STUDENT DETAILS

Name: **Ayush Singh**

AICTE Internship StudentRegistration ID) : **STU64d3642e53aa11691575342**

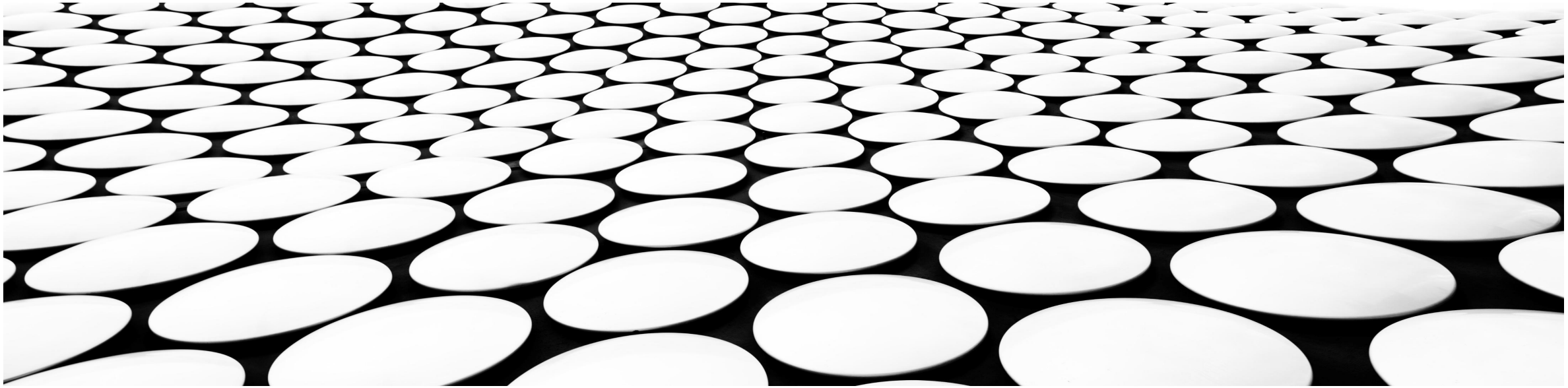
Email Id: **singhayush9179@gmail.com**

College Name: **Kanpur Institute of Technology, Kanpur**

College State: **Uttar Pradesh**

Domain: **Computer Science & Engineering**

Start Date & End Date: **06 july 2024 to 06 Aug 2024**



SENTIMENT ANALYSIS OF RESTAURANT REVIEWS

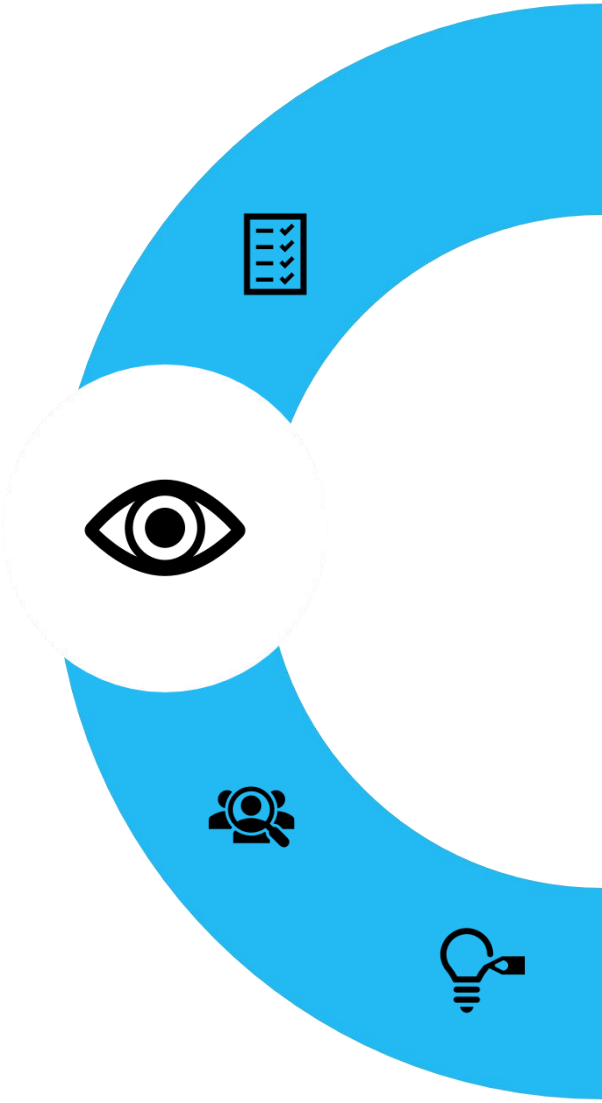
In today's world social media have a huge impact on our lives whether its personal life or professional life. People are tend o make decisions based on what they see and hear on social media nowadays which is making a huge impact on businesses. So, the industries are tend to find ways to analyse and interpret insights from these information present over the social media, blogs and other internet platforms. One of the ways to utilise these data is analysing the sentiment of a particular user from the reviews he/she does over the internet on websites or blog posts. So, we will

