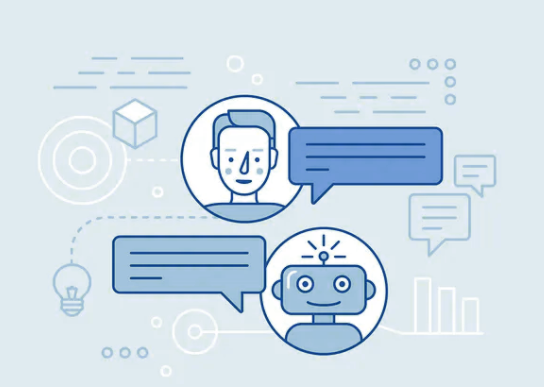
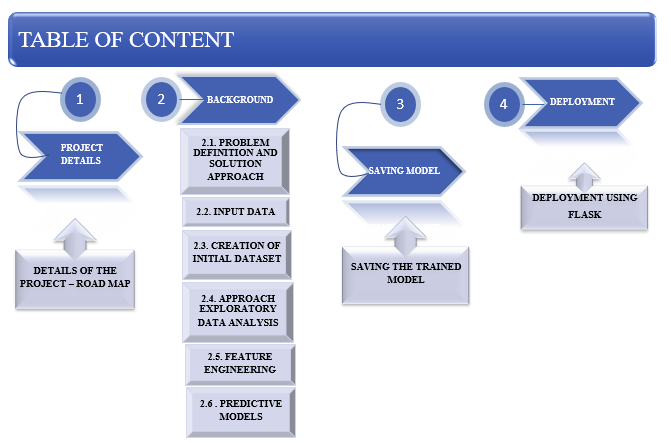
**Banking Query Classification**

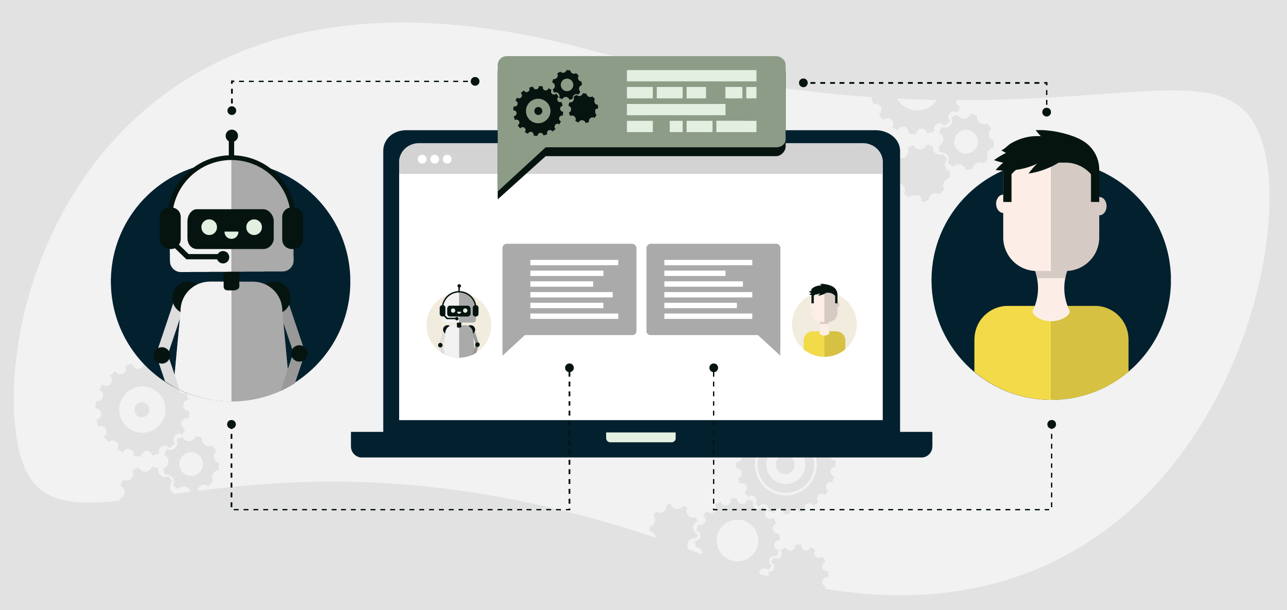


****

**1. PROJECT DETAILS**

This project is an end-to-end process of how machine learning techniques can help us to classify the intent of the query asked by a bank customer without any human intervention. This can be helpful for a chat-bot to decide what should be the reply message to the customer after understanding the intention of the machine with high level accuracy.

This is to improve a chat bot that was already launched to provide customer service in a large bank’s website using text classification in NLP (Natural Language Processing). Here we are not only going to focus on the customer side query clarification, also focusing on showing useful insights to the final user according to the trained data.



Now let’s get into the detailed project…

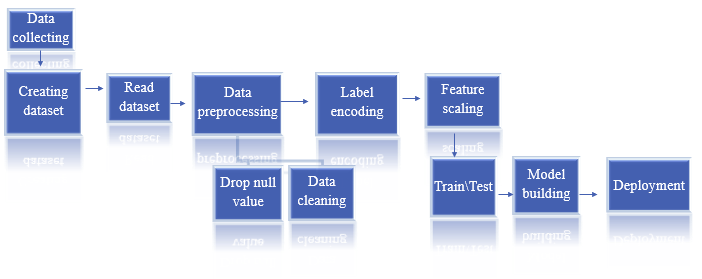
**2. BACKGROUND**

The background contains relevant techniques to do this project. In today's digital age,we can get a massive amount of data in the form of text from various sources. To extract valuable insights and accuracy, it's essential to analyze this text data using machine learning models by processing the data. Which involves finding the category of a given text document.

