KCC Query Analysis

CS685: Final Project Report [Group 1]

Aniket Sanghi
Roll no: 170110
sanghi@iitk.ac.in

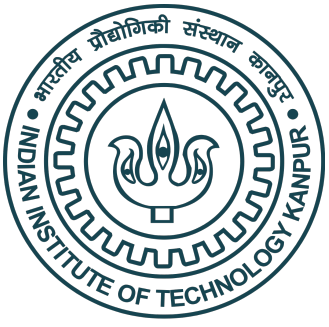
Neil Rajiv Shirude
Roll no: 170429
neilrs@iitk.ac.in

Paramveer Raol
Roll no: 170459
paramvir@iitk.ac.in

Sarthak Singhal
Roll no: 170635
ssinghal@iitk.ac.in

Aman Kumar Thakur
Roll no: 170083
amankt@iitk.ac.in

6 December 2020



Contents

1 Abstract **3**

2 Problem Statement **3**

3 Introduction and Motivation **3**

4 Dataset used **3**

5 Methodology **4**
5.1 Scraping . . . 4
5.2 Data Preprocessing . . . 4
5.2.1 Narrowing down data . . . 4
5.2.2 Inter-Dataset duplication . . . 4
5.2.3 Intra-Dataset duplication . . . 4
5.2.4 Outliers . . . 4
5.3 FAQ Generation . . . 5
5.3.1 Data Cleaning . . . 5
5.3.2 FAQ Methodology . . . 5
5.4 Multi-Class Classification . . . 6
5.4.1 Classification Pipeline . . . 6
5.4.2 Dimensionality and Numerosity Reduction . . . 6
5.4.3 Feature Extraction . . . 7
5.4.4 Classification Models . . . 7
5.5 Government Schemes Queries . . . 7
5.5.1 General Data Pre-processing . . . 7
5.5.2 Getting Scheme Tags . . . 8
5.5.3 Finding key words for scheme specific analysis . . . 8
5.6 Plant Protection analysis . . . 9
5.6.1 General analysis . . . 9
5.6.2 Per crop disease analysis . . . 9
5.6.3 Statewise analysis . . . 10
5.6.4 Monthwise analysis . . . 10
5.7 Analysis of weather related queries . . . 10

6 Results/Discussion **11**
6.1 FAQ Generation Results . . . 11
6.1.1 FAQ Generator . . . 11
6.1.2 Key Insights from FAQs . . . 11
6.2 Multi-Class Classification . . . 11
6.2.1 Baseline System . . . 11
6.2.2 Results . . . 11
6.3 Top level Analysis . . . 13
6.4 Government Scheme Analysis . . . 15
6.4.1 PM Samman Nidhi Scheme . . . 15
6.4.2 PM Man Dhan Scheme . . . 16
6.4.3 Kissan Credit Card . . . 16
6.4.4 Fasal Bima Yojana . . . 17
6.5 Plant Protection . . . 17
6.5.1 General analysis . . . 18
6.5.2 Per crop disease analysis . . . 19
6.5.3 Statewise analysis . . . 20
6.5.4 Monthwise analysis . . . 21
6.6 Weather related queries . . . 22
6.6.1 Weather queries vis a vis total Queries . . . 22
6.6.2 Temporal Distribution of Queries . . . 22
6.6.3 Spatial Distribution of Queries . . . 23
6.6.4 Analysis From Specific Queries . . . 24

7 Future Direction **25**
7.1 Multi-Class Classification 25
7.2 FAQ Generator 25
7.3 Plant Protection 25
7.4 Government Scheme 25

1 Abstract

Kisan Call Centre was launched to harness the potential of ICT in Agriculture, Ministry of Agriculture & Farmers Welfare. The main aim of this program is to answer farmers' queries efficiently on a telephone call in their own language. The call summaries are recorded, compiled and are freely available online. This dataset provides us a platform to understand the problems faced by farmers in different regions across India, to gauge the level of awareness about different government schemes, agricultural technologies, weather patterns, etc and assist farmers directly by improving the efficiency of the Kisan Call Centre.

