Racial Bias Detection

Aarushi Parashar and **Kabir Walia**

1 Introduction

Implicit racial biases in sports commentary have predominantly been studied from a social sciences perspective - on small-scale datasets and subjective annotation of "racial" language. However, more recently, this area of research has garnered attention due to harmful impacts of implicit racial biases including involuntarily profiling and segregating marginalized groups. Iyyer et al conducted a major study into football commentary from 1960-2019, collected from YouTube transcripts to produce the dataset titled FOOTBALL [1]. Although their work affirms the conclusions of previous social science analyses (that point to the fact that commentators tend to compliment caucasian players for their strategy and intelligence while providing applause to colored players for their strength and other physical attributes), the simplistic NLP approaches applied lack the sophistication to handle the temporal as well as confounding characteristics of the data.

Our project focuses on using more nuanced NLP and Machine Learning methods to improve upon some of the limitations highlighted by Iyyer et al and hence provide deeper insight into implicit racial bias in sports commentary.

2 Data

The dataset is collected from transcripts of 1,455 full game broadcasts from the U.S. NFL and National Collegiate Athletic Association (NCAA) recorded between 1960 and 2019. The dataset consists of mentions of players within these transcripts to information about their race (white or nonwhite) and position (e.g., quarterback). In total, FOOTBALL contains 267,778 mentions of 4,668 unique players, 65.7% of whom are nonwhite[1]. Mentions are a set of words around a players name or a "window". The white and non white denominations are used instead of specifically separating every race because a more balanced dataset will help in the presence of other confounding variables and the nonwhite race category is disproportionately black and many races are not represented. There is no missing data as the dataset has been preprocessed and curated by its authors, however, as illustrated in Figure 1 when investigating the total counts of each race in the dataset there is a significant amount of unknown races which we will omit for the analysis. Visually, in Figure 2 there is no clear difference when plotting the difference in frequency between common words in a word cloud indicating the need for deeper analysis.

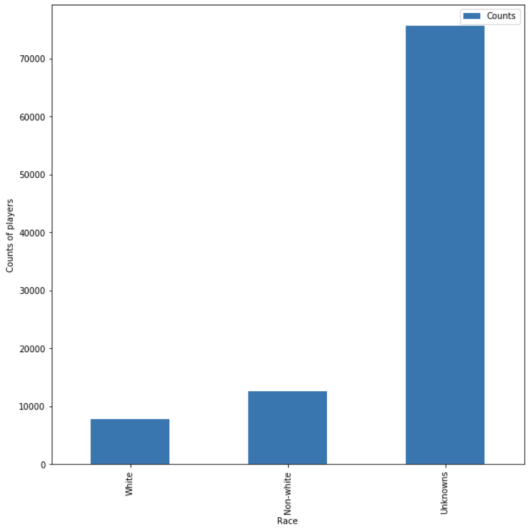

Figure 1: Frequency of each race in the mentions.

Figure 2: Word Clouds of frequent words used to describe each race - a larger word indicates greater frequency.

2.1 Confounding Factors

Through the exploratory data analysis it becomes evident that certain confounding variables need to be accounted for. The first confounding factor is that certain positions are disproportionately white or non white as seen in Figure 3. This affects as the mentions of those players many players' actions ("passes the ball downfield") depend on the position they play. In considering the "brain vs. brawn" debate some positions inherently require more brawn. Hence, to combat this we will compare performance on just the Quarterback Position with all positions to see the effect of the factor. Another confounding variable is description of personality/personal attributes vs. a play (an action made by a player in the game). Negatively referencing a bad play does not indicate racial bias, however referring to a non white player in a derogatory manner when describing personal traits and not referring to a white player with the same criticism would indeed be an indication of racial bias. (Henceforth, we will refer to this problem as the Player/Play confound).To account for this fine line in semantic difference, we will attempt to pre-

dict race from player mentions as opposed to predicting the probability that a word describes a player given the race and position as done in Iyyer et. al.[1]. In reformulating the problem to be a classification problem as well as working under the assumption that on average white and nonwhite players have the same number of good and bad plays, the focus will be primarily on the descriptive phrases that are not based on action but rather based on personality.

