The Bachelor(ette)

By Sarah Bashir, Ethan Ong, and Lindsey Tam

Abstract

Twitter and the TV shows the Bachelor and the Bachelorette are key components of American culture. This projects seeks to better understand the relationship between these two media forms. Using text analysis, we explore the polarity of tweet content as well as associated words that people use to describe the show. These trends are visualized and vary by state, show season, and time. Using all of this information, we begin the process of creating a predictive model. Though the current state of the model yields uninterpretable results, we build the foundation of a potentially promising way to use Twitter data to predict the outcome of each show.

Data

This project utilizes two years worth of Twitter data, which was generated shared with use by Professor Mike Izbicki. Data is stored on a supercomputer as a series of JSON files but parsed into smaller CSVs using a Python script. Data for each season was extracted by looking at the dates between times contestants were announced and one week after the final episode was aired. For each season, a set of key words were used to identify relevant tweets. Key words included the twitter handle of each contestant, as well as 'thebachelor', 'thebachelorette', and 'bachelorabc'. Intially, words like 'bachelor' were included, but this led to high amounts of noise. By narrowing the key words to contestants' twitter handles, the data set became significnatly smaller, but we are more confident that the tweets are relevant to the show of interest. In addition to parsing tweets into geographic location, time, and tweet content, we were also able to measure the sentiment of each tweet. A package called TextBlob was used to assign sentiment from a scale of -1 to 1, with -1 being negative, 0 being neutral, and 1 being positive.

A critical problem of this method is dealing with contestants who do not have Twitter. This was extremely problematic for the Bachelorette 2018, because the winner of the season had no Twitter account. To compensate for this, we parsed a second dataset for this season. This is likely to be a noisy dataset, but the results of the model can be compared to see if the noise made a significant impact. Another flaw in the data was the varying size of each data set. For instance, the Bachelor 2018 was very large, consisting of 1,914 tweets. This is likely due to the fact that this season had a unique ending where the Bachelor ended up leaving the finalist for the runner-up in a very short time frame. Other seasons had very few, nearly 300 tweets. This is something to be mindful of when interpreting results. I

Example of parsed twitter data from the Bachelor 2018:

```r
cleaned_data <- read.csv(file = 'Bachelor_2018.csv')
head(cleaned_data)
```

```
##   X              city                           date polarity state
## 1 0 South Burlington Fri May 18 03:48:18 +0000 2018   0.0000    VT
## 2 1        Manhattan Thu May 17 18:40:39 +0000 2018  -0.0625    NY
## 3 2           Queens Thu May 17 22:43:36 +0000 2018   0.0000    NY
## 4 0            Hooks Sat May 19 01:48:48 +0000 2018   0.0000    TX
## 5 1            Hooks Sat May 19 01:50:59 +0000 2018   0.0000    TX
## 6 2      Lindenhurst Sat May 19 00:03:15 +0000 2018   0.3125    NY
##
## 1                                                             Mike Johnson is not the her
## 2 This is the dumbest show in the world. IT's all fake everyone saying they love each other after sc
```

```
## 3                                                                                     @JohnPaulWebe
## 4 @MarkWoodsmall @packersfan86 @mushwear @jimmyjamny @Babchik @EvCoRadio @jpwilson1982 @JohnPaulFAL @
## 5 @MarkWoodsmall @packersfan86 @mushwear @jimmyjamny @Babchik @EvCoRadio @jpwilson1982 @JohnPaulFAL @
## 6                 The most underrated Led Zeppelin member! John Baldwin aka John Paul Jones. Musical
```

For the visualizations, the data was wrangled this way. . . .

```
#CHANGE THIS
#wrangled_data <- read.csv(file = 'data/car-speeds.csv')
#head(wrangled_data)
```

For the predictive model, we used a dataset that summarized contestant information of total tweet count and average tweet sentiment. We only looked at 3 seasons of data, which included the Bachelor (2018-2019) and the Bachelorette (2019). Example of predictive model data.

```
generated_data <- read.csv(file = 'model_data.csv')
head(generated_data)
```

```
##   X              X0        X1  X2 status
## 1 0        thebkoof 0.1840278 297 winner
## 2 1 laurenburnham91 0.1250230  66  loser
## 3 2  KendallPatrice 0.2456223 186  loser
## 4 3     tiarachel91 0.1430435 501  loser
## 5 4  whats_ur_sign_ 0.1433547 439  loser
## 6 5   seinnefleming 0.2112450  66  loser
```

Visualizations

The visualizations associated with this project were generated in Tableau.

