Text Classification using Support Vector Machine

Group Members:-
Nidheesh Pandey (IIM2015501)
Abhishek Pasi (ICM2015002)
Ayush Agnihotri (IIM2015004)
Vishal Kumar Singh (IIT2015141)
Shreyansh Gupta (IIM2015001)

Under the guidance of-

Dr K.P. Singh

Motivation

- Document Classification can be automated using machine learning. Here, we categorize a text/document in one of the multiple predefined classes.
- By assigning categories to various documents (like customer reviews, emails, articles) we can use it in various application like spam detection, sentiment analysis, News categorisation, genre classification, etc.
- Our model uses supervised Machine Learning model to classify the documents into different categories.

Objective

- Applying various Natural Language Processing techniques for feature extraction from dataset.
- Applying SVD(Singular Value Decomposition) for feature reduction.
- To implement various kernel functions of SVM(Support Vector Machine) for classification:
  - linear kernel
  - rbf (Gaussian) kernel
  - Polynomial kernel
  - Sigmoid kernel

- Analyze the accuracy and training time for each mentioned kernel function of SVM by varying the parameters of SVC (Support Vector Classifier).

Literature Survey

S.no.	Paper Title	Year	Journal/Conference	Abstract	Author
1.	Inductive learning algorithms and representations for text categorization	1998	7th International Conference on Information and knowledge management 1998-11-01	In this paper five different algorithms for text classification have been compared like find similar, decision trees, naive bayes,bayes nets and SVM.The best result were found using SVM . So it guided us to choose SVM for classification.	Susan Dumais, Mehran Sahami, John Platt, David Heckerman
2.	A Statistical Learning Model of Text Classification for Support Vector Machines	2001	24th annual international ACM SIGIR conference on Research and development in information retrieval 2001-09-01	In this paper the statistical properties of text-classification tasks is conneccted with generalizaton performance of SVM. It explained why and when SVMs perform well for text classification.	Thorsten Joachims
3.	High-performing feature selection for text classification	2002	11th International Conference on Information and knowledge management 2002-11-04	This paper mainly focuses on the importance of feature selection and feature reduction in text classification. Here various techniques have been discussed on different machine learning algorithms for feature reduction. In this paper CHI_MAX and IG methods have been combined for giving weights. For reducing redundancy μ co-occurence method is used here.	Monica Rogati, Yiming Yang
4.	Transductive Inference for Text Classification using Support Vector Machines	1999	16th International Conference on Machine Learning 27-06-1999	In this paper the concept of Transductive SVM (TSVM) is introduced. It explains us the limitations of normal SVM in some cases where TSVM are better than SVM. The experiments here tells us about significant improvement of TSVM over inductive methods. TSVMs work very well in cases where there is smaller training dataset than the test set.	Thorsten Joachims
5.	An Optimal SVM-Based Text Classification Algorithm	2006	International Conference on Machine Learning and Cybernetics 13/09/2006	This paper describes new algorithms for feature selection which highly optimize the efficiency of classification. The new algorithm applies entropy weighing scheme for feature selection in a newer way. Also optimal parameter settings have been used to get better results.	Zi-Qiang Wang, Xia Sun, De-Xian Zhang, Xin Li

Methodology

1. Getting the dataset
The Reuters 21578 dataset is loaded and then sent for parsing to get it into a usable format.

2. Parsing
In this step we convert the dataset into a usable format which can be classified for our experiment. This conversion process is known as Parsing. Here we have made SGML parser class which overrides HTMLParser.

3. Stop Words Removal
Stop words are set of commonly used words in any language. It is important for us to filter these words in order to focus on more important words. For example : a, an, is, it, that, the, with, from, has, were, was, its, of, be, will, with, and, etc.

4. Stemming
It is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form.
[inflect (dictionary meaning) :- change the form of (a word) to express a particular grammatical function or attribute]
Example :- (i) banks and banking become bank
(ii)investing and invested become invest

5. Vectorisation

- It is used to convert raw text into a numerical data representation which can be used for classification.
- Firstly we create list of tokens and normalise them using Bag of Words approach.
- So entire dataset can be represented as a large matrix, each row representing one of the documents and each column representing token occurrence in that document. This is the process of vectorisation.

6. tf-idf(Term-Frequency Inverse Document-Frequency)

- It gives us better weighting scheme of the tokens used for classifying the documents.
- We get a high TF-IDF value if its frequency is high in that document and low frequency in collection of all documents

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

7. Singular Value Decomposition (SVD)

- We have used SVD here to reduce the number of features (Feature Reduction).
- SVD achieves this task by creating new features which are linear combination of the existing ones.
- Singular value decomposition reduces a matrix of *R* rank to a matrix of K rank.
- Let A be a rank *R* matrix with *m* rows and *n* columns. SVD tells us

$$A = U\Sigma V^T$$

where:

U = square orthonormal *m* x *r* matrix

Σ = diagonal *r* x *r* matrix

V^T is a square orthonormal *r* x *n* matrix.

