

Comparative Analysis of SVM Kernels for Large-Scale Sentiment Analysis on Sentiment140 Dataset

Shivangi Mishra, Pawan K Tiwari, Abhishek Singh, Deepanshu S Yadav

Department of Computer Science and Engineering, IET, Lucknow-UP (226021), India

2400521635017@ietlucknow.ac.in pawantiwari@ietlucknow.ac.in asingh.csed.cf@ietlucknow.ac.in
dyadav.csed.cf@ietlucknow.ac.in

Abstract- social media is the tool where people express their emotions and opinions on daily basis. There are various social media tool where earlier twitter now X is the most used social media tool used to express people emotions and therefore it generates large amount of data which are noisy and is bit challenging. This study present the analysis of support vector machine (SVM) kernels observe the impact of kernel function and dataset sizes on the classification problem. For the experiment sentiment140 dataset is used. This dataset is further divided into ten subset of varying size from 5k sub dataset to 1.6m (full dataset) then four SVM kernels linear, polynomial, sigmoid and radial basis function is evaluated. For the evaluation, accuracy, precision, recall, f1 and training cost is used mainly focusing on accuracy and training cost. The observation shows that polynomial kernel is best in terms of accuracy for shorter size of data whereas fail over the computational cost. Apart from that, linear kernel is best suited for large- scale sentiment analysis, provide scalability and strong generalization. These observation emphasize on the importance of performance of the model as well as model computational cost in real world machine learning applications.

Keywords – sentiment analysis, sentiment140, SVM, kernel, linear, polynomial, RBF, sigmoid.

1. Introduction

In recent years, there has been huge rise in social media users, specially twitter. It becomes important to analyse the opinion of the people for the purposes like building business strategies, political analysis etc., based on their sentiment analysis. The characteristics of twitter data set such as short and noisy text, uses of slangs, abbreviations and special characters makes twitter bit challenging for opinion mining or sentiments analysis.

To facilitate large scale sentiment analysis, Go, Bhayani, and Huyang introduced sentiment140 dataset in 2009 using distant supervision. The size of this dataset is 1.6 million labelled as positive and negative automatically based on emoticons. This dataset become benchmark dataset because of its size and accessibility. This dataset is also suitable for supervised machine learning algorithms.

When compared several machine learning algorithms, Support Vector Machine (SVM) performed better due to its ability to handle high dimensional sparse feature spaces[3]. Different kernels of the SVM provide different boundaries and computational trade off but due to data set size and training complexity, the kernel performance becomes the critical concern.

This study tells how different SVM kernels, behave across multiple sub data sizes of sentiment140 dataset. This study also focus on training time and computational feasibility with minimal pre-processing and also suggest practical implication.

Although there are various study on SVM model, but they are lacking in comparing SVM kernels while progressively increasing dataset and analysing accuracy and computation time, therefore it become little bit tough to decide best kernel for large scale real world deployment.

The objective of this research is as follows:

1. To train SVM on various kernel like linear, RBF, polynomial and sigmoid.

2. To compute accuracy and training time of various kernel of SVM.
3. Identifying best outperforming kernel for the real world sentiment analysis.

2. Literature review

Pang, Lee, and vaithyanathan demonstrated machine learning models such as naïve bayes, SVM and maximum entropy on movie review classification outperform traditional classification approach. Their work set a separate challenge of sentiment classification from standard text categorization[1].

A major breakthrough came when Go, Bhayani and Huyang introduces the sentiment140 dataset in 2009 which automatically label tweets as positive and negative using emoticons such as :) and :(, removing the need of manual annotation[2]. This dataset has binary polarity and is of large scale makes it suitable for supervised machine learning.

Cortes and Vapnik introduces the Support Vector Machine (SVM) which become the powerful algorithm for classification task. The reason why SVM is the powerful for text classification task is that it uses TF-IDF for feature generation. TF-IDF stands for Term Frequency-Inverse Document Frequency, is the statistical method to evaluate the importance of the word to a document in relation to a larger collection of the documents[4].