In this project we analysed the data and present before you the various insights and tools that we were able to extract and develop from the data. These tools and key insights can not only help us understand farmer problems better and help them but also help in improving the efficiency of Kisan Call Centre by better managing their resources, targeting maximum throughput.

2 Problem Statement

We aim to analyze KCC queries and extract key insights from it. Also, develop some tools that can benefit in increasing effectiveness of KCC and understand farmers better. Our problem statement can be divided into 5 sub-statements as follows

- Assisting KCC executives by implementing multi-class classification algorithms that enable "auto-fill" of query type and sector fields
- Providing a tool to generate FAQs for any sub-data based on the the constraints. Also, derive any key insights from important FAQs.
- Providing frequency of queries of various schemes in various states and to get prevalence of various types of queries(eg registration, contact related etc.) in various schemes.
- Compute the major problems in the most popular crops so that focus can be laid on majority of diseases rather than very less prevalent issues. Also, find the states and months in which plant protection queries for major crops are more prevalent.
- Finding major weather related issues plaguing the farmers in various states across all months.

3 Introduction and Motivation

The feeders, comprising nearly half of the population, suffer from several hardships. To relieve them with some of their hardships several IT initiatives were started by the Indian government. It is generally believed that data collection by government is unorganised and is marred with errors. With this in mind, we explored several government websites with the aim of finding a dataset that can be analyzed in an impactful way.

The Kisan Call Centre dataset is a well documented dataset that contains summaries of queries asked by farmers, provides information of the region where farmers face the particular query along with features such as sector and query type that can be derived from the query text. The vast amount of information and the possible insights which can be used to positively support the Indian agricultural ecosystem that can be gathered from this data motivated us to pursue this project.

Also, we observed that the KCC executives have made many mistakes in documentation, which manifest as missing values and inconsistencies in our dataset. Hence, we were also motivated to develop tools that can be used to partly automate the process of entering data, thus increasing the efficiency and effectiveness of the Kisan Call Centre.

4 Dataset used

- We scraped our complete dataset from the website https://data.gov.in/resources-from-web-service/6622307.
- There are 14k+ webpages where each page has 6 csv files.
- Each csv file contains queries asked to KCCs of a particular district over a particular month of the year.
- Total size of the downloaded dataset was ∼8GB.

5 Methodology

5.1 Scraping

The complete dataset is scraped from the website https://data.gov.in/resources-from-web-service/6622307. Selenium package in python is used to automate the download process. The high level view of major steps involved is given below:

- Iterate over all the page numbers and open the webpage corresponding to the current page number.
- Fetch all the csv elements from the webpage using appropriate heuristics after waiting for the page to load.
- Iterate over the fetched csv elements and wait till the browser reaches the first tab because the tab changes when download button is clicked.
- Click on the "EXPORT IN: CSV" button to open the form for downloading csv and fill in the relevant details automatically before submitting the request for download.
- Switch to the first tab to download other csv files present on the current page.
- After downloading all the csv files from the current page, open the next page and continue the process.

5.2 Data Preprocessing

Query Texts' represent summary of the calls between the farmer and the Kisan Call Centre employee. Since the data is updated in real-time and by the KCC employees in hurry, it has lots of noise, ambiguities and disturbances. Following methodologies were used for processing the data before any further analysis on it

5.2.1 Narrowing down data

- ***Problem*** **-** Justifications derived from large sets of data are generally poor in precision
- ***Solution*** **-** Reduced the data, limiting by time, from 1st Oct, 2018 to 30th Sept, 2020

5.2.2 Inter-Dataset duplication

Each dataset on the portal represents data for a particular district for a particular month

- ***Problem*** **-** Analysis revealed various exact duplicate datasets in the datasets that we scraped
- ***Solution*** **-** We deleted all but one duplicate datasets in pre-processing phase since the duplicates add no extra value

5.2.3 Intra-Dataset duplication

- ***Problem*** **-** Analysing revealed that many datasets had duplicate entries with just the time-field differing
- ***Possible Reason*** **-** Network issues(Multiple requests) when sending data to update in the database
- **Attemps -** Tried various threshold to choose an appropriate time difference to call 2 data as duplicates
- ***Solution*** **-** Finalised 6 secs as an appropriate threshold since at values greater than it, the increase in count of duplicate entries was very low and reasonable. Disregarded/Deleted all but one duplicate entries.

