Recommendation System Manish Garg

October 5th, 2019

# Definition

**Project Overview**

Recommendation systems are information filtering systems that deal with the problem of digital data overload to pull out items or information according to the user's preferences, interests, and behavior, as inferred from previous activities.

In machine learning one of the most common use case is recommendation to users that they'll be interested in. For example, the way Amazon recommends books based on the book that you are currently Browsing or Netflix recommending movies for you.

A recommendation system is one that learns about what items might be of interest to a user, and then recommends those items for buying, renting, listening, watching, and so on. Recommendation systems are broadly classified into two categories:

* content-based filtering
* collaborative filtering.

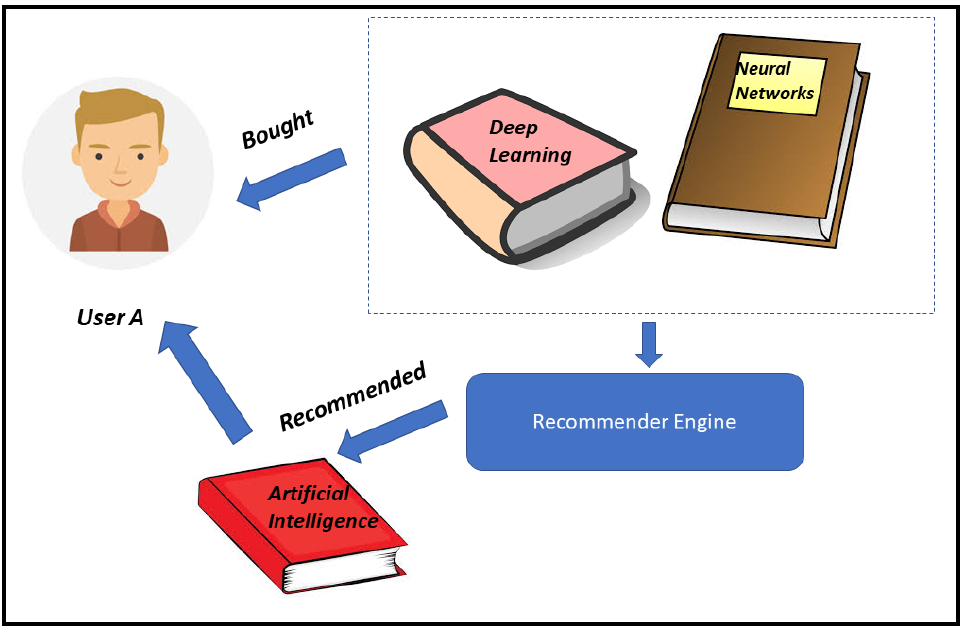
**Content-based filtering**

Content-based filtering is based on creating a detailed model of the content from which recommendations are made, such as the text of books, attributes of movies, or information

about music.

The content model is generally represented as a vector space model. Some of the common models for transforming content into vector space models are TFIDF, the Bagof-words model, Word2Vec, GloVe, and Item2Vec.

Along with the content model, a user profile is also created using information about the user. Content is recommended based on matching the user profile with the content model.



User **A** has bought books named **Deep Learning** and **Neural Networks**. Since the content of the book **Artificial Intelligence** is similar to the two books, the content-based recommender system has recommended the book **Artificial Intelligence to User A.** As we can see, in content-based filtering, the user is recommended items based on their preferences. This doesn't involve how other users have rated the book.

**Advantages of content-based filtering algorithms**

The following are the advantages of content-based filtering algorithms:

* Eliminates the cold-start problem for new items: If we have enough information about the users, and detailed information about the new content, then the cold start problem found in collaborative filtering algorithms does not affect content-based algorithms. The recommendation can be made based on the user profile and the information about the content.
* The recommendations are explainable and transparent: Using content representation models, we'll be able to explain how certain items are selected for recommendations.