SVM performs better in text classification task compared to other traditional probabilistic methods demonstrated by Joachims in his paper[5]. SVM maximizes the margin between classes thus improving generalization by constructing an optimal hyperplane.

Several recent studies compare traditional machine learning models to several deep learning and transformer model, however, SVM always remains in focus due to its low computational requirements, high compatibility and strong baseline performance[6-11]. Few of the literature review are listed in table 1.

Table 1: literature review of the related work

Author(s)	Dataset Used	Key Findings
B. Pang, L. Lee, and S. Vaithyanathan (2002)[1]	Movie Reviews Dataset	One of the earliest sentiment analysis studies; SVM outperformed Naive Bayes and Maximum Entropy
A. Go, R. Bhayani, L. Huang (2009)[2]	Twitter Dataset (Sentiment140, 1.6M tweets)	Introduced distant supervision; SVM showed strong performance for tweet sentiment classification
V. N. Vapnik (1995) / Applied in sentiment studies[3]	Multiple text classification datasets	Established SVM as highly effective for high-dimensional sparse text classification
Thorsten Joachims (1998)[5]	Standard text categorization datasets (e.g., Reuters news corpus and similar high-dimensional datasets)	Suggested SVM as highly effective for text classification, shows svm outperform naïve bayes
Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin (2010)[6]	Various benchmark dataset	Provide step by step guide to apply svm, data scaling, parameter tuning, introduced grid search

Aliaksei Alessandro (2015)[7]	Severyn, Twitter dataset Moschitti	Proposed dl model using cnn for sentiment analysis
Yoon kim(2014)[8]	Multiple benchmark dataset	Provide simple yet effective cnn model
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019)[9]	Large unlabelled corpora	Introduced bert
Seyedali Minaee, Kalchbrenner et al. (2021)[10]	Nal Multiple dataset	Review paper, provide comparison of CNN, RNN and LSTM
Harjule et al. (2020)[13]	Sentiment140 + Twitter Airline	LSTM achieved best results, but SVM remained a strong and efficient classical baseline
Kumar et al. (2025)[16]	Twitter Sentiment Dataset	TF-IDF + SVM achieved strong classification accuracy and practical real-time performance
A. M. Rushdi-Saleh et al. (2011)[19]	OCA Movie Reviews Dataset	SVM achieved high accuracy and outperformed NB in Arabic sentiment classification
Apoorv Agarwal et al. (2011)[20]	Twitter Corpus	Compared tree kernel and feature- based models; SVM provided strong baseline performance

3. Methodology

After reviewing some sentiment analysis method in literature review section, we identified that there is a lack of comparing the kernels of the SVM. The focus of this study is to compare various kernel of SVM model, and for this purpose, TF-IDF word pre-processing technique is employed and the performance of these kernels are evaluated using confusion matrix, accuracy, precision, recall, macro f1 and training time.

3.1 Data set description

In this research sentiment140 dataset is used which is freely available on the website of kaggle. This dataset contains 1.6 million tweets which are labelled automatically using emoticons. Tweets are binary labelled in the form of 0 and 4. 0 is used to represent negative tweets and 4 is used to represent positive tweets. For the research convenience, tweet labels are converted into binary values i.e. 0 and 1. 0 is remain used for negative tweets and 1 used in place of 4 to represent positive tweets.

This dataset contains five attributes which are target, ids, date, flag, user and text and in this research only the target or label or text. The snap shot of the dataset and the customized dataset is shown in the figure 1 and figure 2.

