The Simpson Script Analysis

To classify from whom the scripts are and compare the classification performances of all different models.

吳欣育
資訊管理學系
國立台灣大學
大安區, 台北, 台灣
B07705018@ntu.edu.tw

徐芊綺
資訊管理學系
國立台灣大學
大安區, 台北, 台灣
B07705027@ntu.edu.tw

金民亞
資訊管理學系
國立台灣大學
大安區, 台北, 台灣
B07705043@ntu.edu.tw

陳柄瑞
資訊管理學系
國立台灣大學
大安區, 台北, 台灣
B07705052@ntu.edu.tw

吳昀蔚
資訊管理所
國立台灣大學
大安區, 台北, 台灣
R09725059@ntu.edu.tw

ABSTRACT

Adopting data from Kaggle, we performed a script classification analysis to tell which script is from which character. In the analysis, we built several models to do the classification, including Naïve Bayes classifier, SVM (Linear), SVM (Polynomial), SVM (RBF) and KNN. To compare performances of the models above, we calculated precision, recall, and F1 score to do the comparison. At the same time, we also compare the outcome of the model performance across different characters.

MOTIVATION

Cartoons are the daily joy of our childhood. Whether it is weekend pastimes or art studies, they have a profound impact on our lives. Then our team came up with a question: whether it is possible for us to determine which character is speaking only by the content of the script? The personality of the cartoon characters is especially bright, so maybe the lines of some characters are unique enough to tell. We choose a famous animation series, *The Simpsons*, to do the research. *The Simpsons* is an American adult animated sitcom created by Matt Groening for the Fox Broadcasting Company. The series is a satirical depiction of American life, epitomized by the Simpson family and other characters in the town. The show is set in the fictional town of Springfield and parodies American culture and society, television and the human condition. We consider such a special cartoon may be an interesting subject to the project.

DATA SOURCE

The Simpsons Dataset is adopted from the open source data website Kaggle (https://www.kaggle.com/prashant111/the-simpsons-dataset). The dataset contains the characters, locations, episode details, and script lines for approximately 600 Simpsons episodes.

DATA FORMAT

Originally, the dataset contains 132110 lines, which were spoken by 6722 characters as shown below:

	spoken_words	character_id
0	No, actually, it was a little of both. Sometim...	464
1	Where's Mr. Bergstrom?	9
2	I don't know. Although I'd sure like to talk t...	464
3	That life is worth living.	9
4	The polls will be open from now until the end ...	40
...	...	...
132105	I'm back.	464
132106	You see, class, my Lyme disease turned out to ...	464
132107	Psy-cho-so-ma-tic.	464
132108	Does that mean you were crazy?	119
132109	No, that means she was faking it.	4

32110 rows × 2 columns

Table 1: Original script lines

	id	name	normalized_name	gender
0	7	Children	children	NaN
1	12	Mechanical Santa	mechanical santa	NaN
2	13	Tattoo Man	tattoo man	NaN
3	16	DOCTOR ZITSOFSKY	doctor zitsofsky	NaN
4	20	Students	students	NaN
...	...	...	...	...
6717	5222	Ron Rabinowitz	ron rabinowitz	m
6718	5728	Martha Stewart	martha stewart	f
6719	1770	Officer Goodman	officer goodman	m
6720	1634	Evan Conover	evan conover	m
6721	1868	Agent Johnson	agent johnson	m

6722 rows × 4 columns

Table 2: Original speaking characters

However, we found that most of the lines are from the main character and specific roles around him. We filtered the characters who speak more than 1000 lines. The remaining character_id are 2, 1, 8, 9, 15, 17, 3, 11, 31, 71, 25, 139, 101, and 165. At the same time, compared to the main character who has spoken over 20000 lines, most characters only have spoken several thousands of lines. As a result, for those who speak over 10000 lines, we randomly filtered 3500 from the original data to deal with the issue of data imbalance.

