95-891 Introduction to AI

Homework 2: Classification/Regression Task

Spring 2022

February 7, 2022

Due 11:59 PM EST February 24, 2022

Retweet Prediction during COVID-19

Goal

This homework is adapted from ACM 29th ACM International Conference on Information and Knowledge Management (CIKM2020) Analytics Retweet Prediction challenge (https://www.cikm2020.org/covid-19-retweet-prediction-challenge/). The goal is to "predict the popularity of COVID-19-related tweets in terms of the number of their retweets," meaning, predict a tweet's popularity (as measured by its retweets) via other factors related to it.

Your task is to find relations between features like number of followers, tweet time, number of favorites etc. In other words, the number of retweets may be viewed as a collective effort of many factors from the number of followers, the time one posts it, to the sentiment of a tweet.

Dataset Download

Please download the dataset from Canvas ("hw2.csv") and place it in the same directory of your Jupyter notebook. Specifically, we downsampled 100,000 tweets from the original datasets (more than 8 million samples).

Dataset Description

A detailed feature description is provided below (adapted from https://data.gesis.org/covid19challenge/). We provide some helpful hints in bullet points when applicable.

Tweet Id: Long.

Username: String. Encrypted for privacy issues.

Timestamp: Format ("EEE MMM dd HH:mm:ss Z yyyy").

- Does the post time relate to retweets? For instance, will posting in the morning attract more traffic? You may extract the post hour by pandas datetime method.

#Followers: Integer.

#Friends: Integer.

#Retweets: Integer. [The target variable to predict!]

#Favorites: Integer.

Entities: String. For each entity, we aggregated the original text, the annotated entity and the produced score from FEL library. Each entity is separated from another entity by char ";". Also, each entity is separated by char ":" in order to store "original_text:annotated_entity:score;". If FEL did not find any entities, we have stored "null;". You may skip this feature if it is too involved.

Sentiment: String. SentiStrength produces a score for positive (1 to 5) and negative (-1 to -5) sentiment. We split these two numbers with whitespace char " ". Positive sentiment was stored first and then negative sentiment (i.e. "2 -1").

- You will have to split the original column into two columns like positive_sentiment and negative_sentiment.

```
Sentiment   ->  Positive_entiment    Negative_sentiment
2 -1        ->          2                    -1
3 -2        ->          3                    -2
```

- A quick intro of sentiment analysis can be found here: https://en.wikipedia.org/wiki/Sentiment_analysis
- Do not forget to cast the datatype to integer

Mentions: String. If the tweet contains mentions, we remove the char "@" and concatenate the mentions with whitespace char " ". If no mentions appear, we have stored "null;".

- You may build a dictionary for storing all appeared mentioned and use whether a tweet has specific mentions as features (search for one-hot encoding for more information). However, this may make the feature dimensionality very high (imagine there are many unique)
- You may simply count the number of mentioned of objects as a feature.

Hashtags: String. If the tweet contains hashtags, we remove the char "#" and concatenate the hashtags with whitespace char " ". If no hashtags appear, we have stored "null;".

- You may treat Hashtag similarly as Mentions

URLs: String: If the tweet contains URLs, we concatenate the URLs using ":-: ". If no URLs appear, we have stored "null;" You may safely ignore this feature.

Your Task

1. Get the data from Canvas, and read into Python. I will recommend using pandas dataframe for it but it is your choice for selecting preprocessing procedures. It is up to you how you want to clean and preprocess the data for the following tasks. As long as the following parts could run, you will receive the full credit.
    a. As the first step, you will need to split sentiment feature into Positive_sentiment and Negative_sentiment
    b. You may consider extract other useful information, although it is not required:
        i. The specific hour of the post
        ii. Number of mentions

        iii. Number of hashtags
    c. Do not forget to fill in missing values. How to do imputation? May be mean, mode, or 0.
2. Exploratory Data Analysis
    a. Visualize each of the following variables (#Followers, #Friends, #Favorites) with histograms. You may want to use log to scale the variable like np.log10() so that the plot makes more sense.
    b. Show the correlation matrix between the following features (#Followers, #Friends, #Favorites, Positive_sentiment, and Negative_sentiment) in the dataset with respect to the **#Retweets**. Hint, you may use pandas' built-in corr() function (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html)
    c. What do these plots and correlation matrix signify?
    d. What do the signs (+/-) before the correlation magnitudes signify?
    e. If a feature shows 0 correlation to #Retweets, does that mean it could not contribute to prediction?
3. Split the dataset into train (the first 80,000 samples/rows) and test set (the last 20,000 samples/rows).
4. Build a model to **train a regressor to predict the #Retweets value given the values of other data attributes** (you may use additional features as we describe above but you are expected to use the basic ones including #Followers, #Friends, #Favorites) using at least two methods.
    a. Consider the regression methods taught in class
    b. Scikit learn (https://scikit-learn.org/stable/index.html) is a python library that provides implementation of many standard machine learning algorithms. You could find kNN, SVM, Random Forest, Decision Tree and other types of regressors here; although we discussed these methods in terms of classification in class, they can also be used as regresors (to predict the value of a continuous variable.
    c. XGBoost, lightGBM, and CatBoost are also great tools you may use
    d. We will recommend using at least one decision tree method, e.g., decision tree, random forest, adaboost, xgboost, so you could get a sense of the built-in feature selection effect. Linear regression with L1 regularization has a similar effect. A good tutorial can be found here: https://sebastianraschka.com/faq/docs/feature_sele_categories.html
5. Predict on the test set.
6. Compare the two regression methods you used in the previous part on the training and test sets. What evaluation metrics are appropriate?

