Let’s Gamble – Simulation of Blackjack Card Game

**Prashant Kubsad**

ISYE 6644 | Prof. Goldsman

2021

Table of Contents

[Introduction: 2](#_Toc70974062)

[Background: 2](#_Toc70974063)

[Problem Statement: 3](#_Toc70974064)

[Dataset Overview: 3](#_Toc70974065)

[Twitter Sentiment Training Data 3](#_Toc70974066)

[Coca-Cola Specific Twitter Data*:* 3](#_Toc70974067)

[Methodology: 3](#_Toc70974068)

[Cleansing 4](#_Toc70974069)

[Vectorization 4](#_Toc70974070)

[Sentiment Analysis With Classification 6](#_Toc70974071)

[Sentiment Analysis Results 7](#_Toc70974072)

[Cluster Analysis 8](#_Toc70974073)

[Evaluation: 10](#_Toc70974074)

[Results: 10](#_Toc70974075)

[Sources/References: 13](#_Toc70974076)

[Source Code 13](#_Toc70974077)

## Abstract:

This report investigates simulation of blackjack game and evaluates common strategies used in a game of blackjack. In this study, assumption is Casino is playing with standard house rules and using a standard 52 card deck. This simulation should work fine for single-deck hand shuffled game or 6- deck machine shuffled games. Each card that will be drawn is random, independent, and identically distributed. The simulation consists of outcomes of several hands and several rounds of game for each strategy. It also compares the outcome if those strategies are not used. The major finding of the simulation is how profitable are few of the strategies like doubling down on hard 11, splitting 8s, not splitting 5s yield a very high return. While strategies like not doubling down on 12 when dealer has face up card as 3, will yield in a marginal better return but not as high as other frequently used strategies.

## Background and Description of problem:

Blackjack is the most widely played casino banking game in the world[1]:342. The game is played with one or more decks of 52 cards. Blackjack players do not compete against each other. The game is a comparing card game where each player competes against the dealer. At a blackjack table, the dealer faces five to seven playing positions from behind a semicircular table. Between one and eight [standard 52-card decks](https://en.wikipedia.org/wiki/Standard_52-card_deck) are [shuffled](https://en.wikipedia.org/wiki/Shuffling) together. To start each round, players place bets in the "betting box" at each position. The player whose bet is at the front of the betting box controls the position, and the dealer consults the controlling player for playing decisions; the other bettors "play behind". The dealer deals cards from their left to their far right. Each box gets an initial hand of two cards visible to the people playing on it. The dealer's hand gets its first card face up, and a second card face down (the hole card), which the dealer peeks at but only reveals when it makes the dealer's hand a blackjack.

The objective of the game is to win money by creating card totals higher than those of the dealer's hand but not exceeding 21, or by stopping at a total in the hope that dealer will bust. On their turn, players can choose to do one of the following:

* "hit" (take a card),
* "stand" (end their turn and stop without taking a card),
* "double" (double their wager, take a single card, and finish),
* "split" (if the two cards have the same value, separate them to make two hands)
* "surrender" (give up a half-bet and retire from the game).

Wins are paid out at even money, except for player blackjacks, which are traditionally paid out at 3 to 2 odds.

**Basic Strategy:** Each blackjack game has a basic strategy which prescribes the optimal method of playing any hand against any dealer up-card so that the long-term [house advantage](https://en.wikipedia.org/wiki/House_advantage) (the expected loss of the player) is minimized. Based on the 2 cards you got, and face up card or dealer, there are various options you can play to reduce the edge of casino and increase your chances of winning the round. An example of a basic strategy is shown in the table below:

Table, calendar

Description automatically generated

In this project, we will simulate multiple rounds and see how each of these strategies work in your favor. In the rest of the report, you will see the python code that uses the above table for decision making and determine the earnings. The code will draw cards randomly from a deck for both players and casino. This simulation plays a set of multiple rounds of games (around 100 ) and 100 such sets resulting in 100 outcomes. We have done T-Statistic analysis on the both the sets of outcome(with strategy and without strategy) and visualize that will clearly show how profitable each strategy is.