AGENDA

As the influence of the social media and internet is increasing in ones life there arises a need to analyse the valuable insights from the data scattered over the internet in various forms. Business like restaurants can use this data to find out how they are performing and find out what are the reviews of their customers and know what their customers are really thinking of them. Our goal is to create a AI algorithm that finds the sentiment of a user from its review and distinguish it between two categories either positive or negative. This can be really beneficial for the restaurants to gain insights and also to recognise which dish is performing best, what are the areas to work on, what are the negative points. Using this knowledge they can upgrade themselves so that they

PROJECT OVERVIEW

Restaurant reviews are a valuable source of information for potential customers, but they can be time-consuming and difficult to analyse manually. This project aims to develop a sentiment analysis model that can automatically classify restaurant reviews as positive, negative, or neutral. The model will be developed using a supervised machine learning approach, and it will be evaluated using accuracy, precision, and recall. This project has the potential to benefit both restaurant owners and customers. Restaurant owners can use the model to identify areas where they can improve their service or food quality. Customers can use the model to quickly and easily find restaurants that are likely to meet their needs. The project will be completed in three phases:



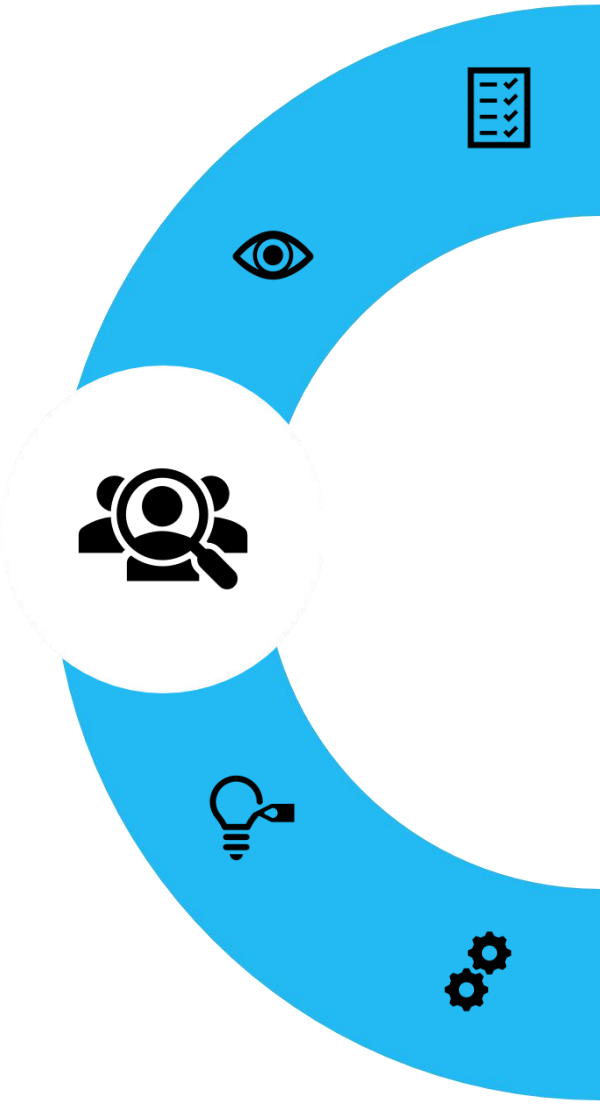
USERS

The users of the sentiment analysis of restaurant reviews project are:

- Restaurant owners: Restaurant owners can use the model to identify areas where they can improve their service or food quality. For example, if the model identifies that many negative reviews mention slow service, the restaurant owner can take steps to improve the speed of service.
- Customers: Customers can use the model to quickly and easily find restaurants that are likely to meet their needs. For example, if a customer is looking for a restaurant with good food, they can use the model to find restaurants that have mostly positive reviews.