Support Vector Machine(SVM)

- Supervised machine learning model.
- Can be used for both classification (SVC) and regression (SVR).

SVM as a Classifier (SVC):

- Discriminative and non-probabilistic classifier.
- It classifies the different groups by finding the decision boundary that best separates the groups based on their known categories.
- This best decision boundary is the one that maximizes the margins between any two groups. (**Maximum Margin Classifier)**.

Fig.1 Maximum Margin Classifier and Support Vectors in SVM

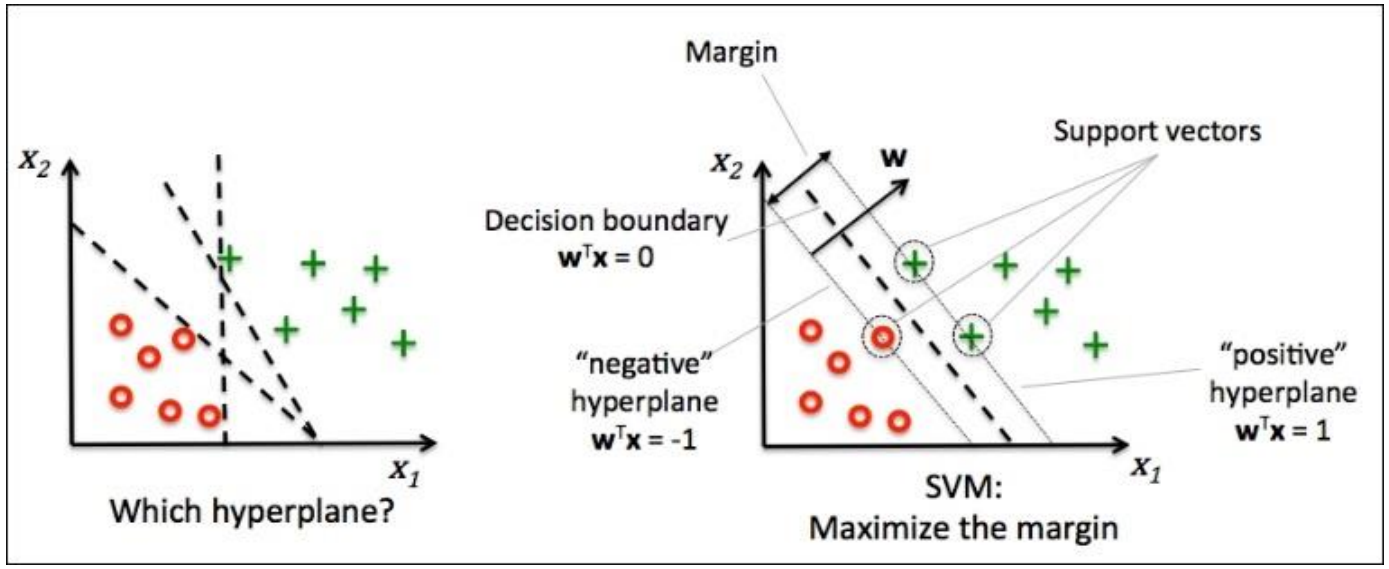
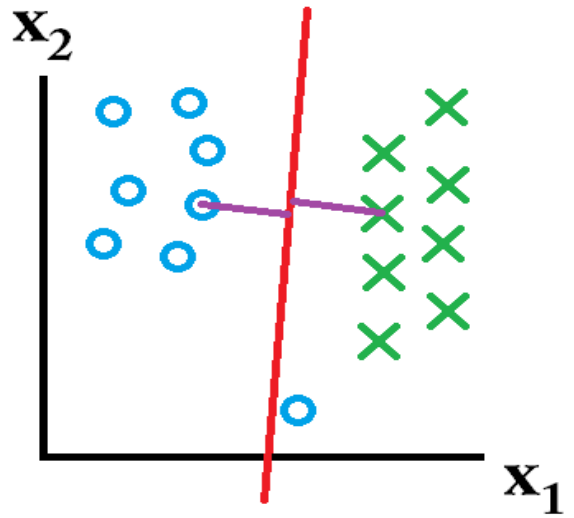


Fig.2 Outlier ignored in general SVM.