Hints

There are a few columns in the HW2 data (Mentions, Hashtags, Entities) that compose a string like "PoliSeance KeepLookingUp6 bennyjohnson" or "null;". These are some internal features provided by the original dataset.

You might want to count the number of entities in these columns as a new feature for prediction. Maybe if a tweet mentions more entities, it is more likely or less likely to be retweeted. You are welcome to try other extraction as well.

There are two ways to count the number of words in Pandas Dataframe column. You could either go row by row but this will be slow or use pandas.apply() function (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.apply.html to apply a custom function to a column. A simple example is here (https://stackoverflow.com/questions/34962104/pandas-how-can-i-use-the-apply-function-for-a-single-column (Links to an external site.)). There are a few examples of using apply function in the Datacamp material

Taking "PoliSeance KeepLookingUp6 bennyjohnson" or "null;" as an example, you could write a custom function like:

```
def count_words(x):
    if str(x)=="null;":
        return 0
    else:
        return len(str(x).split(" "))  # first cast the input to string, and then split it into a list of words separated by " " and count
```

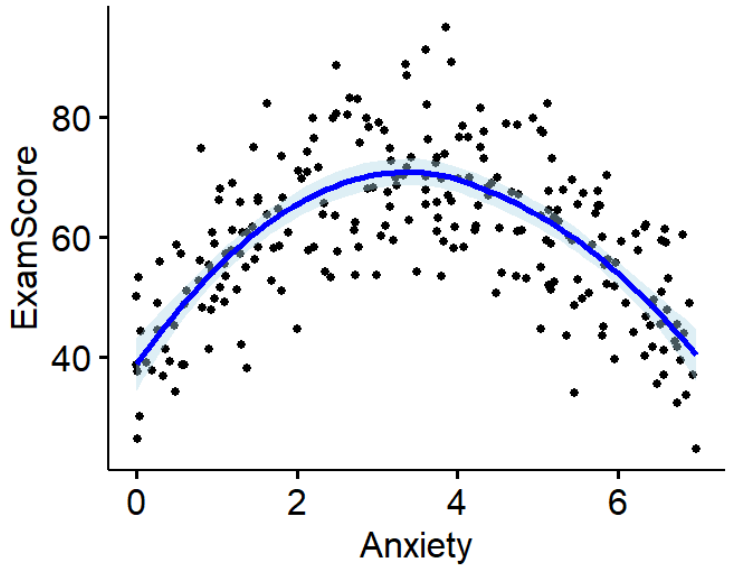
This is a rough example and you need to apply it to specific columns.

If this is too involved, please try your best and use as many features as possible. Skip the ones that are too complicated.

In HW2, you are being asked to create the correlation matrix among the target variable (#retweets) and the input features (e.g., #favoraties and #friends).

One question is to interpret the correlation coefficient, and all the people show the correct answer that a positive correlation suggests the two variables change in the same direction. The magnitude also matters, ranging from -1 to +1, where 0 denotes no linear correlation.

However, corr=0 just means there is no linear correlation, but it does not necessarily rule out the possibility of non-linear relationship. For instance, it can be a quadratic function like y=ax^2+b below, where the linear correlation can be 0.



Another possibility is given x and y are linearly uncorrelated, and z and y are also linearly uncorrelated. However, y may be correlated with {xz} together.

That being said, we should not simply remove zero linear correlation variables because they may be useful in a non-linear way. Hope this helps.

Submission

The submission should be a Jupyter Notebook with the name Intro_to_AI_HW2_<Andrew id>.ipynb.

You can write answers to all of these questions in the first cell of your jupyter notebook in a multiline comment. You could also submit a separate Word or . pdf file if you prefer. NOTE: DO NOT ATTEMPT TO SUBMIT THE DATASET. ONLY THE JUPYTER NOTEBOOK IS REQUIRED.

Other Links

Here are some useful links you may want to refer:

1. An overall example: https://bigdata-madesimple.com/how-to-run-linear-regression-in-python-scikit-learn/
2. General data cleaning: https://machinelearningmastery.com/basic-data-cleaning-for-machine-learning/
3. http://blog.davidkaleko.com/merging-data-sets-pandas.html

- how to extract year/day/hour information from a pandas dataframe. You may use this to extract "hour" information
- how to do correlation analysis

4. https://www.geeksforgeeks.org/python-pandas-split-strings-into-two-list-columns-using-str-split/

- how to split one pandas dataframe column into two columns. You may use this for splitting "sentiment" column

5. https://towardsdatascience.com/introduction-to-pandas-apply-applymap-and-map-5d3e044e93ff

- how to apply a customized function to a pandas dataframe by apply() function

6. https://www.geeksforgeeks.org/python-linear-regression-using-sklearn/

- Python | Linear Regression using sklearn