One of the most famous Coca-Cola consumer research cases dates back to the 1970s in the “Race Against Pepsi”. Though Coca-Cola had a significant lead in the market share, Pepsi released a campaign revealing consumers preferred Pepsi every time in a blind taste test because it was sweeter. As a result of this study, Coca-Cola released “New Coke” a sweeter version of regular Coke, however, it was one of the biggest failures causing a revolt among customers. The reason it failed was because in a blind taste test, the cola isn’t being paired with the meals consumers normally drink it with and only a few sips were given instead of a whole can. Coca-Cola learned that while user studies can be helpful, they do not always paint the full picture.

Fast forward 40 years, the company has learned its lesson to keep constantly monitoring the tastes and preferences of its consumers in order to address and iterate on its current products. Today, consumers are quick to turn to Twitter in order to describe their affinity toward or distaste for the products on the shelves. Live Twitter is the perfect way to understand what consumers are saying, thinking and feeling about Coca-Cola products at real time without spending millions of dollars on focus groups and studies.

## Problem Statement:

How might we derive the areas of Coca Cola consumer dissatisfaction by utilizing Twitter data?

## Dataset Overview:

In order to address this problem statement, we will be requiring two data sets: (1) Twitter sentiment training data and (2) Coca Cola specific Twitter data.

### Twitter Sentiment Training Data

In building our model, developing training and testing data to build the model was one of the biggest challenges faced. We decided on using an already labeled dataset made available by Plural Sight to train/test our initial model for sentiment analysis. This dataset consists of over 400,000 tweets not specific to any company or user and has already been conveniently labeled with a binary value classifying each tweet in the data set as either positive or negative.

### Coca-Cola Specific Twitter Data*:*

After we have trained our model using this prelabeled dataset, we also had to pull the Coca-Cola tweets to answer our decided problem statement. Because we wanted to gather all tweets addressing Coca-Cola, only tweets with the hashtag #cocacola were pulled to avoid any other noise in the dataset. This dataset was pulled on April 28 2021, and contains 2,080 rows of tweet data. This data is the crux of our sentiment and cluster analysis and can be replaced by any company’s dataset of tweets.

## Methodology:

In approaching this problem, the following workflow has been established:

**1.**

**2.**

**3.**

**4.**

**5.**

**7.**

**6.**

**8.**

### Graphical user interface, text, application Description automatically generatedCleansing

After downloading our datasets, we first started by cleansing and normalizing our data because we cannot just pass in raw data. To begin the cleansing process, we followed a series of procedures to prepare the tweet for vectorization. Because tweets are unstructured data, it is crucial for us to be able to create consistency in each tweet and make sure we are valuing the right words and pieces of the tweet for our sentiment analysis. For example, taking the example tweet shown to the right, below is the cleansing process performed:

I love the new cherry flavor!

|  |  |
| --- | --- |
| Cleansing Step | Output |
| Remove URLs | I love the new cherry flavor! The taste is so incredible and refreshing! @CocaCola #cocacola #softdrinks |
| Remove Hashtags | I love the new cherry flavor! The taste is so incredible and refreshing! @CocaCola |
| Remove @ Mentions | I love the new cherry flavor! The taste is so incredible and refreshing! |
| Convert to Lowercase | I love the new cherry flavor! the taste is so incredible and refreshing! |
| Remove Punctuations | I love the new cherry flavor the taste is so incredible and refreshing |
| Remove Stop-words (library provided by NLTK.corpus) | love new cherry flavor taste incredible refreshing |

### Vectorization

Next, the data must then be vectorized, which converts words and word tokens into numbered vectors. This step is crucial in quantifying English words in order to run any machine learning algorithm on them. To perform vectorization, we were found two different effective approaches: (1) Bag of Words model and (2) Term Frequency -Inverse Document Frequency (TF-IDF).

