Rafael Hernandez
IST-736
Final Project Report

TEXT MINING SUPERHERO BACKGROUND AND POWERS

INTRODUCTION

A business organization needs to be aware of the emerging trends in their industry. There are many ways too evaluate the market and one of the most effective ways for a company to predict trends is by analyzing the data they acquire. The challenge comes from record keeping and data maintenance practices. It is likely that in a lifetime of an organization, records will be lost or destroyed. The task at hand is to use existing data to figure out and predict values that are missing or lost. The other task that this project will explore is the examination of different models and word vectorization process to increase the accuracy of prediction.

The data comes from Kaggle.com: https://www.kaggle.com/datasets/jonathanbesomi/superheroes-nlp-dataset. The data is a collection of superhero statistics, descriptions and information identifying who created the superheroes. There are 1449 unique values and 81 columns. The columns of interest are columns that describe the history of the superhero and describe the heroes super powers, and the column identifying which company created the superhero. The text data will be used to classify who created the superheroes. Within the creator column there are over 100 missing values. We do not know who created the superheroes for the missing values and classifying models will be created to predict who are the creators.

The task is to replicate situations that research organizations or companies have to manage. Researchers acquire data and information of records that are incomplete. Companies

lose data and from the information available, it is advantageous to use technology and machine learning to classify the missing data. Accurate classification will help researchers and corporations in recreating records that will provide knowledge, solve problems and inform them about the future.

DATA MUNGING AND PROCESSING

The datagram is uploaded as a csv file on google colab. A quick examination of the data frame and the shape is 1450 by 81. The data has 1450 rows and 81 columns. The focus for the project are two text columns and the column with the name of the creators.

MISSING DATA

```
Number of missing values in history column: 90
Number of missing values in power column: 364
Number of missing values in creator column: 139
```

Handling missing values varied in the three columns of interest. In the creator column, NaN's were replaced with the label unknown. We are interested in classifying who these creators are and dropping the missing values would hinder our objective. The history and power columns are text columns. The columns will be merged together and the column name will become text but before the columns are merged, NaN's are replaced with white space. The columns are merged and all white spaces are removed. A new dataframe is created out of all the unknown creator values. There is one data frame that will be used to vectorize and to train and test the models. There a second data frame of the values that we want to predict.

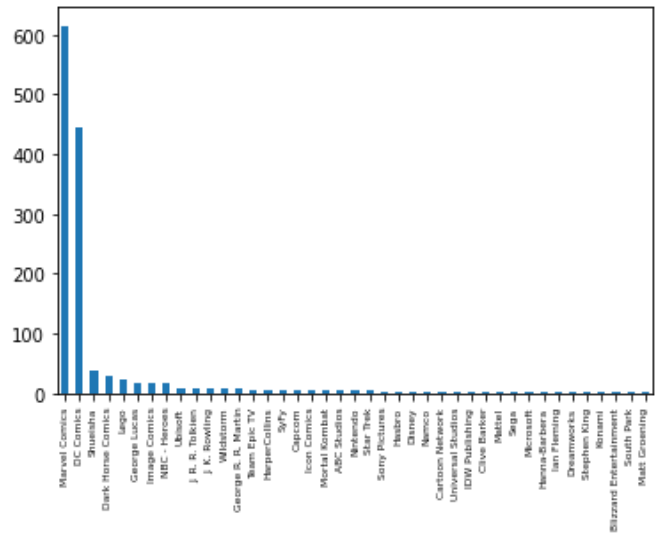
	text	creator
	Delroy Garrett, Jr. grew up to become a track ...	Marvel Comics
	He was one of the many prisoners of Indian Hil...	DC Comics
	Richard "Rick" Jones was orphaned at a young ...	Marvel Comics
	Aa is one of the more passive members of the P...	DC Comics
	Aaron Cash is the head of security at Arkham A...	DC Comics