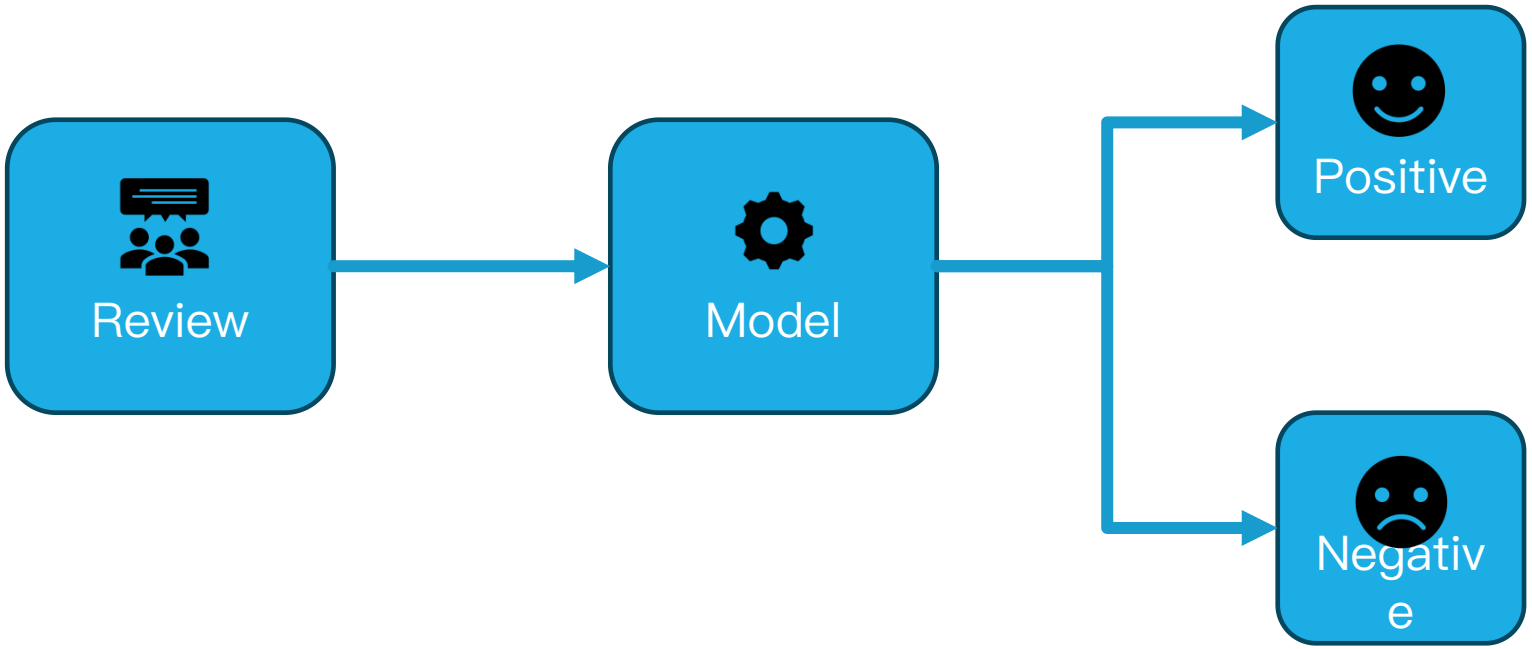
Here is one examples of how the project can help users:

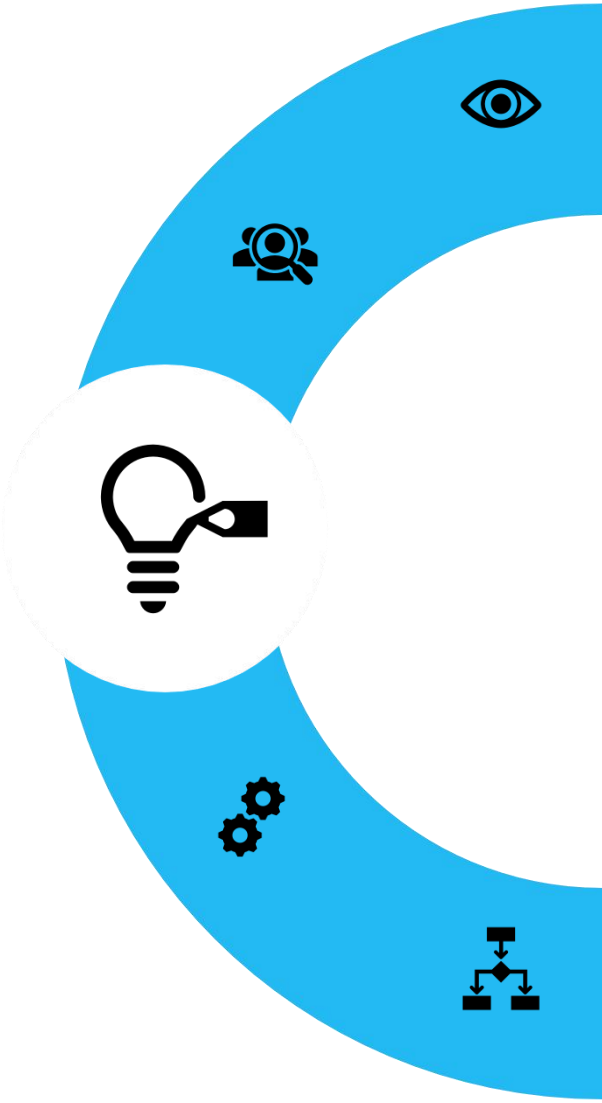
A **restaurant owner** can use the model to identify which dishes are most popular with customers and which dishes need improvement. This information can be used to update the menu or make changes to the way dishes are prepared.



SOLUTION

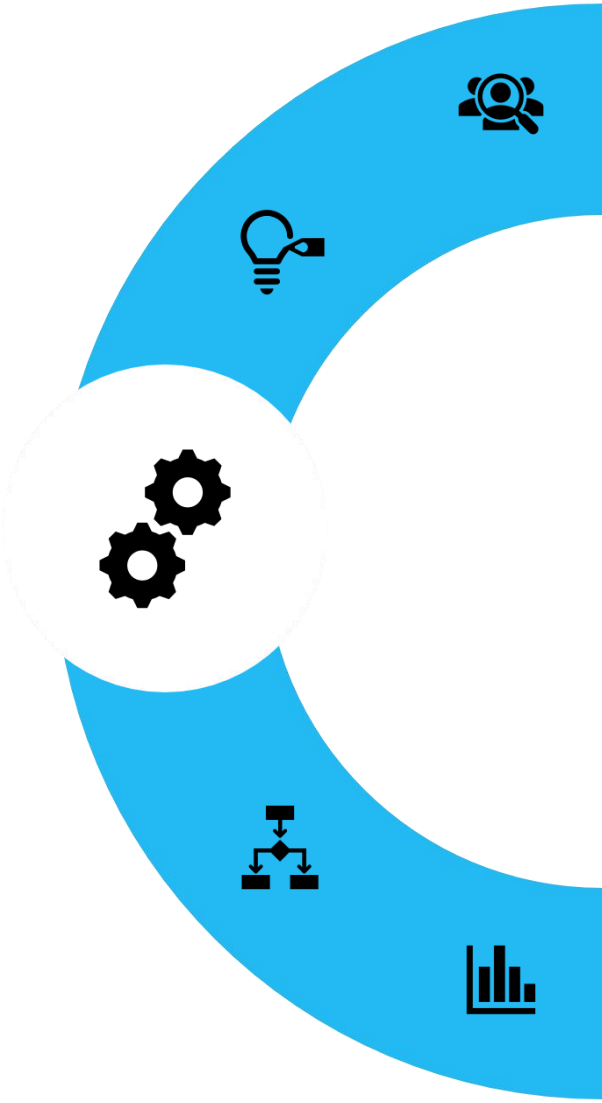
The solution contains a classification model that helps the user to understand whether a review is positive or negative. The user simply has to provide a review or a file or more than one review, the system will accurately give the output result as positive or negative review. This output then can be used to analyse the performance of the restaurant and used it to analyse the areas for improvement and what their customers expect from them.