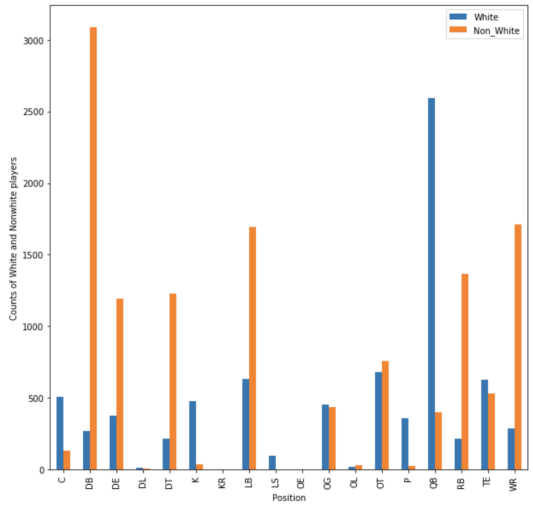

Figure 3: Breakdown of race per position.

3 Preliminary Experiments

For our preliminary modelling, we wanted to see if we could determine the race of the player, by only using the player mentions. We used 2 models - Logistic Regression and Naive Bayes which are discriminative and generative models[1] respectively.

3.1 Feature Transformation

We transform the textual information in the mentions to numeric data using an approach called the Term Frequency - Inverse Document Frequency (TF-IDF). This approach produces a matrix of weights - corresponding to the importance of words across the several documents (or in this case mentions). TF-IDF works on the assumption that more important words occur less frequently. Therefore, instead of simply using word frequencies as features, using the TF-IDF weights allow our models to better learn the impact of important words. This is aligned with the reasoning that racial commentary won't involve only a few specific words. These weights are computed as $tfidf(t,d,D) = tdf(t,d) \cdot idf(t,D)$ where the TFIDF weight of a term t in a document d is given by the product of the term frequency (tf) of t in d and inverse term frequency of t in the collection of all documents D is given as $idf(t,D) = log\frac{N}{|d \in D : t \in d|}$ [7] The outcome is a large matrix of weights per word in each document/mention that are served as features to the models mentioned below. We use the TfidfTransformer in sklearn to perform this transformation[6]. This feature transformation also helps reduce overfitting as the inverse term frequency approach reduces the impact of high frequency words that could impact the model performance.

3.2 Multinomial Naive Bayes

3.2.1 Model Description

The Multinomial Naive Bayes linear classifier based on the Bayes Theorem. It produces a probability distribution over the potential classes C and outputs class c as a label for document d such that class c is most likely (highest probability). Formally, NB chooses the class that maximizes the equation $\widehat{c} = argmax_{c \in C} P(f_1, f_2, ..., f_n|c)P(c)$ where f1, f2, ... , fn represent the terms in a document/mention and c refers to the racial class (White, Nonwhite). NB makes an important assumption here that the occurrence of a word in a document only depends on the class and is independent of other words. Using the independence nature of the probabilities, we get $c_{NB} = argmax_{c \in C} P(c)\Pi_{f \in F} P(f|c)$ In our case, the probabilities $P(f|c)$ are computed using the TFIDF-weights by the sklearn implementation of NB. P(c) for each class c is simply the number of mentions of class c divided by the total number of mentions. Naive Bayes tends to be a high bias model and by default has a lower propensity to underfit.

3.2.2 Model Training

We use the default parameter settings for the MultinomialNB classifier in sklearn[4]. The only parameter to handle is the smoothing parameter, set to 1. Essentially, to avoid the multiplication to lead to zero, which could happen if a term doesn't occur in a class or document, every word in the dataset is given a frequency of 1 in both classes. The window size[2] for context was taken as 15.

