A picture containing text, clipart

Description automatically generated

Ahram Canadian University

Faculty Of Computer Science & Information Technology

Natural Language Processing

Eng. Ahmed Tamer

Project Name : Twitter Sentiment Analysis

Submitted By:

|  |  |
| --- | --- |
| Name | ID |
| Fathy Hesham Fathy | 41910128 |
| Ayman Magdy Yassin | 41910112 |
| Radwa Mohamed Mahmoud | 41910100 |
|  |  |
|  |  |

***Contant***

1. Introduction ……………………………………………………………….
2. Related Work ……………………………………………………………….
3. Main Point ……………………………………………………………….
4. Experiment ……………………………………………………………….
   * Data Exploration ……………………………………………………………….
   * Data Visualization ……………………………………………………………….
   * Data preprocessing ……………………………………………………………….
5. Model Building and Evaluation …………………………………………………….
   * Model Building ……………………………………………………………….
   * Evaluation ……………………………………………………………….

***Introduction***

Sentiment analysis is a method of determining whether a text has a positive or negative feeling. One of the most interesting NLP applications is this. By analyzing the words in a sentence, you can use NLP (Natural Language Processing) to determine whether positive or negative comments were made by the people in the given data set. And from here comes our special idea: to analyze the tweets of giant companies such as FIFA, Facebook, and Google, and analyze whether they are positive or negative.

***Related Work***

Many studies and experiments have been done in analyzing drug reviews using different predictive algorithms. There have been experiments using Logistic Regression (70%), Random Forest (72%), and KNN (63%).

***Main Point***

In this project, we are trying to build models that are able to capture review texts as features, analyze them, and predict whether they are positive or negative reviews for corporate tweets. Several predictive algorithms are used, and a comparison is made between them. We also try to take into account not only textual data, but also numerical and categorical data, such as the tweet's number, the company's name, and the tweet's content.

***Experiments***

1. Data Exploration :

We looked at the data. The data consists of two sets of data, namely the training dataset and the test dataset. The data for training contains 4 columns, they are [**Tweet\_UniqueID , Entity , Sentiment , Tweet\_content**], and 74,682 rows. Also, the data for the test contains 4 columns and 1000 rows.

|  |  |
| --- | --- |
| ***Column Names*** | ***The Definition*** |
| Tweet\_UniqueID | The unique ID of each tweet |
| Entity | The names of the companies that have tweet data |
| Sentiment | The status of tweets if they are positive or negative |
| Tweet\_content | Text content for each tweet |

We have in the training data some of the duplicate rows and it was 2700 rows and they have been removed. In the row [Tweet\_content] there were some empty cells (missing value) that were 326 and they have been removed. Neutral and Irrelevant have been removed.

.

1. Data Visualization :

A plot was made showing the amount and percentage of the data in terms of positive or negative

|  |  |
| --- | --- |
|  |  |

A plot was made showing each company that issued how many tweets were positive and how many were negative

Chart, bar chart

Description automatically generated

1. Data preprocessing

Preprocessing was done for the text content of the tweet in terms of making all letters lower case and any link was removed and a dictionary of abbreviations was created and also for the short word any html tags were removed and the data was cleaned from the special character and all of the ewlines, tabs, and carriage were removed returns, multiple space characters, stopwords, data stemming and tokenizer.

***Model Building and Evaluation***

1. Decision Tree Classifier

It is a supervised learning algorithm. In this algorithm, data are continuously split into smaller parts until it reaches its class. It uses the terminologies like nodes, edges, and leaf nodes. In the Decision Tree classifier, first we compute the entropy of our database. It tells us the amount of uncertainty of our database. The smaller the uncertainty value, the better is the classification results. Each feature’s information gain is calculated. This then tells us how much uncertainty reduces after spitting the database. Finally, all the information gain is calculated for all features, and now, we split the database which has high information gain. The process repeats until all nodes are cleared.



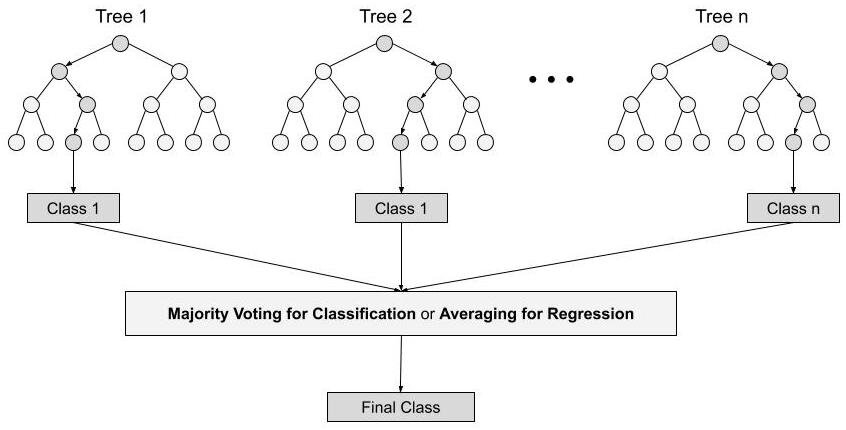
1. K-Nearest Neighbour

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.



1. Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.



***Evaluation The Model***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Recall** | **Precision** | **F1** |
| **Decision Tree** | 0.8895027624309392 | 0.8902038489726121 | 0.8911594242333543 | 0.8894723986321446 |
| **KNN** | 0.7642725598526704 | 0.7654447490567574 | 0.7682921810699588 | 0.7638488719760805 |
| **Random Forest** | 0.8710865561694291 | 0.8723772427458538 | 0.8769819460855706 | 0.8708127141225732 |

**Why are neural networks important?**

Neural networks are also ideally suited to help people solve complex problems in real-life situations. They can learn and model the relationships between inputs and outputs that are nonlinear and complex; make generalizations and inferences; reveal hidden relationships, patterns and predictions; and model highly volatile data (such as financial time series data) and variances needed to predict rare events .As a result, neural networks can improve decision processes.

**Types of Neural Networks**

There are different kinds of deep neural networks – and each has advantages and disadvantages, depending upon the use. Examples include:

* Convolutional neural networks (CNNs) contain five types of layers: input, convolution, pooling, fully connected and output. Each layer has a specific purpose, like summarizing, connecting or activating. Convolutional neural networks have popularized image classification and object detection. However, CNNs have also been applied to other areas, such as natural language processing and forecasting.
* Recurrent neural networks (RNNs) use sequential information such as time-stamped data from a sensor device or a spoken sentence, composed of a sequence of terms. Unlike traditional neural networks, all inputs to a recurrent neural network are not independent of each other, and the output for each element depends on the computations of its preceding elements. RNNs are used in fore­casting and time series applications, sentiment analysis and other text applications.
* Feedforward neural networks, in which each perceptron in one layer is connected to every perceptron from the next layer. Information is fed forward from one layer to the next in the forward direction only. There are no feedback loops.
* Autoencoder neural networks are used to create abstractions called encoders, created from a given set of inputs. Although similar to more traditional neural networks, autoencoders seek to model the inputs themselves, and therefore the method is considered unsupervised. The premise of autoencoders is to desensitize the irrelevant and sensitize the relevant. As layers are added, further abstractions are formulated at higher layers (layers closest to the point at which a decoder layer is introduced). These abstractions can then be used by linear or nonlinear classifiers.

**How Neural Networks Work**

A simple neural network includes an input layer, an output (or target) layer and, in between, a hidden layer. The layers are connected via nodes, and these connections form a “network” – the neural network – of interconnected nodes.

Diagram

Description automatically generated

A node is patterned after a neuron in a human brain. Similar in behavior to neurons, nodes are activated when there is sufficient stimuli or input. This activation spreads throughout the network, creating a response to the stimuli (output). The connections between these artificial neurons act as simple synapses, enabling signals to be transmitted from one to another. Signals across layers as they travel from the first input to the last output layer – and get processed along the way.

When posed with a request or problem to solve, the neurons run mathematical calculations to figure out if there’s enough information to pass on the information to the next neuron. Put more simply, they read all the data and figure out where the strongest relationships exist. In the simplest type of network, data inputs received are added up, and if the sum is more than a certain threshold value, the neuron “fires” and activates the neurons it’s connected to As the number of hidden layers within a neural network increases, deep neural networks are formed.

Deep learning architectures take simple neural networks to the next level. Using these layers, data scientists can build their own deep learning networks that enable Machinelearning , which can train a computer to accurately emulate human tasks, such as recognizing speech, identifying images or making predictions. Equally important, the computer can learn on its own by recognizing patterns in many layers of processing.

So let’s put this definition into action. Data is fed into a neural network through the input layer, which communicates to hidden layers. Processing takes place in the hidden layers through a system of weighted connections. Nodes in the hidden layer then combine data from the input layer with a set of coefficients and assigns appropriate weights to inputs. These input-weight products are then summed up. The sum is passed through a node’s activation function, which determines the extent that a signal must progress further through the network to affect the final output. Finally, the hidden layers link to the output layer – where the outputs are retrieved.

1. **A simple neural network**

How Neural Networks Work. A simple neural network **includes an input layer, an output (or target) layer and, in between, a hidden layer**. The layers are connected via nodes, and these connections form a “network” – the neural network – of interconnected nodes. A node is patterned after a neuron in a human brain.

1. **Convolutional Neural Network**

A convolutional neural network (CNN or ConvNet) is a network architecture for deep learning that learns directly from data. CNNs are particularly useful for finding patterns in images to recognize objects, classes, and categories. They can also be quite effective for classifying audio, time-series, and signal data.

1. **Recurrent Neural Network (LSTM)**

Long Short-Term Memory (LSTM) networks are **a type of recurrent neural network** capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning.