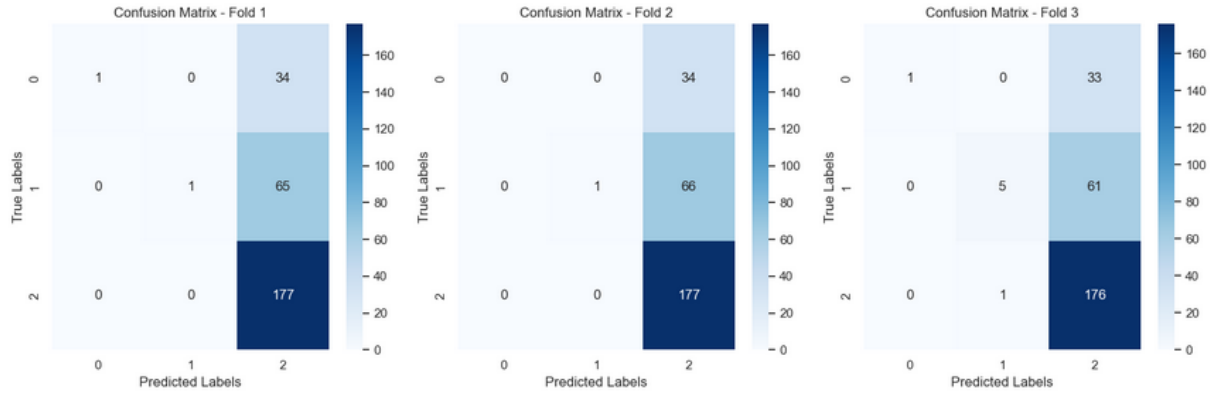
The sentiment classification model achieved an overall accuracy of approximately 65.07%, indicating that it correctly predicted the sentiment (positive, negative, or neutral) of the "Weaknesses_Preprocessed" analysis for about 65.07% of the player profiles in the dataset, on average. This suggests that the model performs reasonably well in categorizing sentiments. However, when examining the fold-wise accuracies, some variations across different subsets of the data were observed.

In the first fold, the model achieved an accuracy of around 64.39%, showcasing its ability to perform adequately on one portion of the dataset. In the second fold, the accuracy was slightly lower, at approximately 64.03%, suggesting consistent but not significantly improved performance. The third fold yielded the highest accuracy among the three, at about 65.70%. While the model's performance is relatively stable across the folds, it's important to note that there is still room for enhancement. Further inspection of each classification type shows that the model performs well for Label 2 but struggles with Labels 0 and 1, particularly in terms of recall.

Figure 11 – Results and Confusion Matrix Model 1.1 Random Forest 3CV Scenario 1

	precision	recall	f1-score	support
Label0	1.00	0.03	0.06	34
Label1	0.83	0.08	0.14	66
Label2	0.65	0.99	0.79	177
accuracy			0.66	277
macro avg	0.83	0.37	0.33	277
weighted avg	0.74	0.66	0.54	277

```
Sentiment Classification Model Accuracy: 0.6507177033492823
Fold 1 Accuracy: 0.6438848920863309
Fold 2 Accuracy: 0.6402877697841727
Fold 3 Accuracy: 0.6570397111913358
```



Model 1.2 Random Forest 3CV Scenario 2 Results:

The results for Random Forest scenario 2 indicate that the distribution of sentiment labels in the dataset is somewhat imbalanced, with Label2 having the highest count (220 samples), followed by Label1 (220 samples), and Label0 with the lowest count (130 samples). This imbalance could impact the model's performance, especially for the minority class (Label0). The imbalance for the neutral class is due to the limitations in availability of neutral classifications. In this study, neutral labels were assigned to scores ranging from 0 to 0.6. However, most of the reviews, analysis and commentary provided by the NBA scouts and analysists were polarized by being strongly positive or strongly negative, thus making the neutral classification a challenge for the models overall.

The complete accuracy of the sentiment classification model for this sample data is 0.5175, indicating that the model correctly predicts the sentiment label for approximately 51.75% of the samples. However, accuracy alone might not provide a complete picture, given the class imbalance. Fold-wise Accuracy: Examining

the fold-wise accuracy, we see that Fold 1 and Fold 3 have the same accuracy of 46.05%, while Fold 2 has a slightly lower accuracy of 44.08%. These variations in fold-wise accuracy suggest some inconsistency in model performance across different subsets of the data.

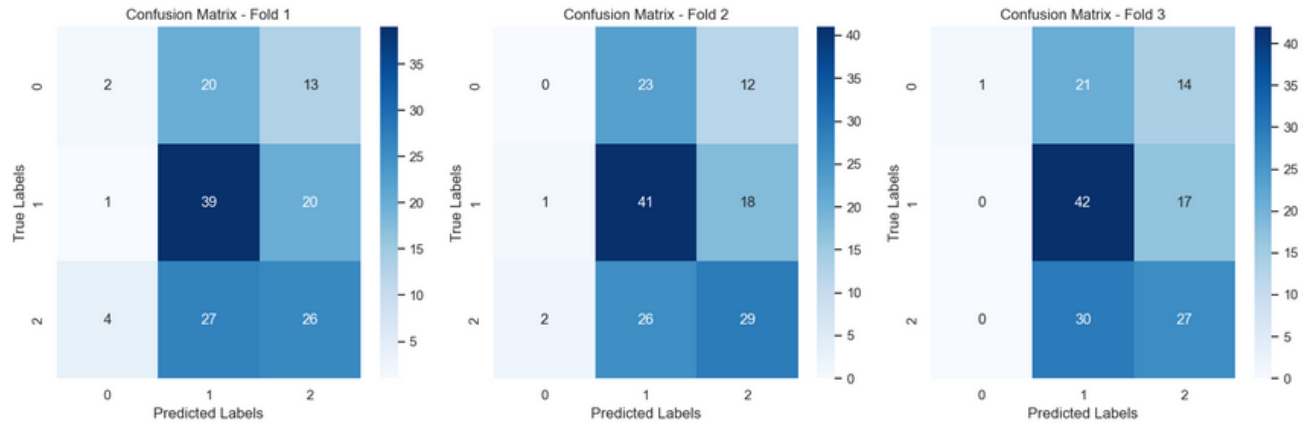
The classification report provides a more detailed breakdown of the model's performance for each sentiment label. For Label0, the precision is high (1.00), indicating that when the model predicts Label0, it is usually correct. However, the recall is low (0.03), indicating that the model misses many actual Label0 instances. This results in a low F1-score (0.05) for Label0. For Label1, the precision is moderate (0.45), indicating that the model's predictions for Label1 are relatively accurate. The recall is higher (0.71), indicating that the model captures a significant portion of the true Label1 instances. This leads to a relatively higher F1-score (0.55) for Label1. For Label2, both precision and recall are moderate (0.47 and 0.47, respectively), leading to a balanced F1-score of 0.47.

