**Individual Project 3: Machine Learning on Streaming Data**

**CS367**

**Cassandra Mason**

**10/28/24**

Table of Contents

[Introduction 2](#_Toc181052430)

[The Dataset 2](#_Toc181052431)

[The Ingest Job 2](#_Toc181052432)

[Exploratory Analysis 6](#_Toc181052433)

[The Machine Learning Algorithm 6](#_Toc181052434)

[Results 6](#_Toc181052435)

[Discussion 6](#_Toc181052436)

[References 7](#_Toc181052437)

# Introduction

This project applies a machine learning algorithm on streaming data from tweets. The goal is to learn useful and actionable insights from the streaming data about tweets.

# The Dataset

The data is a demo from PubNub that streams tweets as they are posted. The subscribe key for the stream and the channel name are publicly available to tie my project to this data (*Demo*, n.d.). Each tweet comes with key value pairs. The keys are:

* Source: all from Twitter
* Text: the tweet content
* Posted from: the app from which the tweet was posted (Android, I phone, Instagram, etc.)
* Tweeted from location: country from which tweet was sent
* User name
* User profile location: Where the user set their home location to
* User follower count: number of followers the user has
* Timestamp: date and time tweet was posted

From these keys, some themes and questions we could ask of the data include: What is the sentiment of each tweet (positive, neutral, negative)? What patterns in location, time and topic can we find? Is there a correlation between follower count and posting frequency?

# The Ingest Job

I used PubNub to ingest the data into my local python environment. My local program subscribed to the ‘pubnub-twitter’ channel with it’s subscribe key. I defined a class, MySubscribeCallback, that formats the tweet data into a dictionary, creates a dequeue with a max length, and appends tweets to the dequeue. The point of the dequeue is to easily add items to one end and remove them from the other. New tweets are pushed onto the end of the queue. Once the queue hits the max limit, an added tweet will cause the oldest tweet to be popped off. This creates a rolling window from which to create a pandas dataframe to use for analysis.

A screenshot of a computer program

Description automatically generated

The results of printing this dataframe is a constantly evolving table of 100 rows. The screenshots show the first section of each of the defined columns (based on the keys in the dictionary). Each screenshot shows a different set of columns.

A screenshot of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The data still needs to be cleaned. There are some null values to remove. I would want to verify the correct datatype for each column. There is an interesting problem of tweets being in different languages, which could make comparisons and analysis more challenging. I’m not sure what to do about that.

# Exploratory Analysis

Exploring the data to better understand the data helps us determine what sort of questions to ask, variables to explore, and machine learning algorithms to use.

Descriptive statistics on the user follower count, such as the mean and median number of followers and the standard deviation explores the only numerical feature in this dataset. We could also calculate frequency of posts by the same user for the current analysis window using the timestamp and username data.

To further explore location, a map with dots representing tweets would be an interesting visual. A frequency plot would show words that are commonly used to help us find popular topics and tweet sentiment. Tweet volumes across timestamps could help us find peak posting times and weekly trends.

# The Machine Learning Algorithm

Supervised vs unsupervised and why chosen.

Based on the exploratory analysis, a machine learning algorithm can help us dive deeper and answer questions we have from the initial analysis. We could use logistic regression to predict how the frequency a user tweets changes with how many followers they have (Arcitura Education, 2021). The K Nearest Neighbors algorithm is a classification algorithm that would help us predict if a tweet’s sentiment is positive, negative or neutral (Arcitura Education, 2021). These are examples of supervised machine learning, where the dependent variable is known and the algorithm is trained using both independent variables that affect the dependent variable, and the resulting dependent variable.

An unsupervised machine learning algorithm is good for data that only contains independent variables. Rather than training and testing the model, the algorithm is left to find its own patterns. For this data set, clustering the tweets by location, follower count, and tweet activity could lead to insights with no supervision from the designers. KMeans clustering is an example of unsupervised machine learning (Brownlee, 2016).

# Results

Accuracy of model (k-fold cross validation). Results of model.

# Discussion

Interpretation of what the numbers in the model mean. Context. Business Insights.

# References

Arcitura Education. (2021, June 16). *2 supervised learning techniques that aid value predictions*. Enterprise AI. https://www.techtarget.com/searchenterpriseai/post/2-supervised-learning-techniques-that-aid-value-predictions

Brownlee, J. (2016, March 15). Supervised and Unsupervised Machine Learning Algorithms. *MachineLearningMastery.Com*. https://www.machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/

*Demo: Easily Connect & Test Real-Time Streaming Data*. (n.d.). PubNub. Retrieved October 28, 2024, from https://www.pubnub.com/demos/real-time-data-streaming/