**Disadvantages of content-based filtering algorithms**

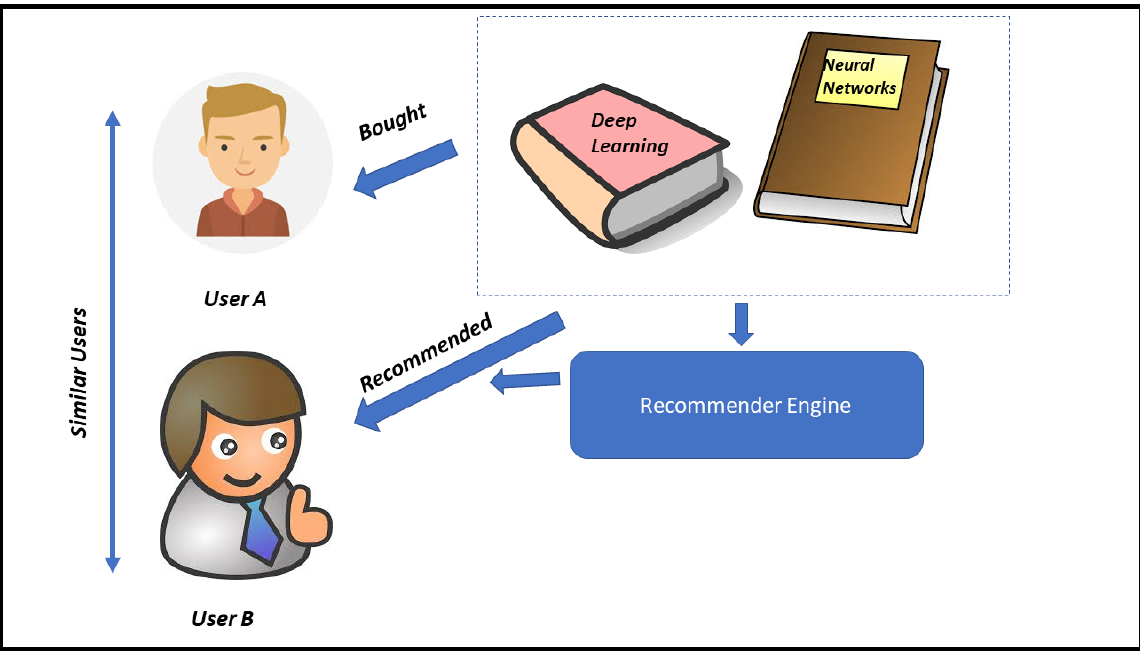
The following are the disadvantages of content-based filtering algorithms:

* Content-based filtering algorithms require detailed information about items and content, which is sometimes not available
* Content-based filtering algorithms are prone to overspecialization

**Collaborative filtering**

Collaborative filtering algorithms do not need detailed information about the user or the items. They build models based on user interactions with items such as song listened, item viewed, link clicked, item purchased or video watched.

It tries to identify similar users pertaining to a given user, and then recommends the user items that similar users have liked, bought, or rated highly. This is generally called **user-user collaborative** filtering. The opposite is to find items similar to a given item and recommend items to users who have also liked, bought, or rated other similar items highly. This goes by the name **item-item collaborative** filtering:



**User** **A** and **User** **B** are very similar in terms of their taste in buying books. User A has recently bought the books **Deep Learning** and **Neural Networks**. Since **User** **B** is very similar to **User** **A,** the user-user collaborative recommender system recommends these books to **User B** as well.

**Problem Statement**

In this project our primary objective is to build a basic **question-answering system (QA systems)**, chatbot.

QA systems can also be of different types like:

**Information retrieval (IR) -based question answering**

It fully relies on the huge amount of information available as text on the Web or in specialized collections such as PubMed. The method processes the question to determine the likely answer type (often a named entity like a person, location, or time)

IR based gets the answer from the documents collected, again it doesn’t generate the answer, it just does copy-paste from the documents, so if the text is not present in the documents, these models can’t give the answer.

They can also **fetch data from large data base** by converting the natural language query into structured database query and returning back the response to the user.

**Story based Question and answering**

This is more like asking the question based on the passage/story given here the input is given in form of triplets for building models. (Story, Question, Answer)

This project intends to use Deep Learning and other Artificial Intelligence algorithms in order to solve the Question Answering System problem.

**Our QA systems chat will be based on Story based Question and answering.**

In this system there will be short story followed by set of questions. The bot will answer the questions by **understating the context, remembering the story and apply logical reasoning.**

Question Answer systems are very useful as they allow users to enter a query based on some facts or stories and the system tries to use the context in the supporting facts and stories to answer the questions effectively instead of just giving out the best suited keywords.

Besides, most of the problems in artificial intelligence and Natural language processing can all be modelled as a question answering problems. For example, the task of text summarization can be modelled as a question answering task in the sense that if the user asks the system "What is the summary of the text?", it can answer the user by providing the appropriate summary.

Metrics

The performance of each classification model is evaluated using statistical measures;

classification accuracy, recall, sensitivity and specificity. It is using true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

We will also summaries the performance of a classification algorithm using confusion

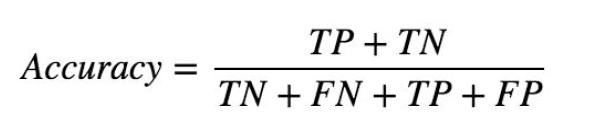
metrics as calculating a confusion matrix can give you a better idea of what your

classification model is getting right and what types of errors it is making.

|  |  |  |
| --- | --- | --- |
|  | **Predicted - Positive** | **Predicted - Negative** |
| **Actual - Positive** | True Positive | False Negative |
| **Actual - Negative** | False Positive | True Negative |

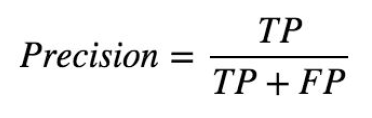
Classification accuracy is defined as the ratio of the number of correctly classified cases and is equal to the

sum of TP and TN divided by the total number of cases (TN + FN + TP + FP).



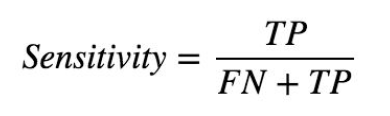
Precision is defined as the number of true positives (TP) over the number of true positives plus the number of

false positives (FP).



Sensitivity refers to the rate of correctly classified positive and is equal to TP divided by the sum of TP and FN.

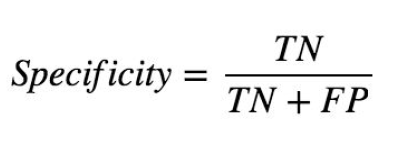
Sensitivity may be referred as a True Positive Rate.



Specificity refers to the rate of correctly classified negative and is equal to the ratio of TN to the sum of TN and

FP

<https://www.sciencedirect.com/science/article/pii/S1877050918308226>



# Data

In order to develop a chatbot, we are using two datasets. These datasets are as follows:

• Cornell Movie-Dialogs dataset

• bAbI dataset

**Cornell Movie-Dialogs dataset**

This dataset has been widely used for developing chatbots. This is available at –

Cornell Movie-Dialogs corpus from this link:

<https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html>

This corpus contains a large metadata-rich collection of fictional conversations extracted from raw

movie scripts.

This corpus has 220,579 conversational exchanges between 10,292 pairs of movie characters. It involves 9,035 characters from 617 movies. In total, it has 304,713 utterances. This dataset also contains movie metadata. There are the following types of metadata:

• Movie-related metadata includes the following details:

* + - Genre of the movie
    - Release year
    - IMDb rating

• Character-related metadata includes the following details:

* + - Gender of 3,774 characters
    - Total number of characters in movies

**The bAbI dataset**

This dataset is built by Facebook AI Research (FAIR), where AI stands for artificial intelligence. This dataset belongs to the bAbI project.

download the dataset from

<https://research.fb.com/downloads/babi/>

The goal of the bAbI project is to try to build an automatic text understanding and reasoning system. This dataset consists of the following sub datasets:

• The (20) QA bAbI tasks

• The (6) dialog bAbI tasks

• The Children's Book Test

• The Movie Dialog dataset

• The WikiMovies dataset

• The Dialog-based Language Learning dataset

• The SimpleQuestions dataset

• HITL Dialogue Simulator

We will be using only one subset here, which is the (20) QA bAbI tasks because it is the one that's most useful for building the chatbot.

If we look at this sub dataset in detail. Here, 20 different tasks have been performed using this (20) QA bAbI dataset. Let's see what these tasks are. These tasks give machines the capacity to perform some reasoning, and based on that, the machine can answer a question. You can refer to the task name given in the following figure:

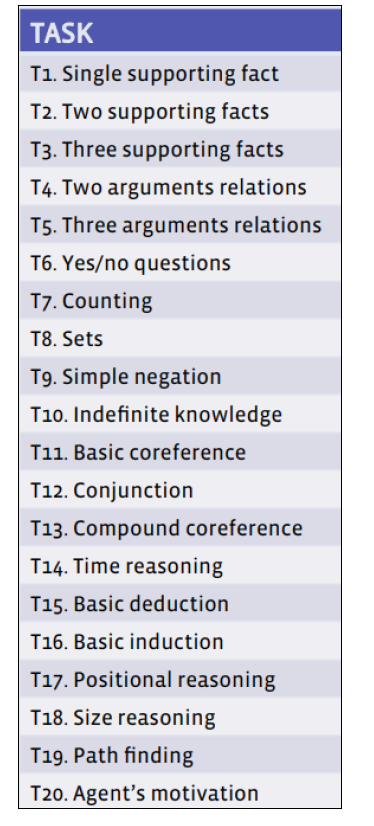


Image source: <http://www.thespermwhale.com/jaseweston/babi/abordes-ICLR.pdf>

# **Algorithms and Techniques**

As the retrieval and rule based chatbot do not work well for the unseen queries and we need to hard code each and every scenario which is not feasible in short duration. Also, it cannot understand the long context so we will not be using this technique.

We will use **generative-based** approaches for our solution. Both the approaches will use **Deep Learning** techniques.

## **Sequence-to-sequence(seq2seq) model**

The sequence-to-sequence model architecture is well suited for capturing the context of the customer input and then generating appropriate responses based on questions just as chatbot would do.

The seq2seq model consists of two Long Short Term Memory (LSTM) recurrent neural networks. The first neural net is an encoder. It processes the input. The second neural net is a decoder. It generates the output.

This model accepts sequence of words in a sentence and outputting a new sequence of words. So, we need a sequence model that can learn data with long range memory dependencies. The LSTM architecture is best suited for this. The encoder LSTM turns the input sentence of variable length into a fixed dimensional vector representation called as **thought vector.**

The reason we are using LSTM is that it can remember words from far back in the sequence; here, we are dealing with large sequence attention mechanisms of the seq2seq model, which helps the decoder selectively look at the parts of the sequence that are most relevant for more accuracy.

## 

(Architecture of the seq2seq model)

Image source: <http://suriyadeepan.github.io/img/seq2seq/seq2seq2.png>

It works by producing answers given by a probabilistic model trained to maximize the probability of the answer given some context.

## End-to-end memory networks

# Benchmark Model

The given dataset is a supervised classification learning problem for which we first tried with simple K-nearest neighbours to get base value of score.

KNN is an algorithm that's used in pattern recognition for object classification based on the characteristics of the nearest objects. An object is classified according to the majority of the votes of its neighbouring k cluster. **KNeighborsClassifier** gave model accuracy for the data as 0.5934.

# **Implementation**

## **Sequence-to-sequence model**

In this implementation, we don't need to generate features because the seq2seq model generates its internal representation for sequences of words given in a sentence. Our implementation part has the following steps:

• Data preparation

• Implementing the seq2seq model

During this implementation, we will be using the Cornell movie-dialogs dataset. First of all, we need to prepare data in a format that we can use for training.

## **5.1.2 Data preparation**

In this step we will be doing following:

• Generating question-answer pairs

• Preprocessing the dataset

• Splitting the dataset into the training dataset and the testing dataset

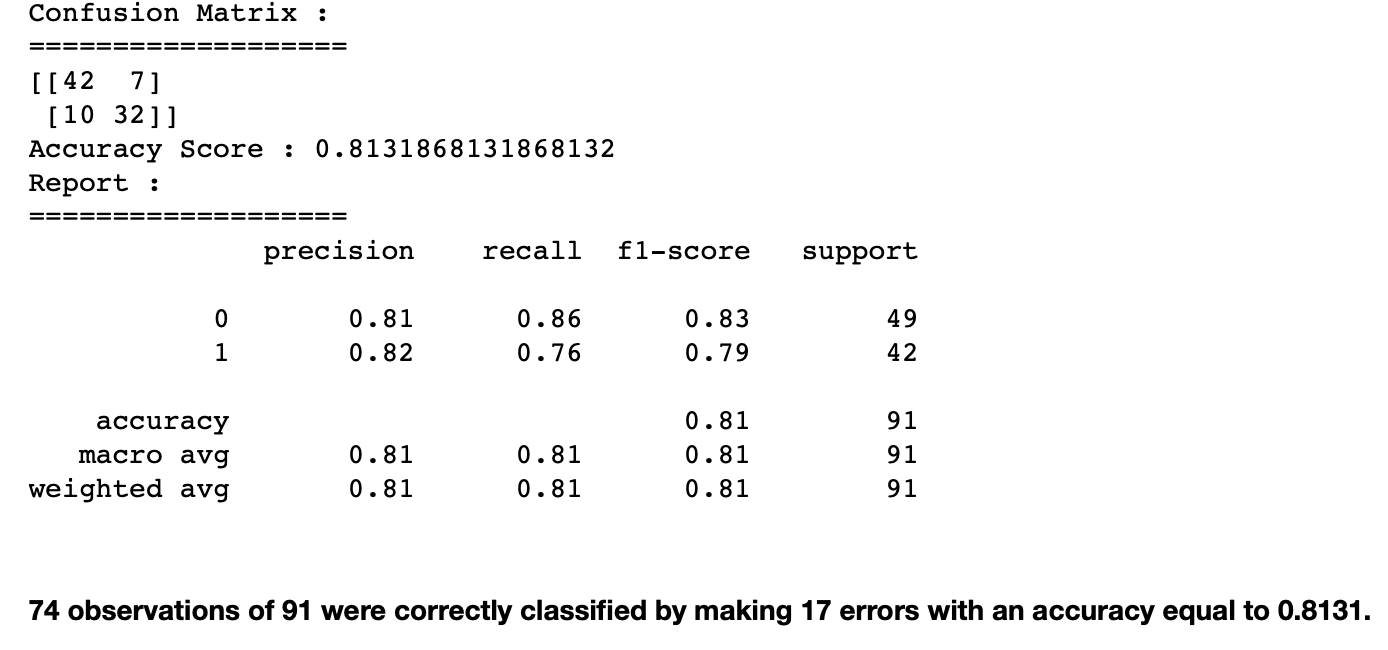
• Building a vocabulary for the training and testing datasets

