**DSC 681-002 - Fall 2024: Applied Machine Learning**

**Group Project Hackathon**

**Professor Yue Han**

**Group 1:**

**Alma Monreal**

**Minh Khue Nguyen**

**Riya Shah**

**Shakif Farhan Shah**

**Table of Contents**

**Project goal and objective……………………………………………………………….3**

**Data Description…………………………………………………..……………………….4**

**Analyses and Results……………………………………………….…………………….8**

**Discussion…………………………………………………………………………………..9**

**Code………………………………………………………………………………………...12**

**Project goal and objective**

The goal of this project is to develop a model to predict the popularity of a tweet by cleaning, comparing and analyzing the data provided. The model will be developed using the general machine learning process.

The objective of this project and of creating the model is to be able to process data provided to better understand the most relevant factors that can make a tweet go viral.

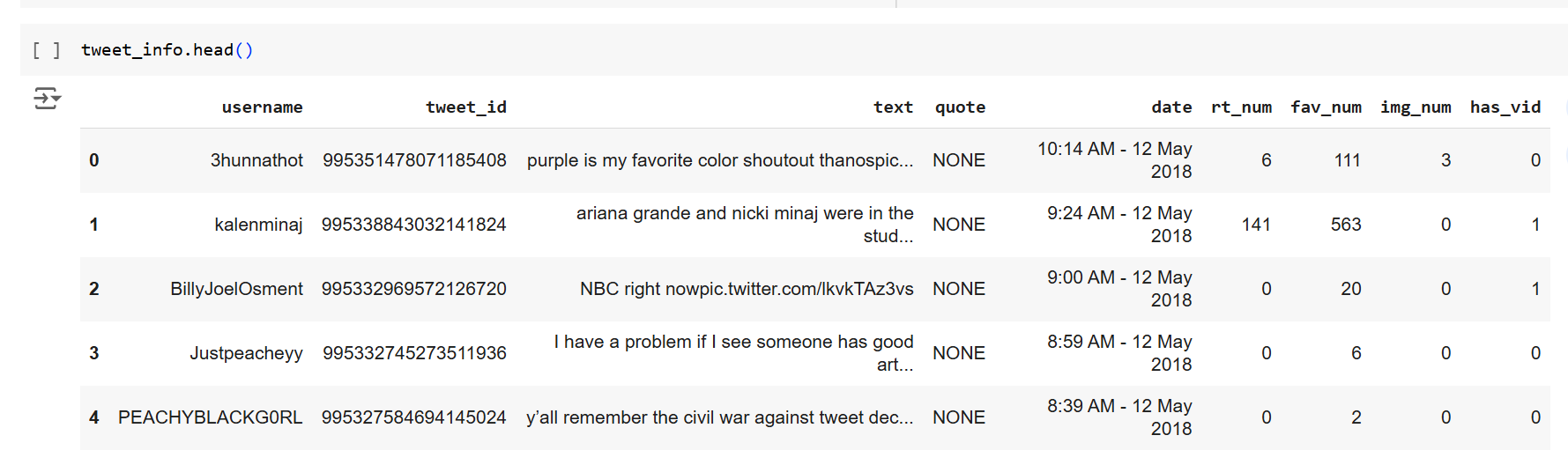
The data provided contains items related to tweet information and user profile information.

**Data Description**

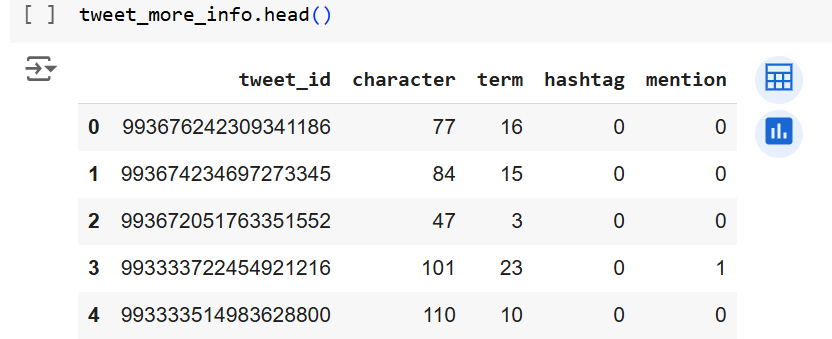
B1. What is the dataset for your project? If there is a link to your dataset, please provide the link. If not, please provide the first 5 rows including the headers of your dataset:

There are three datasets provided. 1) tweet\_info 2) tweet\_more\_info and 3) user\_profile.