In summary, the model performs reasonably well for Label1 and Label2, with higher precision and recall for these classes. However, the performance for Label0 is poor, primarily due to low recall. The overall model accuracy is influenced by the class imbalance, and it's important to consider the specific objectives of the sentiment classification task when evaluating these results. Further analysis and potentially addressing class imbalance are necessary to improve model performance.

Figure 12 – Results and Confusion Matrix Model 1.2 Random Forest 3CV Scenario 2

	precision	recall	f1-score	support
Label0	1.00	0.03	0.05	36
Label1	0.45	0.71	0.55	59
Label2	0.47	0.47	0.47	57
accuracy			0.46	152
macro avg	0.64	0.40	0.36	152
weighted avg	0.59	0.46	0.40	152

```
Sentiment Classification Model Accuracy: 0.5175438596491229
Fold 1 Accuracy: 0.4407894736842105
Fold 2 Accuracy: 0.4605263157894737
Fold 3 Accuracy: 0.4605263157894737
```



Model 1.3 Random Forest 3CV Scenario 3 Results:

The results for Random Forest classification in the scenario of distinguishing between negative (Label0) and positive (Label1) appears to be relatively balanced. The overall accuracy of the sentiment classification

model is quite high at 86.25%. This indicates that the model correctly predicts whether a player's "Weaknesses_Preprocessed" analysis is negative or positive for the majority of samples. The accuracy for each fold shows some variation but remains relatively high across all folds. Fold 1 and 2 have the highest accuracies at 85.05%, followed by Fold 3 at 80.19%. The classification report provides detailed metrics for each sentiment label:

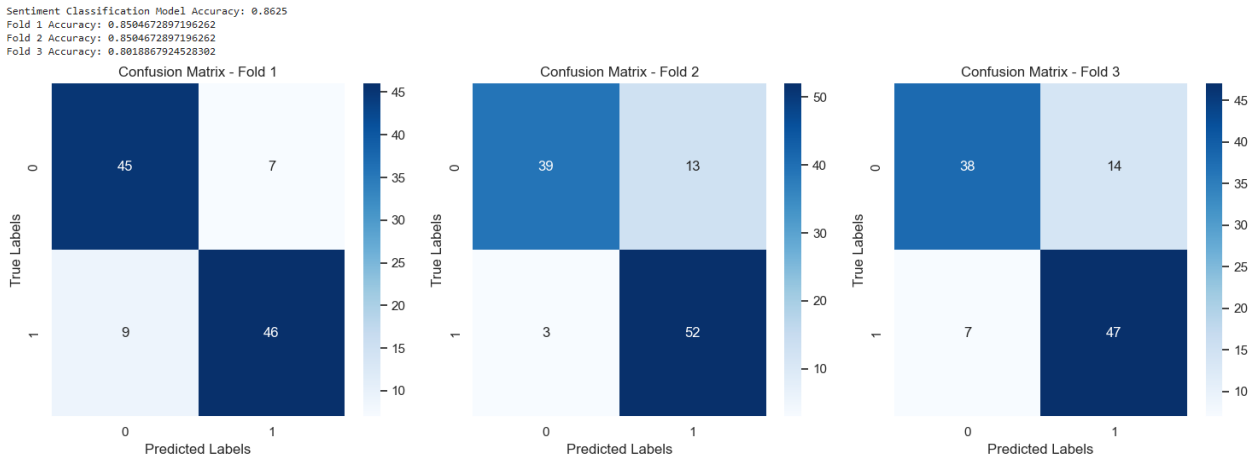
For Label0 (negative sentiment), both precision and recall are balanced at around 0.84 and 0.73 respectively. This indicates that the model accurately identifies negative sentiments and effectively captures most of the true negative instances. The F1-score for Label0 is also 0.78, reflecting a good balance between precision and recall.

For Label1 (positive sentiment), both precision and recall are similarly balanced at around 0.77 and .87. This suggests that the model performs well in identifying positive sentiments and captures most of the true positive instances. The F1-score for Label1 is also 0.82, indicating a good overall performance in distinguishing positive sentiments.

In summary, the Random Forest model performs well in distinguishing between negative and positive sentiments. It demonstrates relatively high accuracy and balanced precision and recall for both sentiment labels.

Figure 12 – Results and Confusion Matrix Model 1.3 Random Forest 3CV Scenario 3

	precision	recall	f1-score	support
Label0	0.84	0.73	0.78	52
Label1	0.77	0.87	0.82	54
accuracy			0.80	106
macro avg	0.81	0.80	0.80	106
weighted avg	0.81	0.80	0.80	106



Model 2. Supervised Vector Machines (SVM Linear Kernel + 3 Cross Validations) Results

The results for SVM classification in the third scenario appear to be relatively balanced. The overall accuracy of the sentiment classification model is the highest thus far at 92.5%. This indicates that the model correctly predicts whether a player's "Weaknesses_Preprocessed" analysis is negative or positive for most of the samples. The accuracy for each fold shows some variation but remains relatively high across all folds. Fold 2 has

the highest accuracy at 89.72%, followed by Fold 1 at 88.79%, and Fold 3 at 82.08%. The classification report provides detailed metrics for each sentiment label:

For Label0 (negative sentiment), both precision and recall are balanced at around 0.81. This indicates that the model accurately identifies negative sentiments and effectively captures most of the true negative instances. The F1-score for Label0 is also 0.82, reflecting a good balance between precision and recall.