#### Bag of words model

This is the simplest form of converting text representation to numbers and extract features from text. It describes occurrence of words within a document. In order to build the bag of words we first developed a vocabulary of known words. This step involves constructing a *document corpus* which consists of all the unique words in the whole of the text present in the data provided. It is often compared to something like a dictionary where each index will correspond to one word and each word is a different dimension.

For example, below are two tweets pertaining to Coca-Cola (note these examples do not reflect cleansed tweets such that stop-words/punctuation/etc. are removed):

**Tweet #1**

**Tweet #2**

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

If we collect the unique words from the above two tweets and assign each to an index, we will have:

*Document Corpus*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Word | This | cherry | flavor | is | very | tasty | and | affordable | not |

After we have constructed a document corpus, we will then record a count of the presence of our known words. For example, if we take Tweet #1 and plot the count of each word, we will have a row that corresponds to the index of the unique words and a row that corresponds to the count.

*Word Count: Tweet 1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Count | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

After converting both tweets into such vectors we can compare different sentences and calculate the Euclidean distance between them to check if two tweets are similar or not. Thus, if there are no words in common the Euclidean distance would be much larger and vice-versa.

Term frequency-inverse document frequency model (TF-IDF)

The next approach we found to vectorizing our data is called TF-IDF with two distinct parts (1) Term Frequency and (2) Inverse Document Frequency.

Term Frequency (TF) is used in connection with information retrieval and shows how frequently a word occurs in a tweet. Term frequency indicates the significance of a particular term within the overall document.

|  |  |
| --- | --- |
|  |  |
|  |  |

Inverse Document Frequency (IDF): The inverse document frequency is a measure of how much information the word provides, i.e., if it’s common or rare across all documents. It is used to calculate the weight of rare words across all documents in the corpus. The words that occur rarely in the corpus have a high IDF score. It is the logarithmically scaled inverse fraction of the documents that contain the word.

|  |  |
| --- | --- |
|  |  |

Thus, TF-IDF is calculated as:

As shown from the equation, TF-IDF gives larger values for less frequent words in the document corpus and vice versa. TF-IDF value is high when both IDF and TF values are high i.e the word is rare in the whole tweet dataset but frequent in a particular tweet.

### Sentiment Analysis With Classification

As mentioned, our approach was to use a predefined dataset that has more than 400K tweets that are already marked with 0 or 1 (0 being negative sentiment, 1 being positive sentiment). After vectorizing these 400K tweets using both above methods of vectorization, we split the dataset into train (80%) and test (20%) data. We trained the data for classification using two different models: (1) Multinomial Naïve Bayes and (2) Logistic Regression.

#### Multinomial Naive Bayes classifier:

Naive Bayes classifier is best suited for text classification, spam filtering and sentiment analysis. Naive Bayes classifier works on the Bayes rule, given by:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

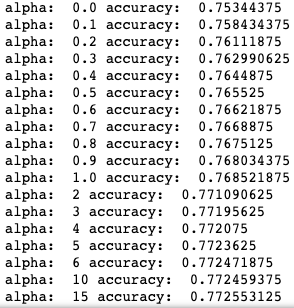
[Multinomial Naïve B](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB)ayes implement the Naive Bayes algorithm for multinomially distributed data. The distribution is parametrized by vectors . The parameters is estimated by a smoothed version of maximum likelihood, i.e., relative frequency counting:

|  |  |
| --- | --- |
|  | each class , where is the number of features (in text classification, the size of the vocabulary) |
|  | the probability of feature appearing in a sample belonging to class . |
|  | is the number of times feature appears in a sample of class in the training set T |
|  | total count of all features for class y |
| α | α ≥ 0 accounts for features that are not present in the learning samples and prevents zero probabilities in further computations. |

We have used the Sk-learn package’s MultinomialNB method to run the above logic with different values for smoothing parameters. We opted to go with a smoothing parameter of 15 because it becomes consistent after α = 2 as shown in the graph below.

