**Group 25** Final Project

**Names:** Ayush Pandey, Badr Arraichi, Dev Kalavadiya and Zaeem Shahzad

**Mentor:** Ziyang Zeng

Natural Language Processing

Professor Adam Meyers

**Sentiment Analysis based on Hotel Reviews**

**Introduction:**

Sentimental analysis is textual mining used to extract valuable subjective data to comprehend societal attitudes toward a brand, which aids in quality assurance or company improvement. From person to person, opinions on the product's utility vary. Customer feedback is crucial for maintaining the reputation of a business and a product. It is used to identify the subject's neutral, positive, and negative emotions.

Like no other sector, hospitality depends on consumer satisfaction and favorable ratings. Reviews are crucial for hoteliers since they have the power to make or break the establishment. For this reason, it's crucial for hotel owners and managers to ensure that they receive as many positive online hotel reviews as they can. In a large-scale business, it is important to be able to efficiently parse through thousands of reviews and determine the overall sentiment around the business.

**Problem Statement:**

To analyze user sentiment, a variety of technologies are used. In this project, we will use deep learning techniques to analyze sentiments on hotel reviews. Support Vector Machines (SVM), Simple Neural Network (SNN), and Long Short-Term Memory (LSTM) approaches are all used for sentiment analysis, and their effectiveness will be explored in this project.

**Evaluation Plan:**

To evaluate the efficiency of the different deep learning techniques we want to explore, we will employ two data sets we found on Kaggle.com ([Set A](https://www.kaggle.com/datasets/harmanpreet93/hotelreviews), [Set B](https://www.kaggle.com/code/algord/trip-advisor-hotel-reviews-gru/data)). Both Set A and Set B contain Sentiment Analysis measures for reviews left on hotel booking websites. The format is in the form of two columns for every row in the set; one for the Sentiment Analysis, and the other for the review text.

There is one difference between the sets. Set A has Sentiment Analysis recorded as either “Happy” or “Not Happy” while Set B has Star Ratings. Thus, in pre-processing the data, we will standardize the first column in both data sets to only include a boolean value (0 - Not Happy, 1 - Happy). Set A can be converted to this format easily, however for Set B we will select Ratings of 4 and lower as Unhappy, and 6 and higher as Happy.

We will merge Set A and Set B into one large data set. This larger data set will be split in the following proportions to create the Training, Development, and Test Corpus:

* 80% allocated to Training/Development
* 20% allocated to Test

In the above splits, Training and Development are considered one corpus because we plan to implement a cross-validation approach in the training process. For the 20% allocated to Test, we will create two files: one file will retain the Sentiment Analysis column while the other will exclude it. This is because we want to compare our system’s output of Sentiment Analysis to the analysis in the original data set to evaluate its performance.

To explain how we will define and score our system output properly, we will first explain the processing of the data. In pre-processing, after tokenization, we want to remove all stopwords from the reviews, lemmatize/stem each word in a review, and truncate all punctuation, numbers, and/or special characters. After this, we will maintain TF-IDF scores for each word in all the reviews based on the Training/Development cross-validated corpus. After this, we will apply all of the approaches discussed in the Introduction section to attain the Sentiment Analysis on the Test corpus via all approaches - 3 output files, one for each approach.

Finally, we will score the output of the system on the Test corpus by evaluating five metrics in comparison with the Test corpus with the Sentiment Analysis values retained. The metrics are Precision, Recall, F1-measure, Accuracy, and Mean Squared Error. After this scoring, we can compare these metrics obtained from the different approaches being employed in the system sequentially to determine which approach performs best.

**Approaches:**

**Support Vector Machines (SVM)** is a classification algorithm used to find a hyperplane having a maximum distance between the data points of two classes. The data points falling on each side of the hyperplane help us differentiate between two classes, which is how we aim to classify positive and negative sentiments.

A **Simple Neural Network (SNN)** is a computational learning algorithm used to find relationships between the input and output data. Data is passed through multiple layers: the input layer, the hidden layer, and the output layer to reach a specific result. It works like the biological neural network, where each neuron carries information and the network learns from experience.

