Support Tickets Classification using

Machine Learning and Deep Learning

By

Musale Krushna Pavan (Reg. No- 0000150)

Sanath Singavarapu (Reg. No- 0000188)

**Supervisor**: Dr. Bhaskar Biswas



*Bachelor Thesis submitted to*

Indian Institute of Information Technology Kalyani

*for the partial fulfilment of the degree of*

**Bachelor of Technology**

**in**

**Computer Science and Engineering**

**April 2020**

**Certificate**

This is to certify that the thesis entitled “Support Tickets Classification using Machine Learning and Deep Learning” being submitted by Sanath Singavarapu and Musale Krushna Pavan, an undergraduate student, Reg. No: 188 and 150, in the Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, for the award of Bachelor of Technology in Computer Science and Information Technology is an original research work carried by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of Indian Institute of Information Technology Kalyani and in my opinion, has reached the standards needed for submission. The work, techniques and the results presented have not been submitted to any other University or Institute for the award of any other degree or diploma.

Dr. Bhaskar Biswas,

Faculty

Indian Institute of Information Technology Kalyani

Declaration

I hereby declare that the work being presented in this thesis entitled, “Support Tickets Classification using Machine Learning and Deep Learning”, submitted to Indian Institute of Information Technology Kalyani in partial fulfilment for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the period from Jan, 2020 to May, 2020 under the supervision of Dr. Bhaskar Biswas, Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, does not contain any classified information.

Name of the Candidate: Sanath Singavarapu, Musale Krushna Pavan.

Reg No/Roll No:188,150.

Name of the Department: Computer Science and Engineering.

Institute Name: Indian Institute of Information Technology Kalyani

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

..................................

Dr. Bhaskar Biswas,

Faculty

Indian Institute of Information Technology Kalyani

Place: Kalyani

Date:

Acknowledgments

First of all, I would like to take this opportunity to thank and to express our gratitude to our supervisor Dr. Bhaskar Biswas for his guidance, and instruction that he has given us from the time we have been doing thesis under his supervision through the completion of our undergraduate project. The skills we have been able to learn from her during our Thesis work have benefited us immensely and will continue to do so throughout our future endeavours.

Name of the Candidate: Sanath Singavarapu, Musale Krushna Pavan.

Reg No/Roll No: 188, 150.

Name of the Department: Computer Science and Engineering.

Institute Name: Indian Institute of Information Technology Kalyani

Place: Kalyani.

Date:

**Abstract**

A support ticket is a service request from an end-user that is received by a ticketing system. Emails are often referred to as support tickets. [1]

IT tickets is the generalized term used to refer to a record of work performed (or needing to be performed) by your [IT support](https://freshservice.com/it-support-software) organization to operate your company’s technology environment, fix issues and resolve user requests. Tickets may represent many different types of tasks or activities depending on the nature of your IT environment and the focus of your support team. They may go by other names like “service requests”, “trouble tickets” or “support cases” but most organizations and users are familiar with the term “IT ticket” so we will use it for simplicity.

IT tickets are important to your company because they keep a record of each of the operations and support activities that take place to keep your IT environment up and running, adding value to the business. Tickets are typically captured in an IT Service Management (ITSM) system where they are stored, managed and updated as the issue or activity is resolved. IT help desks use tickets as a means of capturing and recording interactions with users. Operations teams use tickets to track technical issues that need to be addressed. IT management uses ticket data to understand the workload of their teams, make resourcing decisions and facilitate vendor partnerships.

Managing IT tickets effectively is an important part of ensuring your business receives the full value of your company’s IT investments. This is an important concept at the core of your IT operations and leveraging IT ticketing best practices is a good way to help your IT function manage costs, provide better systems and services to users and mitigate the impact of business disrupting events.

[2]

Now we can use the tools and algorithms of machine learning and consider the records of the previous IT Tickets and we can classify the ticket into its class. Here in this paper, we have discussed various algorithms of Machine Learning and Architectures of Deep Learning to classify the ticket.

