**Analysis of Customer Reviews using Big Data**

**A Project Report**

**Submitted by**

**Pavansai Pottimuthi (811235129,** [**ppotimu@kent.edu**](mailto:ppotimu@kent.edu)**)**

**Shirish Kumar Tallapelly (811229216,** [**stallape@kent.edu**](mailto:stallape@kent.edu)**)**

**Shiva Kumar Peddapuram (811235874,** [**speddapu@kent.edu**](mailto:speddapu@kent.edu)**)**

Under The Guidance of

**Ruoming Jin**

Professor

Department of Computer Science

**Kent State University**

**1. Introduction**

**Motivation examples of this project:**

* Nowadays, we usually buy products online. To buy an item we take a lot of parameters into consideration like shirt color, ratings, reviews for that specific product, comments, price, and lot more. In this growing era, all brands are trying to get the digital presence in different forms like maintaining their own social media pages (selling their products), e-commerce sites, and official websites. All the consumers or the buyer of that specific product gives their feedback in different firms like comments, ratings, and reviews.
* So, what we think of it is the most importance for all kinds of companies is to understand their customer impression in digital platforms as early as possible. If they learn positive, it may help them to publish their products in that model. If they get any negative response, then they can work upon those traits and then improve their products.
* Building automated bots or software can undergo all the reviews and then analyze them for that companies, this process is known as sentimental analysis. For this type of software, we use different machine learning algorithms to develop the model so that it can work on its own.
* Sentiment analysis plays a key role in the field of Natural Language Processing, and it is a technique that extract emotions from the raw texts.

**2. Project Description**

* The first challenge is integrating all the customer’s data from various sources, thus collecting datasets from online sources helps us to analyze data.
* To get a quick overview of the data set we use the dataframe.info () function. Python is an excellent language for data analysis, due to the solid ecosystem of data-centric Python tools.
* The second issue is cleaning all the null values and converting the data into upper or lower case, removing emojis and all unwanted data.
* Visualization of data and by using sentimental intensity analyzer thus converting data from text to vectors into polarity scores and plotting the data word cloud, training the data with different parameters.
* Final step is to find the performance and evaluations of the project.
* The main issue is regarding the privacy and protection of the user data.

**3. Background**

**Related papers:**

Nandal in this paper classified amazon product reviews for sentiment analysis using SVM (Support Vector Machine) Tool. The study examined how words can shift in meaning depending on the context in which they are used, and how this impacts the overall evaluation of a product and its specific features.

Humera Shaziya classified movie reviews for sentiment analysis using WEKA Tool. They enhanced the earlier work done in sentiment categorization which analyzes opinions which express either positive or negative sentiment.

Ahmad Kamal designed an opinion mining framework that facilitates objectivity or subjectivity analysis, feature extraction and review summarizing. He used a supervised machine learning approach for subjectivity and objectivity classification of reviews. The various techniques used by him were Naive Bayes, Decision Tree, Multi-layer Perception and Bagging. He also improved mining performance by preventing irrelevant extraction and noise.

To estimate the semantic orientation polarity and its intensity for phrases, which serves as a foundation for sentiment-based computing, Orestes Appel used natural language processing (NLP) fundamental techniques, a sentiment lexicon improved with help from SentiWordNet, and fuzzy sets. Three different datasets are subjected to the suggested hybrid method, and the outcomes are contrasted with those attained by applying the Maximum Entropy and Naive Bayes techniques.

**Required hardware:**

* **Required Technical Knowledge: Python** Programming Language (Object-Oriented programming), Sentiment analysis, Natural Language Processing (for training the model).
* Integrated Development Environment PyCharm and Google Colob.
* Different Python libraries are used to analyze the data (clean unnecessary data).
* **Dataset**: Here dataset used in this project is Realtime customer data collected from Kaggle source.
* **Required hardware:** System with 64-bit ROM and 8GB RAM.
* **Input:** customer data with reviews and review score.
* **Output:** Use qualitative and quantitative prediction models to forecast the Customer Review Analysis.

**4. Problem Definition**

The challenge is to create a platform that compiles all of the important indicators for a company, such as the most current reviews, overall rating, sentimental distribution, trending keywords, etc. Many evaluations, ideas, and complaints are left on a company's website by customers. Reading and understanding all of this takes a lot of physical labor, money, and time.