In this project, we aim to build a text classification model. To accomplish this task, we are going to follow the steps, including data preprocessing, data cleaning, exploratory data analysis, feature engineering and model building. In the data preprocessing step, we will perform various text cleaning techniques to remove any distortions and standardize the text data. Then, we will perform exploratory data analysis to gain insights in our balanced dataset.

After that, we will perform feature engineering, where we will transform the text data into features that can act as inputs for our machine learning model. We will explore various methods such as word count vectors, TF-IDF vectors, word embeddings, text-based or NLP-based features and topic models. In this we need to choose the most suitable one based on our needs. Next, we will label code our target variable, i.e., "enquiry" and "support," as numerical IDs to enable our machine learning model to provide predictions.

Finally, we will build our machine learning model using a supervised learning approach and evaluate its accuracy. According to the accuracy, we need to choose the best model. In conclusion, this project aims to develop a text classification model that can accurately predict whether a given text document belongs to the "enquiry" or "support" category. By doing this we can automate the process of classifying text documents, enabling organizations to improve their efficiency and production.



**2.1 PROBLEM DEFINITION AND SOLUTION APPROACH**

**Problem Definition:**

Here the Bank has a large number of customer base and also the internal associates. The website ‘s Chat-bot should be able to provide the satisfactory answer to them within its domain-based knowledge otherwise that should be forwarded to Bank’s customer service to retain the customer.

However, the current state of the chat-bot is under development and displays undesirable behaviour in its interactions with humans, which needs to be resolved.

To resolve the problem of unwanted behaviour of the chat-bot, we need to train the model with a supervised learning algorithm - text classification using NLP techniques. The chat-bot should be able to give satisfactory answers to the question that belongs to domain-specific knowledge or else need to forward it to the customer service.

**Solution Approach:**

* Why do we need machine learning for this problem?

The existing approach to do this is to use a human to get the intention & give the solution which is what the customer care support does.

But for the simple & repeated queries where we just have to give the selected answers. In this case, machines could be useful without depending on customer service staff every time. In the meantime, they can answer high-level queries. Here we need to consider the customer base also.

* Also, the chat-bot can be available 24 x 7 and automated too.

**To achieve this, the project will involve the following steps:**

* The first step in this project is to gather and pre-process the dataset. This we can get from customer queries and responses from the bank's website or using a pre-existing dataset. The next step is to clean and pre-process the text data.
* The next and most important step is to represent the text data into numerical format that the machine learning model can understand. This can be achieved using word embeddings such as Count Vector and TF-IDF Vector, which can capture the semantic properties of words and phrases.
* After text representation, the next step is to train a model using supervised machine learning text classification algorithms such as Logistic Regression, Naive Bayes, or Support Vector Machines.

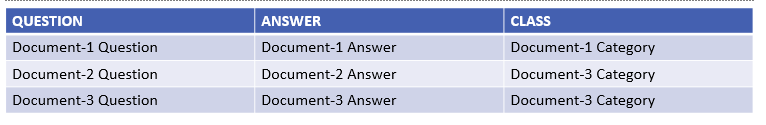
According to the accuracy, we need to select the correct model.

* The selected model will be deployed in a real-time web application using Flask that can gather data from multiple customers and classify their questions in real-time. And we can continuously improve the performance of chat bot using feedback and success rate of conversations.
* Finally, the results of the text classification model can be visualized in a real-time web application that gathers data from several customers and shows a summary of the process that we did. This web application can provide valuable insights to the bank's management and improve the overall customer experience.

**2.2 INPUT DATA:**

To develop The Bank query classification first we need to import some basic libraries. Such as pandas, Numpy and Matplotlib. Then we need to load the dataset. And we need to acquire some details of the dataset. i.e, Column name, Size of the columns and rows, type of the data and how many null values are there in the dataset.

The dataset used in this project is the Banking FAQ Raw Dataset. It consists of Questions & Answers from the Banking website. By reading the dataset we can find the following features of the dataset. It contains three columns i.e Question, Answer & Class. And 1773 rows. Customer’s query as a question and support team’s response as an answer and which category it belongs to is a class. Type of the data is an object. That is the dataset contains text format values. And also it contains no null-values.



**2.3. CREATION OF INITIAL DATASET**

This part in this project contains two steps. One is creating a data set by collecting data from different sources and merging that with existing dataset i.e, BankFAQs. And another one is cleaning the data set that includes removing any irrelevant information, such as special characters, punctuations, and stop words. By using various text preprocessing techniques, such as tokenization, stemming, and lemmatization, we need to clean the data before going to the next step.

Before creating any feature from the raw text, we must perform a cleaning process

to ensure no distortions are introduced to the model. Otherwise it can affect the performance of our machine learning model.

Now let's see these steps in detail…

**Creating initial dataset:**

The first step in the project is to create an input dataset with columns for filename, content, and category, and rows for document content. The BankFAQs Dataset is used for this project. This dataset consists of raw text data in the form of questions asked by customers on the banking website. Already we saw the details of the dataset in the above paragraph.

This part contains collecting the data from different sources. This could be obtained from customer service chat logs or publicly available datasets Or else directly from the bank's website.

Here we took only 300 rows of the other dataset to add with the existing one. For this we need to read the dataset i.e, CSVfile. Then we need to take the first 300 rows of the file and then need to merge the file with the existing one using the cancating function. Now by checking the size of the new file we can see the merged data that contains 2073 files. That should be in the format of the main dataset with the same kind of columns and details.Then we need to check for the null values. If we have any null values, then we need to fill it. In this dataset there are no null values. Next we need to perform a data cleaning process.