	A	B	C	D	E	F
1	1467810300	Mon Apr 08 22:18:45 PDT 2009	NO_QUERY	..TheSpecialOne..	@valetfoot http://twitpic.com/2j1d - Awaw, that's a bummer. You shoulda got David Cam of Third Day to do it.	
2	1467810372	Mon Apr 08 22:19:49 PDT 2009	NO_QUERY	notthemelon	is upset that he can't update his facebook by texting it... and might cry as a result. :sob: today also. :Bw/	
3	1467810917	Mon Apr 08 22:19:53 PDT 2009	NO_QUERY	mattpus	@Kerican I Blvd many times for the bus. Managed to save 50%. The rest go out of bounds	
4	1467811384	Mon Apr 08 22:19:57 PDT 2009	NO_QUERY	ElecTF	my whole body feels tizzy and like its on fire	
5	1467811991	Mon Apr 08 22:19:57 PDT 2009	NO_QUERY	karol	@mattmillerless no, it's not behaving at all. I'm dead. why am I here? because I can't see you all over there.	
6	1467811372	Mon Apr 08 22:19:00 PDT 2009	NO_QUERY	jay_wild	@Kerican wait the whole cono	
7	1467811592	Mon Apr 08 22:20:03 PDT 2009	NO_QUERY	mybitch	Have a hug	
8	1467813394	Mon Apr 08 22:20:03 PDT 2009	NO_QUERY	cozz	@LLOrTish how long time no see! Yes.. Have a bit only a bit LOL. I'm fine thanks.. how's you?	
9	1467813795	Mon Apr 08 22:20:05 PDT 2009	NO_QUERY	2HoodHollywood	@Seriana, it hope they didn't have it	
10	1467813525	Mon Apr 08 22:20:09 PDT 2009	NO_QUERY	jay_wild	@Kerican get me means?	
11	1467812410	Mon Apr 08 22:20:10 PDT 2009	NO_QUERY	ericdlaaneno	spring break in plain city... it's cooling	
12	1467812570	Mon Apr 08 22:20:17 PDT 2009	NO_QUERY	perdonkaren	I just re-posted my own	
13	1467812725	Mon Apr 08 22:20:19 PDT 2009	NO_QUERY	tlac	@mattmillerless I couldn't bear to watch it. And I thought the UK was embarrassing	
14	1467812771	Mon Apr 08 22:20:19 PDT 2009	NO_QUERY	redoublerobert	@cedric16 H it counts, lol why I did it like you never talk to me anymore	
15	1467812784	Mon Apr 08 22:20:20 PDT 2009	NO_QUERY	bayofwishes	@seriana I would've been the first, but I didn't have a gun. not really though, no. myler's just a douchebag.	

Fi. 1: Snap shot of the original dataset

	A	B	C	D	E	F	G	H	I
1	text	label							
2	Ate breakfast, getting ready. going to get my new phone. yay!		1						
3	I wish it would stop raining.		0						
4	@moolintheamix it still hasn't come		0						
5	@kcs584 Good enough reason to fly with Jetstar. If they go where you want and the price is good.		1						
6	@lain7 it's ok goodnight.		1						
7	@iam41 Haha! I love how kids say whatever comes to mind with zero filter. They're unintentionally hilarious. Yours sound like a fool!		1						
8	@willywintfr Oh really... u stay busy huh. thats good		1						
9	Working on your birthday isn't so bad when you get to work with Ruby on Rails		1						
10	@miley Cyrus Aww Well Wer Behind Ya All Our Support But Mother Nature Is Kind To Us Today We Hav A 10Pack Guy Wit Us Mmmmm xxx		1						
11	@yahalom4 O so its like mine		1						
12	@danthmusic whats the next movie review going to be? You're like wospy of brum.		1						
13	Just more controversial		1						
14	All is well		1						
15	I am buying a fly ticket to Milan...		1						
16	@stickefingerz sle... lu bel di oyoyon ig. hahah... itu ex housemate gw tuh... hahaha.. impressed with IG1 setup that has successfully forwarded a call when mobile signal dropped. sadly means am constantly contactable		0						
17	Going to coffee shop to get lots of work done on my report. at least it's cloudy		0						
18	@jaylink11 And I feel fine! Thanks, buddy.		1						
19	@jordanknight I guess u missed all my tweets yesterday I was basically beggin 4 a Happy Bday shout out. My Bday is 1 month after urs.		0						
20	Passing thru picturesque Chester. Locals get the joke		1						
21	Going to get by beautiful car washed		1						
22	@em_oak love a good Batista challenge!		1						
23	@MrsNewlywed what happened? I missed 4 hours of updates		0						
24	@Stewart come you've caught the cyclone bus. Hope you had the henna on lol		1						

Fig. 2: snap shot of the customized dataset

3.2 Division of dataset

The sentiment140 dataset is divided into ten sub dataset to study the scalability of the data. The subset have the size of 5k, 10k, 20k, 50k, 75k, 100k, 200k, 500k, 1m and 1.6m(full dataset). Each subset is sampled while preserving class balance.