- Linearly Separable Dataset → Simple
- Non-Linearly Separable Dataset → Not so simple (**kernel trick** used)

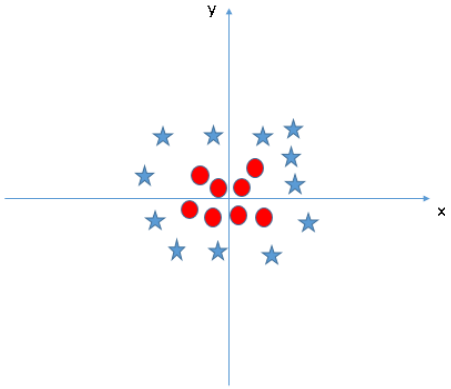


Fig 3. Non-linearly separable data

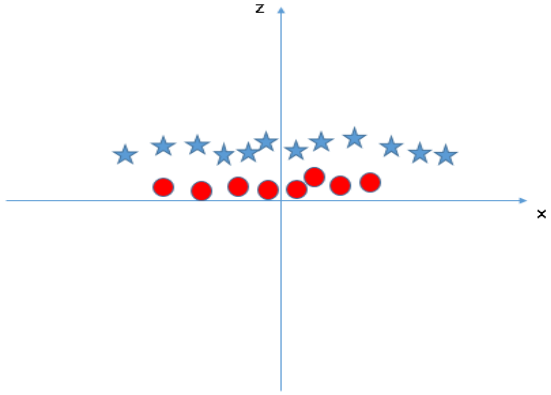


Fig 4. Mapping into higher dimension (adding z-axis component)

$$Z = x^2 + y^2$$

- SVM does not need the actual vectors to work on it, it can do it only with the **dot products** between them. This dot product is called a **kernel function**.

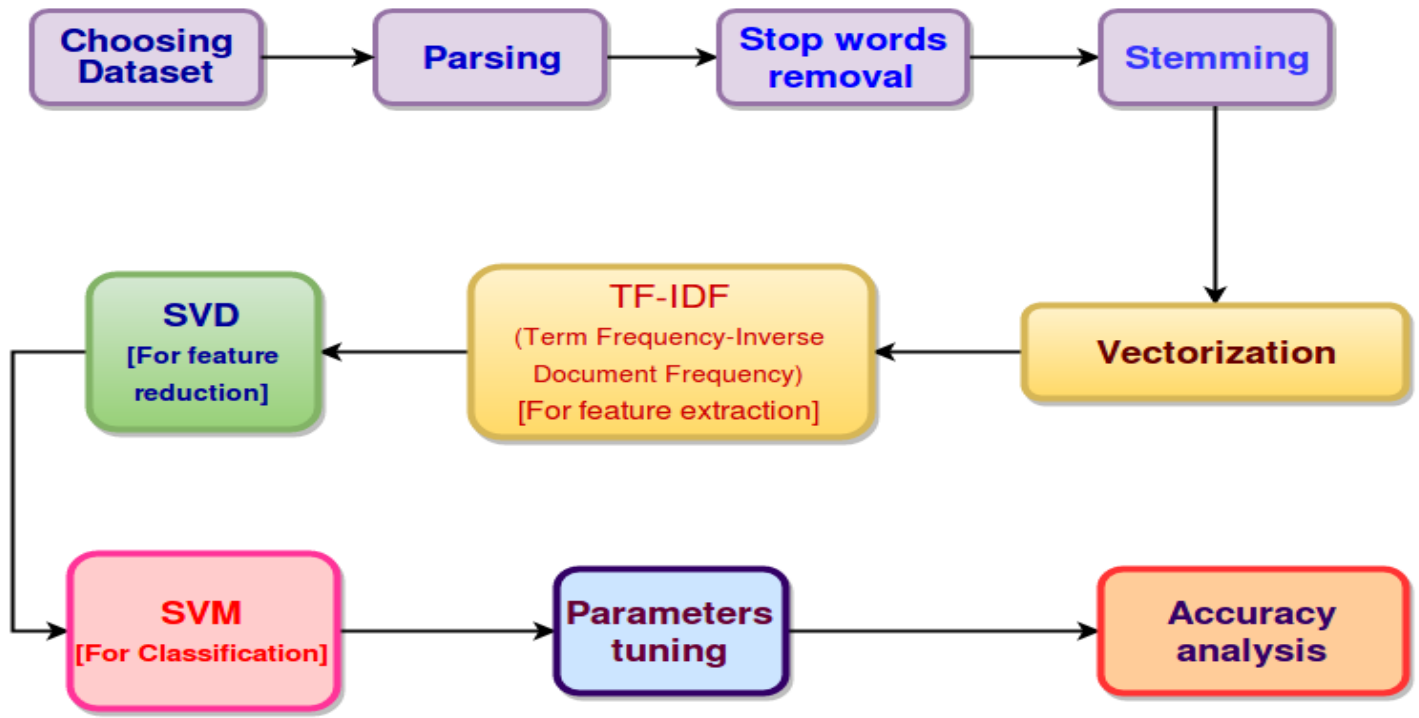
Kernel Functions

$$K(\mathbf{X_i}, \mathbf{X_j}) = \begin{cases} \mathbf{X_i} \cdot \mathbf{X_j} & \text{Linear} \\ (\gamma \mathbf{X_i} \cdot \mathbf{X_j} + C)^d & \text{Polynomial} \\ \exp(-\gamma \mid \mathbf{X_i} - \mathbf{X_j} \mid^2) & \text{RBF} \\ \tanh(\gamma \mathbf{X_i} \cdot \mathbf{X_j} + C) & \text{Sigmoid} \end{cases}$$

where $K(\mathbf{X_i}, \mathbf{X_j}) = \phi(\mathbf{X_i}) \bullet \phi(\mathbf{X_j})$

that is, the kernel function, represents a dot product of input data points mapped into the higher dimensional feature space by transformation ϕ