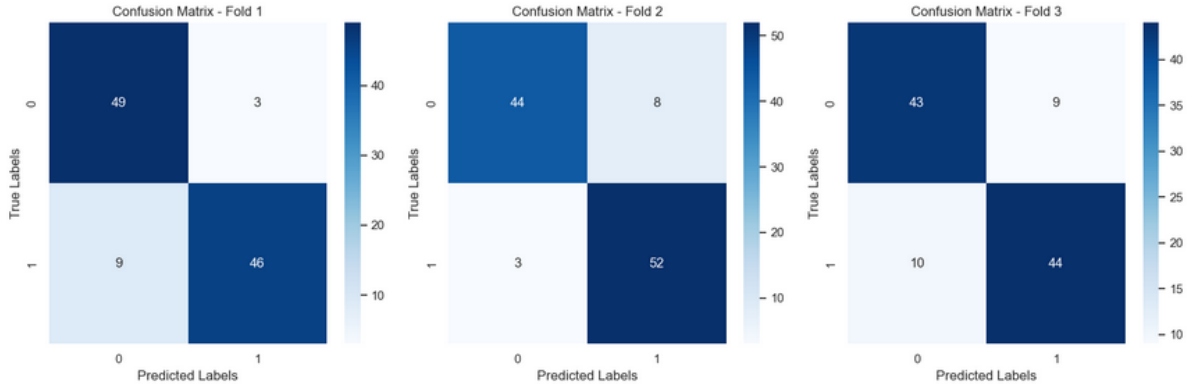
For Label1 (positive sentiment), both precision and recall are similarly balanced at around 0.83. This suggests that the model performs well in identifying positive sentiments and captures most of the true positive instances. The F1-score for Label1 is also 0.82, indicating a good overall performance in distinguishing positive sentiments.

In summary, the SVM model with a linear kernel performs remarkably well in distinguishing between negative and positive sentiments. It demonstrates high accuracy and balanced precision and recall for both sentiment labels. These results suggest that the model is effective in classifying sentiment in the context of negative versus positive sentiments for the "Weaknesses_Preprocessed" analysis of NBA draft players.

Figure 13 – Results and Confusion Matrix Model 2 SVM 3CV Scenario 3

	precision	recall	f1-score	support
Label0	0.81	0.83	0.82	52
Label1	0.83	0.81	0.82	54
accuracy			0.82	106
macro avg	0.82	0.82	0.82	106
weighted avg	0.82	0.82	0.82	106

```
Sentiment Classification Model Accuracy: 0.925
Fold 1 Accuracy: 0.8878504672897196
Fold 2 Accuracy: 0.897196261682243
Fold 3 Accuracy: 0.8207547169811321
```



Model 3: Supervised Vector Machines (SVM + Lemmatization + 3 Cross Validations) Results

Although no significant improvements were observed after the addition of lemmatization, model 3 produced impressive results. The detailed results for the SVM with a linear kernel and lemmatization in a 3-fold cross-validation scenario for sentiment classification, focusing on distinguishing between negative (Label0) and positive (Label1) sentiments are as follows:

Model Accuracy: The overall accuracy of the sentiment classification model is very high at 91.25%. This indicates that the SVM model with a linear kernel and lemmatization correctly predicts whether a player's "Weaknesses_Preprocessed" analysis is negative or positive for most samples.

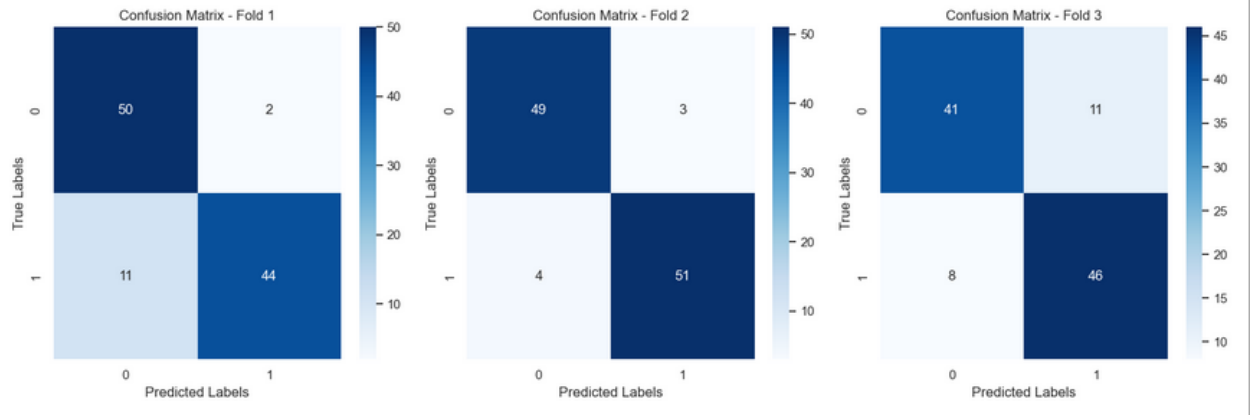
Fold-wise Accuracy: The accuracy for each fold varies slightly but remains high across all folds. Fold 2 has the highest accuracy at 93.46%, followed by Fold 1 at 87.85%, and Fold 3 at 82.08%.

These results demonstrate that the SVM model with a linear kernel and lemmatization performs exceptionally well in distinguishing between negative and positive sentiments. It achieves a very high overall accuracy and balanced precision and recall for both sentiment labels, further highlighting its effectiveness in sentiment classification for the "Weaknesses_Preprocessed" analysis of NBA draft players.

Figure 14 – Confusion Matrix heatmap for SVM 3CV + Lemmatization

	precision	recall	f1-score	support
Label0	0.84	0.79	0.81	52
Label1	0.81	0.85	0.83	54
accuracy			0.82	106
macro avg	0.82	0.82	0.82	106
weighted avg	0.82	0.82	0.82	106

```
Sentiment Classification Model Accuracy: 0.9125
Fold 1 Accuracy: 0.8785046728971962
Fold 2 Accuracy: 0.9345794392523364
Fold 3 Accuracy: 0.8207547169811321
```



Model 4: Supervised Vector Machines (SVM + Stemming + 3 Cross Validations + Bigrams) Results

The results for the SVM with a linear kernel, stemming, and bigrams in a 3-fold cross-validation scenario for sentiment classification produced a very high overall accuracy of the sentiment classification of 85%. This indicates that the SVM model with a linear kernel, stemming, and bigrams correctly predicts whether a player's "Weaknesses_Preprocessed" analysis is negative or positive for the majority of samples. The accuracy for each fold varies slightly but remains high across all folds. Fold 2 has the highest accuracy at 87.85%, followed by Fold 3 at 83.96%, and Fold 1 at 83.18%.