**Sample raw dataset:**

| NO | QUESTION | ANSWER | CLASS |
| --- | --- | --- | --- |
| 1 | What are the documents required to open a salary account, | To open a Salary Account you have to: Submit your identity proof and address proof along with your latest passport size photographs Proof of employment (Company ID Card / Appointment letter / Company Introduction through Introduction Letter OR attestation on Account opening form) Submit self-attested copies and originals for verification | accounts |
| 2 | On which mobile number will I receive the OTP | OTP will be sent to the registered mobile number (updated in Credit card system)., | security |
| 3 | Is it necessary to register the card for an OTP transaction | Prior registration is not necessary for an OTP transaction | security |
| 4 | Can a salary account have a joint applicant | Yes, parents, spouse, child or sibling can be a joint applicant to an account. The joint applicant will need to submit a valid photo ID and address proof | accounts |
| 5 | What is the procedure to add/modify or delete a nominee for all my accounts and deposits " | ,"To add/modify or delete a nominee you will need to fill the nomination form and submit it to your nearest branch. The branch will assist you after the form has been submitted by you.For more details, Contact Us | ,accounts |
| 6 | How to get NOC for Vehicle loan | No, each IVR password can be used only for a maximum of 3 attempts (including decline attempts) within the specified validity period. For further transaction attempts, a new IVR password must be generated | security |
| 7 | Can I transfer my Current Account from one branch to another | Yes, You can transfer by submitting the following document: Request form for account transfer at bank branch. In case a bearer submits the request on your behalf, a bearer authorisation along with KYC documents(self attested copy and original for validation) of the bearer, i.e. Identity and Signature Proof like PAN Card, Aadhar Card will also need to be provided. | loans |
| 8 | How do I renew my Accidental Protection Plan - Hospital Cash policy | Yes, Current Accounts can be transferred from one branch to another. However, there are certain restrictions. Please visit your nearest branch for details. | accounts |
| 9 | How to change existing fixed deposit account tenure | We will send you renewal reminders. The number of days in advance the reminder is sent depends on the pay plan you have opted for. Pay plan break up: Cheque – 45 days Credit Card – 60 days View more | insurance |
| 10 | Does the Accidental Protection Plan - Hospital Cash policy cover natural death and daily sickness as well | We regret to inform you that tenure selected for the fixed deposit account cannot be changed once the account is opened. In this case we suggest that you can close your existing fixed deposit account and open a new account with a desired tenure. | accounts |
| 11 | Is there any maturity benefit under this product | This policy only covers accidental death. | insurance |
| 12 | Why is the principal amount and the fixed deposit maturity amount of my Fixed Deposit the same | There is no maturity benefit available under this product. | insurance |
| 13 | How do I repay my Business Loan | You can pay the loan in equal monthly instalments (EMIs). The loan will be paid through post-dated cheques. You can also pay through Electronic Clearing System (ECS ) or a standing instruction to debit your HDFC Bank account with the EMI amount. | Loan |



**Data cleaning:**

Data cleaning is the process of identifying and correcting errors, inconsistencies, and missing values in the dataset to improve the quality and accuracy of the data. It is an important step in the data preprocessing of a machine learning project, as the quality of the data has a significant impact on the performance of the machine learning model.

**Handling Text Data:** Text data can contain a lot of noise, such as duplicate values, stop words, punctuation marks and special characters. We need to preprocess text data by removing stop words, punctuation marks, and special characters, and applying techniques such as tokenization, lemmatization, and stemming to clean the dataset.

Let's see these steps one by one.

**Removing Duplicates:** Duplicate data can occur due to data entry errors or system errors. Duplicate data can skew the distribution of the data and cause the model to overfit. We need to identify and remove duplicate data to improve the quality of the dataset. After removing duplicates, Our dataset contains 1581 total number of rows. There are 274(from the Main dataset BankFAQs)+77(From the collected dataset) duplicate files we have removed from our dataset.

**New Line Character Removal:** New line characters, also known as line breaks(\n) are removed from text data to ensure that the text is in a single continuous string format.

**Lower Case/Upper case:** Lower casing/Upper casing is the process of converting all the text data to the same type of case. This is done to ensure that words with the same spelling but different cases are treated as the same. There is no different in that. Here we have changed the text into Lower case.

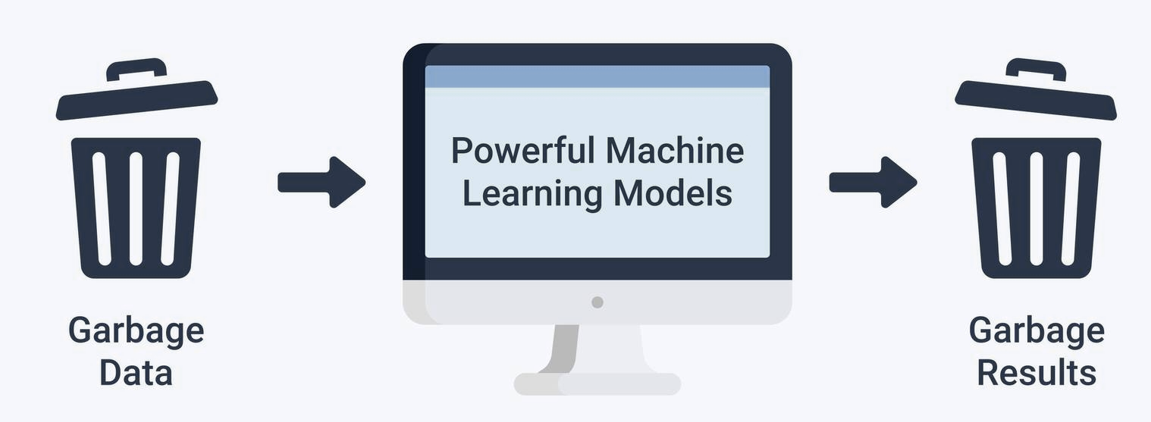
**Possessive Pronouns:** Possessive pronouns such as "his", "her", and "their" are removed from text data during preprocessing, as they do not add any significant meaning to the text.