Ethical Considerations for predictive models

It is critical to point out that no interpretation should be made about these models. We were only able to train this model on 3 seasons worth of data (not 4 because one season's winner had no twitter handle). A better model would have included data from all seasons of the bachelor and bachelorette. We are framing this model as an approach to how the model would look with all of the seasons of data.

The 5 classification models

These models were implemented using Python's scikit learn library. A quick overview of each model is attached below:

Algorithim for predictive models

1. Bootstrapp parsed twitter data for one season
2. Get average tweet sentiment and total tweet count per contestant
3. Shuffle contestant order
4. Repeat steps 1-2
5. Repeat steps 1-3 for all seasons of interest

Therfore, if we repeat step for 1000 times for 3 seasons, the data used for the model will have 3000 entries.

Based on the way dataset was created, the classification model will have 30 classes (contestants), 60 dimensions/features (avg sentiment + total tweets for each contestant), and a binary outcome (winner or not winner).

The 5 classification models

These 5 classification models were implemented using Python's scikit learn library. A quick overview of each model is attached below:

- Naive Bayes is a supervised classification model that uses Bayes' theorem of conditional probability and also uses the 'naive' assumption that feautures are independent. In this case, the Naive Bayes model is trying to predict whether or not a contestant will given their tweet count and average sentiment, under the assumption that tweet count and sentiment are independent (which they are not). Therefore, as more data is inputted into a model, accuracy is expected to go down because the independence assumption does not always hold true, as in this case. In general, a Naive Bayes model creates overly-simplified assumptions, which make it a poor model to use on real world data. It is included in this project in order to compare performance with other models.
- Linear Discriminant Analysis is a supervised classification model that creates a linear decision boundary that seperates multiple classes. LDA has two main uses: dimensionality reduction and linear classification. For dimensionality reductions, LDA tries to project the data into another space/dimension that minmimzes distance from the mean while minimizing variance of each class; this projection can make the data linearly seperable in a lower space when it was not linearly serperable in its original space.
- Logistic Regression is a classification algorithim that generates the probability of binary dependent variables. In our case, the dependent variable is whether or not a contestant will win or not win based on total tweet count and avergae tweet sentiment. It is similar to linear regession, except the outcome is binary (not continuous) and the seperating boundary is 'S' shaped.
- Linear Support Vector Classification tries to linearly seperate training data into classes. A linear SVC can be though of as a SVM. with a linear kernel. Linear SVC tends to perform better with larger sample sizes and is more computationally efficient because it is using a linear kernel.
- Stochastic Gradient Descent Classifier is a linear classifier that utilizes SGD for training. This means that for each data point, the gradient (or minima) of loss is estimated. The SGD classifier has a similar loss function as compared to Logistic Regression, but the solver each one uses is different. Since Logistic Regression uses GD and not SGD, we expect the SGD classifier to be the most computationally efficient out of these 5 models. Like the Naive Bayes model, SGD Classifier is not commonly used in practice.

Model with 3 seasons worth of data

Based on the way dataset was created, the classification model will have 89 classes (contestants), 2 dimensions/features (avg sentiment + total tweets), and a binary outcome (winner or not winner).

The results are highly questionable.

```
knitr::include_graphics("chart.png")
```

	name	accuracy
0	nb	[0.9444444444444444]
1	lda	[0.8888888888888888]
2	lr	[0.9444444444444444]
3	svc	[0.8888888888888888]
4	sgd	[0.9444444444444444]

```r
knitr::include_graphics("chart.png")
```

	name	accuracy
0	nb	[0.9444444444444444]
1	lda	[0.8888888888888888]
2	lr	[0.9444444444444444]
3	svc	[0.8888888888888888]
4	sgd	[0.9444444444444444]

Conclusions

Reminder that **no conclusions** can be drawn from the predictive model. It is mainly here in order to setup a model that has access to more seasons of data.

From the visuals, we gain a clear understanding on the geographic distribution of the tweets and how polarity varies across states.

Next Steps

For the predictive model, more seasons' worth of data can be added. This would require Twitter data from 2002 - 2017 to be available. Furthermore, hyperparameters in each classification model can be tuned to yield better results.

New Insights

Completing this project required gaining a better understanding of the 5 classification models mentioned above. Furthermore, generating the visualizations required learning Tableau, which none of the team members had experience in. Accessing data through a supercomputer (ssh-ing through the machine) was also new to the group members abd represeted a major hurdle in intially collecting data.