3.3 Logistic Regression

3.3.1 Model Description

Logistic regression is a linear classification algorithm that assigns probabilities to all potential classes (RACE) and outputs the one with a higher probability. The logistic function is defined as $\sigma(t) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$ Where t = -B0 + B1x and there is only a single feature x. In our case, the features are the TFIDF-weights of the words.

[1] See Midterm.ipynb for code.

[2] Window size is the length of the mention/context surrounding the player. See Appendix for results with various window sizes.

3.3.2 Model Training

We use the LogisticRegression class in sklearn[5]. Due to this being a preliminary study, we stick to the default hyperparameter settings. To prevent overfitting we employ an L2 regularization with log loss function. The default optimization solver is the liblinear solver that uses a variant of gradient descent called coordinate descent. Instead of updating all parameters together, this algorithm updates one parameter at a time. We set the regularization parameter C = 5 which is not too high hence preventing underfitting.

4 Evaluation

To evaluate the model, in the beginning we took out about 20% of the data for a test set. This subset has 14,241 examples when looking solely at the quarterback and 51,552 examples when looking at the set of all positions. Accuracy gives us a picture of how easily we are able to predict race given the mention of the player. A higher accuracy would be extremely indicative of racism if given the just commentary we are able to predict the race. This indicates a discernable pattern in the commentary for whites vs non whites which should not exist. However, we recognise that it is possible for the classifier to achieve high accuracy due to the slight class imbalance - there is a majority of nonwhite players (roughly 65%). So even if the model predicts every player in the test to be white, it would still achieve an estimated 65% accuracy. This is reflected in the table below. Therefore, we use F1 Score which is $\frac{2(precision)(recall)}{(precision+recall)}$. We see that the F1 score is higher for logistic regression. This is analyzed using the confusion matrices[3]. We see the impact of the higher majority nonwhite players as the true positives and false positives (right side of confusion matrix) are larger than the true negatives and false negatives. The Naive Bayes model is more impacted than the logistic regression. In the final project, we will deeply analyze this effect using precision and recall measures. Comparing the results for training and evaluating on just the Quarterback Subset versus all positions there is an improvement in both metrics for both models. This consistent improvement validates quantitatively the confound between players positions and their races and emphasizes the need to deal with this confound by isolating each position or in another manner. Hence, for future experiments we should use isolated positional data.

Window Size =15		Accuracy	F1 Score
Multinomial Naive Bayes	QB	0.7604	0.4492
	All Positions	0.6669	0.4291
Logistic Regresssion	QB	0.7886	0.6200
	All Positions	0.6801	0.5812

Table 1: Results for both models with a window size of 15.

[3]See Appendix.

Comparing the F1 Scores and Accuracy between the Naive Bayes and Logistic Regression model it is clear that there is definitely a detectable difference in the commentary of actions of white players as compared to actions of non white players. Logistic Regression slightly outperforms Naive Bayes in Table 1 indicating that the "racism" that the classifier is detecting does not follow the assumption that the words are conditionally independent i.e. words occurring in the statement depend solely on the category of the statement and not each other. Logistic regression is clearly able to use the dependencies between words to extrapolate meaning.

5 Next Steps

While our preliminary evaluations have produced high accuracies, we see the inability of our models to handle the class imbalance. Further, we also need a method that can help tackle the confounds in the dataset. Word embeddings are useful features to capture meanings of words and sentences. We believe that the embeddings can be used to resolve the confounding factors, especially the Player/Play confound. We plan on developing word embeddings by one of the following methods:

1. Unsupervised word embeddings: using a model like bag of words to get vector representations
2. Using supervised word embeddings: getting word representations by training a model on the data so that the embeddings pick up information about the class (RACE) of the player mentioned
3. Using contextual word embeddings: contextual word embeddings are an industry standard when it comes to vector representations of text. They allow the information stored in the vectors to better showcase the context words are used within our specific dataset.