**Stop Words:** Stop words are commonly used words in a language that are not considered informative for text analysis. Examples of stop words include "the", "a", "an", "and", and "in". Stop words are removed from text data to reduce noise and improve the efficiency of text analysis.

**Tokenization:** Tokenization is the process of breaking down text data into smaller units called tokens. Tokens can be individual words, phrases, or even sentences. Tokenization is a fundamental step in text analysis, as it allows the computer to understand the structure of the textPunctuation Removal: Punctuation marks such as periods, commas, and exclamation marks are removed from text data to reduce noise and make the text more uniform.

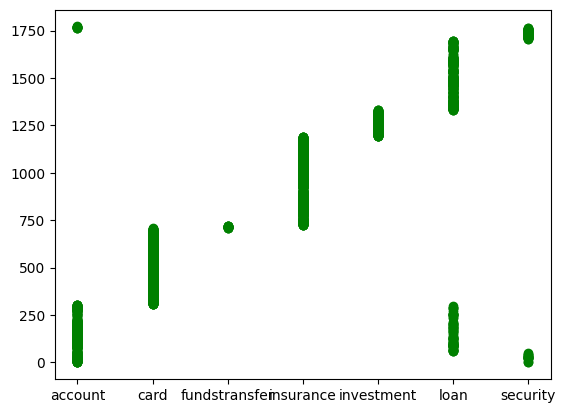
**Lemmatization:** Lemmatization is the process of converting words into their base form, or lemma. For example, the lemma of the word "running" is "run". Lemmatization is used to reduce the dimensionality of text data and to ensure that words with similar meanings are treated as the same.

**Stemming:** Stemming is the process of reducing words to their root form by removing suffixes and prefixes. For example, the stem of the word "running" is "run". Stemming is a simpler version of lemmatization and is often used in applications where speed is a priority.



By doing these all the above steps we have cleaned the data. And it can be used for the next step.

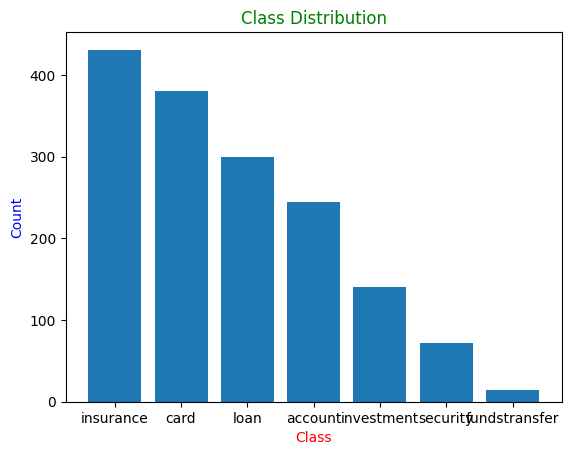
**2.4. EXPLORATORY DATA ANALYSIS**



Before creating the machine learning model, it is essential to perform exploratory data analysis (EDA) to gain insights from the data. It helps to understand the dataset's characteristics, such as the distribution of data, outliers, and class imbalance. This step involves visualizing the data using histograms, bar plots, box plots, and scatter plots.

The output of the class distribution gives us insights into the balance or imbalance of our dataset. In this case, we can see that the categories are not evenly distributed, with some categories having significantly fewer instances than others. This may have implications on the performance of our classification model, as the model may be biased towards the categories with more instances.

Plotting the distribution of the 'Class' column helps us to understand the balance of the dataset. If the number of questions in each class is roughly equal, then we have a balanced dataset, which is good for training a classification model. If one class has significantly more questions than the others, then we have an imbalanced dataset, which can cause problems when training a classification model. We may need to use techniques such as oversampling or undersampling to balance the dataset or use different evaluation metrics other than accuracy.

By analyzing the distribution of the 'Class' column, we are getting the below data,