Flow diagram



Experiments
&
Observations

Change in **Accuracy** and **Training time** with change in SVC parameters **C** and **gamma**

S.No.	Penalty Parameter (C)	Kernel Coefficient (gamma)	Training Time (in seconds)	Accuracy (% age)
1.1	1	0.0	14.717918573825267	53.51637764932563
1.2	1	5.0	20.259658244591563	88.294797687861271
1.3	1	10.0	49.77984757556541	86.801541425818884
2.1	10	0.0	10.91928373088761	80.973025048169556
2.2	10	5.0	19.581059889092217	88.969171483622356
2.3	10	10.0	50.659539527844345	87.235067437379576
3.1	100	0.0	7.2436635577647195	86.705202312138729
3.2	100	5.0	19.243339823695365	88.294797687861271
3.3	100	10.0	50.66085798509113	87.186897880539505
4.1	1000	0.0	6.068772320865371	88.150289017341044
4.2	1000	5.0	19.179974332207593	88.294797687861271
4.3	1000	10.0	50.54288278987775	87.186897880539505
5.1	10000	0.0	6.277793767285239	87.668593448940269
5.2	10000	5.0	19.17688324917816	88.294797687861271
5.3	10000	10.0	50.95072527978192	87.186897880539505
6.1	100000	0.0	7.173925166539156	87.572254335260113
6.2	100000	5.0	19.18787737791652	88.294797687861271
6.3	100000	10.0	50.592198137224784	87.186897880539505
7.1	1000000	0.0	9.732850540490489	87.186897880539505
7.2	1000000	5.0	19.2118927152992	88.294797687861271
7.3	1000000	10.0	50.54131843158757	87.186897880539505

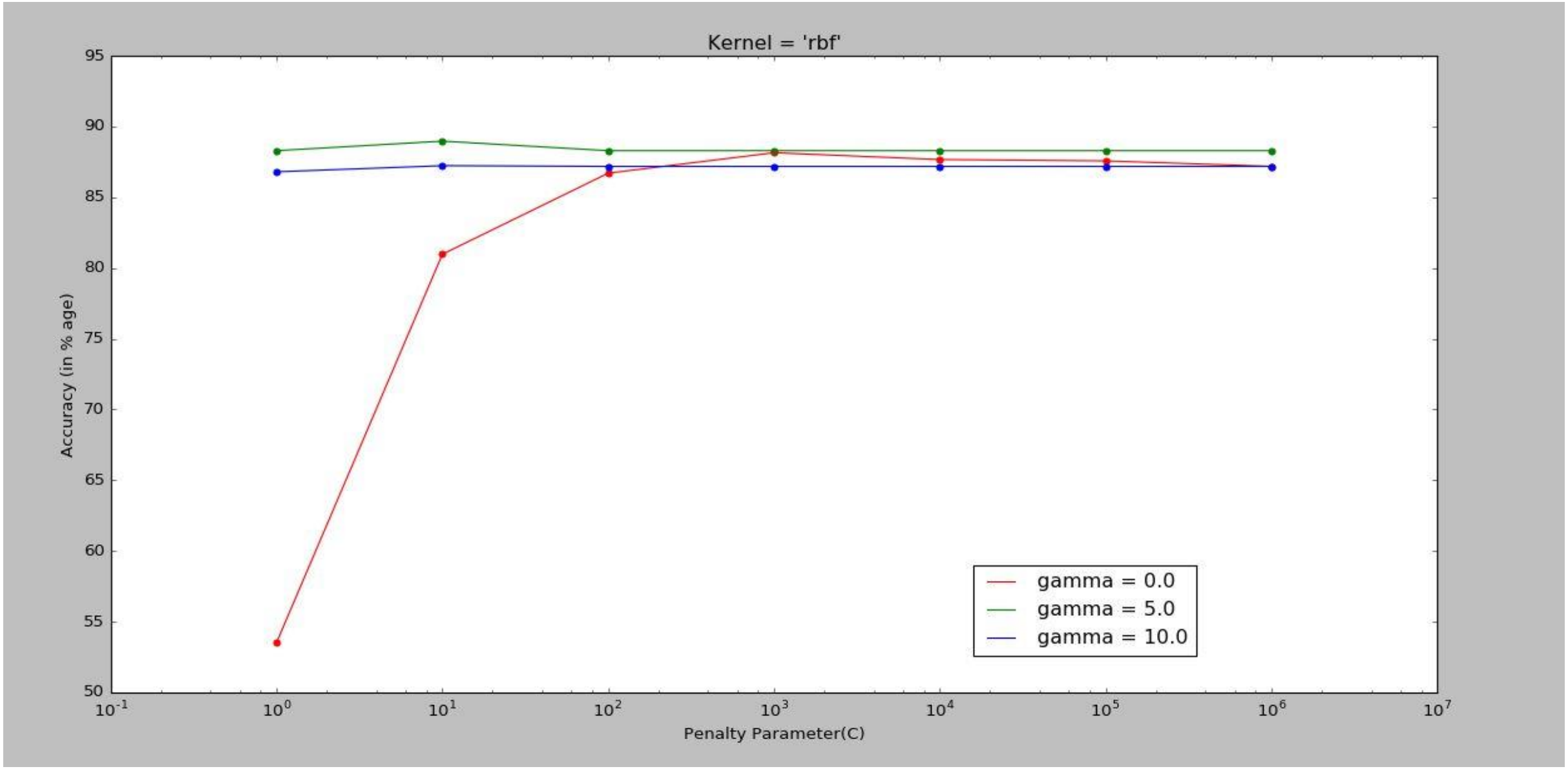
Kernel Function :
Radial Basis Function (rbf)