Figure 1: Training and testing data frame

	text	creator
	During the Serpentine Wars Acidicus and the ot...	unknown
	As a top agent of the DEO Alex is in top physi...	unknown
	Alita known in Japan as Gally and originally n...	unknown
	The Anti Metahuman Adaptive Zootomic Organism ...	unknown
	The Anacondrai Serpent would help the Serpenti...	unknown

Figure 2: Unknown creator

```
Marvel Comics              615
DC Comics                  444
Shueisha                    37
Dark Horse Comics           29
Lego                        22
George Lucas                18
Image Comics                17
NBC - Heroes                16
Ubisoft                      9
J. R. R. Tolkien             8
J. K. Rowling                8
Wildstorm                    7
George R. R. Martin          7
Team Epic TV                 6
HarperCollins                6
SyFy                         6
Capcom                       5
Icon Comics                  4
Mortal Kombat                4
ABC Studios                  4
Nintendo                     4
Star Trek                    4
Sony Pictures                3
Hasbro                       3
Disney                       3
Namco                        3
Cartoon Network              2
Universal Studios            2
IDW Publishing               2
Clive Barker                 2
Mattel                       1
Sega                         1
Microsoft                    1
Hanna-Barbera                1
Ian Fleming                  1
Dreamworks                   1
Stephen King                 1
Konami                       1
Blizzard Entertainment       1
South Park                   1
Matt Groening                1
```



Figures 3 and 4: Superhero creators and the numbers of superheroes created by each creator

The final cleaning of the text columns removes all punctuation in both data frames. The text is ready for vectorization.

TEXT VECTORIZATION

Two different bag of words are created, CountVectorization and TF-idf Vectorization. The settings in both vectorizations are similar. Stop words are set to English. Encoding is set to latin-1 and min_df is set to two. The vectorizers have lower case words as the default which lowers all capital words. A lemmatizer was added to both vectorizers.

Model Building Problem

A problem kept taking place with the data that required for some of the creators to get dropped. The decision tree, Naïve Bayes and SVM models kept on setting precision and f-score at 0.0 for a lot of the categories. Regardless of the model and vectorization, precision and f-score were ill-defined. To rectify the problem, the training and testing splits were changed from 70/30 and 60/40 and regardless of the changes, the ill-defined warnings continued. The values that got a 0.0 on all the models were dropped. The final data frame contained six creators in total.

Marvel Comics	615
DC Comics	444
Shueisha	37
Dark Horse Comics	29
George Lucas	18
J. K. Rowling	8

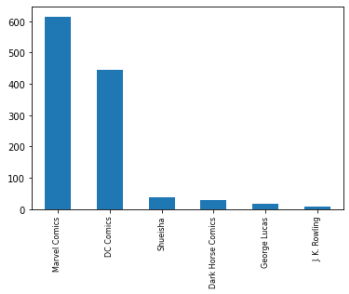


Figure 5 and 6: Creators in final training and testing data frame and their values.

The data was converted into training and testing data utilizing train_test_split. Labels were given the y values and text was given X. The training and testing split was set at 70/30. The data was randomized to make sure that the training and testing sets had all of the values with a respectable representation.

Four models were created. Two Multinomial Naïve Bayes models and two SVM models. The count vectorizer and the TF-idf vectorizers were used in Naïve Bayes and SVM, thus the naïve bays and the SVM contain one model that uses count vectorizer and another model that with TF-idf. The SVM models are linear and contain no C.