There are problems while taking the dataset, all the reviews will not be in the same language, so we must change the data into machine readable. There are some companies that ask media managers to give false reviews for their benefits thus we cannot analyze data properly. Thus, in cleaning and preparation step we change the data. By analyzing the data, companies can know where they are lagging and how to improve their products and how they can change their products and to analyze whom they must target.

In our model we have analyzed the data by using traditional classifier random forest versus Transformers by using RoBERTa. we have got some challenges like cleaning the data and runtime for transformers, so we have taken subset of our dataset in order to calculate the score. The traditional classifier has less accuracy while comparing with transformers, but we have used pre-trained model while training gave us less accuracy when compared to traditional method.

After getting the review sentiment we can know wither review is positive/negative or neutral. we have done performance metrics which tells if there is any load imbalance in the dataset or not which gives model accuracy.

**5. Proposed Model**

After examination of many research papers and reviewing the articles on sentiment analysis. Examine popular classification techniques such as Nave Bayes, Random Forest, k-nearest neighbour, Decision Tree Induction, and Support Vector Machine and Transformers using RoBERTa.

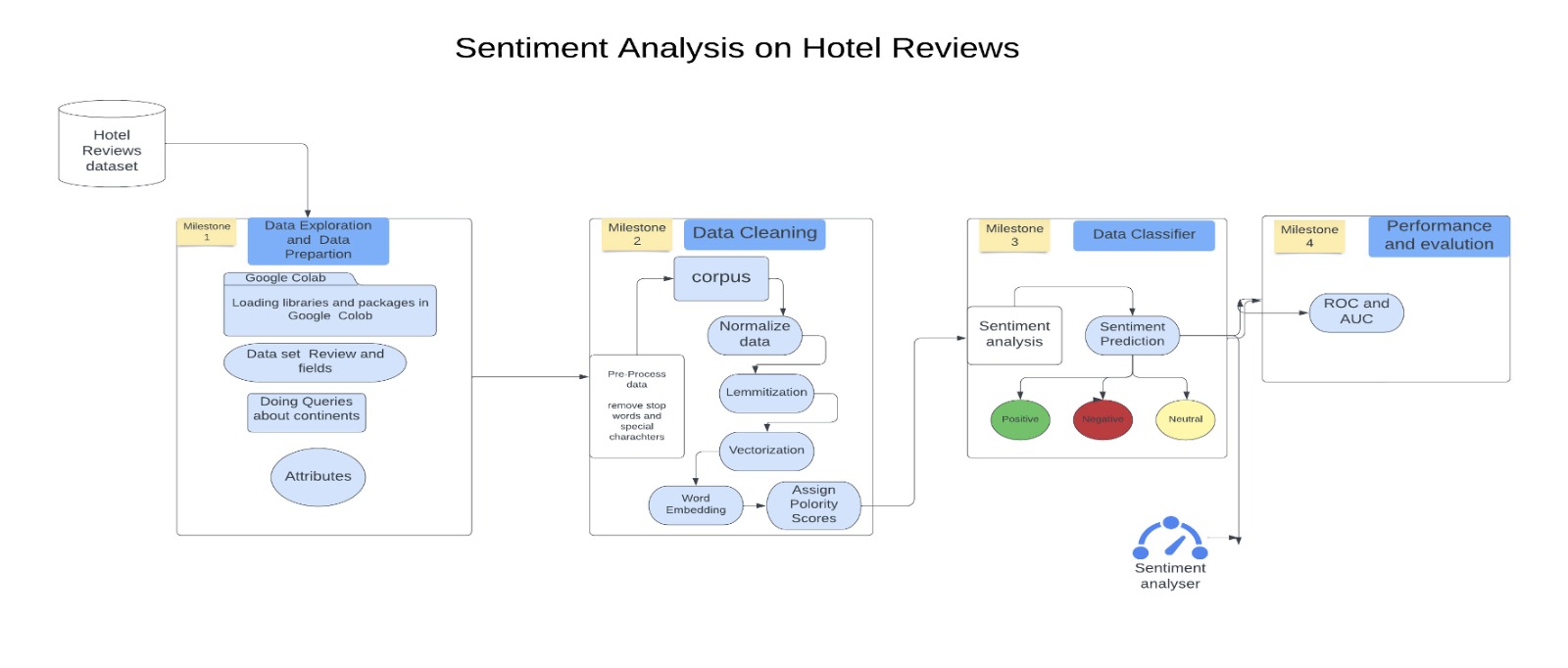
Dataset is obtained from Booking.com. This dataset includes 515,000 customer reviews and ratings for 1493 premium hotels throughout Europe and other countries.