3.3 Minimal pre-processing

In this research, minimal pre-processing is done to maintain scalability and avoid unnecessary computational cost. In minimal pre-processing, removal of null values, urls, mention, hashtags, removal of punctuation and special symbols and optimal stop words handling, lowercasing conversion is done. Figure 3 and figure 4 shows the dataset before and after pre-processing.

	A	B	C	D	E	F	G	H	I
	text	label							
1	Ate breakfast, getting ready. going to get my new phone. yay!	1							
2	I wish it would stop raining.	0							
3	@moolinhamix it still hasn't come	0							
4	@kc3584 Good enough reason to fly with jetstar If they go where you want and the price is good.	1							
5	@laln7 it's ok goodnight.	1							
6	@lam4 Haha! I love how kids say whatever comes to mind with zero filter. They're unintentionally hilarious. Yours sound like a hoot!	1							
7	@willywilly Oh really.. u stay busy huh. thats good	1							
8	Working on your birthday isn't so bad when you get to work with Ruby on Rails	1							
9	@miley Cyrus Aww Well Wer Behind Ya All Our Support But Mother Nature Is Kind To Us Today We Hav A 10Pack Guy Wit Us Mmmmm xxx	1							
10	@yahalomia O so its like mine	1							
11	@danhmusic whats the next movie review going to be? You're like wossy of brum. Just more controversial	1							
12	All is well	1							
13	I am buying a fly ticket to Milan..	1							
14	@stickytngers tle.. lu bell di oyoyn jg.. hihhi.. itu ex housemate gw tuh.. hahaha.. impressed with #G1 setup that has successfully forwarded a call when mobile signal dropped. sadly means am constantly contactable	1							
15	Going to coffee shop to get lots of work done on my report. at least it's cloudy	0							
16	@JayLinkIII And I feel fine! Thanks, buddy.	0							
17	@Jordanknight I guess u missed all my tweets yesterday I was basically beggin 4 a Happy bday shout out. My bday is 1 month after urs.	1							
18	Passing thru picturesque Chester. Locals get the joke	1							
19	Going to get by beautiful car washed	1							
20	@em_oak love a good Batista challenge!	1							
21	@MnNewlywed what happened? I missed 4 hours of updates	0							
22	@Mewar come you've caught the cyclone bus. hope you had the hcrja on lol	1							

Fig. 3: dataset before pre-processing

	A	B	C	D	E	F	G	H	I
	text	label							
1	ate breakfast, getting ready. going to get my new phone. yay!	1							
2	I wish it would stop raining.	0							
3	it still hasn't come	0							
4	good enough reason to fly with jetstar If they go where you want and the price is good.	1							
5	it's ok goodnight.	1							
6	haha! I love how kids say whatever comes to mind with zero filter. they're unintentionally hilarious. yours sound like a hoot!	1							
7	oh really.. u stay busy huh. thats good	1							
8	working on your birthday isn't so bad when you get to work with ruby on rails	1							
9	aww well wer behind ya all our support but mother nature is kind to us today we hav a 10pack guy wit us mmmmm xxx	1							
10	o so its like mine	1							
11	whats the next movie review going to be? you're like wossy of brum. just more controversial	1							
12	all is well	1							
13	i am buying a fly ticket to milan..	1							
14	tle.. lu bell di oyoyn jg.. hihhi.. itu ex housemate gw tuh.. hahaha.. impressed with #G1 setup that has successfully forwarded a call when mobile signal dropped. sadly means am constantly contactable	0							
15	going to coffee shop to get lots of work done on my report. at least it's cloudy	0							
16	and i feel fine! thanks, buddy.	1							
17	i guess u missed all my tweets yesterday i was basically beggin 4 a happy bday shout out. my bday is 1 month after urs.	0							
18	passing thru picturesque chester. locals get the joke	1							
19	going to get by beautiful car washed	1							
20	love a good batista challenge!	1							
21	what happened? i missed 4 hours of updates	0							
22	come you've caught the cyclone bus. hope you had the hcrja on lol	1							

Fig. 4: dataset after pre-processing

Figure 5 shows the cloud of the dataset produced from the dataset described under the dataset section.