Icon

Description automatically generated



**MultinomialNB with Different Smoothing Parameters**

#### Logistic Regression

Logistic regression is a classification algorithm, used when the value of the target variable is categorical in nature. As learnt in the classes, logistic regression classifies data using probability of a data belonging to a category. We define a threshold for this probability of data, and if it crosses the threshold, we can safely classify it to that category. Logistic regression follows sigmoid function:

### Sentiment Analysis Results

Combining all the approaches we tried the following combinations and recorded the accuracy of each:

|  |  |  |
| --- | --- | --- |
| Vectorization | Classification | Accuracy |
| TF-IDF | Multinomial NB | 0.772553125 |
| TF-IDF | Logistic Regression, max -iter 250 | 0.787696875 |
| TF-IDF | Logistic Regression, max -iter 500 | 0.788209375 |
| Bag of words | Multinomial NB | 0.773078125 |
| Bag of words | Logistic Regression | 0.78565 |

As shown, all the models with their respective combinations returned very similar accuracy rates. Looking at these five, we chose the model that yielded the highest accuracy even if just by a small margin. This combination was TF-IDF vectorization and Logistic Regression classification with a *max-iter* of 500. Next, we apply this model to our Coca-Cola Twitter data which has been cleansed and vectorized per the methods above, we can see that the model classifies tweets into 369 *negative* Tweets and 1124 *positive* tweets. See below for examples of positive and negative tweets.

|  |  |
| --- | --- |
| Negative Tweets | Positive Tweets |
| RT @glbl301 : These are only a few instances, but there are many, many more. The more I looked up, the more cases there seemed to be of #Nestlé and other beverage giants like #CocaCola and #Pepsi depriving water poor communities around the world of access to clean drinking water. | 4PM ET - The New Retail master class and networking w/ #CocaCola Barry Thomas Microsoft @ricardo\_belmar #Salesforce @MGTalksRetail Proximis @KhamtanThavy @cathymccabe Proximity Insights @Winston\_W\_Ma global tech investor and more @rwthurston1 Retail Pride |
| "RT @PatrioticSocia1 : It’s okay to be white.There is nothing wrong with it.#cocacola#BoycottCocacola" | Beverage giant joins a growing list of global corporations sponsoring a protected reserve in the Amazon. #FMTNews #CocaCola |
| RT @fordmb1 : The right amount of #CocaCola is zero. Not Coke Zero, but Zero Coke. #Woke #WokeBreakingPoint #boycottcoke #BoycottCocacola #Hypocrites https://t.co/7lme0lTsmv | RT @OriginalFunko : This Pop! is sure to be the cherry on top of your collection. Coming Soon: Pop! Funko: @CocaCola - Cherry Coca-Cola can🍒. Pre-order yours now, Cheers! |

### Cluster Analysis

Clustering falls under the realm of unsupervised machine learning algorithms – We do not know before-hand the different clusters that are present in the data, and we expect the algorithm to give us insights into the different clusters available. While it is not straightforward to identify the best way to cluster data, there are multiple algorithms readily available to assist with this – We chose the K-means algorithm as our clustering algorithm to help perform the clustering of similar tweets. As we learned, the efficiency of the K-means clustering is sensitive to the initial starting points for each of the cluster centroids – it is easy to start with some bad cluster centers and have clustering results which are not easily decipherable. Thus, we tried two different ways to assign the initial cluster centroids for the K-means algorithm and compared the outcomes from both: (1) Having K-Means automatically select the best cluster centers randomly and (2) Use the Jaccard Distance metric to identify the tweets which are extremely uncorrelated with each other to identify the initial center points.