Steps to calculate sentiment score of the model

* Data preparation: Panda’s methods will give some statistical parameters of the data set like count, mean and standard deviation.
* Data Cleaning: It is a fundamental step in any NLP techniques, Data Cleaning and pre-processing steps. Using genism module to convert the text to vectors using DOC2VEC function and by using SentimentIntensityAnalyzer we find the polarity scores and plot word cloud.
* Data Classification: We are using Random forest and Transformers Using RoBERTa.

.

* Performance Metrics: We are using a receiver operating characteristic (ROC) and area under the curve (AUC). Precision Recall curve(PR) for performance and evaluation.



**Key Concepts**

1. **NLTK**

The popular open-source library Natural Language Toolkit (nltk) makes working with human language data in Python straightforward. It provides a wide range of NLP methods, including as sentiment analysis, part-of-speech tagging, tokenization, stemming, lemmatization, and named entity recognition, among others.

1. **SentimentIntensityAnalyzer**

It is a rule-based sentiment analyser, and depending on their semantic orientation, sentences are frequently labelled as either positive or negative. Lastly, we use the polarity scores technique to determine the emotion.

1. **Gensim**

The gensim library is an open-source Python library for topic modelling and similarity detection in large and complex text datasets. The Gensim algorithms Word2Vec, Fast Text, Latent Semantic Indexing where it automatically recognize the semantic structure of documents by examining statistical occurrences patterns within a corpus of training texts. There is no need for human input because these algorithms are unsupervised.

1. **Bag Of Words (BOW)**

Raw text is initially reprocessed and after processing and the cleaning techniques. Here, it Remove all unnecessary special characters, if there are words of other accent like polish, German, Spanish etc. Remove or replace them or add the right Unicode to make them readable for machine.

1. **Normalize all Data**

Make the data properly in a single case letter, either upper or lower. Preferred lower using .lower() function

1. **Stemming and lemmatization**

In this methods used by search engines and chat bots to analyse the meaning behind a word. Stemming uses the stem of the word, while lemmatization uses the context in which the word is being used.

1. **Term Frequency and Inverse Dense Frequency(TF-IDF )**

Inverse document frequency looks at how common (or uncommon) a word is amongst the document.

IDF =Log [(# Number of documents) / (Number of documents containing the word)]

Term frequency works by looking at the frequency of a particular term you are concerned with relative to the document.

TF = (Number of repetitions of word in a document) / (# of words in document).

For combined TF – IDF = TF(t, d) \* IDF(t)

So, using TF and IDF machine makes sense of important words in a document and important words throughout all documents.

1. **Word2vec**

It is a combination of models where it represents distributed of words in a corpus C. Here, it is an algorithm which accepts input corpus and outputs as vector representation for each word. Word embeddings are numerical representations of words that capture their semantic meaning, used in various NLP tasks such as text classification, sentiment analysis, and machine translation.Word2Vec creates a dense vector representation for each word, allowing us to explore semantic relationships between words.

1. **Random Forest Classifier**

It is popular algorithm for classification and regression issues is the supervised machine learning technique known as random forest. It creates decision trees from several samples, using the majority vote for classification and the average for regression. The ”forest” it creates is an ensemble of decision trees, often trained using the ”bagging” approach.

Bagging: The final result is based on majority vote and a separate training subset is created using replacement from a sample of the training data.

**Data exploration**

1. **Overview of Dataset**

As we have taken dataset from booking.com. The CSV file contains 17 fields. The description of each field is as below.