Fig. 5: cloud of the dataset

3.4 Feature extraction

For feature extraction, TF-IDF method was used. TF-IDF was used to convert text into numerical form. TF-IDF was used because it works well with SVM and also handles sparse data efficiently.

Working of TF-IDF: in TF-IDF, the term frequency measures how often any word appears in a document. It is calculated as the ratio of number of times term appear in the document and total number of terms in the document[4].

$$TF = \frac{\text{number of words } w \text{ in document } d}{\text{Total number of words in document } d}$$

While inverse document frequency is the ratio of total number of document in corpus C and the number of document containing term w[4].

$$IDF = \frac{\text{Total no. of document in corpus } C}{\text{No. of documents containing term } w}$$

3.5 Train-test split

Every subset of the dataset is divided into two parts i.e. train and test data. The ratio of train and test split is 80:20. 80% of the data is used to train the model and 20% of the data is used to test the model. Also to preserve class balance stratified sampling is used.

3.6 Model selection

In this research, Support Vector Machine (SVM) is used due to its capability of effective text classification. SVM is the machine learning algorithm designed for two-group classification problem. SVM achieves high accuracy by using support vectors to define decision boundary and by maximizing the margin between classes founded by Cortes and Vapnik[21].

Working of SVM: the SVM algorithm takes input data in labelled form such as:

$$i/p = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

where, x_i is for feature vector and y_i for label class (for eg, 0 for negative and 1 for positive).

After taking an input SVM searches for decision boundary which separate the classes using the formula:

$$w^T x + b = 0$$

where w is the weighted vector, x is for input feature and b is for biasness. SVM selects the hyperplane with largest margin for better classification problem and to reduce errors.

$$\text{Margin} = \frac{2}{\|w\|}$$

The data points nearest to the hyperplane is called support vectors and these support vectors define the hyperplane position. Hyperplane is dependent on support vectors. Now, the SVM checks the data point on the either side of the hyperplane and marks the points under class 1 and on the other side mark the point as class 2 as illustrated in figure 5.

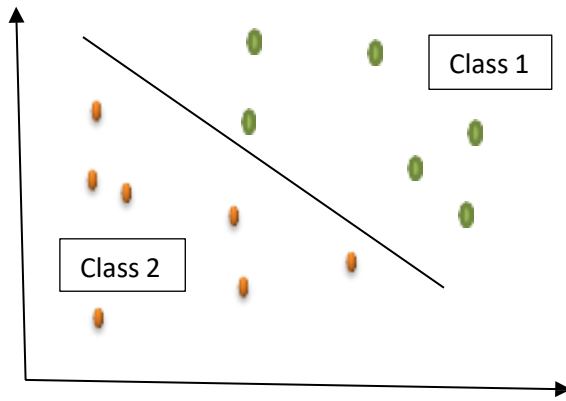


Fig. 5: data points on the hyperplane

Why SVM: SVM is used because of high-dimensional spaces, strong performance, robust generalization ability and suitable for binary classification.

In this research, various kernel of SVM is used. Kernel is mathematical function used to transform data into higher dimension space so it become easier to separate classes. The description of the kernel is given below:

Linear kernel: it is used when data is already linearly separable. It uses dot product of two input vectors. Formula for this kernel is,

$$K(x_i, x_j) = x_i^T x_j$$

Polynomial kernel: it is used to create curved decision boundaries when classes are not linear but follow polynomial pattern. It is calculated as,

$$K(x_i, x_j) = (x_i^T x_j + c)^d$$

Where c stands for constant and d stands for degree of polynomial.