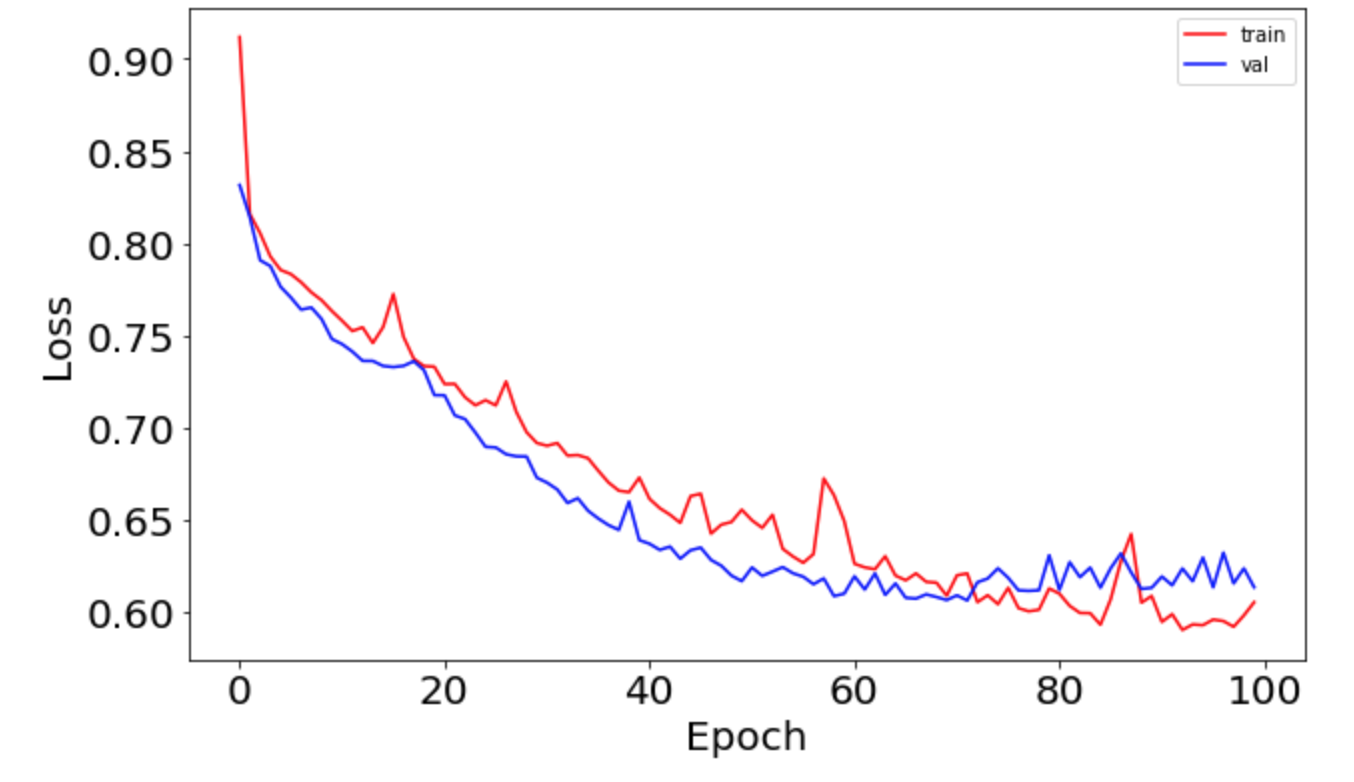
## Refinement and Results

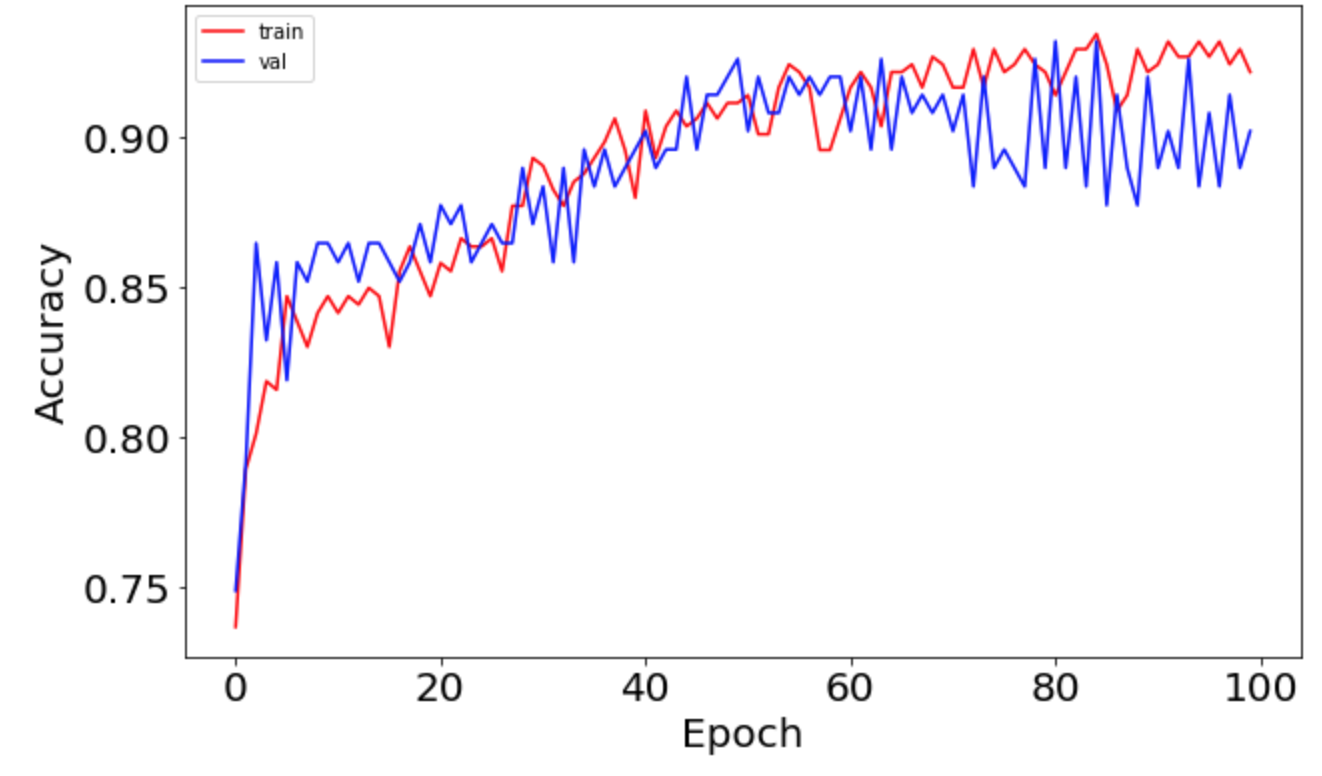
### Deep Neural Network

We performed hyper tuning of parameters of **Deep Neural Network** for binary classification by implementing the interface *tensorflow.keras.wrappers.scikit\_learn.KerasClassifier* and using the *sklearn.model\_selection.RandomizedSearchCV*

|  |  |  |
| --- | --- | --- |
| Parameter | Values Tested | Best Value |
| batch\_size | [10, 20, 40] | 40 |
| epochs | [100, 500, 1000] | 100 |
| optimizer | ['SGD', 'RMSprop', 'Adagrad', 'Adam'] | Adam |
| learning\_rate | [0.001, 0.01, 0.1] | 0.001 |
| init\_mode | ['uniform', 'normal', 'zero'] | 'normal' |