* Hotel Address: Address of hotel.
* Review Date: Date when reviewer posted the corresponding review.
* Average Score: Average Score of the hotel, calculated based on the latest comment in the last year.
* Hotel Name: Name of Hotel
* Reviewer Nationality: Nationality of Reviewer.
* Negative Review: Negative Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: ’No Negative’.
* ReviewTotalNegativeWordCounts: Total number of words in the negative review.
* Positive Review: Positive Review the reviewer gave to the hotel. If the reviewer does not give the negative review, then it should be: ’No Positive’.
* ReviewTotalPositiveWordCounts: Total number of words in the positive review.
* Reviewer Score: Score the reviewer has given to the hotel, based on his/her experience.
* TotalNumberofReviewsReviewerHasGiven: Number of Reviews the reviewers has given in the past.
* TotalNumberof Reviews: Total number of valid reviews the hotel has.
* Tags: Tags reviewer gave the hotel.
* dayssincereview: Duration between the review date and scrape date.
* AdditionalNumberof Scoring: There are also some guests who just made a scoring on the service rather than a review. This number indicates how many valid scores without review in there.
* lat: Latitude of the hotel.
* lng: longitude of the hotel.

1. **Queries**

After data is explored we have found that we don’t had any region specific data in the data set so we have taken a different data set and added new column to original dataset that is ‘region. so now, we can see the in the data from which country, continent and region data in the dataset and from which place customer visited.

Table

Description automatically generated

we have also explored how many null values that is no data found the fields using “isnull”function and found total sum, duplicated data, converting the reviewe\_score into integer, finding numeric and non-numeric columns. We have also dropped duplicates in the data set.

Graphical user interface, application, table

Description automatically generated

**Data Cleaning**

After exploring dataset the next step is Data Cleaning. In this step data is cleaned in many steps. Data cleaning is the fundamental step in any NLP techniques. The below code represents that we are doing the following processes.

Graphical user interface, text, application

Description automatically generated

* Convert the sentences to lower case letter as a step for pre-processing.
* Remove special symbols like “”, !,@,#,$ in our data.
* Remove stop words and remove tokens like tags.
* Tokenization, lemmatization.
* Remove no negatives and no positives from data.
* Clean textual data using parts of speech.

After cleaning the text, After this we will create Doc2vec vector using gensim model, thus text is transferred to vectors each individually. The next step is to add a sentiment analysis column’s for this we have used vader which is part of NLTK. Thus, we get polarity scores for each individual review in the dataset.

Text

Description automatically generated with medium confidence

**6. Visual Applications**

1. **Distribution of Reviewer Score:** Here, we have seen Distribution of Reviewer Score, we plotted on x label as Reviewer score and Y label as total count. We found that highest count is with reviewer score 10.

**Chart, histogram

Description automatically generated**

1. **Correlation between the different variables:** Pearson Correlation Coefficient it is a way to show the relationship between two variables. It is used when there is a measure of the linear association between two variables*.*It always takes on a value between -1 and 1 where:

* -1 indicates a perfectly negative linear correlation between two variables.
* 0 indicates no linear correlation between two variables.
* 1 indicates a perfectly positive linear correlation between two variables.

The further away the correlation coefficient is from zero, the stronger the relationship between the two variables.