Eventually, the data that we took as the input of the following steps are as follows:

	spoken_words	character_id
0	I don't care if he was filling in for Mel Zetz...	2
1	I spit on your corpse, advertiser-supported te...	2
2	Lisa? Is this what I've come to? Fighting over...	2
3	And I wouldn't have to pay it back for three m...	2
4	I'll just hide here.	2
...	...	...
33851	No children, your not seeing things, this, my ...	3
33852	Are you the substitute?	3
33853	Are you insane?	3
33854	Well, all right. Play friendly with your new t...	3
33855	Yay, Bart!	25

3856 rows × 2 columns

Table 3: Filtered script lines

id	name	count
1	Marge Simpson	3500
8	Bart Simpson	3500
9	Lisa Simpson	3500
17	Moe Szyslak	2810
3	Seymour Skinner	2390
11	Ned Flanders	2057
31	Grampa Simpson	1875
71	Chief Wiggum	1796
25	Milhouse Van Houten	1798
101	Nelson Muntz	1145
165	Lenny Leonard	1144
2	Homer Simpson	3500
15	C. Montgomery Burns	3121
139	Krusty the Clown	1720

Table 4: Filtered speaking characters

SOLUTIONS

1.1 Naïve Bayes - Bernoulli Model (NB-Bernoulli)

Naïve Bayes Classification is a probabilistic classification method which uses Bayes' theorem and adopts conditional and positional independent assumptions to perform classification. There are two ways of Naïve Bayes Classification, which are the Multinomial model and the Bernoulli model. The main difference between the two models is that the Bernoulli model determines a term's posterior probability based on whether they occur in a document, and the Multinomial model focuses on term frequencies. In the research, we applied the Bernoulli Naïve Bayes Classification model.

From Figure 1, we can see that the overall accuracy of the Bernoulli Naïve Bayes model is 0.39, which would be regarded as our baseline model for evaluating the performances of other models.

Furthermore, we took every classes' performances into account and found that classes with larger samples usually generated higher recall than small classes. For example, class 2, which has the most samples, reached 0.9 recall score. However, class 25, 101 and 165, which have the lowest samples all get the recall scores of 0. That means the imbalance data affects the model performance to some extent and we should take action to deal with the problem. Therefore, we applied random undersampling to larger classes. After the adjustment, we can see from Figure 2 that the recall scores for larger classes (e.g. class 2) decreases and those for small classes increases, which implies the balanced data mitigate the model's preference of assigning samples to larger classes.

We decided to use the balanced data for other models in later research.

	precision	recall	f1-score	support
1	0.47	0.27	0.34	2652
2	0.38	0.90	0.53	5474
3	0.43	0.04	0.08	480
8	0.39	0.23	0.29	2587
9	0.45	0.14	0.21	2247
11	0.44	0.01	0.02	396
15	0.53	0.10	0.17	627
17	0.21	0.02	0.03	557
25	0.00	0.00	0.00	339
31	1.00	0.01	0.02	381
71	0.67	0.01	0.01	371
101	0.00	0.00	0.00	255
139	0.44	0.01	0.02	335
165	0.00	0.00	0.00	238
accuracy			0.39	16939
macro avg	0.39	0.12	0.12	16939
weighted avg	0.41	0.39	0.31	16939

Table 5: Metrics of Bernoulli Naïve Bayes Classification Model with Original Data

	precision	recall	f1-score	support
1	0.22	0.49	0.31	685
2	0.18	0.27	0.22	718
3	0.64	0.17	0.27	495
8	0.15	0.54	0.24	703
9	0.25	0.28	0.26	735
11	0.82	0.05	0.10	424
15	0.49	0.34	0.40	630
17	0.35	0.22	0.27	543
25	0.00	0.00	0.00	366
31	0.56	0.02	0.04	411
71	0.72	0.03	0.07	381
101	0.00	0.00	0.00	241
139	0.61	0.04	0.08	337
165	0.00	0.00	0.00	204
accuracy			0.23	6873
macro avg	0.36	0.18	0.16	6873
weighted avg	0.36	0.23	0.20	6873

Table 6: Metrics of Bernoulli Naïve Bayes Classification Model with Balanced Data

1.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a kind of large-margin classifier. Its goal is to find a decision boundary (hyperplane) between two classes that is maximally far from any point in the training data. When the data is difficult to separate linearly in original data, then use a linear classifier in the higher dimensional space to separate the data. However, mapping to high dimensional space and computing dot products are computationally expensive.