WOW IN SYSTEM

- Accurate results
- Accurate classification
- Higher accuracy
- Flexible
- Scalable

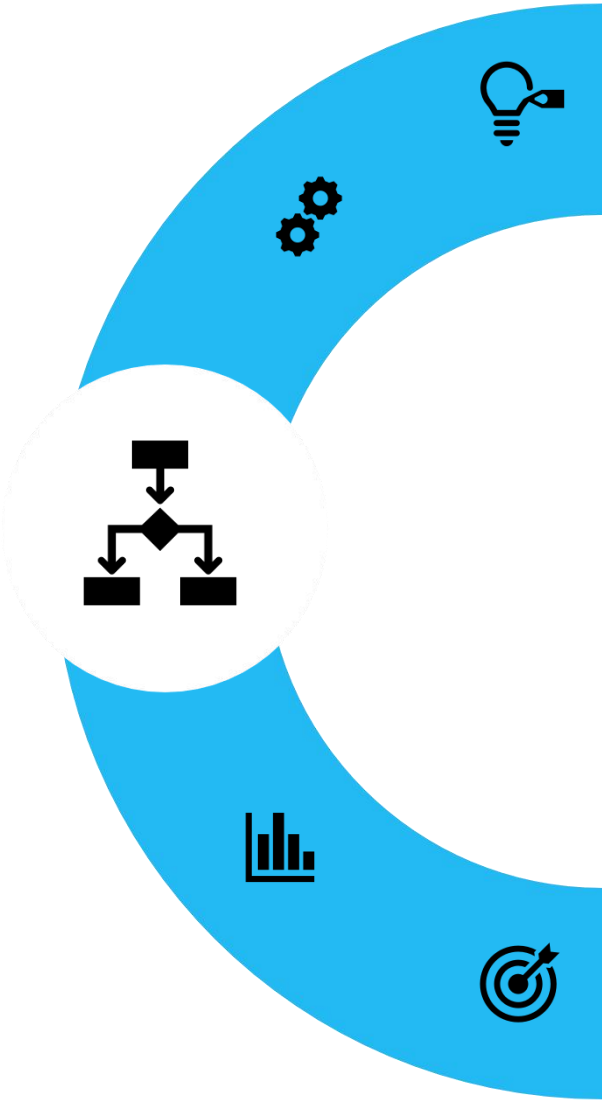


MODELLING

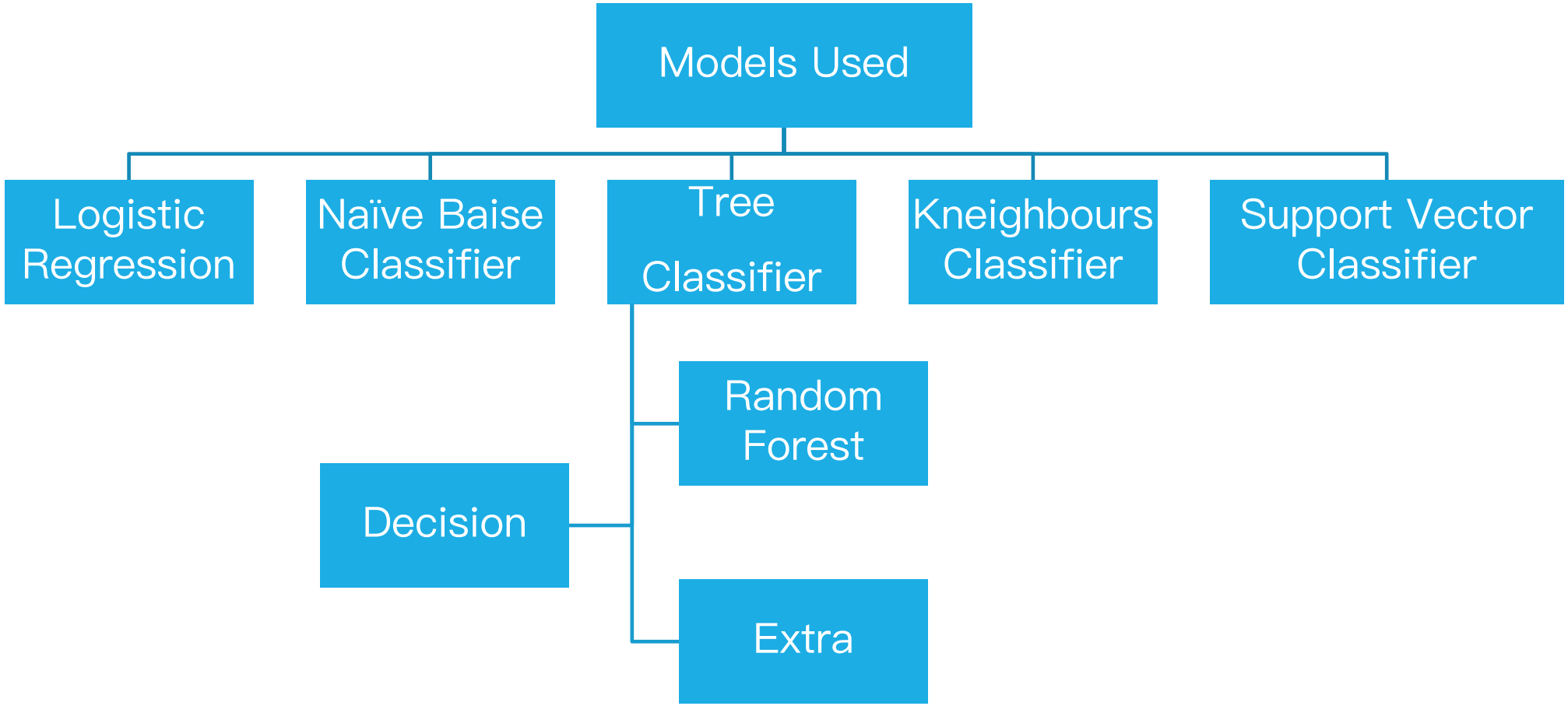
The algorithm was developed in four stages

1. Data Cleaning : Before using data for model training its essential to clean the data. Cleaning data involves steps like making data into correct format for used for training, removing blanks, and removing unwanted data. In this case we have stopwords (the words that doesn't contribute in meaning of sentence) so we will remove them first before training the model.

```
[ ]  corpus = []

     for i in range(0,1000):

       review = re.sub(pattern='[^a-zA-Z]',repl=' ',string=data['Review'][i])
       review = review.lower()
       review_words = review.split()
       review_words = [ word for word in review_words if not word in set(stopwords.words('english'))]
       review = [ps().stem(word) for word in review_words]
       review = ' '.join(review)
       corpus.append(review)

     corpus[:1000]
```



MODELLING


Models Used
Logistic Regression
Naïve Baise Classifier
Tree Classifier
Kneighbours Classifier
Support Vector Classifier
Random Forest
Decision
Extra

MODELLING

2. Model Training & Evaluation : After the cleaning is completed we train the model using the data. The training corpus is divided into two sections of 70% and 30%. The 70% data is used for model training and 30% data is used for model testing. After the training of model is done the model is evaluated using three performance testing parameters accuracy, precision and recall.

```
from sklearn.naive_bayes import MultinomialNB
classifier =  MultinomialNB()
classifier.fit(X_train,y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

score1 = accuracy_score(y_test,y_predict)
score2 = precision_score(y_test,y_predict)
score3 = recall_score(y_test,y_predict)

print(f"The accuracy score for the model is {score1*100}%")
print(f"The precision score for the model is {score2*100}%")
print(f"The recall score for the model is {score3*100}%")
```

```
The accuracy score for the model is 76.5%
The precision score for the model is 76.41509433962264%
The recall score for the model is 78.64077669902912%
```