Using different values for the **Penalty Parameter (C)** and **Kernel coefficient (gamma)**, the best results we found from our experiments:

- C = 10.0
- gamma = 5.0
- Training Time taken = 19.581059889092217 seconds
- Accuracy on test dataset = 88.969171483622356 %

Accuracy vs. **Penalty Parameter (C)** (for **rbf kernel**)

Standard deviation = 12.660401(for gamma = 0.0), 0.254889(for gamma = 5.0), 0.149765(for gamma = 10.0)



Change in Accuracy and Training time with change in SVC parameter C

S.No.	Penalty Parameter (C)	Training Time (in seconds)	Accuracy (% age)
1	1	5.744766691228165	85.597302504816952
2	10	4.48024516547855	88.439306358381498
3	100	4.467598581728339	87.909441233140651
4	1000	5.394875593316442	87.475915221579958
5	10000	10.462791323242755	87.186897880539505
6	100000	101.26224597887654	87.090558766859349

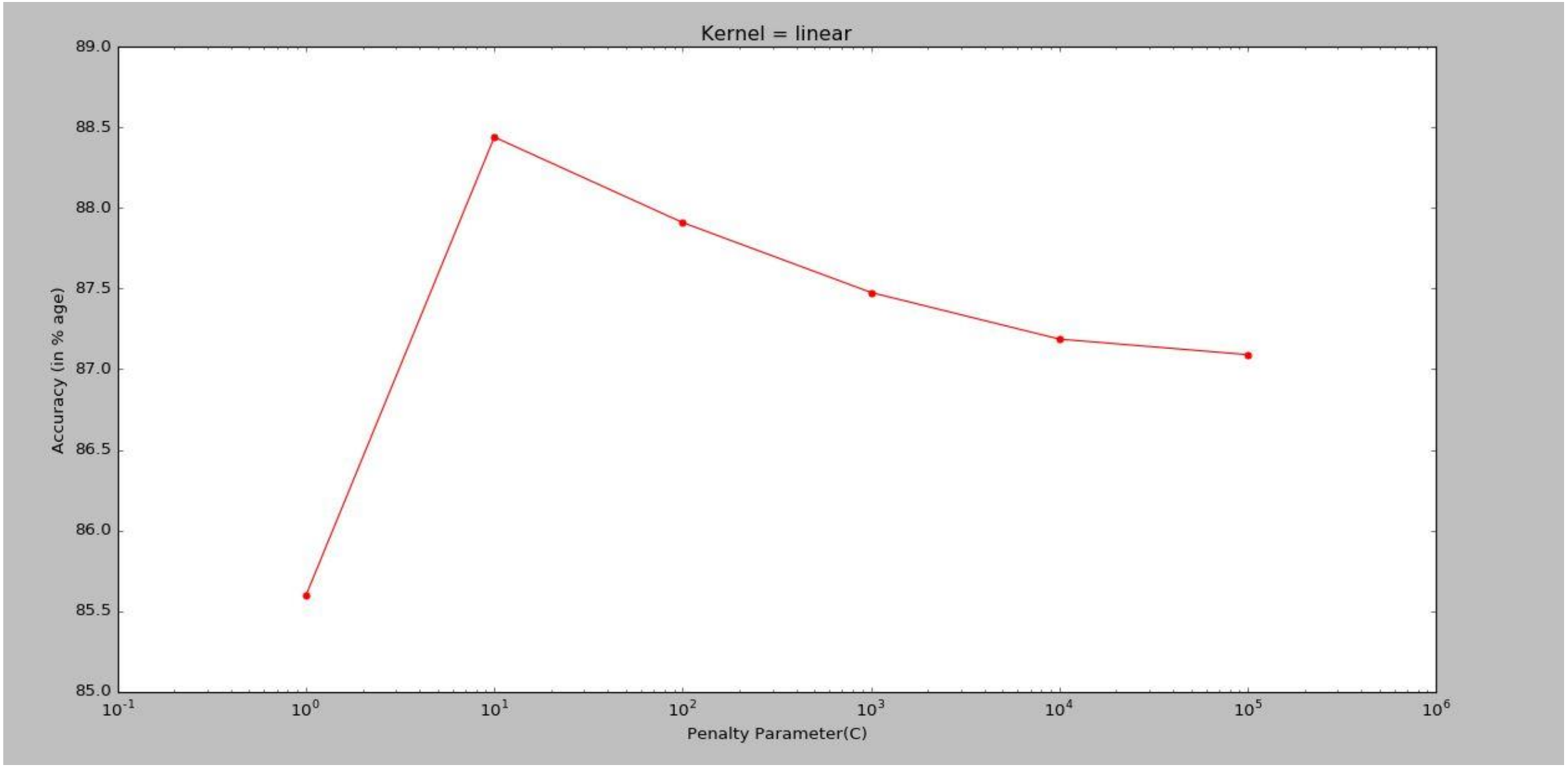
Kernel Function :
Linear

Using different values for the **Penalty Parameter (C)** and **Kernel coefficient (gamma)**, the best results we found from our experiments:

- C = 10.0
- Training Time taken = 4.48024516547855 seconds
- Accuracy on test dataset = 88.439306358381498 %

Accuracy vs. **Penalty Parameter (C)** (for **linear kernel** with **gamma** = 0.0, 5.0 and 10.0)

Standard deviation = 0.964835



Average Accuracy and **Average Training time** with different random splits of training and test

S.No.	Penalty Parameter (C)	Kernel Coefficient (gamma)	Average Training Time (in seconds)	Average Accuracy (% age)	Minimum Accuracy (% age)	Maximum Accuracy (% age)
1.	10	5.0	19.59759051189758	87.591522157996149	86.849710982658956	88.198458574181116
2.	1	5.0	19.828029212188813	88.236994219653186	88.053949903660889	88.342967244701354
3.	100	5.0	19.10364374027886	86.531791907514455	85.9344894026975	86.705202312138729
4.	1000	5.0	18.75711862138014	87.976878612716758	87.186897880539505	88.198458574181116
5.	10000	5.0	20.28234682823986	88.737957610789986	87.861271676300579	89.25818882466281

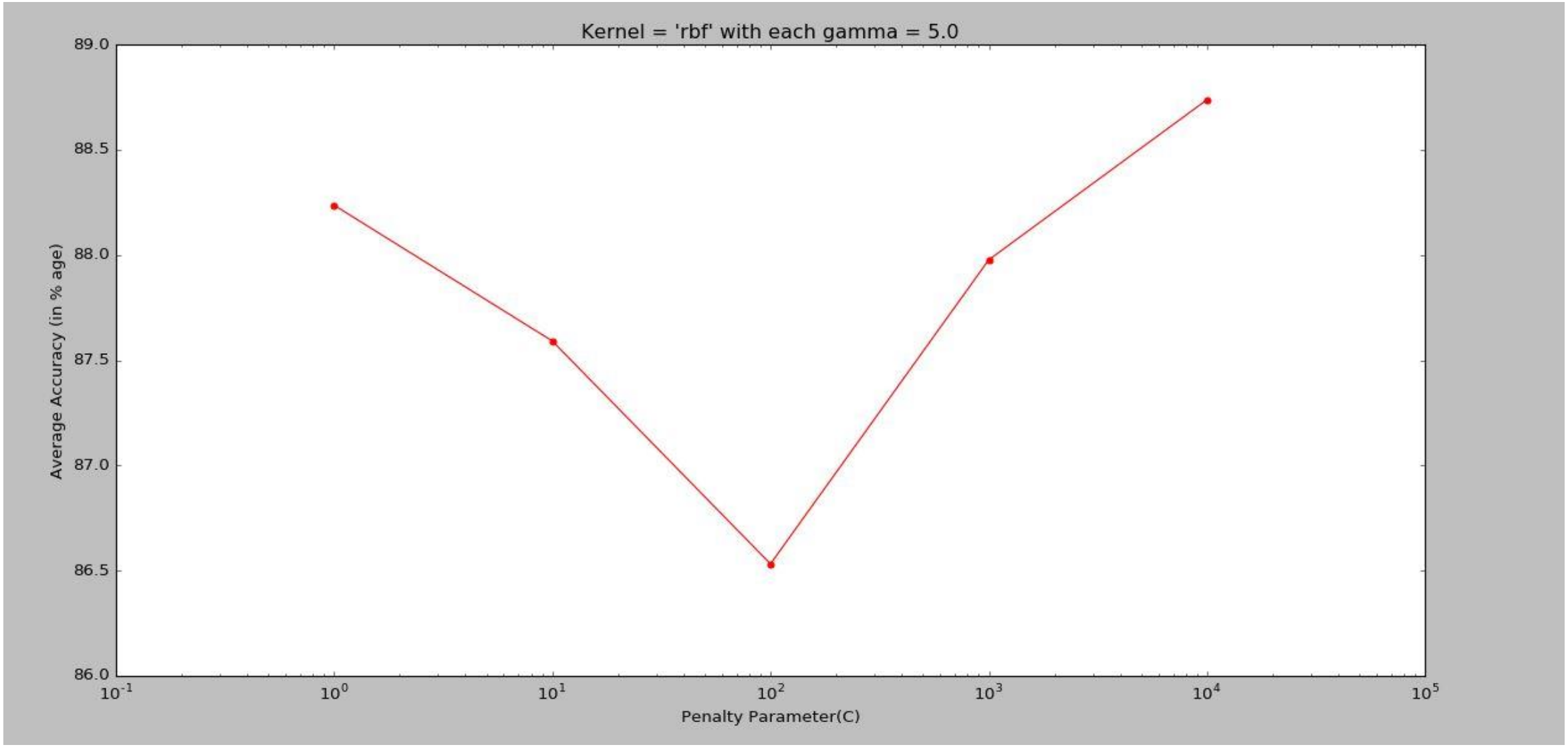
Kernel Function : Radial Basis Function (rbf)