**Class** **No.of Q&A**

accounts 315

cards 403

funds transfer 14

insurance 469

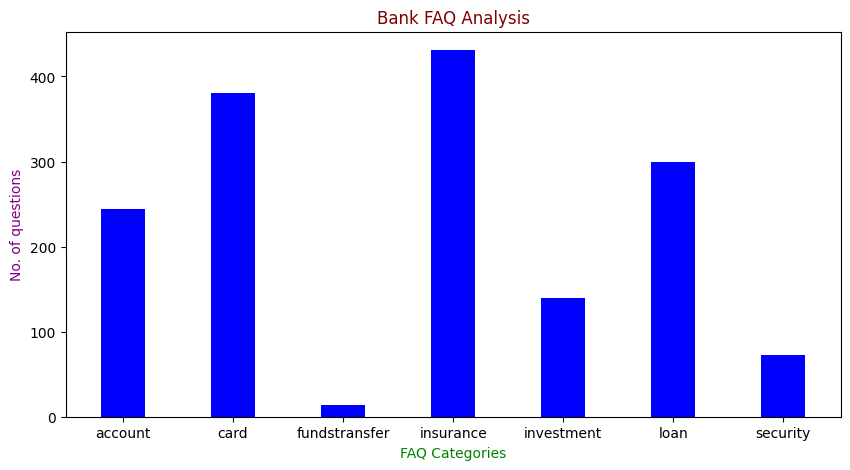
investments 140

loans 618

security 114

As per the data, we have 7 types of categories in our ‘Class’ column. i.e, Accounts, Cards, Funds transfer, Insurance, Investments, Loans and Security. These categories indicate that the customers asked queries related to these topics. And the No.of Questions asked by the customer as per the categories that we got are not equal in this dataset. For example, if we observe that the 'Loans' class has significantly more questions than other classes and the ‘Funds transfer’ category has the lowest number of questions. Then we have an imbalanced dataset. One more thing here we can see that the customers are looking for more answers on loan related queries and then Insurance, Card & Account related queries according to this dataset. We can consider undersampling the 'Loans' class or oversampling the other classes to balance the dataset. We can also consider using metrics such as precision, recall, and F1-score to evaluate the performance of our classification model instead of accuracy

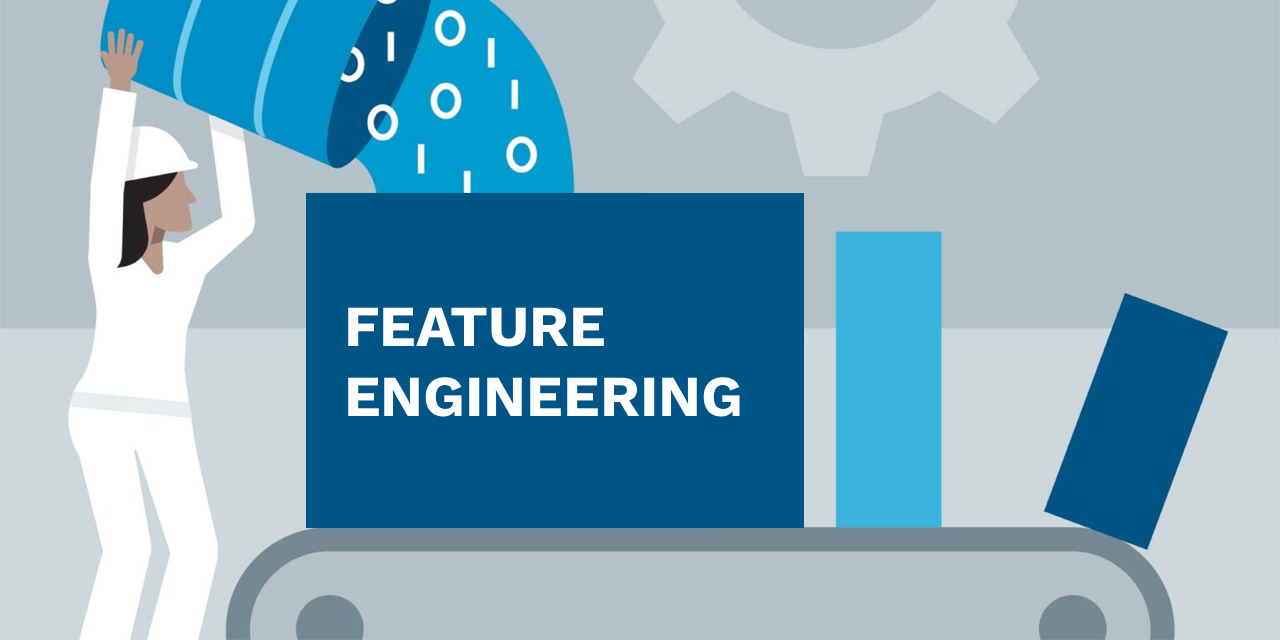
In addition, we can also perform further exploratory data analysis on the text data itself, such as creating word clouds or analyzing the most common words used in each category. This can give us insights into the language and vocabulary used in each category, which we can perform in our feature engineering and preprocessing steps.



**2.5. FEATURE ENGINEERING**

Feature engineering is the process of transforming data into features to act as inputs for machine learning models such that good quality features help in improving the model performance.

When dealing with text data, there are several ways of obtaining features that represent the data. Such as Text representation, Label coding, Text cleaning, etc., Following are some of the most common methods and then choose the most suitable for our needs.



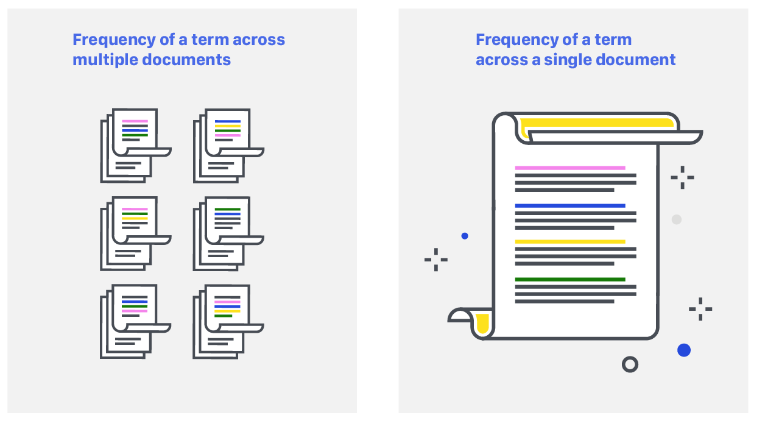
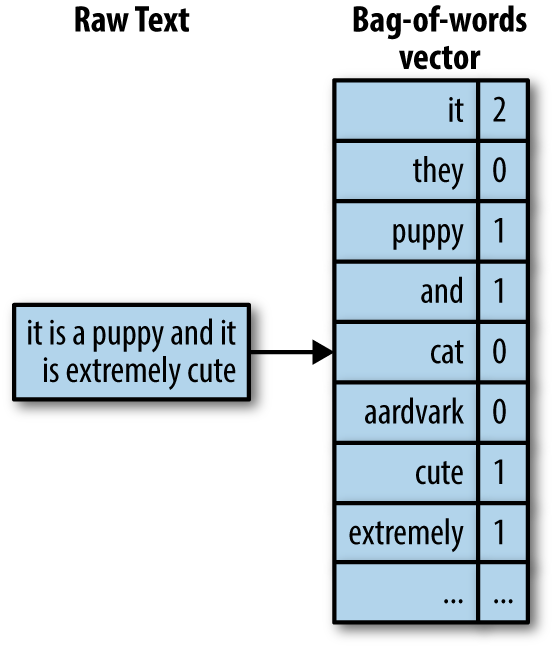
**2.5.1. Text representation**