RBF kernel (Radial Basis Function): RBF is the most commonly used non-linear kernel, used for complex data to map infinite-dimensional spaces. The formula for calculating the RBF kernel is,

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

Sigmoid kernel: it is used for NLP tasks and it also behaves like neural network activation function. It can be defined as,

$$K(x_i, x_j) = \tanh(\alpha x_i^T x_j + c)$$

Where α is the slope meter and c stands for constant value.

3.6 Evaluation matrices

The following matrices are used for the model evaluation

Confusion matrix: it compare actual value with predicted values to determine the correct and incorrect prediction. It is represented in table 2,

	Positive predicted	Negative predicted
Actual positive	TP	FN
Actual negative	FP	TN

Accuracy: it is used to measure how correctly any model measures the output. It is calculated as ,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision: it is the evaluation matrix that measures how many predictive positives are actually positive. It is calculated as,

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall: it measures how many positive cases are correctly identified by the model. It is calculated as,

$$\text{Recall} = \frac{TP}{TP+FN}$$

Macro F1: it is the twice of the ratio of product of precision and recall and summation of precision and recall. The formula for the macro f1 is,

$$F1 = \frac{2 \text{ precision} * \text{recall}}{\text{Precision} + \text{recall}}$$

Where, TP stands for true positive, TN stands for true negative, FP stands for false positive and FN stands for false negative.

Apart from all these matrix, training time is also calculated to determine practical stability.

4. Result and discussion

To conduct the experiment, pandas, time, sklearn, matplotlib and seaborn libraries are used. These libraries help to perform an experiment as well as help in plotting a graph.

As the size of the dataset is already very large, we make a subset and perform an experiment. The table from 3 to 12 shows the performance of the various kernel on varying data size.

Table 3: kernel result on 5k dataset

Kernel	Accuracy	Precision	Recall	F1_Score	Training Time(s)
linear	0.813	0.813209	0.813	0.812921	2.27933
rbf	0.86	0.860134	0.86	0.859963	3.006689
poly	0.861	0.861149	0.861	0.861005	2.249743
sigmoid	0.79	0.790151	0.79	0.789921	1.817924

Table 4: kernel result on 10k dataset

Kernel	Accuracy	Precision	Recall	F1_Score	Training Time(s)
linear	0.8705	0.870506	0.8705	0.870493	6.857043266
rbf	0.96	0.96014	0.96	0.959992	9.878001213
poly	0.9845	0.984541	0.9845	0.984501	17.1238203
sigmoid	0.8065	0.806519	0.8065	0.806475	7.055343628

Table 5: kernel result on 20k dataset

Kernel	Accuracy	Precision	Recall	F1_Score	Training Time(s)
linear	0.81775	0.817789	0.81775	0.817675	24.71330953
rbf	0.931	0.931167	0.931	0.930971	65.83527756
poly	0.97775	0.977781	0.97775	0.977752	76.09317064
sigmoid	0.76325	0.76321	0.76325	0.763195	24.72503138

Table 6: kernel result on 50k dataset

Kernel	Accuracy	Precision	Recall	F1_Score	Training Time(s)
linear	0.841916	0.8371	0.841916	0.824192	193.2781312
rbf	0.916308	0.92021	0.916308	0.910794	452.9104307
poly	0.975602	0.975735	0.975602	0.975295	538.2059083
sigmoid	0.80252	0.784161	0.80252	0.78169	234.6084135

Table 7: kernel result on 75k dataset

Kernel	Accuracy	Precision	Recall	F1_Score	Training Time(s)
linear	0.835411	0.828313	0.835411	0.816796	406.267915
rbf	0.913072	0.916072	0.913072	0.907448	817.5923522
poly	0.970869	0.970895	0.970869	0.97049	1306.315927
sigmoid	0.795947	0.776667	0.795947	0.778154	679.4543464

Table 8: kernel result on 100k dataset

Kernel	Accuracy	Precision	Recall	F1_Score	Training Time(s)
linear	0.829359	0.821436	0.829359	0.80946	559.2820282
rbf	0.899755	0.90282	0.899755	0.892374	1074.13109
poly	0.946803	0.947499	0.946803	0.945171	1771.538613
sigmoid	0.789861	0.769624	0.789861	0.771597	746.4273911