Chart

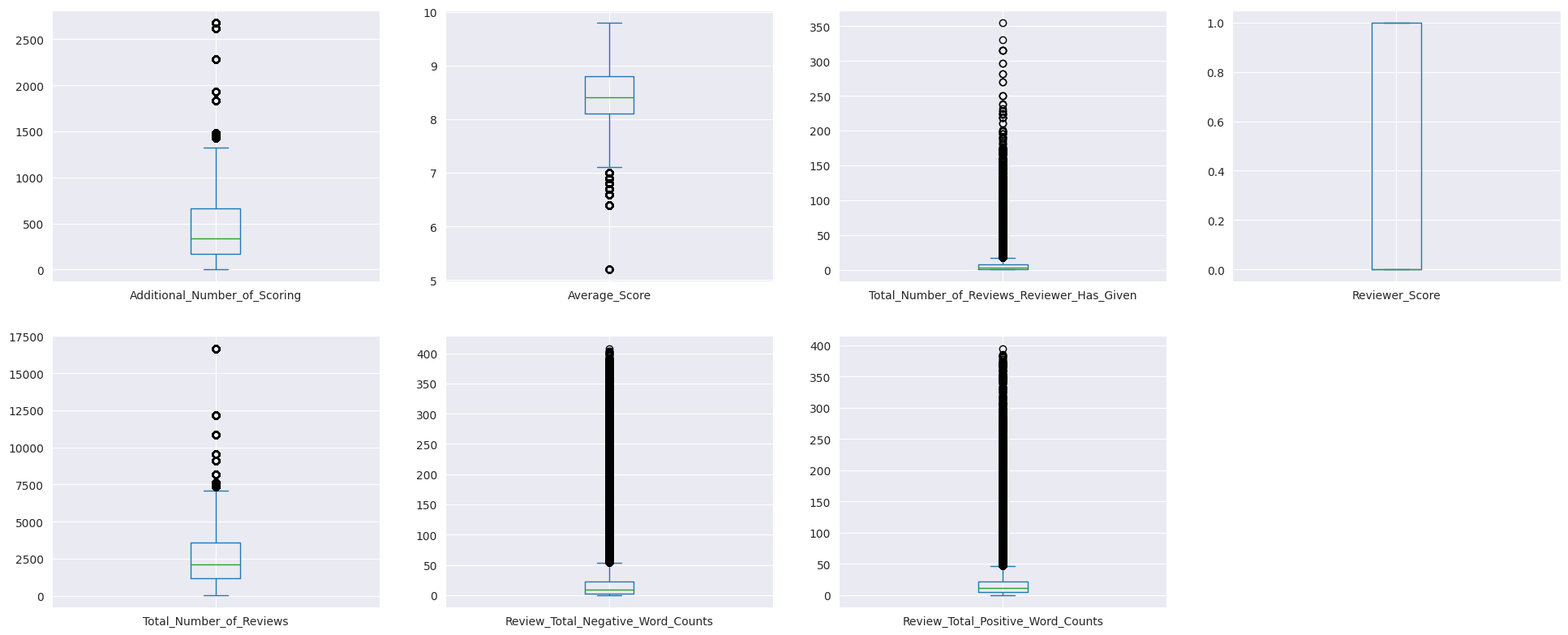
Description automatically generated

From the above heatmap, we can see there is negative correlation between lat and lng, lat and reviewerscore, and many. This gives that if one increase the other value also increase positive correlation function.

1. Boxplot: In this example, we have a overview of a set of data values with characteristics like minimum, first quartile, median, third quartile, and maximum is shown using a box plot or whisker plot. A box with a vertical line across it at the median is drawn between the first and third quartiles. The data to be shown is represented by the x-axis, and the frequency distribution is shown on the y-axis.

Box plots are used to show how numerical data values are distributed, especially when comparing them between different groups. They provide comprehensive information on the symmetry, skew, variance, and outliers of a set of data and are intended to convey high-level information at a glance.

from below plots we can see only reviewer score and counts does not have outliers, remaining all have outliers.



1. **Number of reviews given by nationality**: Horizontal bar plot here gives the quantitative data are shown in a horizontal bar plot using rectangular bars with lengths proportional to the values they represent. In below graph we can see number of reviews given by United Kingdom is highest while least is with Hong Kong with the dataset which we had.

Table

Description automatically generated

1. Checking who gave the worst reviews according to nationalities who have given at least 1000 reviews in the data. From, below visualization we can see average worst reviews are given by the Iran, Oman, and UAE. whereas good average reviews are given by the USA, New Zealand and Israel.

Background pattern

Description automatically generated

1. **Distribution of Hotel City and Country:** In the below visualization we can know distribution of hotels in the dataset. we can see all are European countries as our dataset has European. Highest distribution is there in UK, while least has Milan Italy.

**Chart, bar chart

Description automatically generated**

1. **Plotting Word Cloud**: To find what are the words frequently used in the reviews we will import word cloud library and plot the word cloud using Mat plot library. From above visualization we can see words like hotel,welcoming,bathroom etc are frequently used.