In order to represent our text, every row of the dataset will be a single document of the corpus. Here the columns are considered as the features. That will be different depending on which feature creation method we choose. For example Word Count Vectors, TF–IDF Vectors, Word Embeddings, Text based or NLP based features & Topic Models are the most commonly used methods. Here we have used the TF-IDF Vectorizer for this project.

TF-IDF Vectorizer():

TF-IDF stands for Term Frequency-Inverse Document Frequency.

The TF-IDF vectorization process involves the following steps:

* Count the number of times each word appears in a document (term frequency).
* Calculate the frequency of each word across all documents (document frequency).
* Compute the TF-IDF value for each word in each document.
* The term frequency is simply the number of times a word appears in a document divided by the total number of words in the document. The document frequency is the number of documents that contain the word.
* The TF-IDF value is calculated by multiplying the term frequency

Here the TF-IDF Vectorizer() function from Sklearn is used to convert the tokenized 'Question' column into a sparse matrix of TF-IDF vectors. This step transforms the text data into a numerical representation that can be used by the machine learning algorithm.

**2.5.2. Text cleaning:**

We already saw these steps in the previous chapters. This function is defined to clean the text data by removing newline characters, punctuation, and numbers. Then, this function is applied to the 'Question' and 'Answer' columns of the DataFrame using the Pandas.

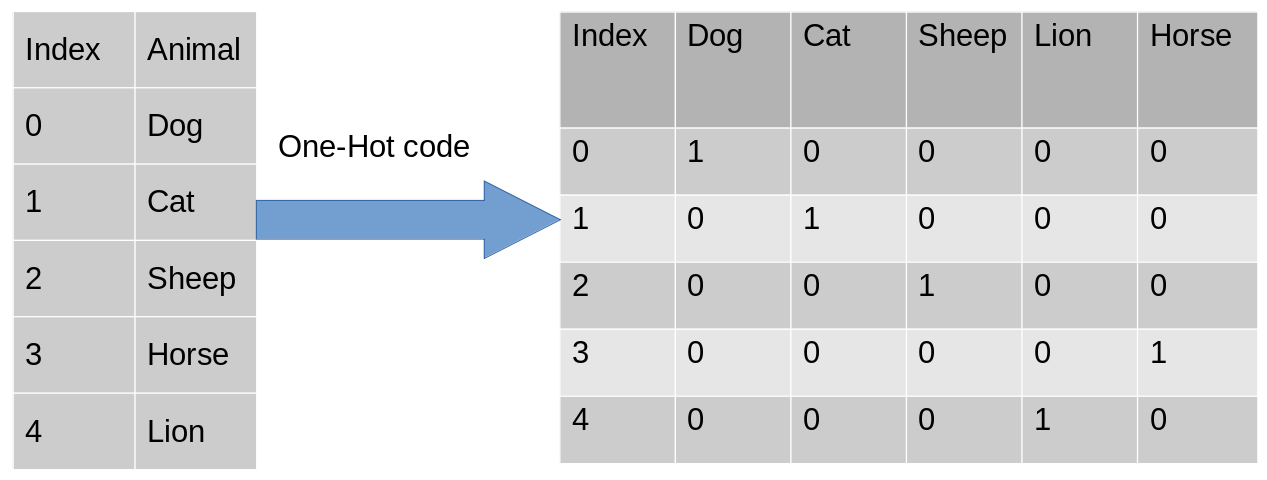
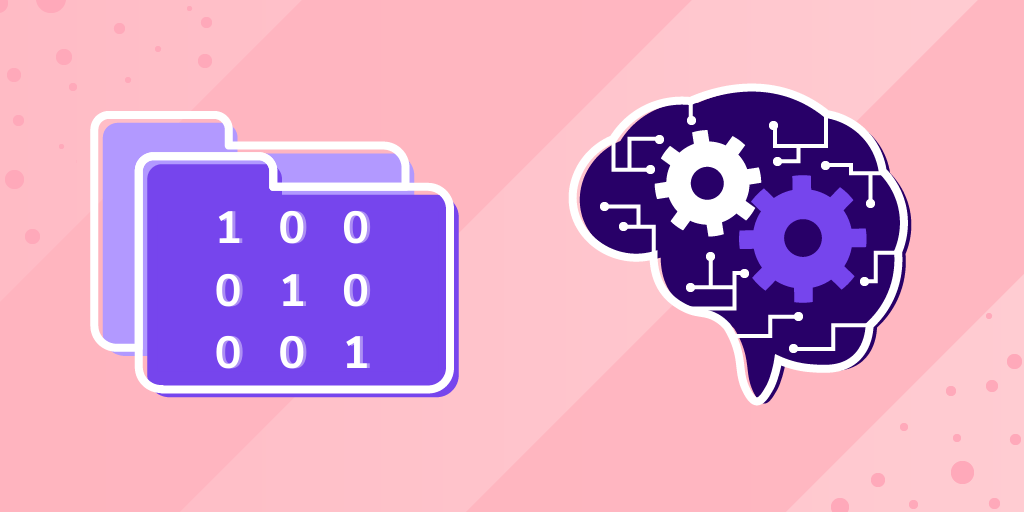
This function takes a text input and performs some pre-processing steps that we mentioned above such as removing punctuations, numbers, and converting all characters to lowercase, Possessive pronouns removal, Stemming or Lemmatization & Stop words removal.

Tokenization:

Tokenization is an important step in text preprocessing. Word tokenization is the process of splitting a sentence into individual words or tokens. Here The 'Question' column is tokenized using the NLTK word\_tokenize() function, which splits the text into individual words.

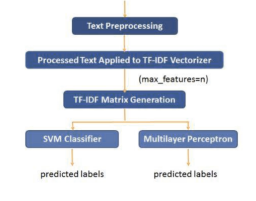
The resulting tokens are then passed to the TF-IDF Vectorizer for feature engineering.

**2.5.3. Label coding**



Label encoding is a process of converting categorical labels into numerical values that can be used as inputs for machine learning algorithms.

In this project, label encoding is used to convert the categorical class labels into numerical values so that they can be used as input for the machine learning algorithm. The Label Encoder function from scikit-learn library is used for this purpose. It assigns a unique integer value to each class label in the 'Class' column of the dataframe.

**2.5.4. Train — test split**

Train-test split is a technique used in machine learning to evaluate the performance of a model. It involves splitting a dataset into two separate sets: one for training the model and the another one for testing the model. The training set is used to fit the model to the data, while the testing set is used to evaluate how well the model can predict the outcomes of unseen data. The purpose of this split is to check how well the model can generalize to new, unseen data.

It helps in detecting overfitting, which is when a model performs well on the training data but poorly on the testing data. By splitting the data, we can avoid overfitting and ensure that our model is not memorizing the training data.

Here 80% of the data used to train our machine learning model, and 20% of the data is used to test the model.

**2.6. Predictive Models**

****

The first step is to import the necessary modules, which in this case are CountVectorizer and train\_test\_split from Scikit-learn. Next, we load the data from Columns. For this project, we only take the Question and class columns, which we extract into the X and y variables, respectively.

Then split the data into training and testing sets using the train\_test\_split() method from Scikit-learn. The test\_size parameter specifies the proportion of the data to use for testing (in this case, 20%), and the random\_state parameter is used to ensure that the data is split in a reproducible manner.

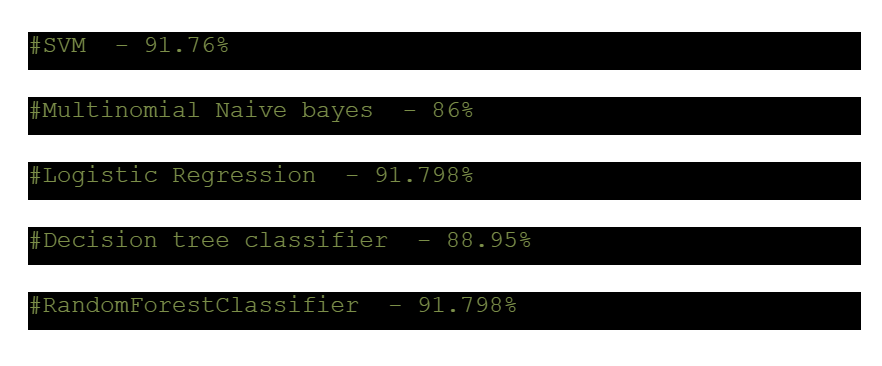
The next step is to create an instance of CountVectorizer and use it to transform the training and testing data. CountVectorizer is a Scikit-learn class that converts a collection of text documents into a matrix of token counts. In this case, we use it to convert the Question column into a matrix of token counts. The fit\_transform() method is used to fit the CountVectorizer to the training data and transform it, while the transform() method is used to transform the testing data using the fitted CountVectorizer.

Next, we can apply any of the classification algorithms to the transformed data. In this project we used 5 different algorithms: logistic regression, Naive Bayes, Decision tree classifier, Random forest classifier and support vector machine (SVM).

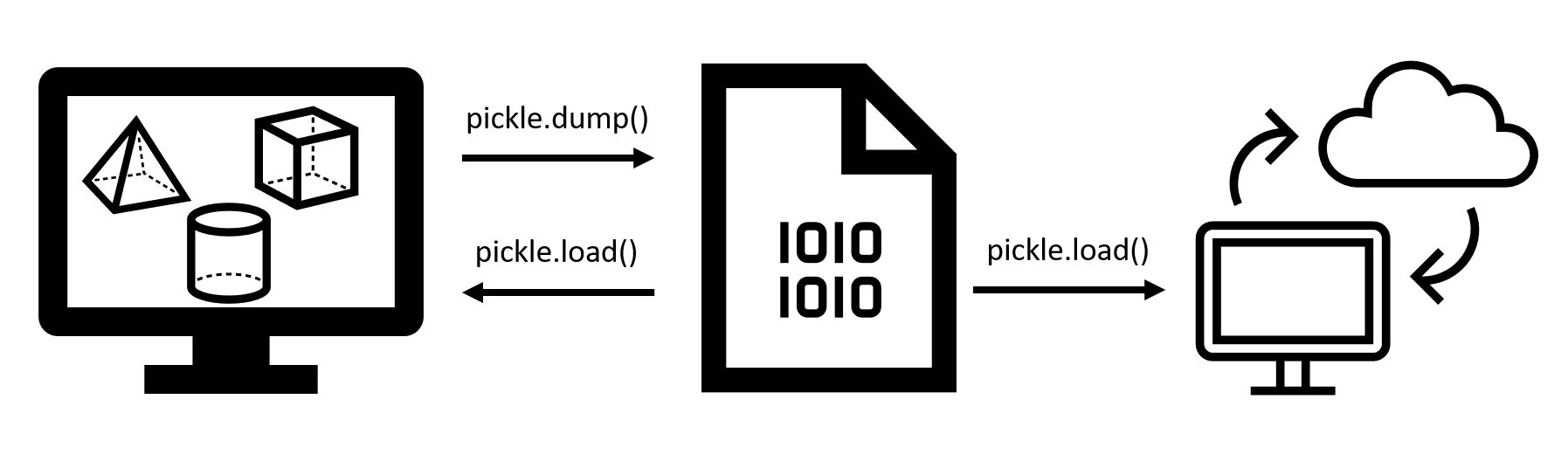
Each algorithm is initiated as an object of the respective class (i.e, LogisticRegression(), MultinomialNB(), and SVC()), and then fitted to the training data using the fit() method. The accuracy of each algorithm is then computed using the score() method, which returns the mean accuracy on the given test data and labels.