Table 9: kernel result on 200k dataset

Kernel	Accuracy	Precision	Recall	F1_Score	Training Time(s)
Linear	0.8164	0.8042	0.8164	0.7923	1566.73
RBF	0.8195	0.8079	0.8195	0.7970	3676.60

Table 10: kernel result on 500k dataset

Kernel	Accuracy	Precision	Recall	F1 Score	Training Time(s)
Linear	0.841529	0.800393	0.727943	0.752923	1646.157

Table 11: kernel result on 1m dataset

Kernel	Accuracy	Precision	Recall	F1 Score	Training Time (s)
Linear	0.844623	0.806817	0.73114	0.757066	1712.4168

Table 12: kernel result on 1.6m dataset

Kernel	Accuracy	Precision	Recall	F1 Score	Training Time (s)
Linear	0.844061	0.805348	0.73107	0.75665	1813.295

Discussion: the performance of the SVM is calculated using four kernels, linear, RBF, polynomial and sigmoidal and multiple sub set of the dataset is used.

For the small dataset size almost 5k to 20k, the polynomial kernel achieved the highest accuracy, then followed by the RBF kernel. If we talk about the performance of kernel in terms of training time, then sigmoid kernel is fastest but in terms of the accuracy sigmoid kernel is the worst performer. This shows that non-linear kernel are especially polynomial kernel for smaller dataset is effective in capturing dataset.

For the medium dataset size almost 50k to 100k, again the polynomial kernel achieved the highest accuracy. But the observation also suggested that although polynomial kernel has highest accuracy, with increase in data size the training cost is also become significantly high.

If we see overall model performance from the view of training cost then liner kernel is the most suitable and state of the art kernel achieving the lowest training cost for all subset dataset.

Therefore, For larger dataset from 200k to 1.6m, experiment were conducted only for linear kernel. Kernel such as RBF and polynomial depend on pairwise similarity calculation between data points therefore, they require significantly higher computational resources i.e. excessive memory consumption and very long execution time. This can be clearly understandable from earlier experiment (1771 seconds for 100k data size using polynomial kernel and 1074 seconds for 100k data size using RBF kernel). This means if we conduct experiment for larger data size, then for normal hardware conditions it become computationally impractical.

Thus we can conclude that linear kernel is the kernel which is not only the limitation for this study but it also provide computational feasibility and scalability.

5. Conclusion

The performance of the SVM is studied in this experiment. This experiment was performed on the various data size of the sentiment140 dataset ranging from 5k to 1.6m(full dataset) and various kernel of SVM were used such as linear, RBF, polynomial and sigmoid. Also for this experiment TF-IDF vectorization is used and for evaluation matrices such as accuracy, precision, recall, f1 and training time.

The result of an experiment shows that polynomial kernel achieve the highest accuracy meanwhile the computational cost increases in polynomial with increase in data size. Also the RBF kernel shows strong performance and can be used as an alternative with slight compromise in accuracy.

However, when the size of the dataset is increased, the computation cost of non-linear kernel such as polynomial and RBF increases which make impractical for large scale dataset in real world. Where as in linear kernel, stable performance and scalability is provided with lower computational cost and it maintain consistent accuracy around 84% shows strong generalization for larger dataset.

Another finding is that after some data size, increasing training sample produced slight improvement in accuracy thus showing saturation point in model learning.

Overall the conclusion is that the kernel selection of the SVM should not only depend on the evaluation matrices but computational cost should also be kept in mind. As the polynomial kernel achieve the highest accuracy, with increase in data size it becomes impractical for computation in normal circumstances. Whereas the linear kernel is observed as the most suitable for real world sentiment application as it balances between the computational cost and accuracy and produced scalability.

6. Future work

While this study only provide the importance of balancing accuracy and computation cost for real world sentiment application there are several future work can be done in this direction. To address the problem of scalability in non-linear kernel, approximation algorithm such as random Fourier features can be used, other polynomial approximation can be also explored as well as parallel and distributed SVM training can be implemented. Apart from all these work, hyperparameter tuning can also be done, techniques like random search, grid search can improve the model.