Using different random splits of training and test datasets for fixed value of svc parameters the best results we found are:

- C = 10000.0
- gamma = 5.0
- Average Training Time taken = 20.28234682823986 seconds
- Average Accuracy on test dataset = 88.737957610789986 %

Accuracy vs. **Penalty Parameter (C)** (for **rbf kernel** with different splits of training and test data)

Standard deviation = 0.829563



Average Accuracy and **Average Training time** with different random splits of training and test

S.No.	Penalty Parameter (C)	Average Training Time (in seconds)	Average Accuracy (% age)	Minimum Accuracy (% age)	Maximum Accuracy (% age)
1.	10	26.454119013539277	86.310211946050097	85.308285163776498	87.668593448940269
2.	100	19.094978187508424	87.360308285163779	86.657032755298646	88.872832369942201
3.	1000	26.599559003206924	85.462427745664737	84.489402697495186	86.897880539499039
4.	10000	19.12901739225981	87.283236994219648	86.416184971098264	88.535645472061653
5.	100000	22.862347076279367	88.169556840077068	87.475915221579958	89.0655105973025

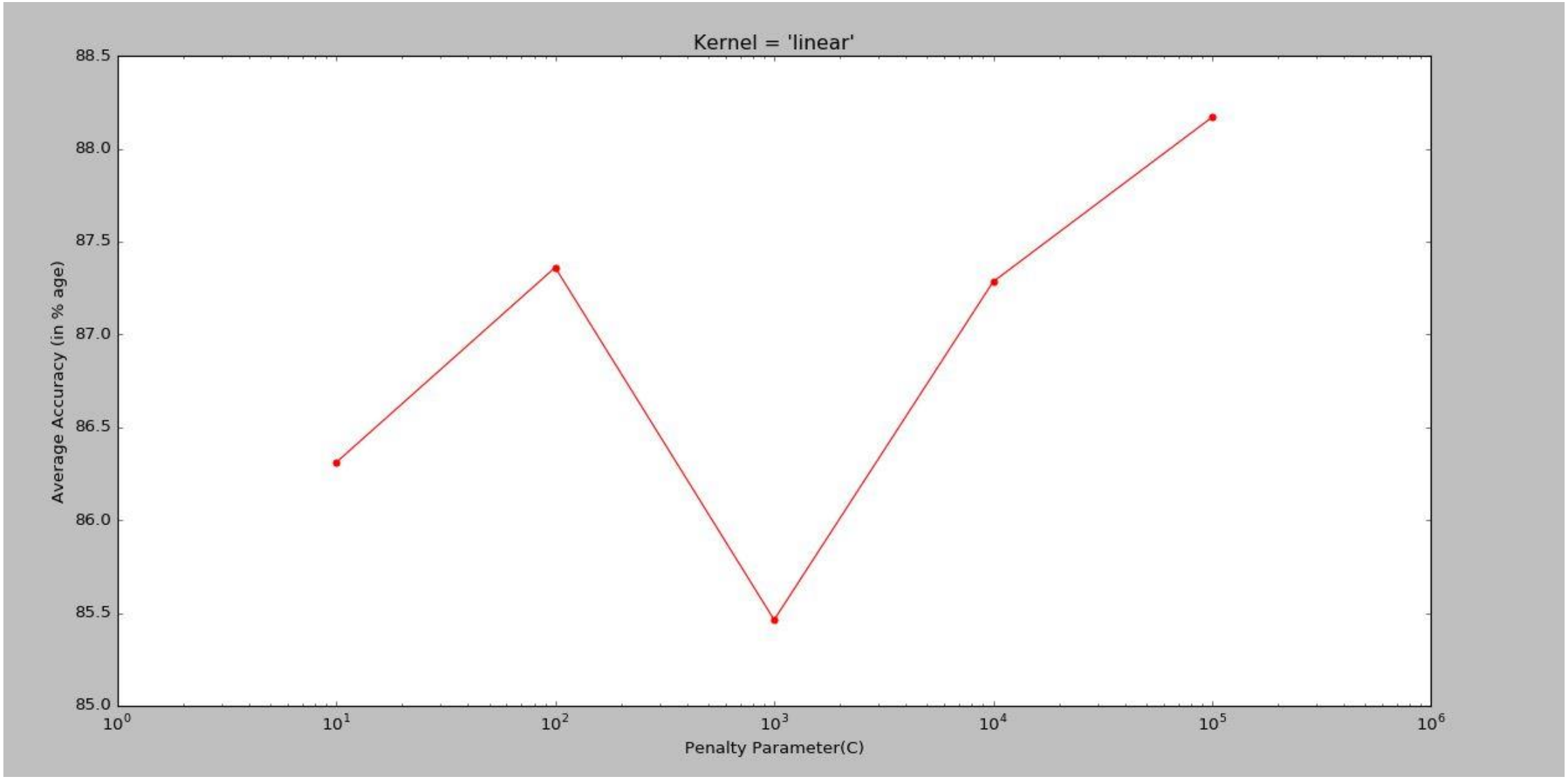
Kernel Function : Linear

Using different random splits of training and test datasets for fixed value of svc parameters the best results we found are:

- C = 100000.0
- Average Training Time taken = 22.862347076279367 seconds
- Average Accuracy on test dataset = 88.169556840077068 %

Accuracy vs. **Penalty Parameter (C)** (for **linear kernel** with different splits of training and test data)

Standard deviation = 1.046843



Future Work

- To find the most optimal values of C and Gamma , we did various experiments and obtained some interesting observations and results.
- However, for end-semester evaluation we will be using various techniques for parameter search like Exhaustive Grid Search, Randomised Parameter Optimisation , Tips given in scikit-learn documentation and Alternatives to brute-force parameter search.
- We will also apply various other kernels in our SVM(like) and analyse and find the optimum one.

Softwares Used : Spyder 3 (Scientific PYthon Development EnviRonment)