The classification report for Label0 shows a precision of 0.84, indicating that the SVM model with stemming and bigrams accurately identifies negative sentiments. The recall for Label0 is 0.83, suggesting that the model captures most of the true negative instances. The F1-score for Label0 is 0.83, reflecting a good balance between precision and recall. For Label1 the precision is 0.84, indicating that the model performs well in identifying positive sentiments with stemming and bigrams. The recall for Label1 is 0.85, suggesting that the

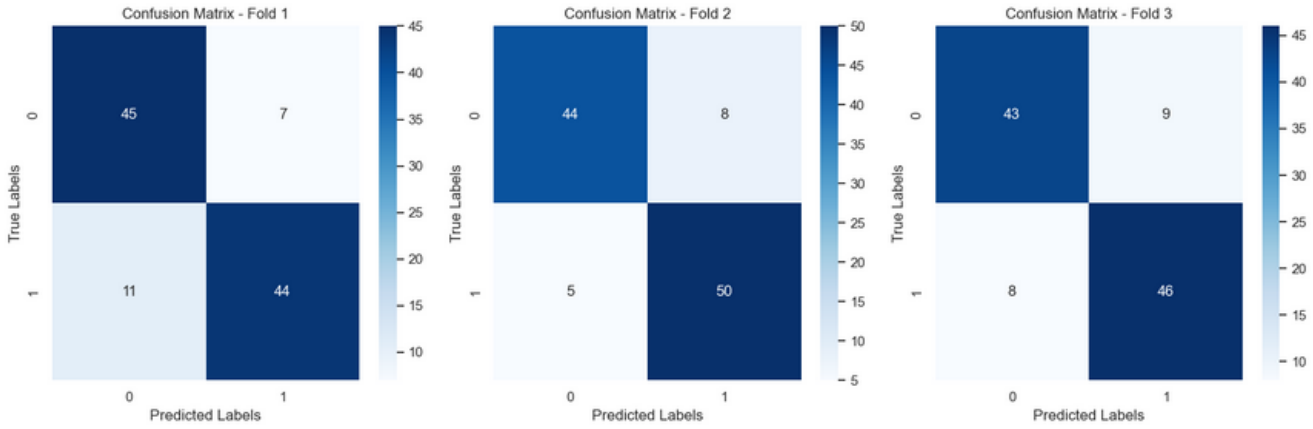
model effectively captures most of the true positive instances. The F1-score for Label1 is 0.84, indicating a good overall performance in distinguishing positive sentiments with stemming and bigrams.

These results, although not as good as the results from the simple SVM with linear kernel, demonstrate that the SVM model with stemming, and bigrams performs exceptionally well in distinguishing between negative and positive sentiments. It achieves a very high overall accuracy and balanced precision and recall for both sentiment labels, further highlighting its effectiveness in sentiment classification for the "Weaknesses_Preprocessed" analysis of NBA draft players.

Figure 15 – Confusion Matrix heatmap for SVM 3CV + Stemming + Bigrams Results:

	precision	recall	f1-score	support
Label0	0.84	0.83	0.83	52
Label1	0.84	0.85	0.84	54
accuracy			0.84	106
macro avg	0.84	0.84	0.84	106
weighted avg	0.84	0.84	0.84	106

```
Sentiment Classification Model Accuracy: 0.85
Fold 1 Accuracy: 0.8317757009345794
Fold 2 Accuracy: 0.8785046728971962
Fold 3 Accuracy: 0.839622641509434
```



Sentiment Analysis Summary of Results:

Scenario 1: Original Data with all 1042 rows highly skewed towards positive sentiment

Model Type: Random Forest ensemble with 3 Cross Validations TF-IDF vectorization.

Accuracy: Achieved an accuracy of approximately 65%, indicating a moderately successful classification model.

Sentiment Labels: Distinguished between three sentiment labels: Label0 (negative), Label1 (neutral), and Label2 (positive).

Precision and Recall: Displayed varying levels of precision and recall for each sentiment label. Notably, Label2 (positive) had the highest precision and recall.

Scenario 2: Sample Data with 570 rows still skewed towards positive and negative sentiment scores (220 label2,220 label1, and 130 label0)

Model Type: Random Forest with TF-IDF vectorization and imbalanced sentiment labels.

Accuracy: Achieved an accuracy of around 46%, indicating a lower-performing model due to imbalanced data.

Sentiment Labels: Classified into three sentiment labels: Label0, Label1, and Label2.

Precision and Recall: Displayed imbalanced precision and recall, with Label2 (positive) having the highest values.

Scenario 3: Sample Data with top 200 negative and top 200 positive sentiment scores

Model Types: Random Forest with 3CVs and SVM with linear kernel and various preprocessing techniques (lemmatization, stemming, and bigrams).

Accuracy: Achieved consistently high accuracy across all preprocessing conditions, ranging from 82% to 91%.

Sentiment Labels: Focused on binary classification (negative vs. positive) due to preprocessing variations and data constraints.

Precision and Recall: Demonstrated balanced precision and recall for both negative and positive sentiments in all scenarios.

Overall Conclusions:

Random Forest vs. SVM: SVM consistently outperformed Random Forest in sentiment classification, particularly when distinguishing between negative and positive sentiments.

Imbalanced Data Challenge: Scenarios 1 and 2 struggled due to imbalanced sentiment labels, resulting in lower accuracy. This highlights the importance of addressing class imbalance in classification tasks.

Preprocessing Matters: Preprocessing techniques such as lemmatization, stemming, and bigrams did not significantly improve model accuracy and balance in SVM-based sentiment classification.

Binary vs. Multi-label Classification: The choice of binary (negative vs. positive) or multi-label (negative, neutral, positive) classification should depend on the research question and dataset characteristics. Binary classification with SVM yielded the best results.