RESULTS

	precision	recall	f1-score	support
DC Comics	0.91	0.93	0.92	136
Dark Horse Comics	0.67	0.33	0.44	6
George Lucas	1.00	0.25	0.40	4
J. K. Rowling	1.00	0.50	0.67	2
Marvel Comics	0.94	0.96	0.95	186
Shueisha	1.00	0.92	0.96	12
accuracy			0.92	346
macro avg	0.92	0.65	0.72	346
weighted avg	0.92	0.92	0.92	346

Figure 7: naïve Bayes classification report for count vectorizer

	precision	recall	f1-score	support
DC Comics	0.98	0.71	0.82	136
Dark Horse Comics	0.00	0.00	0.00	6
George Lucas	0.00	0.00	0.00	4
J. K. Rowling	0.00	0.00	0.00	2
Marvel Comics	0.75	1.00	0.86	186
Shueisha	0.00	0.00	0.00	12
accuracy			0.82	346
macro avg	0.29	0.28	0.28	346
weighted avg	0.79	0.82	0.78	346

Figure 8: naïve Bayes classification report for TFidf vectorizer

Figures 7 and 8 are the classification reports for both naïve Bayes models. The model that used count vectorizer had bet accuracy and better precision, recall and f-score performances.

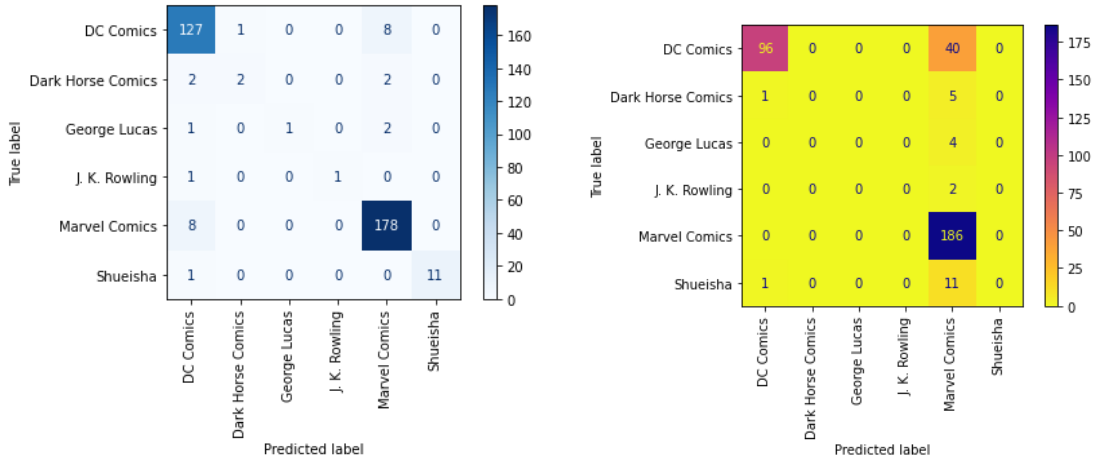


Figure 9(left): count vectorizer confusion matrix. Figure 10(right): tfidf confusion matrix

Figures 9 and 10 provide a break down of the number of predicted labels and true labels that were predicted accurately and inaccurately.

	precision	recall	f1-score	support
DC Comics	0.79	0.79	0.79	136
Dark Horse Comics	0.50	0.17	0.25	6
George Lucas	1.00	1.00	1.00	4
J. K. Rowling	1.00	0.50	0.67	2
Marvel Comics	0.85	0.88	0.86	186
Shueisha	0.80	0.67	0.73	12
accuracy			0.82	346
macro avg	0.82	0.67	0.72	346
weighted avg	0.82	0.82	0.82	346

Figure 11: SVM classification report for count vectorizer

	precision	recall	f1-score	support
DC Comics	0.90	0.92	0.91	136
Dark Horse Comics	1.00	0.33	0.50	6
George Lucas	1.00	0.50	0.67	4
J. K. Rowling	1.00	0.50	0.67	2
Marvel Comics	0.92	0.95	0.94	186
Shueisha	1.00	0.83	0.91	12
accuracy			0.92	346
macro avg	0.97	0.67	0.76	346
weighted avg	0.92	0.92	0.91	346

