

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

```
[ ] tweet_info.head()
```

	username	tweet_id	text	quote	date	rt_num	fav_num	img_num	has_vid
0	3hunnathot	995351478071185408	purple is my favorite color shoutout thanospic...	NONE	10:14 AM - 12 May 2018	6	111	3	0
1	kalenminaj	995338843032141824	ariana grande and nicki minaj were in the stud...	NONE	9:24 AM - 12 May 2018	141	563	0	1
2	BillyJoelOsment	995332969572126720	NBC right nowpic.twitter.com/lkvkTAz3vs	NONE	9:00 AM - 12 May 2018	0	20	0	1
3	Justpeacheyy	995332745273511936	I have a problem if I see someone has good art...	NONE	8:59 AM - 12 May 2018	0	6	0	0
4	PEACHYBLACKG0RL	995327584694145024	y'all remember the civil war against tweet dec...	NONE	8:39 AM - 12 May 2018	0	2	0	0

2) tweet_more_info

```
[ ] tweet_more_info.head()
```



	tweet_id	character	term	hashtag	mention
0	993676242309341186	77	16	0	0
1	993674234697273345	84	15	0	0
2	993672051763351552	47	3	0	0
3	993333722454921216	101	23	0	1
4	993333514983628800	110	10	0	0



3) user_profile

```
[ ] user_profile.head()
```

	username	name	tweets_num	following_num	followers_num	location	join_date	user_desc
0	00Exodus	In FBG I Trust 🚀	105545.0	267.0	773.0	SoCal	11/19/09 17:34	you should have never doubted me
1	10bucks_10bucks	Bruce	2193.0	231.0	157.0	Florida, USA	9/30/16 13:48	average joe
2	12News	12 News	300590.0	12185.0	203653.0	Arizona, USA	11/5/08 4:15	#Arizona's team for #breakingnews, #traffic, #...
3	1800BUDDHA	AUNTIE MAJIC	56868.0	493.0	1775.0	NaN	4/11/09 17:21	NaN
4	1800SADDAD	Error 404	73389.0	358.0	21795.0	coffin shopping	12/10/12 9:06	aquarian antisocialite / ENTP-A

Next steps: [Generate code with user_profile](#) [View recommended plots](#) [New interactive sheet](#)

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 Observatio n	Min	Max	Mean	Standard deviation	
Retweets (Dependent variable)	10000	0	21536 1	853.83 7	7046.603497	
Hashtags (independent variable)	10000	0	7	0.0746	0.358117	

Followers number (independent variable)	1.000000e+0 4	0	8.8266 66e+0 7	1.7012 20e+06	8.553986e+06	
Fav Number (independent variable)	10000	0	55994 8	2207.5 9	14517.38	
Tweets Number (independent variable)	10000	0	51224 2	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](#)

```
# 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"sampladata:\n{sampledata.info()}")

print(f"\nsampladata.describe: \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}")
```