Aside from using embeddings to train our classifiers, we want to account for sentiment analysis. Prior approaches for sentiment include a method where windows were marked as positive or negative to calculate the proportion of positive and negative instances for each window. However, due to the limited number of terms used to identify the sentiment, these experiments don't have lucrative results. An alternative method which we will use to supplement our use of embeddings is a distance metric to capture similarity in word embeddings. The distance metric will allow us to identify how closely related a word is to some set of positive/negative words - thereby allowing us to produce a sentiment score for all mentions. The sentiment score will serve as an additional feature in our classification model.

6 Appendix

6.1 References

[1]https://www.aclweb.org/anthology/D19-1666.pdf

[2]https://medium.com/better-programming/understanding-racial-bias-in-machine-learning-algorithms-1c5afe76f8b

[3]https://theundefeated.com/features/tony-dungy-some-announcers-biased-language-perpetuates-black-qb-stereotypes/

[4]https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

[5]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[6] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html?highlight=tfidfsklearn.feature_extraction.text.TfidfTransformer

[7] https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html

6.2 Diagrams

Naive Bayes - Quarterback		
Window Size	Accuracy	F1 Score
5	0.7664	0.4658
6	0.7655	0.4600
8	0.7657	0.4580
10	0.7657	0.4579
12	0.7644	0.4510
15	0.7604	0.4492

Table 2: Effect of changing window size for Naive Bayes when trained and evaluated on QB position.

Naive Bayes - All Positions		
Window Size	Accuracy	F1 Score
5	0.6718	0.4659
6	0.6709	0.4594
8	0.6715	0.4513
10	0.6688	0.4409
12	0.6684	0.4355
15	0.6669	0.4291

Table 3: Effect of changing window size for Naive Bayes when trained and evaluated on all positions.

Logistic Regression- Quarterback		
Window Size	Accuracy	F1 Score
5	0.7737	0.5536
6	0.7731	0.5615
8	0.7757	0.5782
10	0.7788	0.5875
12	0.7842	0.6055
15	0.7886	0.6200

Table 4: Effect of changing window size for Logistic Regression when trained and evaluated on QB position.

Logistic Regression- All Positions		
Window Size	Accuracy	F1 Score
5	0.6818	0.5755
6	0.6825	0.5804
8	0.6797	0.5790
10	0.6805	0.5825
12	0.6815	0.5834
15	0.6801	0.5812

Table 5: Effect of changing window size for Logistic Regression when trained and evaluated on all positions.

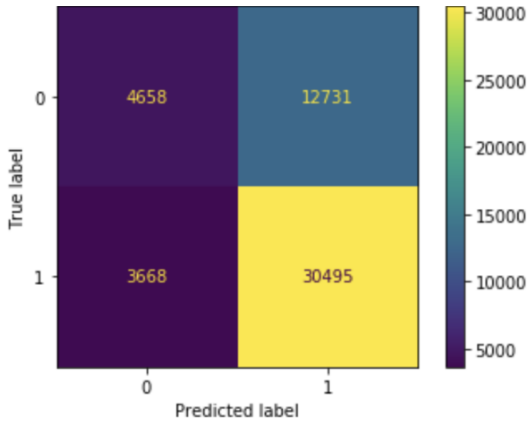

Figure 4: Confusion Matrix for Logistic Regression on All Positions.

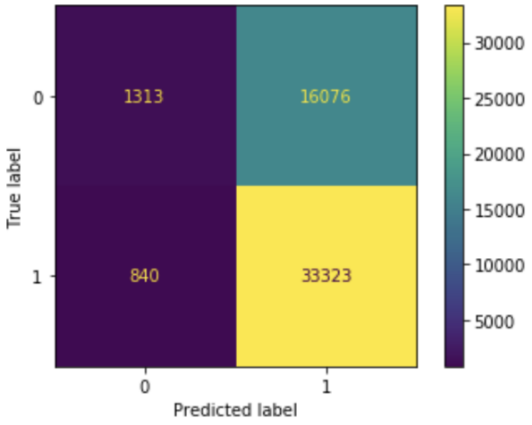

Figure 5: Confusion Matrix for Naive Bayes on All Positions.