Language Used : Python 3.5

Libraries Used :

• **pandas** (Python Data Analysis Library) for inputting the Dataset (.tsv file).

• The module pyplot of matplotlib library is used for graph plotting re library for the usage of Regular Expressions for text cleaning

• Stopwords list from the ‘corpus’ package of nltk (Natural Language Processing Toolkit) data package

• **PorterStemme**r algorithm from the package nltk.stem.porter for stemming of words.

• **CountVectorizer** class from sklearn.feature_extraction.text submodule for building the matrix (sparse) of word counts from text documents.

• **TfidfTransformer** class from sklearn.feature_extraction.text submodule to convert the count. matrix (sparse) to a matrix of TF_IDF features.

• **TruncatedSVD** class from sklearn.decomposition module fordimensionality reduction of the feature space.

References

- [1] Susan Dumais, Mehran Sahami, John Platt, David Heckerman "Inductive learning algorithms and representations for text categorization" published in CIKM '98 Proceedings of the seventh international conference on Information and knowledge management(Pages 148-155) in Bethesda, Maryland, USA — November 02 - 07, 1998

- [2] Thorsten Joachims "A statistical learning model of text classification for support vector machines" published in SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval Pages 128-136 New Orleans, Louisiana, USA

- [3] Monica Rogati, Yiming Yang "High-performing feature selection for text classification" published in CIKM '02 Proceedings of the eleventh international conference on Information and knowledge management Pages 659-661 McLean, Virginia, USA— November 04-09,2002

- [4] Thorsten Joachims "Transductive Inference for Text Classification using Support Vector Machines" published in ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning Pages 200-209 June 27-30,1999

- [5] Zi-Qiang Wang, Xia Sun, De-Xian Zhang, Xin Li " An Optimal SVM-Based Text Classification Algorithm“ published in Proceedings of 2006 International Conference on Machine Learning and Cybernetics Dalian, China

Thank You