In summary, SVM models with appropriate preprocessing techniques, especially when focusing on binary sentiment classification, proved to be the most effective in accurately classifying sentiment in the NBA draft data. The choice of preprocessing techniques and the handling of imbalanced data are critical considerations for improving sentiment analysis model performance.

Sentiment Analysis Scores vs. "Bust" Frequency Correlation Analysis:

To explore potential correlations between sentiment scores derived from NBA Draft analysis data and a player's "bust" status, we selected the top 50 players with the most positive sentiment scores and the top 50 players with the most negative sentiment scores for comparison. We then examined the distributions based on four different criteria used to classify a player as a "bust" or not.

The analysis revealed interesting insights. While some players with high positive sentiment scores were classified as "busts," the prevalence of "busts" was notably higher among players with top negative sentiment scores. This observation suggests that sentiment analysis of scouts' and analysts' reviews could offer valuable additional insights for predicting a draft player's likelihood of becoming a "bust."

Figure 16 – Positive Bust Distribution – Top 50 Sentiment Scores from Weaknesses_Preprocessed

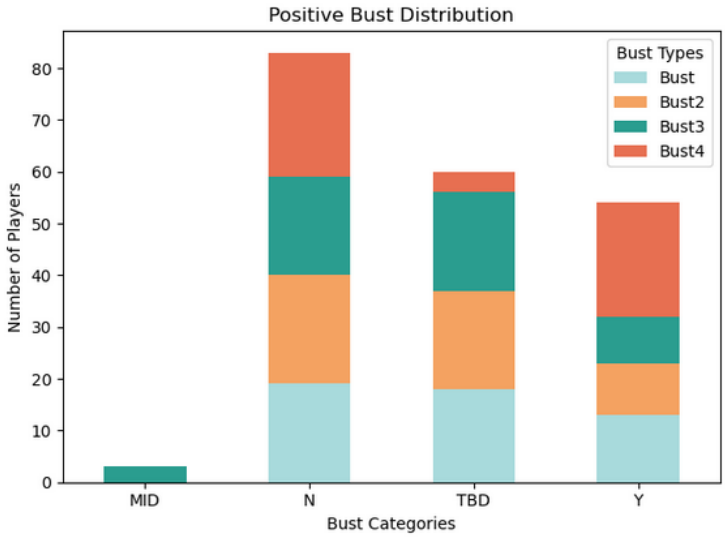
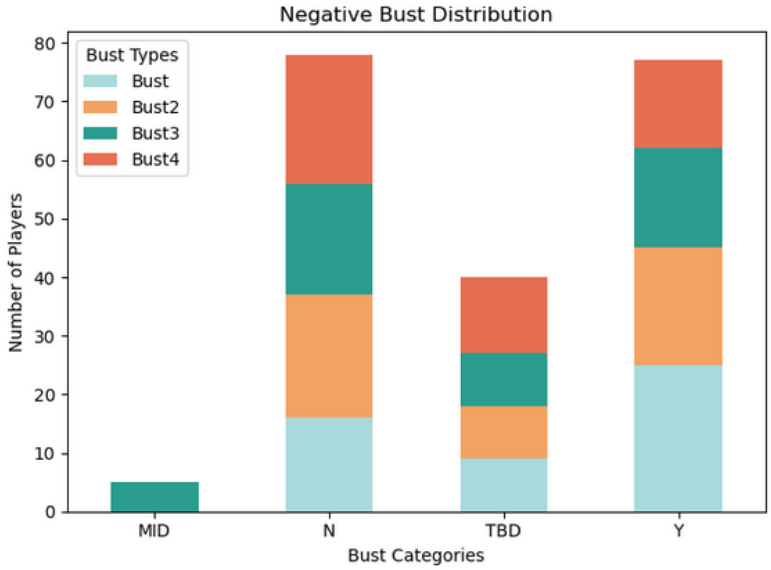


Figure 17 – Negative Bust Distribution – Top 50 Sentiment Scores from Weaknesses_Preprocessed



Model 5: Latent Dirichlet Allocation (LDA) – Topic Modeling Results

In Model 5 adopted several approaches to unveil hidden topics within NBA Draft player analyses and commentaries. In one approach, all reviews, and commentaries from the "Strengths," "Weaknesses," and "Outlook" columns were combined into a single text string and stored in a column named "Concatenated_Preprocessed." In the second approach, only commentaries from the "Weaknesses" column were used due to their balanced representation of negative and positive player assessments.

The primary goal was to analyze word co-occurrence patterns in these texts to identify latent topics and assess their prevalence across the dataset. This method allowed for a deeper understanding of the underlying themes in the "Weaknesses_Preprocessed" textual data. The same data cleaning and preprocessing techniques from the sentiment analysis were applied.

Initially, when the LDA model was applied to the combined "Strengths," "Weaknesses," and "Outlook" columns, it produced ten topics, but these exhibited repetitive themes and words, suggesting a need for further refinement. To optimize the model's performance and identify the most appropriate number of topics for the dataset, a grid search was employed. In this grid search, the LDA model was trained multiple times, each time with a different number of topics ranging from 5 to 50 possibilities.

The key to understanding how the grid search for coherence scores works lies in the coherence score itself. Coherence scores are a measure of how interpretable and semantically meaningful the topics are. Higher coherence scores indicate that the topics are more distinct and representative of the underlying themes in the text data. During the grid search, the LDA model is trained repeatedly with varying numbers of topics, and for each iteration, the coherence score is calculated. The grid search systematically explores the entire range of topic numbers, and as the number of topics increases, the coherence score is tracked.

Figures 16 and 17 visually represent this process, with Figure 17 specifically showing the top 15 topic coherence scores. The blue line in the coherence score graph demonstrates how the coherence score changes as the number of topics increases. Analysts look for a point on this graph where the coherence score exhibits a sharp increase, followed by a decrease or stabilization. This point, often referred to as an "elbow" point, indicates the ideal number of topics. The sharp drop in the blue line (coherence scores) serves as a clear indicator of where this optimal number of topics can be found. This approach ensures that the selected topics are both informative and coherent, ultimately enhancing the utility of the LDA model's results for further analysis and interpretation.