**Table of Contents**

1. Introduction
2. Machine Learning
   1. Supervised Learning
   2. Unsupervised Learning
3. Deep Learning
4. Natural Language Processing
5. Dataset Description
6. Contribution
7. Classification
   1. Random Forest Classifier
   2. Support Vector Machine
   3. Naive Bayes Classifier
   4. Logistic Regression Classifier
   5. Deep Learning Classifier
8. Comparing Results
9. References

**Introduction**

The contents of this paper mainly focus on various machine learning, deep learning algorithms and Natural Language Processing practices that are valuable in IT Support Tickets Classification. Getting Support Tickets of various departments and separating each department tickets to their and solving them is quite difficult. Searching for a ticket in all these support tickets is not an easy task. In order to solve this problem, we came up with a solution by including machine learning, deep learning and Natural Language Processing Techniques**.**

IT Tickets are of various types, some of them are Database Tickets,

Access Tickets, Help Tickets, Application Tickets, Network Tickets and many more types.

We consider some past tickets which are already labelled and worked on it classify a new It Ticket into its respective category.

**Machine Learning**

“Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.”

Machine Learning is mainly classified into two different groups:

**Supervised Learning:**

“Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples.”

For every given input X there is a labelled output Y. The model learns according to data by the programmed algorithm. The trained model predicts the output with an input given.

In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyses the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way (see inductive bias).

Given a set of ‘N’ training examples of the form ‘{x1, y1), ..., (xn,yn)}’ such that ‘x{i}’ is the feature vector of the i-th example and ‘ y{i}’ is its label (i.e., class), a learning algorithm seeks a function ‘g:X -> Y’ , where ‘X’ is the input space and ‘Y’ is the output space. The function ‘g’ is an element of some space of possible functions ‘G’, usually called the hypothesis space. It is sometimes convenient to represent ‘g’ using a scoring function ‘f: X x Y ->R’ such that ‘g’ is defined as returning the ‘y’ value that gives the highest score: ‘g(x)=arg max f(X,Y)’ . Let ‘F’ denote the space of scoring functions. [3]

Ex: K-Neighbours Classifier, Linear SVC etc.

**Unsupervised Learning:**

“Unsupervised learning is the training of an artificial intelligence (AI) algorithm using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance.”

The model learns features according to the given data. For every given input X there is no labelled output available.

This is a type of self-organized Hebbian learning that helps find previously unknown patterns in a data set without pre-existing labels. It is also known as self-organization and allows modelling probability densities of given inputs.

Two of the main methods used in unsupervised learning are principal component and cluster analysis. Cluster analysis is used in unsupervised learning to group, or segment, datasets with shared attributes in order to extrapolate algorithmic relationships. Cluster analysis is a branch of machine learning that groups the data that has not been labelled, classified or categorized. Instead of responding to feedback, cluster analysis identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data. This approach helps detect anomalous data points that do not fit into either group. [4]

Ex: K-Means Clustering, Mixture Models, DBSCAN etc.

**Deep Learning**

**Deep Learning** is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called **artificial neural networks**.

**Deep learning** (also known as **deep structured learning** or **differential programming**) is part of a broader family of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) methods based on [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_networks) with [representation learning](https://en.wikipedia.org/wiki/Representation_learning). Learning can be [supervised](https://en.wikipedia.org/wiki/Supervised_learning), [semi-supervised](https://en.wikipedia.org/wiki/Semi-supervised_learning) or [unsupervised](https://en.wikipedia.org/wiki/Unsupervised_learning).

Deep learning architectures such as [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks), [deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network), [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks) and [convolution neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_networks) have been applied to fields including [computer vision](https://en.wikipedia.org/wiki/Computer_vision), [speech recognition](https://en.wikipedia.org/wiki/Automatic_speech_recognition), [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [audio recognition](https://en.wikipedia.org/wiki/Audio_recognition), social network filtering, [machine translation](https://en.wikipedia.org/wiki/Machine_translation), [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics), [drug design](https://en.wikipedia.org/wiki/Drug_design), medical image analysis, material inspection and [board game](https://en.wikipedia.org/wiki/Board_game) programs, where they have produced results comparable to and in some cases surpassing human expert performance.

[Artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological [brains](https://en.wikipedia.org/wiki/Brain). Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.

A deep neural network (DNN) is an [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a [linear relationship](https://en.wikipedia.org/wiki/Linear_relationship) or a non-linear relationship. The network moves through the layers calculating the probability of each output.

**Natural Language Processing**

**Natural language processing**  (**NLP**) is a subfield of [linguistics](https://en.wikipedia.org/wiki/Linguistics), [computer science](https://en.wikipedia.org/wiki/Computer_science), [information engineering](https://en.wikipedia.org/wiki/Information_engineering_(field)), and [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of [natural language](https://en.wikipedia.org/wiki/Natural_language) data. Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software. Natural Language processing is considered a difficult problem in computer science. It’s the nature of the human language that makes NLP difficult. The rules that dictate the passing of information using natural languages are not easy for computers to understand.

