Project Name: Banking Query Classification

Problem Statement:

Since we know that various natural language techniques are used to determine grammar rules and word meanings. Semantic analysis involves deriving meaning and is used to generate human language. Semantic analysis is challenging because human language rules are very difficult to understand for any machines. Words and phrases take on different meanings in different contexts. So, it is very complicated to understand any query and give the proper answers according to the customers' needs for machine.

The business problem we are looking into query in text format of customer whether they have account in same bank or not and we have to find those keywords in query and classify those things in various classes to provide better response.

Solution Approach:

The current approach we are using is human interactions like as call centres' employee. But for simple and easy queries where we just have to give a selected and pre-loaded answer. So Chatbot would be very useful to tackle all these simple and repeated queries in one go and in the case of complicated queries this Chatbot will directly transfer those calls to Customer Care Executive.

To accomplish these tasks, we have to make machine learning model for customer interactions such as customer service automation through Chatbots.

Bag of Words and related algorithms are popular natural language techniques that classify phrases and document by category or type. Bag of Words simply counts how often each word appears in a document. The algorithm then compares documents and determine the topic of each document.

Word2vec is another popular natural language model. It is a two-layer neural network that classifies text to determine meaning. It converts words to mathematical 'vectors' that computers can understand. Vector conversion is required because neural networks work better with numerical inputs.

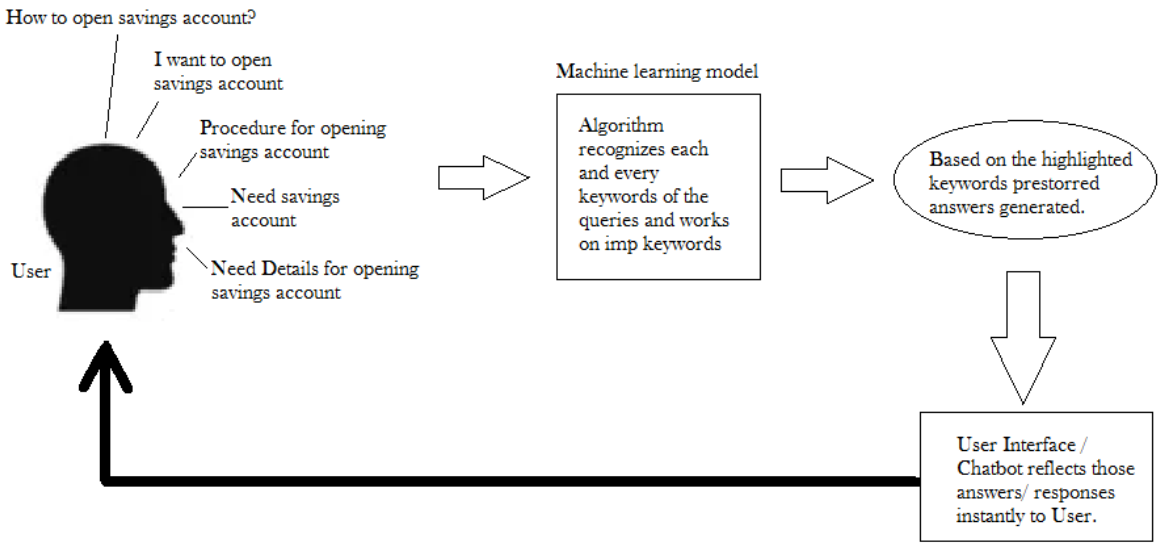
Human-computer 'conversations' can be used here:

1. We provide text input such as typing into a chatbot interface.

2. The computer converts the text into a format it understands i.e., text and words to vectors. This helps computer to make clusters using cluster algorithm and classify different words using classification algorithm.

3. Then computer figures out the datasets after processing and allocates all the keywords with proper responses in its memory location to give spontaneous response.

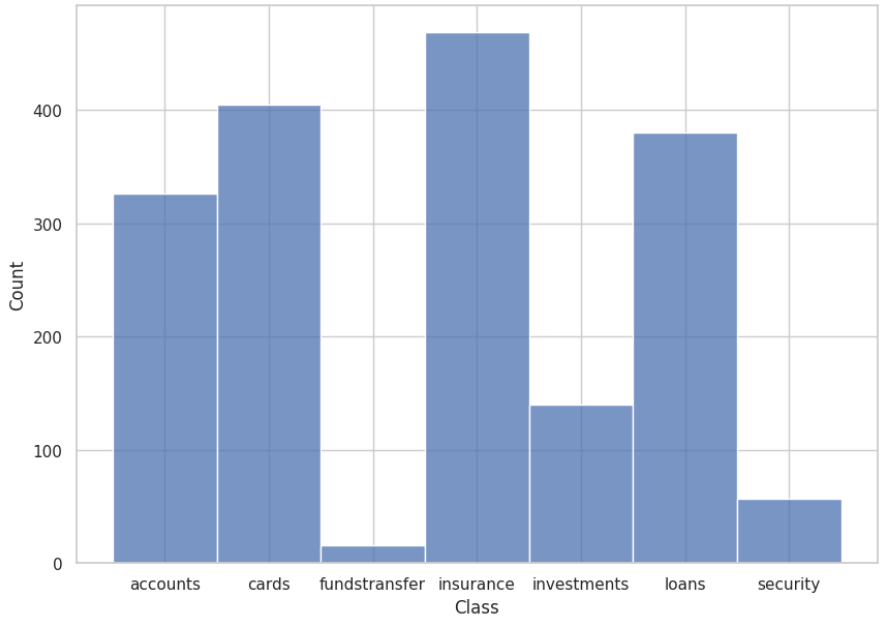
4. After determining an appropriate response computer will respond us with text output on chatbot interface.



Flow Chart

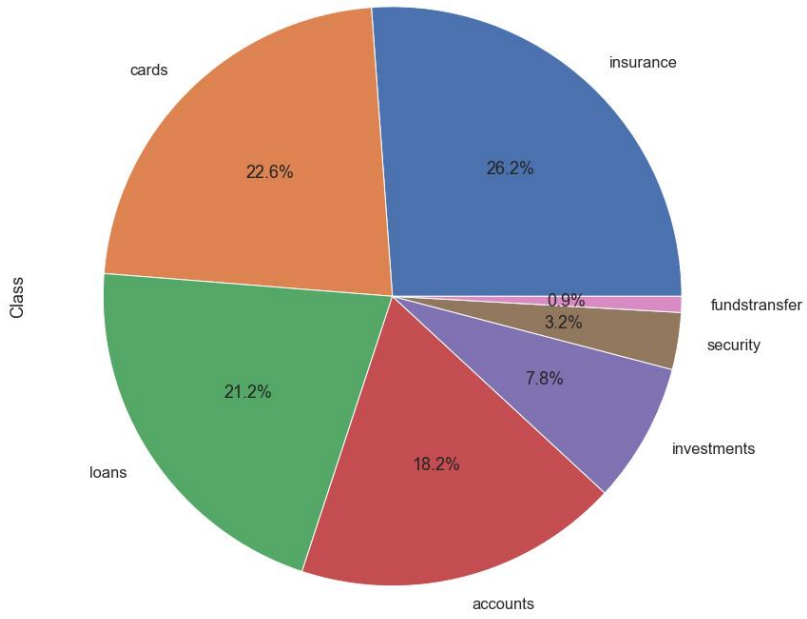
Exploratory Data Analysis:

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.



Bar Graph of various class enquiries

- Maximum Query arises in Insurance Class while least query comes under Fund Transfer Section.
- Cards, Account and Loan Enquiries are also high enough.
- Fewer Enquiries can be seen in Investments and Security.



Pie Chart of Class Enquiries in Total Enquiries

- For Insurance 26.2% enquiries, For Cards 22.6% enquiries, For Loans 21.2% enquiries and For Accounts 18.2% enquiries are there.
- Security and Fund Transfer are having combined enquiry of 4.1% only.
- More than 88% enquiries are under 4 classes out of 7 classes.

Feature Engineering:

Text Cleaning:

- Imported String than used string punctuation to remove punctuations from Questions and Answers both column using remove_punc function.
- Then we used nltk (Natural Language ToolKit) to find to find out those stop words from nltk library.
- WordNetLemmatizer is used lemmatize a word based on its context and its usage within the sentence.
- Then we appended all cleaned words in new list.

Text representation:

TF IDF (Term frequency-Inverse document frequency) Vectorizer:

The Tf-idf Vectorizer uses an in-memory vocabulary (a python dictionary) to map the most frequent words to feature indices and hence compute a word occurrence frequency (sparse) matrix.

Actually TF-IDF Vectorizer enables us to gives us a way to associate each word in a document with a number that represents how relevant each word is in that document. Then, documents with similar, relevant words will have similar vectors, which is what we are looking for in a machine learning algorithm.

Label Encoding:

Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering. So we used label encoding technique for class column.

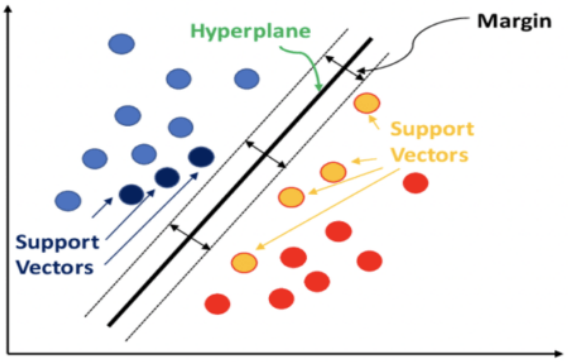
Train Test Split:

A train test split is when you split your data into a training set and a testing set. The training set is used for training the model, and the testing set is used to test your model. This allows you to train your models on the training set, and then test their accuracy on the unseen testing set.
So we used train_test_split from sklearn.model_selection to split the data into 75:25 ratios for training and testing respectively.