Text

Description automatically generated

1. **plot sentiment distribution for positive and negative reviews:** In this plot we can see that sentimental distribution score has been showed that is if it is good review then the sentimental score will be 1 and if it is bad review then it will show as –1 if it is neutral then it shows as 0. Blue line shows good review whereas orange shows bad review.

**Chart, line chart

Description automatically generated**

7. **Experimental Performance and Evaluations**

In our method we have used two methods to classify the data they are Random Forest Classifier and RoBERTa using Transformers.

1. **Random Forest Classifier:** The supervised machine learning method known as random forest is a popular strategy for classification and regression issues.

It constructs decision trees from a range of samples using the average for regression and the majority vote for classification. The "forest" that it creates is made up of an ensemble of decision trees that were often trained using the "bagging" approach.

Bagging: A unique training subset is produced from a sample of the training data by replacement, and by the final conclusion is determined by a majority vote. Diversity, parallelization, train-test split, and stability are some of the advantages of a random forest, which uses CPU power, minimizes feature space, and depends on majority vote.

Steps involved in training our dataset are we have imported necessary libraries from Sklearn ensemble Random Forest classifier and model select to train split our dataset features selected.

We have used hyperparameters like n estimators and random state and trained the dataset with different set of values.

From below we can see feature selection we trained the random forest and queried the top 20 features to see.

Graphical user interface, text, application, email

Description automatically generated

**Performance of Random Forest**

1. **ROC Curve :** A receiver operating characteristic (ROC) curve shows how well a model can categorize binary outcomes. An ROC curve is created by displaying a model’s false positive rate vs its true positive rate for each feasible cut-off value. The area under the curve (AUC) is frequently measured and used as a statistic to demonstrate how well a model can identify data points.

The ROC (Receiver Operating Characteristic) curve is frequently employed to summarize the quality of our classifier. The better predictions are, the higher the curve is above the diagonal baseline. Although the AUC ROC (Area Under the Curve ROC) is excellent, we should not use it to evaluate the quality of our model here. Because our dataset is imbalanced, the Negatives corresponds to our number of positive reviews, which is very high.

This means that even if there are some instances of false positives, our FPR will remain very low. Our model will be able to forecast many false positives while maintaining a low false positive rate, while raising the genuine positive rate and therefore artificially improving the accuracy.

Our model has shown average 0.88 with different set of Hyper parameters.

**Chart, line chart

Description automatically generated**

1. **PR Curve:** Precession and recall are used to calculate the PR Curve. Precision (also known as positive predictive value) is calculated by dividing the number of true positives by the total number of positive predictions. As a result, precision measures the fraction of valid positive predictions:

How accurate were our model's favorable predictions. The number of true positives divided by the total number of true positives and false negatives (i.e. all genuine positives) is described as recall (also known as sensitivity). As a result, recall measures what proportion of true positives you were able to identify.How accurate our model was in detecting positives.

Chart, histogram

Description automatically generated

After experimenting with various N-Estimators and Random State values, the Random Forest Classifier achieved the highest accuracy of 0.89 for the ROC and 0.34 for the PR curves.

Table

Description automatically generated

1. **Transformers using RoBERTa:** RoBERTa is a BERT variation that alters the pretraining procedure. Among the modifications are larger batch sizes for training models.

Training on longer sequences, deleting the next sentence prediction target, and dynamically altering the masking pattern used on the training data are all options. By dynamically changing the masking pattern used on the training data and removing the goal training for longer sequences.

We used pretrained **cardiffnlp/twitter-Roberta analysis** to train our dataset.

Steps involved in Roberta process are installing necessary libraries which are required for our model. They are AutoTokenizer, AutoModelForSequenceClassification, and SoftMax.

Here we have taken a subset of 1000 reviews which are there in the dataset. Next step is to encode the text with tokenizer, using pretrained model we are running the dataset and then applying the SoftMax in order to get polarity scores.

**Performance evaluations**

Another strategy using Transformers yields an accuracy of 0.99 in Vader and 0.94 in Roberta.Here, at first we have taken subset of 1000 reviews and done sentimental analysis on all the dataframe subset taken from the dataset.

Here,we compared the first 1000 reviews which is taken as subset with both the classifiers in order to check the accuracy of the model and which is performing better.

A screenshot of a computer

Description automatically generated with low confidence