After performing the predictive part here Linear regression & Random forest classifier have same accuracy i.e, 91.79%. For further project we are going to take the Linear Regression algorithm for future use.

Accuracy in each model:



**3. SAVING THE MODEL**

****

After the predictive model we need to save the best performing model for the further process. Here we have stored the Linear Regression model as the best model using pickle.

Then we train the best model on the full dataset. Finally, we save the best model as a file using pickle. We specify the filename as best\_model.pkl and open the file in write binary mode ('wb'). Then we use the pickle.dump() function to serialize the best\_model object and write it to the file.

Later, we can use this to load the saved model object from the file and use it for making predictions.

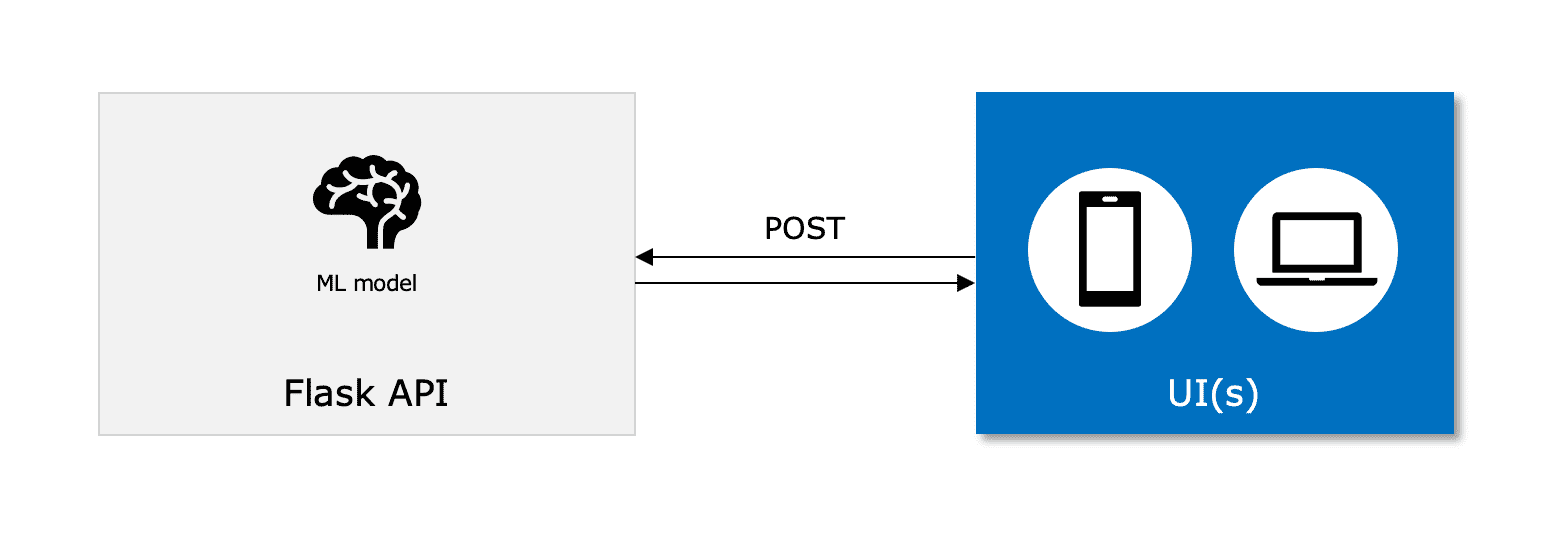
**4. DEPLOYMENT USING FLASK**

1. **Creating a flask application which will accept the text and return the category of text.**



After saving the predicted model using pickle, we need to create the Flask framework to deploy the model. For that first we need to load the saved machine learning model using the pickle.load() method. Then we need to create a Flask app and create a route for the endpoint that will accept a JSON payload with “Question” and text fields. When a user sends a POST request to this endpoint with “Question” and “test” fields in the JSON payload, the predict() function will use the saved model to predict the answer to the input text. The predicted answer, along with the original question will be returned as a JSON response.

**2. Creating the flask API**

****

To run the Flask API, We need to save the code in app.py file. Then the API will start running on <http://127.0.0.1:5000/> in this URL. We can send a POST request to the predict endpoint using a tool like postman or cURL, Or else by creating a simple HTML form that gives the input text as a POST request to the predict endpoint.

Using this we can predict the answer for the given question.