NLP entails applying algorithms to identify and extract the natural language rules such that the unstructured [language data](https://blog.liveedu.tv/a-quick-introduction-to-text-summarization-in-machine-learning/) is converted into a form that computers can understand. When the text has been provided, the computer will utilize algorithms to extract meaning associated with every sentence and collect the essential data from them.

Syntactic analysis and semantic analysis are the main techniques used to complete Natural Language Processing Tasks.

**1. Syntax**

Syntax refers to the arrangement of words in a sentence such that they make grammatical sense. In NLP, syntactic analysis is used to assess how the natural language aligns with the grammatical rules. Computer algorithms are used to apply grammatical rules to a group of words and derive meaning from them.

Here are some syntax techniques that can be used:

* [**Lemmatization**](https://en.wikipedia.org/wiki/Lemmatisation): It entails reducing the various inflected forms of a word into a single form for easy analysis.
* **Morphological** **segmentation**: It involves dividing words into individual units called morphemes.
* **Word segmentation**: It involves dividing a large piece of continuous text into distinct units.
* **Part-of-speech tagging**: It involves identifying the part of speech for every word.
* **Parsing**: It involves undertaking grammatical analysis for the provided sentence.
* **Sentence breaking**: It involves placing sentence boundaries on a large piece of text.
* **Stemming**: It involves cutting the inflected words to their root form.

**2. Semantics**

Semantics refers to the meaning that is conveyed by a text. Semantic analysis is one of the difficult aspects of Natural Language Processing that has not been fully resolved yet. It involves applying computer algorithms to understand the meaning and interpretation of words and how sentences are structured. Here are some techniques in semantic analysis:

* **Named entity recognition (NER):** It involves determining the parts of a text that can be identified and categorized into preset groups. Examples of such groups include names of people and names of places.
* **Word sense disambiguation:** It involves giving meaning to a word based on the context.
* **Natural language generation**: It involves using databases to derive semantic intentions and convert them into human language.

**Dataset Description**

We considered a collection of data with 3001 rows and 2 columns

i.e. Description and Category. Each corresponds to a single ticket logged by the user.

1. Description: The content of the ticket. The issue of the person who has logged that ticket

2. Category: The Class of the Ticket which the ticket belongs to. The following are the categories

'Application’, 'Database’, 'Network’, ‘User Maintenance’, ‘Security'

**Stages over the Flow**

**Data Pre-processing:** It is an important step in the [data mining](https://en.wikipedia.org/wiki/Data_mining) process. If there is much irrelevant and redundant information present or noisy and unreliable data, then [knowledge discovery](https://en.wikipedia.org/wiki/Knowledge_discovery) during the training phase is more difficult. Data preparation and filtering steps can take a considerable amount of processing time. Data pre-processing includes [cleaning](https://en.wikipedia.org/wiki/Data_cleaning), [Instance selection](https://en.wikipedia.org/wiki/Instance_selection), [normalization](https://en.wikipedia.org/wiki/Data_normalization), [transformation](https://en.wikipedia.org/wiki/Data_transformation), [feature extraction](https://en.wikipedia.org/wiki/Feature_extraction) and [selection](https://en.wikipedia.org/wiki/Feature_selection), etc. The product of data pre-processing is the final [training set](https://en.wikipedia.org/wiki/Training_set).

Data pre-processing may affect the way in which outcomes of the final data processing can be interpreted.

**Steps Involved in Data Pre-processing:**

**1. Data Cleaning:**  
The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling missing data, noisy data etc.

**2. Data Transformation:**  
This step is taken in order to transform the data in appropriate forms suitable for the mining process. This step involves normalization, attribute selection etc

**3. Data Reduction:**  
Since data mining is a technique that is used to handle huge amounts of data. While working with a huge volume of data, analysis became harder in such cases. In order to get rid of this, we use data reduction techniques. It aims to increase the storage efficiency and reduce data storage and analysis costs.

We used various techniques of natural language processing such as Stemming with Porter Stemmer Algorithm, Tokenization, Stop Word Removals, Contraction Extractions, Vectorizing data with Count Vectorizer algorithm. Even applied many cleansing techniques all over the flow.