**Long Short-Term Memory (LSTM)** resolves the shortcomings of the traditional neural networks. It is a recurrent neural network that is capable of learning from long term experiences i.e. they can remember information for a long period of time. We believe that this persistence of information will help us generate sentiments by understanding the context of each sentence in a review.

**Collaboration plan:**

To efficiently tackle this project, we are planning to schedule weekly work sessions to get together, discuss the next steps, and resolve any issues that might arise. We will be dividing the tasks equally among us as per the following:

**Preprocessing** (*Zaeem, Ayush*)

1. Text Cleaning
2. Text Exploration
3. Text Tokenization

**Feature Engineering**

1. TF-IDF Vectorization *(Badr, Dev)*

**Building the Models**

1. SVM *(Badr)*
2. SNN (*Dev)*
3. LSTM (*Zaeem*)

**Cross-validation** (*Ayush*)

**Model Evaluation/Scoring** (*Badr, Dev, Zaeem, Ayush*)

**Gathering the conclusions and writing the final paper** *(Badr, Dev, Zaeem, Ayush)*

**References**

[1] Jenq-Haur Wang, Ting-Wei Liu, Xiong Luo, and Long Wang. 2018. [An LSTM Approach to Short Text Sentiment Classification with Word Embeddings](https://aclanthology.org/O18-1021). In Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018), pages 214–223, Hsinchu, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

This paper examines how word embedding and long short-term memory (LSTM) affect the classification of sentiment in social media. A word embedding model is used to first turn the words in posts into vectors. The LSTM is then used to learn the long-distance contextual dependency among words from the word order in phrases. Since most of the hotel reviews are from popular travel blog sites and comments on social media, we can use this RNN extension to learn sequential data and their long term connections to precisely carry out better sentimental analysis compared to other models.

[2] Tony Mullen and Nigel Collier. 2004. [Sentiment Analysis using Support Vector Machines with Diverse Information Sources](https://aclanthology.org/W04-3253). In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pages 412–418, Barcelona, Spain. Association for Computational Linguistics.

This paper will be of major help when it comes to using Support Vector Machines (SVM) as an algorithm for our sentiment analysis. It introduces a rigorous approach to bringing together various sources of pertinent information including but not limited to favorability measures for adjectives, phrases and knowledge of the topic covered. It showcases different experiments that were held to get an idea of the efficiency of SVM when it comes to sentiment analysis.

[3] Maria Karanasou, Christos Doulkeridis, and Maria Halkidi. 2015. [DsUniPi: An SVM-based Approach for Sentiment Analysis of Figurative Language on Twitter](https://aclanthology.org/S15-2120). In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 709–713, Denver, Colorado. Association for Computational Linguistics.