Thus, kernel functions provide an easy and efficient way to map and compute the dot products which is called “kernel trick”. It does not really need to transfer data to high dimensional space, and the dot products are computed in terms of original data. In this project, we used three models of SVM. One is linear SVM, and the other two are polynomial, and radial basis function (RBF) by using kernel functions. Polynomial maps data into 3-dimensional space. RBF maps data into an infinite dimensional space. Usually, SVM is considered as a good model to train. In SVM, there are two important hyper parameters that we can tune which are C and gamma (in RBF). C determines soft-margin or hard-margin. The smaller C is, the more accepted outliers. When C is too small, it might occur under-fitting. On the other hand, if C is too big, it might occur over-fitting. Gamma determines curve rate to determine boundary.

1.2.1 SVM-Linear

We set hyper parameter C as 1 which means it tends to have a soft-margin that allows some outliers. The overall accuracy was 0.27 which was higher than our baseline model’s accuracy (0.23). It implies that Linear SVM is suitable for our classification problem.

	precision	recall	f1-score	support
1	0.27	0.38	0.31	713
2	0.18	0.31	0.23	666
3	0.33	0.26	0.29	457
8	0.21	0.28	0.24	682
9	0.21	0.35	0.27	711
11	0.51	0.21	0.30	429
15	0.35	0.43	0.39	611
17	0.32	0.37	0.34	549
25	0.29	0.08	0.13	357
31	0.39	0.14	0.21	388
71	0.39	0.18	0.24	353
101	0.81	0.15	0.26	229
139	0.42	0.14	0.21	336
165	0.62	0.06	0.11	245
accuracy			0.27	6726
macro avg	0.38	0.24	0.25	6726
weighted avg	0.33	0.27	0.27	6726

Table 7: Metrics of SVM Linear Model with Balanced Data

1.2.2 SVM-Polynomial

Similar to Linear, we set C as 1, and the other parameters set as default values. The overall performance was quite similar to Linear SVM. Its accuracy was 0.27 which was higher than our baseline model (0.23). We also trained before under-sampling (imbalanced) data via using the Polynomial SVM model. It implies that Polynomial SVM is suitable for our classification problem.

	precision	recall	f1-score	support
1	0.28	0.40	0.33	726
2	0.20	0.28	0.23	643
3	0.37	0.29	0.32	488
8	0.20	0.27	0.23	662
9	0.22	0.34	0.26	686
11	0.39	0.21	0.27	419
15	0.35	0.38	0.37	572
17	0.30	0.30	0.30	524
25	0.22	0.12	0.15	373
31	0.27	0.16	0.20	368
71	0.36	0.23	0.28	352
101	0.43	0.13	0.20	233
139	0.35	0.16	0.22	354
165	0.23	0.10	0.14	212
accuracy			0.27	6612
macro avg	0.30	0.24	0.25	6612
weighted avg	0.29	0.27	0.26	6612

Table 8: Metrics of SVM Polynomial Model with Balanced Data

1.2.3 SVM-RBF

RBF is usually considered as one of the best models. It also gave us the best results. We tried to adjust some hyper-parameters, such as C and gamma, in this model. We tried both C = 1 and C = 10 while gamma is a default value. Though, it didn't have a significant improvement or a worsening. However, when we tuned gamma into a big number (70), the results got a lot worse than before. It shows the best result when hyper-parameters are set as default-values. Thus, we came to a conclusion that we set the hyper-parameters as default values.

	precision	recall	f1-score	support
1	0.26	0.42	0.32	672
2	0.19	0.32	0.24	688
3	0.45	0.27	0.34	476
8	0.21	0.28	0.24	674
9	0.22	0.42	0.29	705
11	0.51	0.18	0.26	405
15	0.37	0.50	0.42	602
17	0.33	0.33	0.33	573
25	0.39	0.05	0.09	390
31	0.39	0.11	0.17	349
71	0.61	0.18	0.27	352
101	0.90	0.13	0.23	212
139	0.39	0.09	0.14	368
165	0.68	0.05	0.10	239
accuracy			0.28	6705
macro avg	0.42	0.24	0.25	6705
weighted avg	0.36	0.28	0.26	6705