**Set up training data and Testing data:**

## Training Data

The observations in the training set form the experience that the algorithm uses to learn. In supervised learning problems, each observation consists of an observed output variable and one or more observed input variables.

## Test Data

The test set is a set of observations used to evaluate the performance of the model using some performance metric. It is important that no observations from the training set are included in the test set. If the test set does contain examples from the training set, it will be difficult to assess whether the algorithm has learned to generalize from the training set or has simply memorized it.

**Training the Model:**

The process of training an ML model involves providing an ML algorithm (that is, thelearning algorithm) with training data to learn from. The term ML model refers to the model artifact that is created by the training process.

The training data must contain the correct answer, which is known as a target*or*target attribute*.* The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

You can use the ML model to get predictions on new data for which you do not know the target.

**Contribution**

Ticket classification is a use case of text document classification in which each ticket description is considered as one single document and the category of the ticket is the label of the document. Main objectives of this paper are-

1. Building the IT service desk ticket classifiers using Machine Learning and Deep learning classifier methods such as 1.Random forest classifiers 2. Support vector Machine 3. Naive Bayes classifiers 4. Logistic Regression 5.Neural Networks methods
2. Evaluation and comparison of various different classifiers using various classification performance evaluation metrics.

The advantages of developing such automated ticket classifier systems are simplified user interface, faster resolution time, effective resource utilization, improvement in customer satisfaction and growth in business. The IT infrastructure ticket data considered in this research work had a huge amount of unwanted and unclean data and hence handling data related challenges is one of the key issues in developing such a ticket classifier.

We firstly remove the unnecessary words, then convert each word to its root form. Then we are using the Bag of Words model to convert the words to the numerical vector format. We will use these vectors for the training of all the different models proposed above.

There are few research papers which use the different machine algorithms like Decision tree, Random Forest etc. There are also some papers which overlap with the current research work. But the approach for the data pre-processing is the most important step to get the better results. We have used accuracy as the metrics for the comparison as our dataset contains an equal number of samples for each class.

The most important in the context of text classification is the pre-processing the data and the deep learning model that is built. We have followed the following data cleaning process to cleaning the data

* Convert to lower case
* Expand the contracted words (isn’t 🡪 is not)
* Remove unnecessary punctuations
* Remove special characters
* Remove unnecessary digits
* Removing the hyperlinks and email addresses
* Replacing Contractions
* remove after regards
* Replacing multiple white spaces
* Lemmatizing

The above steps clean the data set makes it ready to fit in the word to vector converter that converts them into numbers.

The Architecture of the deep learning model also stands unique in the way consisting of the following layers.

* Embedding
* SpatialDropout1D
* LSTM
* Dense

The above layers are chosen after doing a lot of research about the layers that needed to be used to get the better results.

**Classification**

**1.Random Forest Classifier:**

Decision trees are the building blocks of the random forest model. A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continuous value)”.

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, and utility. It is one form to show an algorithm that simply contains conditional control statements.

A decision tree looks like a flowchart-like structure in which each internal node represents a “test” or “condition” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the result of the test we then explore the branch in which the output falls in, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

“Random forest classifiers create a set of decision trees from a randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.”

Some of the parameters for this classifier are number trees to form and the parameters that are used in making decision trees. In most of the cases it is better than a general Decision tree classifier as it is generating many different decision trees and chooses the best one.

“A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.”

The low correlation between models is the key. Uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don’t constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

There needs to be some actual signal in our features so that models built using those features do better than random guessing.

The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the most separation between the observations in the left node vs. those in the right node. In contrast, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification.

So, in our random forest, we end up with trees that are not only trained on different sets of data but also use different features to make decisions.

Point needed in order to make accurate class predictions

We need features that have at least some predictive power. After all, if we put garbage in then we will get garbage out.

The trees of the forest and more importantly their predictions need to be uncorrelated (or at least have low correlations with each other). While the algorithm itself via feature randomness tries to engineer these low correlations for us, the features we select and the hyper-parameters we choose will impact the ultimate correlations as well.

The accuracy that we get using this model: 71.34%

**2.Support Vector Machine:**

“A SVM performs classification by finding the hyperplane that maximises the margin between two classes. The vectors that define the hyperplane are the support vectors''.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Given an input X and labelled target Y the algorithm outputs an hyperplane which then further classifies the new inputs

There may be different types of margin to deal with. If there is a linear boundary then it is a linear support vector machine. There are other types of margin (hyper planes) like polynomial, gaussian etc.