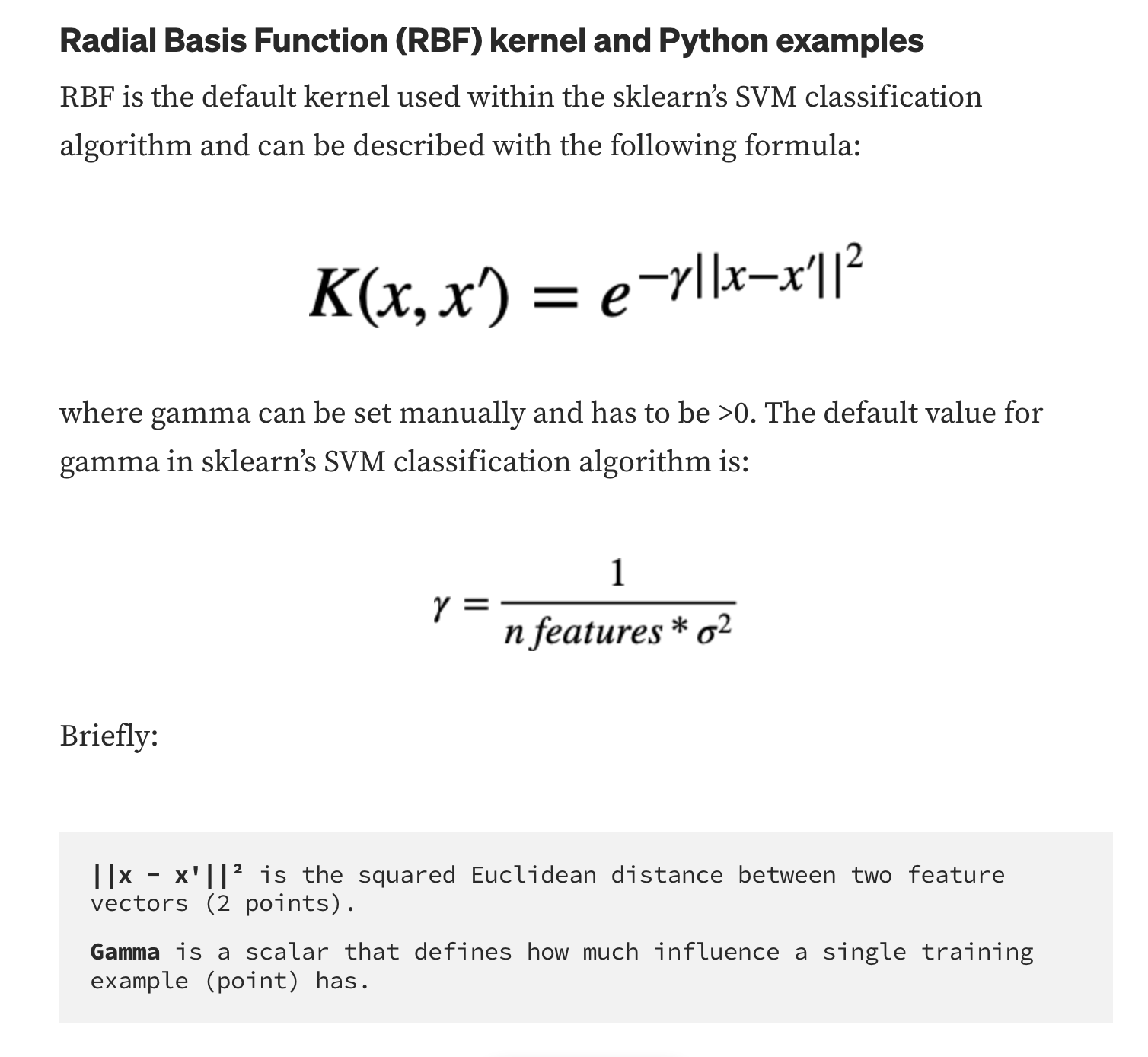
This paper will also be a reference in understanding Support Vector Machines (SVM) as it addresses the problem of sentiment analysis on figurative language. Identifying underlying sentiment in such texts is difficult due to their limited size and features such as the use of abbreviations and slang. It uses Twitter text that contains implications to different meanings using irony, metaphors, #hashtags, emoticons and the use of intensifiers such as capitalization and punctuation, and creates a classifier that assigns labels to each tweet.

[4] Georgios Paltoglou and Mike Thelwall. 2010. [A Study of Information Retrieval Weighting Schemes for Sentiment Analysis](https://aclanthology.org/P10-1141). In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1386–1395, Uppsala, Sweden. Association for Computational Linguistics.

This paper is specifically of relevance as it challenges one of the topics discussed in previous papers which is the use of SVM. It explores a different approach (as many of the sentiment analysis experiments solely rely on SVM classifiers with binary unigram weights) and looks into whether more sophisticated feature weighting schemes from Information Retrieval can enhance the sentiment classification accuracy. These techniques are tested and analyzed throughout the entire paper to come to a conclusion of whether an increase in the accuracy happens.

[5] Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. [Sentiment Analysis Is Not Solved! Assessing and Probing Sentiment Classification](https://aclanthology.org/W19-4802). In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 12–23, Florence, Italy. Association for Computational Linguistics.

This paper gives an general overview of the neural methods for sentiment analysis and the ones that produced quantitative gains over other methods. The study describes the difficulties that sentiment classifiers for English still face when presented with a difficult dataset. Most importantly, shows how the dataset can be used to examine how well a particular sentiment classifier performs in relation to language events. Allowing us to research into the possible models we can explore for an accurate sentiment analysis.



https://towardsdatascience.com/svm-classifier-and-rbf-kernel-how-to-make-better-models-in-python-73bb4914af5b