MODELLING

3. Hyperparameter Tuning: After training we have to tune the hyperparameter to find the best model.

```
best_score = 0.0
alp = 0.0
for i in np.arange (0.1,1.1,0.1):
  classifier = MultinomialNB(alpha = i)
  classifier.fit(X_train,y_train)
  y_predict = classifier.predict(X_test)
  score = accuracy_score(y_test,y_predict)
  if score > best_score:
    best_score = score
    alp = i

print("Best accuracy Score is "+str(best_score*100)+" for alpha "+str(alp))
```

```
Best accuracy Score is 78.5 for alpha 0.2
```

MODELLING

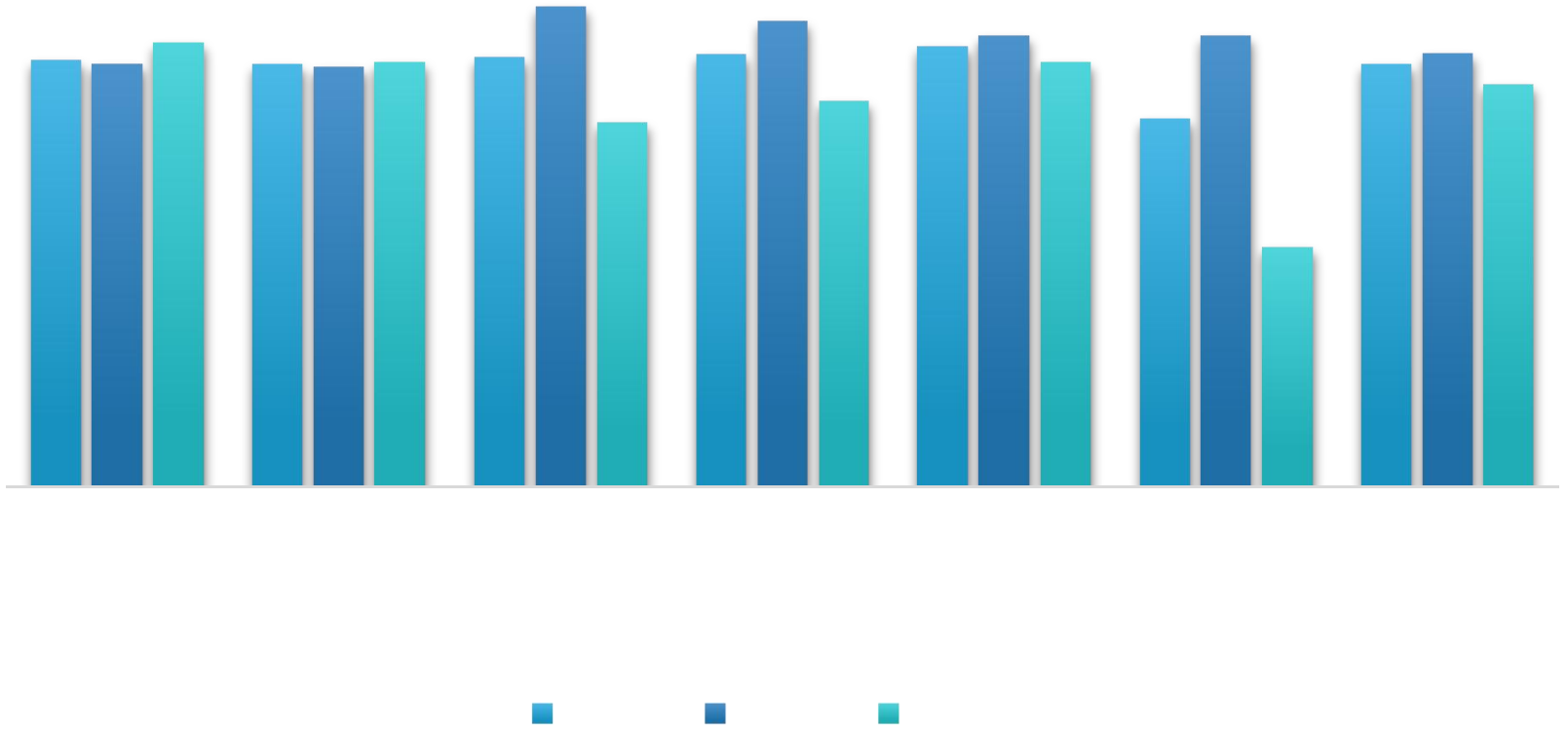
3. Model Testing : Testing the model with five different inputs.

```
def predict_sentiment(sample_review):
  sample_review = re.sub(pattern='[^a-zA-Z]',repl=' ',string = sample_review)
  sample_review = sample_review.lower()
  sample_review_words = sample_review.split()
  sample_review_words = [ word for word in sample_review_words if not word in set(stopwords.words('english'))]
  final_review = [ps().stem(word) for word in sample_review_words]
  final_review = ' '.join(final_review)

  temp = cv.transform([final_review]).toarray()
  return classifier.predict(temp)
```

```
sample_review = ['The food is really wonderful',
                 'The food is bad and service is also not good',
                 'Not tasty and the texture was just nasty',
                 'Highly recommended',
                 'The worst was the salmon sashimi']
i=1
for sample in sample_review:
  if predict_sentiment(sample):
    print(f'The review {i} is Positive')
  else:
    print(f'The review {i} is Negative')
  i+=1
```