Kernel, Regularization, Gamma and Margin are tuning parameters which then decide the algorithm working and accuracy according to the given dataset.

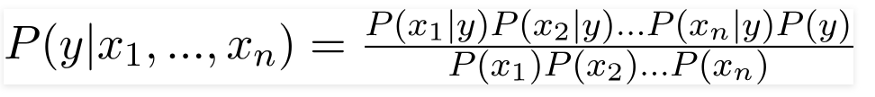
Kernels: Linear, rbf, sigmoid, polynomial

Regularization: The C parameter is the regularization parameter. For higher values of C the optimization will choose a smaller margin.

The accuracy that we get using Liner SVC: 72.77%.

The accuracy that we get using SVM: 68.03%

**Naive Bayes Classifier:**

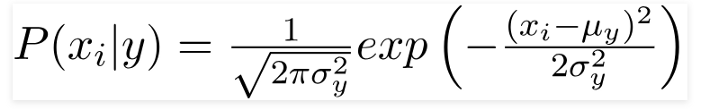
In machine learning, **Naive Bayes classifiers** are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is, the presence of one particular feature does not affect the other. Hence it is called naive.

Multinomial Naive Bayes:

This is mostly used for document classification problems, i.e. whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

Gaussian Naive Bayes:

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution. Since the way the values are present in the dataset changes, the formula for conditional probability changes to

The accuracy that we get using gaussian Naive Bayes: 69.77%.

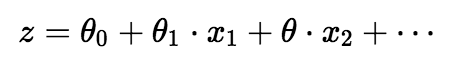
The accuracy that we get using multinomial Naive Bayes: 74.4%.

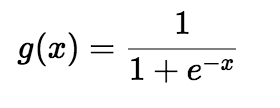
**Logistic Regression:**

Logistic regression is used for classification tasks. A really simple, rudimental and useful algorithm for classification is the logistic regression algorithm.

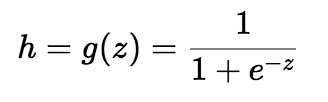
Sigmoid Function (Logistic Function)

Logistic regression algorithm uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be class variable, i.e. 0-no, 1-yes. Therefore, we are squashing the output of the linear equation into a range of [0,1]. To squash the predicted value between 0 and 1, we use the sigmoid function.