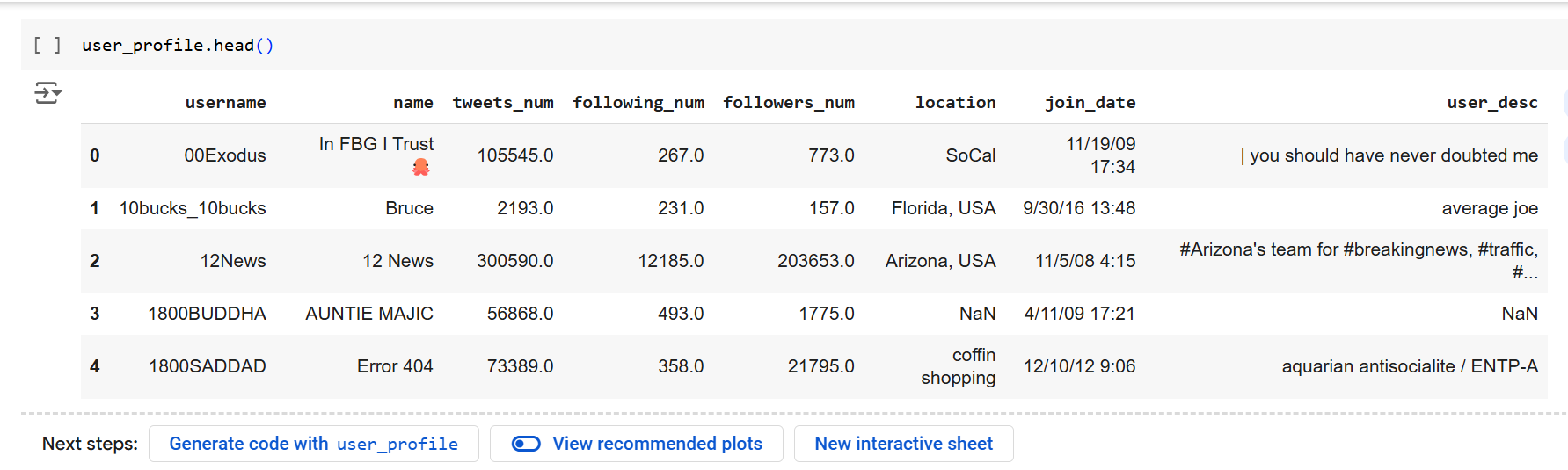
1. tweet\_info



1. tweet\_more\_info



1. user\_profile



B2. What is the dependent/output variable? What are the independent/input variables? Why do you choose these variables?

**Dependent/Output Variable**: The number of retweets (rt\_num). This variable measures the popularity of a tweet.

**Independent/Input Variables**:

* hashtag: Number of hashtags in the tweet.
* img\_num: Number of images included in the tweet.
* has\_vid: Whether the tweet included a video.
* mention: Number of mentions in the tweet.
* followers\_num: Number of followers of the tweet's creator.
* tweets\_num: Total number of tweets posted by the user.

We chose these variables because the number of retweets is the dependent variable that we want to know. Which depends on other factors which are the independent variables mentioned above. Those independent variables are important things to consider towards the popularity of a tweet and therefore it helps to analyze what makes a tweet go viral in order to predict future outcomes of the popularity of a tweet or to recreate those same characteristics in future tweets to make them go viral.

B3. Please provide the statistics table of your cleaned data. Below is an example, yours can be a little different.

| **Variable** | **Number of Observation** | **Min** | **Max** | **Mean** | **Standard deviation** |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Retweets (Dependent variable)** | 10000 | 0 | 215361 | 853.837 | 7046.603497 |  |
| **Hashtags (independent variable)** | 10000 | 0 | 7 | 0.0746 | 0.358117 |  |
| **Followers number (independent variable)** | 1.000000e+04 | 0 | 8.826666e+07 | 1.701220e+06 | 8.553986e+06 |  |
| **Fav Number (independent variable)** | 10000 | 0 | 559948 | 2207.59 | 14517.38 |  |
| **Tweets Number (independent variable)** | 10000 | 0 | 512242 | 29302.4 | 48569.86 |  |

**Analyses and Results**

C1. What models are you using for this project and why?

We use a support vector machine with regression to help us predict the number of retweets the tweet will get. SVR is effective for predicting continuous numerical values, such as the number of retweets. It is also a great tool for handling data and creating predictions even with a limited sample size.

C2. What are the results? Which model works better?

Due to the numerical nature of rt\_num, it is more suitable for a regression model like SVR, designed for predicting continuous outputs.

If the dependent variable were categorical, classification models would be more appropriate. For this project, predicting tweet popularity relies on numerical engagement metrics, justifying the use of regression.

**Discussion**

D1. What are the implications of your analyses and results?

The implications of our analyses include:

**Key Factors of Popularity:** The analysis identifies that features like the number of followers, favorites, hashtags, and total tweets by the user have significant influence on the number of retweets.

**Predictive Power:** The SVR model has the ability to predict the popularity of tweets, providing insights for marketers and influencers to optimize their content for the best engagement.

**Broader Applications:** These results can guide businesses in creating social media strategies and content by focusing on measurable attributes like hashtags, mentions, and visuals to increase their engagement and growth.

D2. What are some potential ways to improve your models in the future?