Figure 18 displays the results from PyLDAvis. PyLDAvis is a Python library that provides an interactive visualization of topic models, particularly those generated using Latent Dirichlet Allocation (LDA). It helps users to explore and understand the topics within a text corpus by visualizing the relationships between topics and the terms that define them. The topic bubbles in the PyLDAvis represent topics as circles or bubbles on the screen. The size of each bubble corresponds to the prevalence of the topic in the corpus.

The inter-topic distance map arranges the topic bubbles in 2D space based on their similarity. Similar topics are placed closer to each other, allowing you to identify topic clusters. When a topic bubble is clicked, PyLDAvis displays a word cloud that shows the most relevant terms for that topic. The terms are selected based on their probability of appearing in the topic.

PyLDAvis also provides a topic matrix with a list of terms associated with each topic, along with their relevance scores and displays a bar chart for each topic, showing the distribution of that topic across the documents in your corpus. In the NBA Draft data case, topic 8 has the most prevalence and words like “good”, “game”, “ball”, “shot”, and “ability” have the highest frequencies.

Figure 18 – 10 Topics from Concatenated Preprocessed Results

```
Topic #1: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #2: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #3: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #4: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #5: good game ball great ability shot player nba shooting level
Topic #6: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #7: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #8: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #9: thing hits amazing character agile learn desire perfect mcdonalds competitor
Topic #10: thing hits amazing character agile learn desire perfect mcdonalds competitor
```

Figure 19 – 15 Topics from Concatenated Preprocessed Results Using Coherence Scores

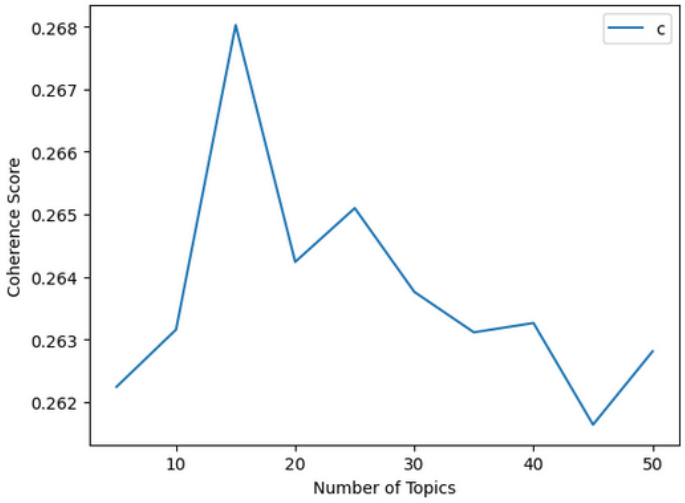
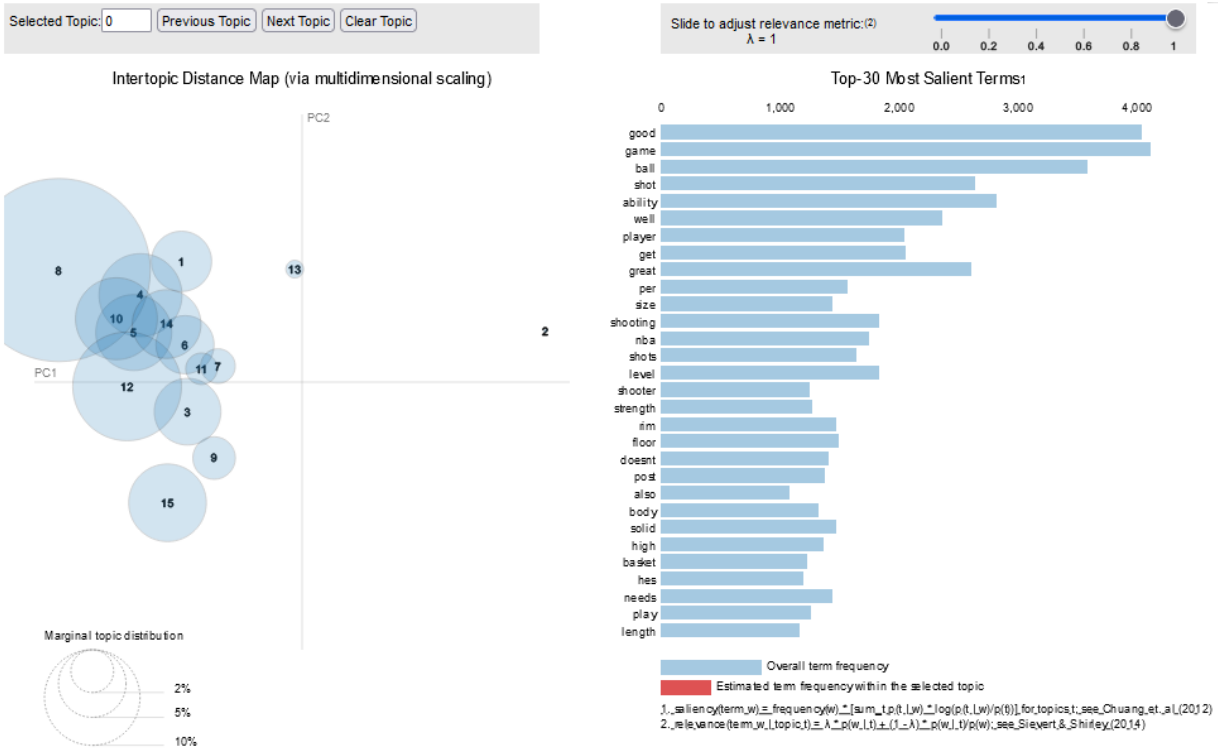


Figure 20 – LDA visualization (PyLDAvis) with 15 Topics from Concatenated Preprocessed Results Using Coherence Scores and Distribution of Most Frequent Words