#### Data Cleansing and Preparation

The data for clustering analysis is the negative tweets identified by sentiment analysis step above. We run similar data cleansing steps as above resulting in a vector matrix like:

|  |  |
| --- | --- |
| word | tf-idf score |
| water | 0.304960 |
| many | 0.269237 |
| are | 0.269237 |
| more | 0.260634 |
| Glbl301 | 0.188627 |
| access | 0.176928 |

#### Finding the K-Value

Chart, line chart

Description automatically generatedWe initially tried multiple K-values manually to identify the proper number of clusters that would make the clustering meaningful using ‘k-means++’ initialization of cluster centroids. We found that K = 2, 3 and 4 were too low of values. Clusters were very unevenly sized, with majority of tweets falling into just one cluster and only a few tweets falling into other clusters. While K = 10, 11, and 12 were too high, as we could not decipher any meaning from these high number of clusters since there were too few tweets falling into each cluster. For a more accurate K-Value we finally utilized the elbow plot method in order to observe the infamous “kink in the curve”. Based on the plot to the right it looks as if K= 5 or K = 6 seems like the appropriate number of clusters. In trying both, we can see that K=5 is a more evenly distributed cluster. Below is a table of the distribution of each cluster at each of the K values:

|  |  |  |
| --- | --- | --- |
| Cluster Number | Number of Tweets Per Cluster (K=5) | Number of Tweets Per Cluster (K=6) |
| 0 | 276 | 276 |
| 1 | 61 | 93 |
| 2 | 36 | 61 |
| 3 | 30 | 36 |
| 4 | 104 | 30 |
| 5 | - | 11 |

#### Centroid Initialization

In choosing the method of centroid initialization for these clusters, there were three main possible routes to take:

1. **Choosing ‘k-means++’ as the initialization parameter of SK-Learn K-means to assign centroid points:** As per SK-learn documentation, the ‘k-means++’ centroid initialization selects initial cluster centers for k-mean clustering in a smart way to speed up convergence. We tried this method to see if the K-means algorithm out of the box produces a good clustering or not.
2. **Choosing initial centroid points using the Jaccard Distance calculation of centroid points:** TheJaccard index is also called the Jaccard similarity coefficient is used in understanding the similarities between sample sets. While the Jaccard Index calculates the SIMILARITY between two sets the Jaccard Distance measures the DISSMILIARITY between two sets and is defined as Jaccard Distance = (1 - Jaccard Index). Formulaically, it can be written as:

In this method, we calculated the Jaccard Distance between pairs of tweets from the entire tweets collection to identify tweets which were totally dissimilar from each other. As a rule of thumb, pairs of tweets with the least Jaccard Distance estimate proved to be reasonable selection for our initial cluster centers. Once the cluster centers were identified, our K-means algorithm performed the below:

1. Initialize cluster centroids to the 5 most dissimilar tweets using Jaccard’s distance
2. For each incoming document (i.e tweet vector), calculate the cosine similarity between the tweet and the tweet at each of the cluster center.
3. Assign the tweet to the cluster which has the most similarity with the cluster centroid
4. Repeat this process until all the tweets have been processed.
5. **Choosing ‘random’ as the initialization parameter of SKLearn Kmeans to assign centroid points**: As per SKLearn documentation, the ‘random’ centroid initialization selects K rows at random from data for the initial centroids. We also tried this method to identify if the clustering gets better if the centroids were chosen totally at random.

## Evaluation:

The output of the K-means clusters were fed into a Word Cloud package, which helped us visualize the different key words that were prominent across the different clusters. Based on our experiments above, we observed that the overall clustering performed by the below 3 methods were similar – but the K means algorithm where we identified initial clusters using Jaccard Distance metric did a better job of clustering more tweets into each cluster, thus giving the word cloud better interpretability. Interestingly, the K means algorithm where centroids were chosen at random did a similar clustering as the K-means algorithm with ‘k-means++’ method of centroid initialization.