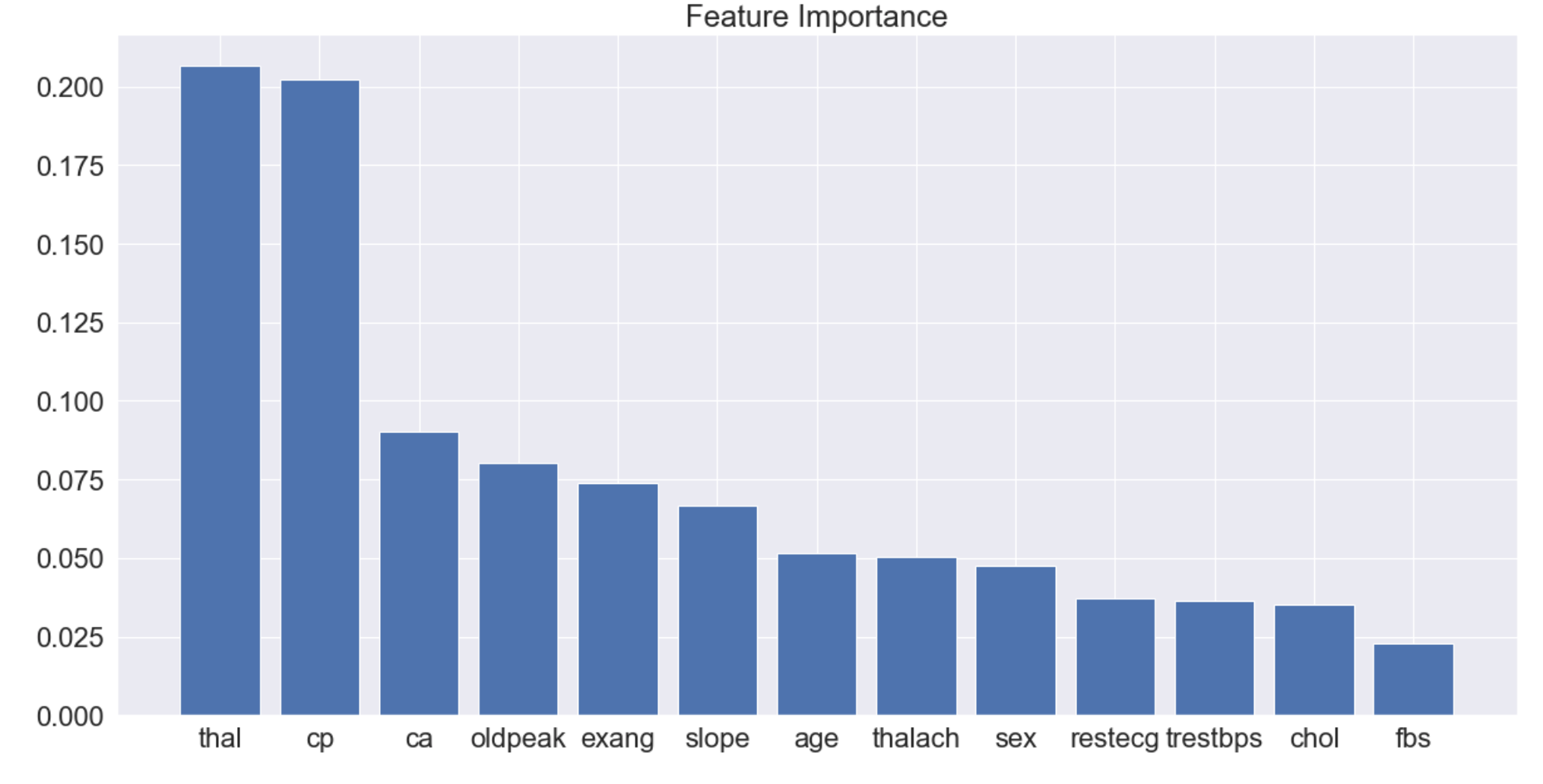






## Conclusion

The feature importance is visualized as:



The graph depicts the most influencing factors for the Heart Disease. Higher values mean the feature is more important in determining the Heart Disease.

As per the above graph 'thal','cp' and 'ca' are the highest influencing factor for the Heart Disease. This also matches with our initial finding in the data visualization section.

* 'cp' is Chest pain type, which determines the type angina. This is one of the major issues reported by the Heart Disease patients. Therefore, Heart Disease dependency on this feature is natural.
* 'thal' Thalassemia is an inherited blood disorder characterized by less hemoglobin and fewer red blood cells in your body than normal. Thalassemia is one of the main causes of Heart disease as Low levels of hemoglobin may cause anemia. Anemia is associated with a special risk in proatherosclerotic conditions and heart disease.
* 'ca' A noninvasive method for coronary artery diseases diagnosis. It indicates Number of major vessels to be get narrowed and are not able to convey enough fresh blood to this blood-pumping organ. This is also quite major factor to determine the heart disease.

**Model evaluations:**

We have used three different models for classification- KNN, Neural Network binary classifier, and XGBoost classifier. Based upon various evaluations matrix we see the XGBoost classifier performs best with highest accuracy of cross validation accuracy score 83.40%. Also, it is much easier to tune XGBoost classifier for the given problem and the dataset.

### Reflection

The most important and time-consuming part of the problem was data cleansing and processing as we need to ensure that we have correctly prepared the data well for the algorithm to work in its full potential.

Once the data was prepared and ready, the next challenge was to pick the optimized parameters for the selected algorithm and the data set.

As the data set was complex but small so may be Deep Neural Network was not the correct algorithm for the problem as it has not given any major advantage and XGBoost classifier has given better results.

The other issue I see is the inconsistency of the models to given consistent results in-spite of similar data set, parameters, setting of initializers and seed values.

There are many other ways to improve the model further like :

* Having more training data.
* Better model tuning using various hyper parameters
* Trying more different Algorithms

<https://arxiv.org/pdf/1506.05869v3>.