7. References

- [1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," Proceedings of EMNLP, 2002. Link: <https://aclanthology.org/W02-1011/>
- [2] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," Stanford University, 2009. Link: <http://help.sentiment140.com/for-students/>
- [3] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995. Link: <https://link.springer.com/article/10.1007/BF00994018>
- [4] <https://www.geeksforgeeks.org/machine-learning/understanding-tf-idf-term-frequency-inverse-document->

frequency/

- [5] T. Joachims, "Text categorization with support vector machines," ECML, 1998. Link: <https://link.springer.com/chapter/10.1007/BFb0026683>
- [6] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," National Taiwan University, 2010. Link: <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [7] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," Proceedings of SIGIR, 2015. Link: <https://dl.acm.org/doi/10.1145/2766462.2767830>
- [8] Y. Kim, "Convolutional neural networks for sentence classification," Proceedings of EMNLP, 2014. Link: <https://aclanthology.org/D14-1181/>
- [9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proceedings of NAACL, 2019. Link: <https://aclanthology.org/N19-1423/>
- [10] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning-based text classification: A comprehensive review," ACM Computing Surveys, vol. 54, no. 3, 2021. Link: <https://arxiv.org/abs/2004.03705>
- [11] A. Hassan, A. Mahmood, and M. B. Khan, "Comparative analysis of machine learning and deep learning techniques for Twitter sentiment analysis".
- [12] M. Alharbi and A. de Doncker, "Twitter sentiment analysis with machine learning: A survey," Information Processing & Management, 2023.
- [13] P. Harjule, M. Gurjar, and S. Chaudhary, "Text Classification on Twitter Data," 2020. Discussed in survey paper: Link: <https://www.mdpi.com/2076-3417/13/7/4550>
- [14] V. Dang, M. Moreno-García, and F. De la Prieta, "Hybrid Deep Learning Models for Sentiment Analysis," *Complexity*, vol. 2021, Article ID 9986920, 2021. Link: <https://gredos.usal.es/bitstream/handle/10366/166830/Complexity%20-%202021%20-%20Dang%20-%20Hybrid%20Deep%20Learning%20Models%20for%20Sentiment%20Analysis.pdf>
- [15] K. L. Tan, C. P. Lee, and K. M. Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," *Applied Sciences*, vol. 13, no. 7, p. 4550, 2023. Link: <https://www.mdpi.com/2076-3417/13/7/4550>
- [16] J. Gnanaprakasam and J. Sivakumar, "Understanding Public Opinion through Sentiment Analysis: A Case Study Using the Sentiment140 Dataset," 2025. Link: https://www.researchgate.net/publication/391632660_Understanding_Public_Opinion_throug_h_Sentiment_Analysis_A_Case_Study_Using_the_Sentiment140_Dataset
- [17] M. S. Asyaky et al., "Sentiment Analysis on Short Social Media Texts Using DistilBERT," 2025. Link: <https://itscience-indexing.com/jurnal/index.php/CNAPC/article/view/5836>
- [18] G. Anish Kumar et al., "Twitter Sentiment Analysis Using Machine Learning Techniques," *IJSRST*, vol. 12, no. 4, 2025. Link: <https://www.ijsrst.com/index.php/home/article/download/IJSRST251241/IJSRST251241>
- [19] A. M. Rushdi-Saleh, M. T. Martín-Valdivia, L. A. Ureña-López, and J. M. Perea-Ortega, "OCA: Opinion Corpus for Arabic," *Journal of the American Society for Information Science and Technology*, vol. 62, no.

10, pp. 2045–2054, 2011.

Link: <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21582>

[20] A. Agarwal, F. Biadsy, and K. R. Mckeown, “Contextual Phrase-Level Polarity Analysis Using Lexical Affect Scoring and Syntactic N-Grams,” in *Proceedings of the 12th Conference of the European Chapter of the ACL*, 2009.

Link: <https://aclanthology.org/E09-1017.pdf>