Below are how the cluster sizes looked like for the negative tweets of Coca Cola after each of these algorithms. Please note that we cannot compare the cluster counts 1:1 between each of the methods due to randomness incorporated in the K-means heuristic.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **K-Means++ centroid** | **Jaccard** | **Random** |
| *Cluster1* | 276 | 35 | 104 |
| *Cluster2* | 61 | 67 | 36 |
| *Cluster3* | 36 | 10 | 276 |
| *Cluster4* | 30 | 34 | 61 |
| *Cluster5* | 104 | 366 | 30 |

## Results:

We looked at the different tweets which were on each of the clusters from the K-means algorithm to identify what the defining themes were within each of the clusters. Below were the 5 different themes we were able to capture out of the clustering from the negative tweets for Coca Cola – Cluster numbers in no order

#### Cluster 1

* Text

  Description automatically generated
* The most common tweet/retweet within this cluster was
  + *"RT @PatrioticSocia1 : It’s okay to be white.There is nothing wrong with it.#cocacola#BoycottCocacola"*
* It looks like this retweet/negative sentiment was broadly due to a training conducted by Coca Cola where they suggested employees be “less white” (<https://www.entrepreneur.com/article/366132>)

#### Cluster 2

* Text

  Description automatically generated
* The tweets within this cluster were all along the lines of investing and suggestions about selling Coca Cola’s stock.
* The tweets seem to suggest that even Coca Cola’s stock price is going to be more expensive, and buying crypto currencies is the cheapest option today.
* One of the sample tweets from this cluster is below:
  + *RT @CryptoJoker12 : don't be trapped in the short-term volatility. just a year ago #Bitcoin was $5k and #xrp at $0.17. Inflation is coming. Even #CocaCola gonna be more expensive. In such a market, I don't think #Gold  #Silver  #BTC #DGB #xrp #xlm  #cryptocurrency gonna be any cheaper.*

#### Cluster 3

* Text

  Description automatically generated
* Most of the tweets seem to have fallen within this cluster and so this cluster seems to show a variety of patterns
* Some of the negative sentiments within this cluster seems to be on the high price of some of the coke products

#### Cluster 4

* Text

  Description automatically generated
* This one was an interesting cluster – Most of the tweets within this cluster were in response to a marketing campaign by Coke where they asked folks on twitter what the right amount of ice in Coca Cola is. The Tweet is below for reference:
* Text

  Description automatically generated
* A sample tweet from this cluster
  + *RT @fordmb1: The right amount of #CocaCola is zero. Not Coke Zero, but Zero Coke. #Woke #WokeBreakingPoint #boycottcoke #BoycottCocacola #Hypocrites* [*https://t.co/7lme0lTsmv*](https://t.co/7lme0lTsmv)

#### Cluster 5

* Text

  Description automatically generated
* The tweets within this cluster were mostly centered around a comment made by Chinese Foreign Minister Wang Yi, which was retweeted multiple times.
* Below is the tweet for reference
  + *RT @SpokespersonCHN: #Democracy is not #CocaCola, which, with the syrup produced by the United States, tastes the same across the world. The world will be lifeless and dull, if there is only one single model and one single civilization. --State Councillor & FM Wang Yi*

For comparison, below are the clusters we got for K-Means ++ method:

**Text

Description automatically generated**

Text

Description automatically generated

## Sources/References:

* <https://scikit-learn.org/stable/modules/naive_bayes.html#multinomial-naive-bayes>
* <https://www.pluralsight.com/guides/building-a-twitter-sentiment-analysis-in-python>
* <https://medium.com/analytics-vidhya/fundamentals-of-bag-of-words-and-tf-idf-9846d301ff22>
* <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>
* <https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/>
* <https://kambria.io/blog/logistic-regression-for-machine-learning/#:~:text=Logistic%20regression%20is%20a%20classification,either%20a%200%20or%201>
* <https://ieeexplore.ieee.org/document/5340335>

## Source Code

<https://github.com/Prazhant/ISYE6740_Project>