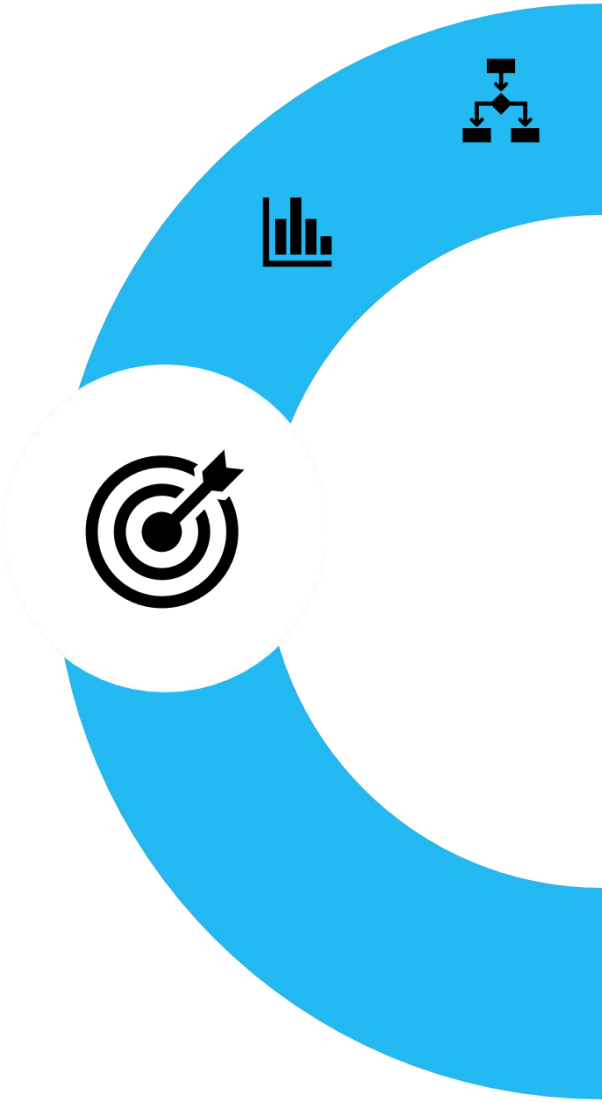
```
The review 1 is Positive
The review 2 is Negative
The review 3 is Negative
The review 4 is Positive
The review 5 is Negative
```

MODEL PERFORMANCE



RESULTS

- Extra Trees Classifier Algorithm is the best algorithm amongst all the algorithm used for this dataset with the highest accuracy score of 81%
- The model was accurate to predict the results for different inputs.
- KNeighbours Classifier Algorithm is the worst algorithm amongst all the algorithm used for this dataset with the lowest accuracy score of 67.5%.



LINKS


Google Collab
Click on shape to open link


GitHub
Click on shape to open link

THANK YOU