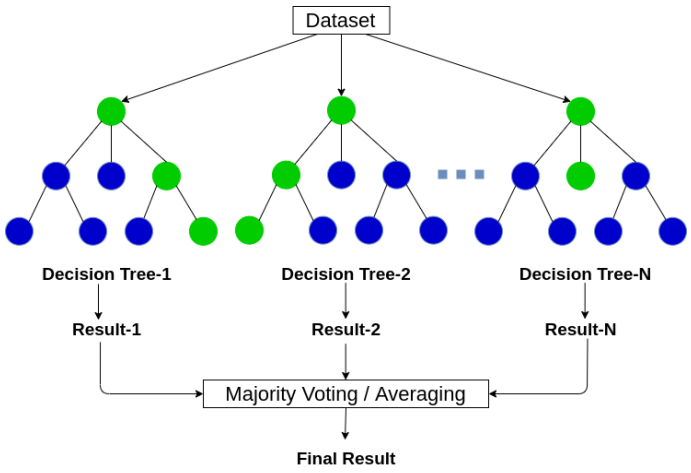
Predictive Models:

We used several machine learning models to find the best accuracy, So we used three ml models.

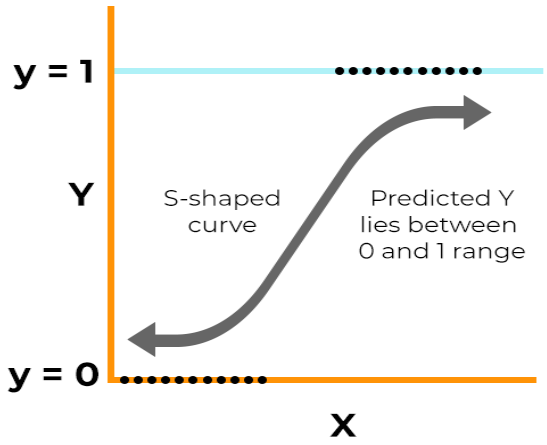
1. SVM (Support Vector Machine): It is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups.



2. Random Forest: It 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.
3.



4. Logistic Regression: It estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables.

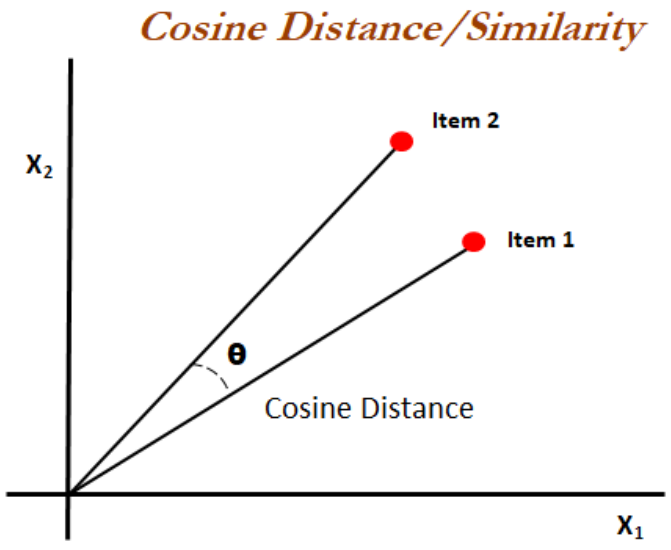


After optimising different machine learning models and tuning some hyper parameters we got that Logistic regression gives highest accuracy 88% with best parameters of C:1 and linear kernel.

Cosine similarity:

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

So we imported cosine_similarity from sklearn.metrics.pairwise to find the distance or relation between input question and answers and then detect the most similar text as the answer to the query (input question from customer).



Deploying App:

We used Streamlit to deploy this Chatbot (Virtual Assistant), fot that we used GitHub. Since streamlit is a faster way to build and share data apps.

Streamlit turn data scripts into shareable web apps in minutes, not weeks. It's all Python, open-source, and free! And once we have created an app, we can use and share this app.

We followed various steps to deploy our app:

- We created a GitHub repository- " V&S Banking Virtual Assistant", which contains the code, data, and other files.
- Copied over the code and data files.
- Added a requirements.
- Pushed the updated repository using gitbash.
- Signed up for Streamlit Cloud (Community Tier)
- Deployed our app.

=*=s=*=