Figure 12:SVM classification report for tfidf vectorizer

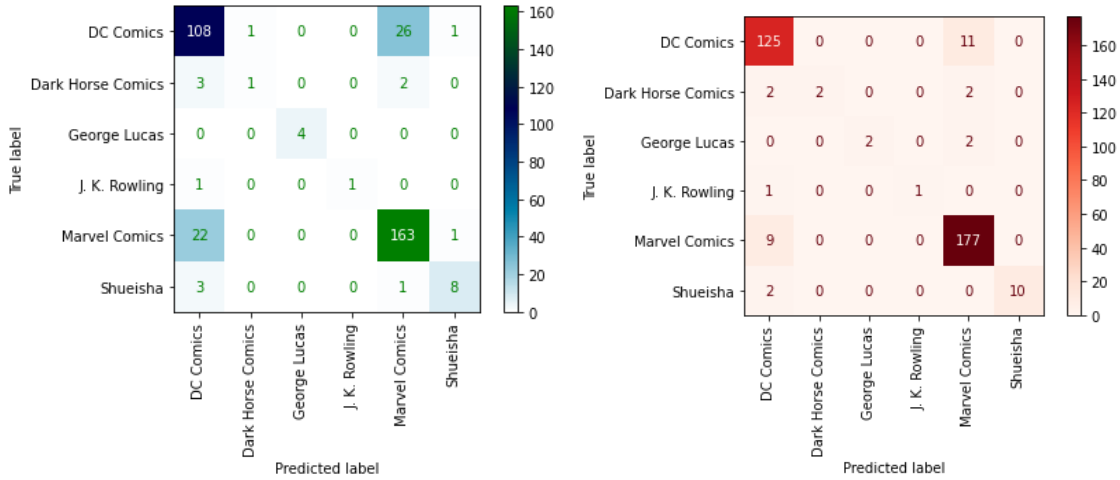
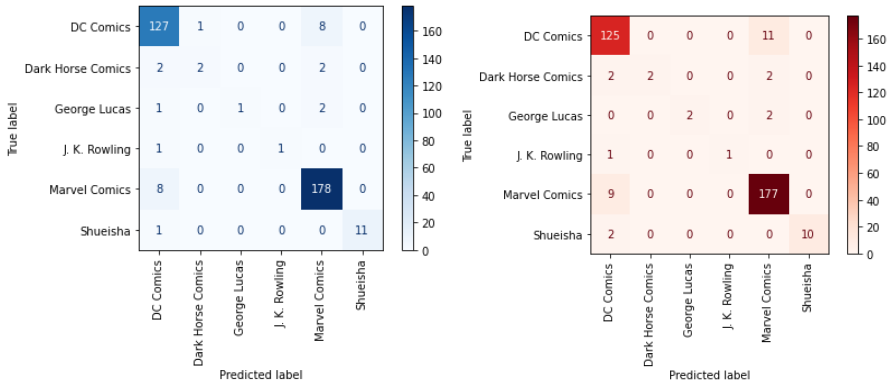


Figure 13(left): SVM confusion matrix for count vectorizer. Figure 14(right) SVM confusion matrix for tfidf vectorizer

CONCLUSIONS

The results are inversed from the Naïve Bayes models to the SVM Models. Both sets have one model with an accuracy of 92% and another with 82%. In the naïve Bayes models, the count vectorizer model has the 92% accuracy whereas in the SVM models it's the TF-idf vectorizer with the 92% accuracy. We can hair split the metrics on the classification report and opinions will differ on the importance of recall versus precision versus f-score.

Naïve Bayes SVM



Examining the confusion Matrix and it appears that the Naïve Bayes models is better at predicting DC Comics and Marvel comics than the SVM models.

WHO ARE THE UKNOWN CREATORS?

who are they	
DC Comics	72
Marvel Comics	63
Shueisha	3
Dark Horse Comics	1

Figure 15: Breakdown of who the unknown creators are using the count vectorizer naïve bayes model.

BIBLIOGRAPHY

https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f

https://stackoverflow.com/questions/54875846/how-to-print-labels-and-column-names-for-confusion-matrix

https://errorsfixing.com/sklearn-adding-lemmatizer-to-countvectorizer/

https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a