**Add More Features**: Include new data points like sentiment analysis or keywords in tweets.

**Use Better Models**: Try other models like Random Forest or Neural Networks and compare performance.

**Fine-Tune Parameters**: Optimize settings like SVR’s kernel or regularization.

**Expand the Dataset**: Use more tweets or create data for training.

**Engineer Features**: Create new variables like retweets per follower or normalized hashtag counts.

**Avoid Overfitting**: Use techniques like cross-validation and regularization to make the model generalize better.

D3. What are your main takeaways of this project (or any of the projects for this course)?

Main takeaways from this course are the process of building and evaluating a machine learning model, from data cleaning to model interpretation and presenting them in an insightful manner. Learning the strengths and limits of Support Vector Regression, especially in predicting numerical outcomes like retweets.

Also, to have the ability for critical thinking. Developing a mindset for questioning, going the extra mile and improving models to achieve better accuracy.

Lastly, gaining insights into how data science can be applied to real-world problems like optimizing content of various business entities for their social media platforms.

D4. What are your suggestions/feedback on how to improve the class experience of this course?

It is a well designed course with a lot of useful information, but I feel that 2 months is too short of a time frame to learn the basics applied machine learning in depth. Would be better if it was a 4 month course.

**Code**

[HACKATHON.ipynb](https://colab.research.google.com/drive/1WmxhV-a9QTi4JPdi-cQbstcshpPZrvV6?usp=sharing)

# Step 1: Setting working directory and importing libraries

from google.colab import drive

drive.mount('/content/drive')

import os

os.chdir('/content/drive/MyDrive/Colab Notebooks/Applied Machine Learning/DSC 681 Hackthon Project Data')

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVR

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import seaborn as sb

# Step 2: Load datasets

tweet\_info = pd.read\_csv('tweet\_info.csv')

tweet\_more\_info = pd.read\_csv('tweet\_more\_info.txt', delimiter='\t') # Assuming tab-separated file

user\_profile = pd.read\_csv('user\_profile.csv')

tweet\_info.head()

tweet\_More\_info.head()

user\_profile.head()

# Step 3: Check for missing values

print(f"tweet\_info missing values: \n{tweet\_info.isnull().sum()}")

print(f"\ntweet\_more\_info missing values: \n{tweet\_more\_info.isnull().sum()}")

print(f"\nuser\_profile missing values: \n{user\_profile.isnull().sum()}")

# Step 4: Remove missing values and duplicates

tweet\_info = tweet\_info.drop(columns=['text', 'quote']) # Drop rows where 'text' or 'quote' columns have NaN values

user\_profile = user\_profile.drop(columns=['name', 'location', 'join\_date', 'user\_desc']) # Drop rows where specified columns have NaN values

# Step 5: Show the first few rows of each dataset

print(f"tweet\_info: \n{tweet\_info.head()}")

print(f"\ntweet\_more\_info: \n{tweet\_more\_info.head()}")

print(f"\nuser\_profile: \n{user\_profile.head()}")

# Step 6: Check for duplicates in the DataFrames

print(tweet\_info.duplicated().sum())

print(tweet\_more\_info.duplicated().sum())

print(user\_profile.duplicated().sum())

# Step 7: Consolidate the data

merged\_data = tweet\_info.merge(tweet\_more\_info, on='tweet\_id', how='left')

merged\_data = merged\_data.merge(user\_profile, on='username', how='left')

merged\_data.head()

# Step 8: Fill missing values and check merged data

print(merged\_data.isnull().sum())

# Step 9: Sampling the data

data = merged\_data[['followers\_num', 'rt\_num', 'fav\_num', 'tweets\_num', 'hashtag']]

sampledata = data.sample(n=10000, replace=False)

print(f"sampledata:\n{sampledata.info()}")

print(f"\nsampledatadescribe: \n{sampledata.describe(include='all')}")

# Step 10: Split the data into dependent (y) and independent (X) variables

y = sampledata['rt\_num']

# Safely select only the available columns in merged\_data

X = sampledata[['followers\_num', 'fav\_num', 'tweets\_num', 'hashtag']]

# Step 11: Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

print(f" X\_train: \n{X\_train}")

print(f"\ny\_train: \n{y\_train}")

# Step 12: Correlation analysis

print(sampledata['rt\_num'].corr(sampledata['followers\_num']))

samplecor = sampledata.corr(method='pearson')

print(samplecor)

#Step 13: Producing correlation through heatmap.

sb.heatmap(samplecor,

xticklabels=samplecor.columns,

yticklabels=samplecor.columns,

cmap='RdBu\_r',

annot=True,

linewidth=0.5)

# Step 14: Standardize the features

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Step 15: Train the SVM model

model = SVR(kernel='linear')

model.fit(X\_train, y\_train)

# Step 16: Make predictions

y\_pred = model.predict(X\_test)

y\_pred

# Step 17: Evaluate the model

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("\nModel Evaluation:")

print(f"Mean Absolute Error: {mae}")

print(f"Mean Squared Error: {mse}")

print(f"R2 Score: {r2}")