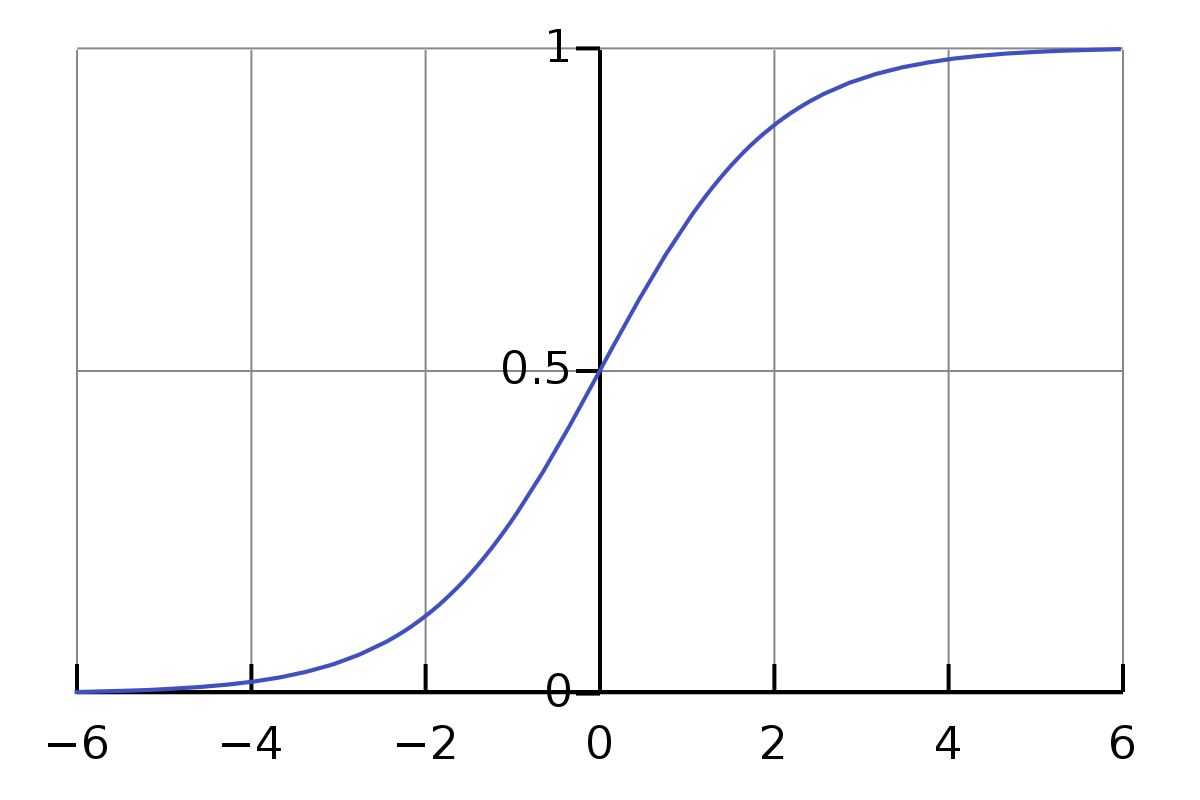


Linear Equation and Sigmoid Function



Squashed output-h

We take the output(z) of the linear equation and give to the function g(x) which returns a squashed value h, the value h will lie in the range of 0 to 1. To understand how sigmoid function squashes the values within the range, let’s visualize the graph of the sigmoid function.



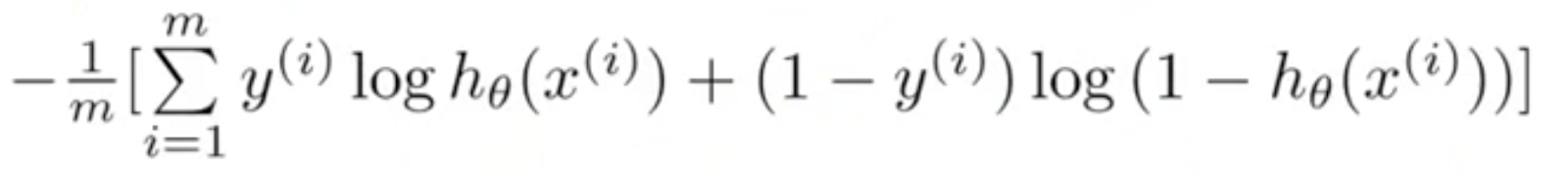
Sigmoid Function graph

As you can see from the graph, the sigmoid function becomes asymptote to y=1 for positive values of x and becomes asymptote to y=0 for negative values of x.

Cost Function

we use a logarithmic loss function to calculate the cost for misclassifying

The above cost function can be rewritten as below since calculating gradients from the above equation is difficult.



Calculating Gradients

The accuracy that we get using this model: 74.74%.

**Architecture Used in Deep Learning Flow**

1. Embedding Layer

2. Drop-Out Layer

3. LSTM Cells

4. Dense Layer

All these layers combinable made the model to train and predict the sentences.

**Embedding Layer**

It requires that the input data be integer encoded, so that each word is represented by a unique integer. The Embedding layer is initialized with random weights and will learn an embedding for all of the words in the training dataset. The Embedding layer is defined as the first hidden layer of a network.

It must specify 3 arguments:

* **Input\_dim**: This is the size of the vocabulary in the text data. For example, if your data is integer encoded to values between 0-10, then the size of the vocabulary would be 11 words.
* **Output\_dim**: This is the size of the vector space in which words will be embedded. It defines the size of the output vectors from this layer for each word. For example, it could be 32 or 100 or even larger. Test different values for your problem.
* **Input\_length**: This is the length of input sequences, as you would define for any input layer of a Keras model. For example, if all of your input documents are comprised of 1000 words, this would be 1000.