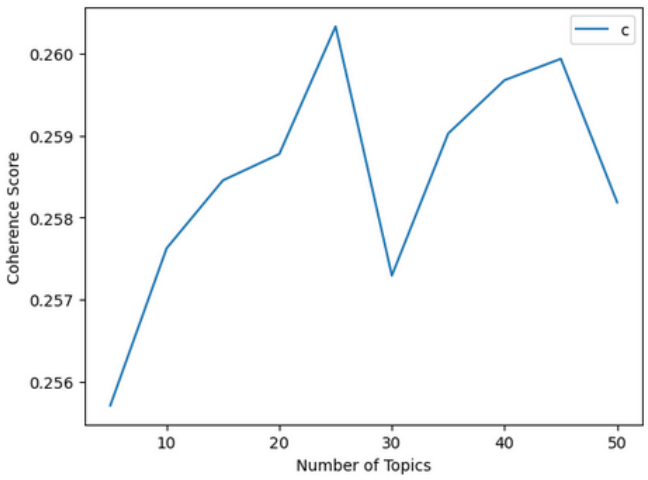


For scenario 2, using only the reviews and commentaries from the "Weaknesses_Preprocessed" column, the first 10 topics were more distinguishable and thus offered more insights into what the main ideas were. The application of a coherence scores search grid yielded 25 total topics which are shown in figure 19 along with top words and scores.

Figure 21 – 10 Topics from Weaknesses_Preprocessed Results

```
Topic #1: battle load distance exploit williams armour incident smith underdeveloped jones
Topic #2: battle load distance exploit williams armour incident smith underdeveloped jones
Topic #3: smith armour association distance 2018 nbpa favor clip creativity load
Topic #4: battle load distance exploit williams armour incident smith underdeveloped jones
Topic #5: shot need game lack ball time nba player doesnt level
Topic #6: battle load distance exploit williams armour incident smith underdeveloped jones
Topic #7: battle exploit williams facilitator underdeveloped league weakness appears scorer playing
Topic #8: incident load distance fight credit risk williams court carry deliberate
Topic #9: 69 pushed injury upper question size body strength doesnt battle
Topic #10: jones flash skillset plenty championship 18 12 deliberate minute vertical
```

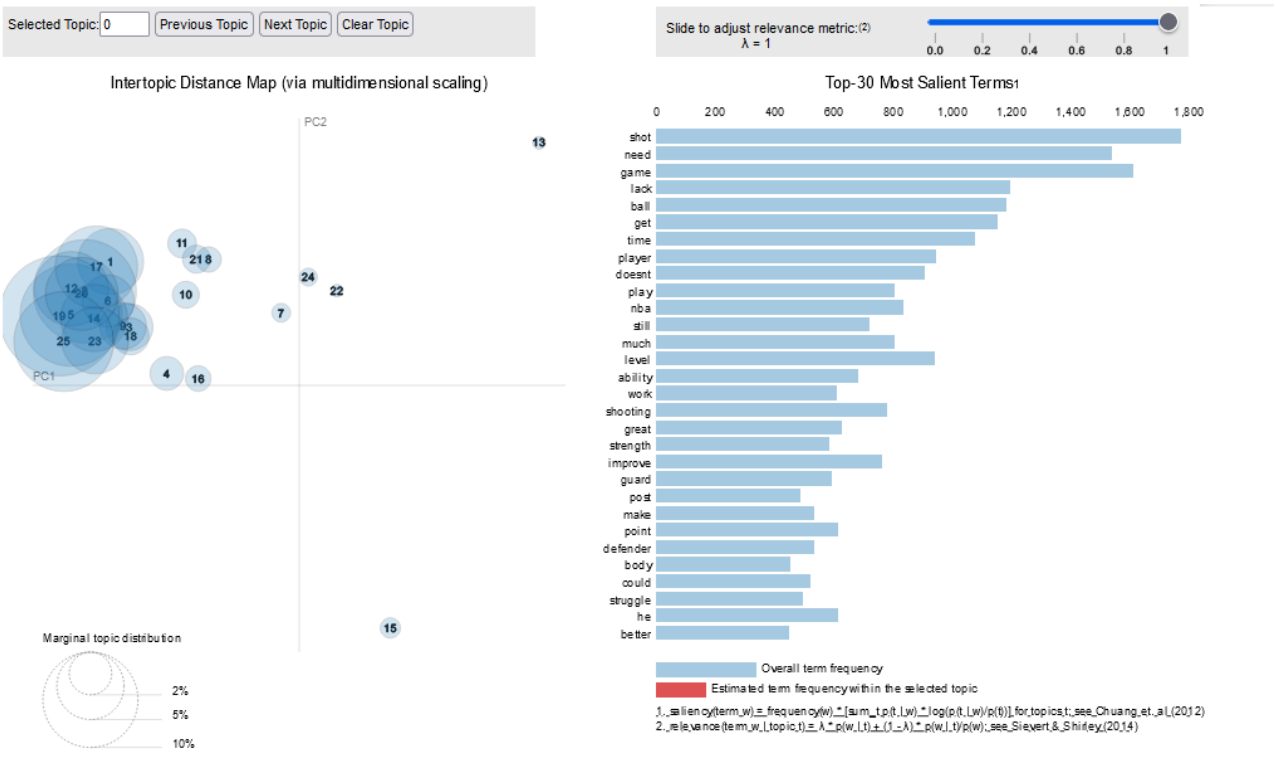
Figure 22 – 25 Topics from Concatenated Preprocessed Results Using Coherence Scores



The PyLDAvis visualization below highlights that for the second scenario, out of the 25 topics found, the distance between bubbles is much smaller when compared to the first scenario. This indicates the degree of similarity or dissimilarity between topics. Specifically, it represents the inter-topic distance, also known as the topic distance map. When two bubbles are positioned close to each other on the PyLDAvis visualization, it suggests that the topics they represent share similar terms and are more related to each other. Conversely, if two bubbles are farther apart, it indicates that the topics they represent are less similar and have fewer overlapping terms. In other words, topics that are close to each other may indicate subtopics or themes within a broader topic, while isolated topics might represent unique or niche themes within the corpus.

For the second scenario the words like “shot”, “need”, “game”, “lack”, “get”, “time” and “player” are the most frequent.