Table 9: Metrics of SVM Polynomial Model with Balanced Data (C = 1, gamma = 'scale') - BEST PERFORMANCE

	precision	recall	f1-score	support
1	0.25	0.38	0.30	693
2	0.19	0.25	0.22	701
3	0.38	0.30	0.33	496
8	0.23	0.31	0.26	679
9	0.22	0.32	0.26	737
11	0.43	0.22	0.29	425
15	0.40	0.45	0.42	655
17	0.33	0.34	0.33	536
25	0.16	0.08	0.10	367
31	0.31	0.18	0.23	402
71	0.39	0.22	0.28	349
101	0.40	0.16	0.22	238
139	0.34	0.16	0.22	331
165	0.15	0.04	0.06	218
accuracy			0.28	6827
macro avg	0.30	0.24	0.25	6827
weighted avg	0.29	0.28	0.27	6827

Table 10: Metrics of SVM Polynomial Model with Balanced Data (C = 10, gamma = 'scale')

	precision	recall	f1-score	support
1	0.24	0.06	0.10	693
2	0.10	0.87	0.18	701
3	0.35	0.02	0.04	496
8	0.19	0.04	0.07	679
9	0.16	0.07	0.10	737
11	0.48	0.03	0.05	425
15	0.48	0.04	0.07	655
17	0.27	0.02	0.03	536
25	0.15	0.02	0.04	367
31	0.39	0.03	0.06	402
71	0.36	0.01	0.02	349
101	0.78	0.06	0.11	238
139	0.29	0.02	0.03	331
165	0.20	0.00	0.01	218
accuracy			0.12	6827
macro avg	0.32	0.09	0.07	6827
weighted avg	0.29	0.12	0.07	6827

Table 11: Metrics of SVM Polynomial Model with Balanced Data (C = 1, gamma = 70)

1.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbors is a type of supervised machine learning algorithm that assigns documents to the majority of their k closest neighbors. The most important step when conducting KNN is to determine an appropriate k. Therefore, we performed 10-fold cross-validation using GridSearchCV in the sklearn package and iterated *k* from 1 to 30 in order to find the one with the highest validation score. We performed KNN classification on balanced data. From Figure 9, we found that $k = 8$ generated the best model for respective data and the model achieved accuracy of 0.14 on testing data. The overall performances are worse than our baseline model (accuracy = 0.23), which means maybe KNN is not a suitable solution for our classification problem.

{'n_neighbors': 8}

	precision	recall	f1-score	support
1	0.12	0.25	0.16	684
2	0.11	0.20	0.14	680
3	0.14	0.09	0.11	480
8	0.13	0.19	0.15	710
9	0.14	0.20	0.17	727
11	0.20	0.06	0.09	400
15	0.21	0.17	0.18	592
17	0.15	0.06	0.08	585
25	0.10	0.06	0.08	367
31	0.12	0.12	0.12	386
71	0.24	0.06	0.09	364
101	0.42	0.15	0.23	240
139	0.21	0.06	0.10	344
165	0.15	0.04	0.07	229
accuracy			0.14	6788
macro avg	0.17	0.12	0.13	6788
weighted avg	0.16	0.14	0.13	6788

Table 12: Metrics of KNN Model with Balanced Data ($k = 8$)

CONCLUSION

We divided our conclusion into two aspects: comparison between models and classes (characters).

1.1 Comparison between Models

We compared the above models based on overall accuracy, recall, precision and F1 score. To get a sense of effectiveness on small classes and to avoid the

performances being dominated by large classes, we computed macro-averaged results instead of micro-averaged metrics.

	NB (Org. data)	NB (Bal. data)	SVM-linear	SVM-P.	SVM-rbf	KNN (8)
Acc	0.39	0.23	0.27	0.27	0.28	0.14
R	0.12	0.18	0.24	0.24	0.24	0.12
P	0.39	0.36	0.38	0.30	0.42	0.17
F1	0.12	0.16	0.25	0.25	0.25	0.13

Table 13: Macro-averaged metrics for each model

First of all, we set NB-Bernoulli as a baseline model. When we have a look at F1 scores, the F1 score of Naive Bayes is 0.16. All the F1 scores of SVM models are 0.25 which are quite larger than the Naive Bayes model. Among SVM models, while recall rates remain all the same, RBF's accuracy and precision rates are higher than the other two SVM models. The F1 score of KNN (0.13) is lower than NB. After evaluating the performances of each model in our classification, we concluded that SVM models, in general, showed good performances. RBF's accuracy was 0.28 while the other two SVM models' accuracy were 0.27. Since RBF maps data to infinite space, we can assume that it might show better performance than others. On the other hand, KNN showed the worst performance.

1.2 Comparison between Classes

We compared the performances between each class (characters). We labeled the 3 characters with best scores in red, and the 3 with the lowest in green for each model. Then, we selected the characters having at least 3 red scores as the ones with relatively better performances, while those who have at least 3 green scores are the ones with relatively worse performances. As shown in Figure 11, the characters with good performances are Marge Simpson (ID: 1), Seymour Skinner (ID: 3), C. Montgomery Burns (ID: 15), and Moe Szyslak (ID: 17). On the other hand, those with bad performances are Milhouse Van Houten (ID: 25), Krusty the Clown (ID: 139), and Lenny Leonard (ID: 165).

Character ID	NB	SVM-linear	SVM-P.	SVM-rbf	KNN (8)	Nc
1	0.31	0.31	0.33	0.32	0.18	3500
2	0.22	0.23	0.23	0.24	0.14	3500
3	0.27	0.29	0.32	0.34	0.12	2387
8	0.24	0.24	0.23	0.24	0.15	3500
9	0.26	0.27	0.26	0.29	0.16	3500
11	0.1	0.3	0.27	0.26	0.08	2054
15	0.4	0.39	0.37	0.42	0.17	3121
17	0.27	0.34	0.3	0.33	0.13	2808
25	0	0.13	0.15	0.09	0.09	1795
31	0.04	0.21	0.2	0.17	0.13	1870
71	0.07	0.24	0.28	0.27	0.08	1796
101	0	0.26	0.2	0.23	0.21	1145
139	0.08	0.21	0.22	0.14	0.06	1702
165	0	0.11	0.14	0.1	0.08	1143

Table 14: F1-score of the characters mapping to the models

We assumed that characters with good performances have a unique style of speaking, frequently using specific terms. On the contrary, those with bad performances may use general terms when talking, leading to a more uniform distribution of the terms in his/ her script lines. Thus, we generated word clouds of each character's whole script lines, aiming to link the performances and the habits of term using.

Table 15: Word Clouds for characters with good and bad performances.

However, after checking the word clouds, we found out that only a few terms act differently between classes such as *Homer*, *Bart*, *Smithers*, and they are mostly other characters' names. In general, we do not find significant differences of term distribution over classes and most characters tend to use common terms such as *Hey*, *Well*, and *Now*. Thus, the result cannot fully support our assumption. There are two possible reasons:

(1) Different sample size

As shown in Figure 11, although we operated down-sampling for the classes with too many samples, the problem of data imbalance still exists. Moreover, classes with larger sample size generally perform better than those classes with less samples. Thus, we may conclude that the imbalanced problem still impacts the performances over classes, leading to biased results.

(2) Not only terms with high frequencies are important.

Though the term distributions with highest term frequency do not show significant differences, it does not mean the whole term distributions over characters are similar. Since we can only observe the term distributions with highest term frequency using word clouds, we may miss the vital terms which impact greatly in classification. It is likely that there are other terms with less occurrence but can help when distinguishing the speaker.

REFERENCES

[1] Prashant B., The Simpsons Dataset, accessed 2021-10-19, https://www.kaggle.com/prashant111/the-simpsons-dataset