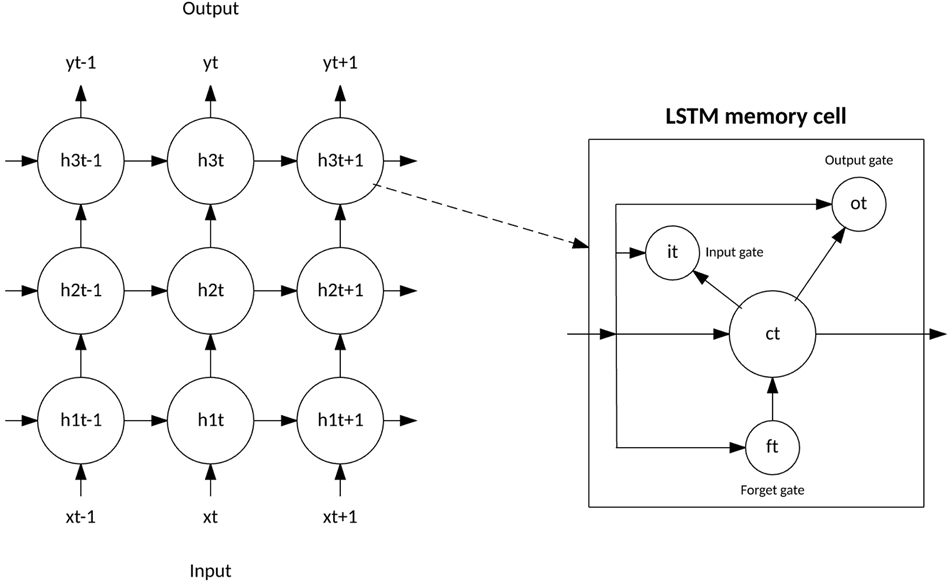
**Drop-Out Layer**

Dropout is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass

The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to over fit the training data.

**Long Short Term Memory (LSTM)**

The LSTM memory cell contains three gates that control how information flows into or out of the cell. The input gate controls when new information can flow into the memory. The forget gate controls when an existing piece of information is forgotten, allowing the cell to remember new data. Finally, the output gate controls when the information that is contained in the cell is used in the output from the cell. The cell also contains weights, which control each gate. The training algorithm, commonly BPTT, optimizes these weights based on the resulting network output error.



**Dense Layer**

A dense layer represents a matrix vector multiplication. (Assuming your batch size is 1) The values in the matrix are the trainable parameters which get updated during back propagation.

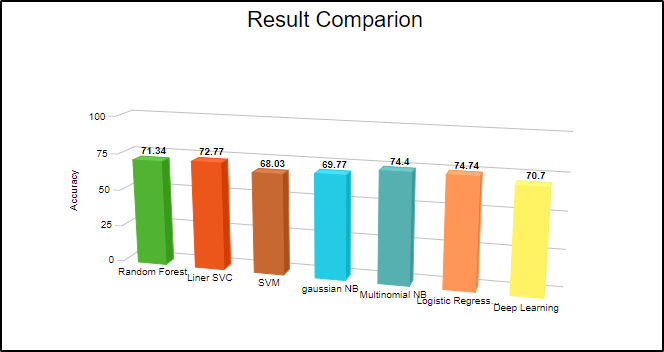
uT.W,W∈Rn×muT.W,W∈Rn×m

So, you get a m dimensional vector as output. A dense layer thus is used to change the dimensions of your vector. Mathematically speaking, it applies a rotation, scaling, translation transform to your vector.

The accuracy that we get using multinomial Naive Bayes: 70.4%.

**Results Comparison:**

The Different results are shown in the following Bar chat



Bar We can observe that most the algorithms are having accuracy around 71%.

Among all the algorithms used we got the highest accuracy for the Logistic regression with 74.4% accuracy and we have the least of 69.77% accuracy with Support vector machine.

With all the different approaches to we have used to increase the accuracy we found that Logistic regression is giving highest accuracy of 74.74% over cross fold validation.

**References**

1. MACHINE LEARNING BASED TICKET CLASSIFICATION IN ISSUE TRACKING SYSTEMS Mucahit Altintas(a,b,c) , A. Cuneyd Tantug (d,b) a maltintas@itu.edu.tr, d tantug@itu.edu.tr b Istanbul Technical University, Istanbul, Turkey c Bayburt University, Bayburt, Turkey
2. S. P. Paramesh, C. Ramya and K. S. Shreedhara, "Classifying the Unstructured IT Service Desk Tickets Using Ensemble of Classifiers," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2018, pp. 221-227, doi: 10.1109/CSITSS.2018.8768734.
3. C. Kadar, D. Wiesmann, J. Iria, D. Husemann and M. Lucic, "Automatic Classification of Change Requests for Improved IT Service Quality," 2011 Annual SRII Global Conference, San Jose, CA, 2011, pp. 430-439, doi: 10.1109/SRII.2011.95.
4. Al-hawari, Feras & Barham, Hala. (2019). A Machine Learning Based Help Desk System for IT Service Management. Journal of King Saud University. 10.1016/j.jksuci.2019.04.001.