Figure 23 – LDA visualization (PyLDAvis) with 25 Topics from Concatenated Preprocessed Results Using Coherence Scores and Distribution of Most Frequent Words



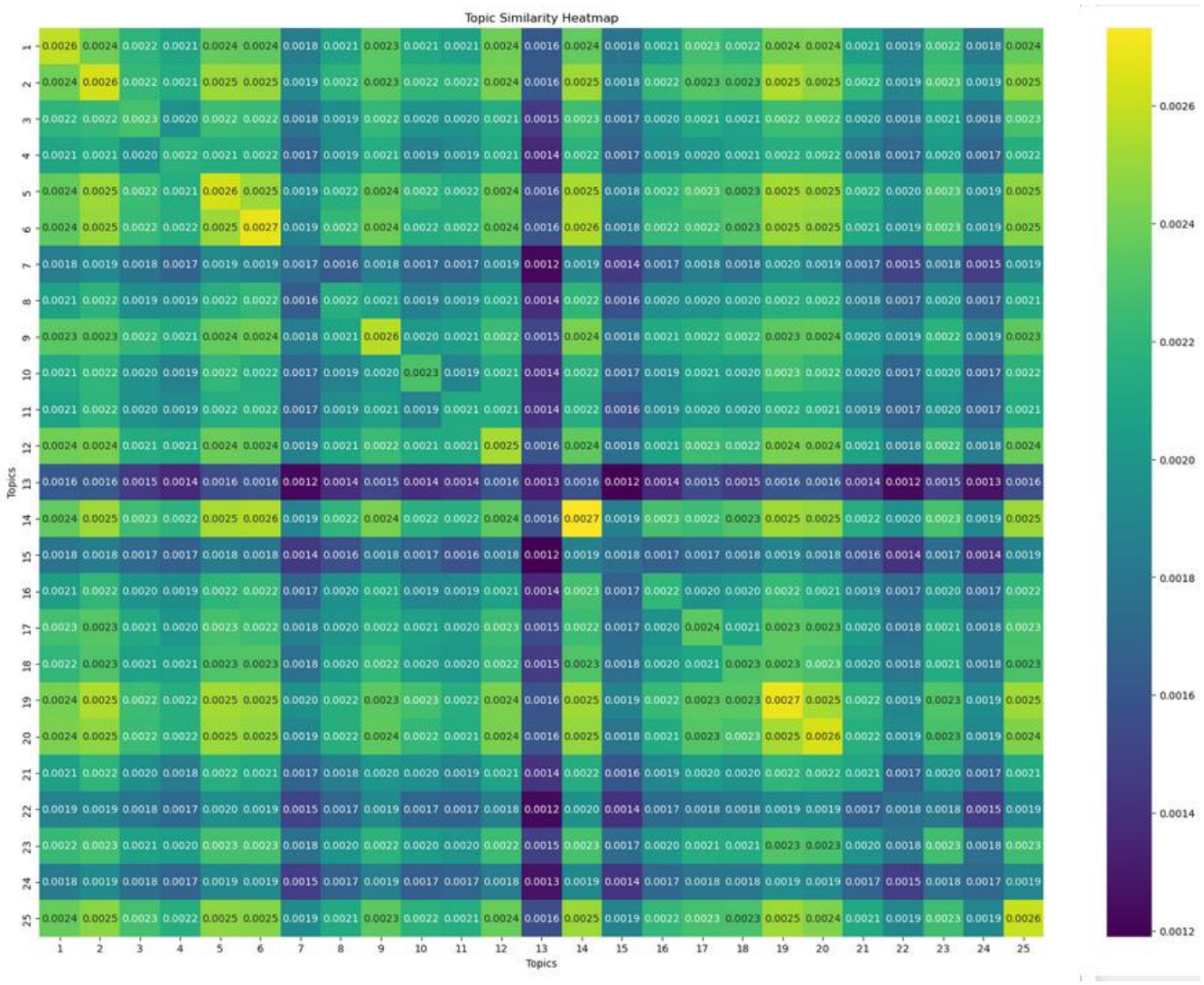
Next, the overall top 10 words from each of the 25 topics observed from the "Weaknesses" analysis was plotted in a series of mini wordClouds as shown in figure 22.

Figure 24 – WordClouds per Topic – top 10 Words from Weaknesses_Preprocessed

The heatmap below provides an overview of topics with the most significant overlapping coherence scores. In this heatmap, darker colors signify stronger relationships between the topics, indicating a higher

degree of word similarity and thematic correlation. For instance, Topics 7, 13, and 15 serve as illustrative examples of topics exhibiting pronounced correlations among their constituent words, suggesting shared themes and subject matter.

Figure 25 – Coherence Scores Heatmap of 25 Topics from "Weaknesses_Preprocessed



LDA Summary of results:

The best version of the model identified 25 distinct topics based on the "Weaknesses_Preprocessed" text data. A few key takeaways can be analyzed from these topics:

Common Themes: Several topics revolve around common themes like the need to improve various aspects of the players' game, such as shooting, ball-handling, and overall skills. These topics also touch upon the player's physical attributes, including strength and athleticism.

Game and Skills: Topics #1, #2, #3, and #6 focus on aspects related to the players' performance in the game, including shooting, scoring, and defense. It appears that the player is analyzed in terms of their skill development and gameplay.

Strength and Physical Attributes: Topics #4, #9, and #15 emphasize the importance of strength, body control, and physicality in the players' game. This suggests a focus on the players' physical attributes and their impact on performance.

NBA and Professional Level: Topics #2, #3, #5, #7, and #17 mention the NBA and professional basketball, indicating that the players' readiness for the NBA or a similar level is a recurring theme.

Shooting and Shooting Range: Topics #10, #12, #19, and #20 highlight the players' shooting abilities and range, including three-point shooting and midrange skills. Improving shooting seems to be a key focus.

Ball Handling and Passing: Topics #13, #16, #21, and #22 touch on the players' ball-handling, passing, and playmaking abilities. These topics suggest an evaluation of the player's ability to distribute the ball and make plays.

By harnessing these extracted topics and their prevalence in player weaknesses, analysts can build predictive models that consider the influence of these themes on players' future success or struggles in the NBA. These topics can serve as essential features in predictive algorithms, allowing analysts to make more informed judgments about whether certain weakness themes are indicative of potential "busts" or players destined for success in their professional careers. This nuanced understanding of player profiles can significantly enhance the accuracy of draft predictions and player evaluations, providing a valuable tool for NBA teams and scouts in their decision-making processes.