5.2.4 Outliers

Following outliers were removed to help our Language Processing/ Analysis algorithms in deriving information from data

- ***Arbitrary Length Queries*** **-** Extremely large (>150 ch) and small queries (0 ch) were removed. Threshold chosen at values where the increase/decrease in count of such queries were low/high and reasonable.
- ***Failed Calls*** **-** All queries such as *Call Disconnected*, *Missed Call* and various others which convey failed calls were found using regex and deleted since they provide no valuable information.
- ***Noisy Queries*** **-** Many queries were very noisy, had a lot of non-alpha characters. Detected and removed such queries by choosing an appropriate threshold ratio of non-alpha and alpha characters in query text.

After pre-processing and storing data in json instead of csv, the final data.json sized at around 650 MB

5.3 FAQ Generation

Here we derived an approach/algorithm/software which helps you generate FAQ for any sub-data based on meta-data provided. For example we can ask it to generate FAQ for all calls of Uttar Pradesh that came in June-2020

5.3.1 Data Cleaning

Data after global pre-processing and formatting still had some disturbances per query which we didn't remove in previous stages as removing them could have served poor result with other analysis

1. Data text had a lot of non alpha-numeric characters at arbitrary non-grammatical places as noises
2. We deleted them and replaced them with spaces/empty string depending on the need
3. This helped in improving the accuracy of NLP algorithms used for grouping similar sentences

5.3.2 FAQ Methodology

There are about 1 million *unique queries* in total of 6 million queries. But even in those unique queries, a lot of queries convey similar semantic meaning and should be counted together. So, to generate FAQs, we first grouped similar queries together using NLP and Clustering. Following methods were used to generate FAQs

1. **Compute Embeddings**

   We first generated the embeddings (n-dimensional numerical vectors) for each of the query text using an available python library called sentence-transformers. We experimented with some of the pre-trained models available that specialises in **semantic textual similarity** like

   Models

   - distilbert-base-nli-stsb-mean-tokens
   - bert-large-nli-stsb-mean-token
   - xlm-r-bert-base-nli-stsb-mean-tokens

   We finalised **xlm-r-bert-base-nli-stsb-mean-tokens**, a **multi-lingual model for semantic similarity** which has been trained on parallel data in more than 50+ languages (as the developers claim).

   Reasons

   (a) *Multi-Lingual* - Many of the queries of our dataset are in hindi and many are written partially in hindi and partially in english. Other models which were only trained with english gave very poor performance for these queries while clustering and computing FAQ

   (b) *Faster* - This model was faster in generating embeddings as compared to others. Like for queries of Kerala took around **1 sec per 32x32 batch** in xlm-r-bert-base-nli-stsb-mean-tokens as compared to **4 secs per 32x32 batch** in bert-large-nli-stsb-mean-tokens

2. **Clustering**

   Now we have converted our list of queries into n-dimensional vector encodings. Next step is to cluster them into groups/clusters where queries in each cluster are semantically similar enough to be considered as a group for computing FAQs.

   Approach

   (a) We first computed cosine similarity scores for each query-embedding with every other query-embeddings.

   (b) Next we formed clusters by applying a threshold on this similarity score which in our case settled at **0.9**.

   (c) We then removed the smaller one for every 2 clusters that had an intersection

   Evaluation

   (a) The major challenge was to choose the appropriate similarity level (or threshold) that groups queries which are similar enough that separating them into clusters won't serve any more benefit for FAQ-gen

   (b) Accuracy of the clusters were tested based on the following 2 measures

   - *Homogeneity* - Analysed if they carry common context and is that context sufficient for FAQ
   - *Completeness* - Analysed if any two so-formed clusters can serve any benefit if merged together

3. **Generate FAQs**

   (a) We computed the final FAQ by first computing the exact frequency of each cluster by summing up each queries frequency within it.

   (b) Then, sort the clusters in decreasing order of the cluster-frequency and give the centroids of the top-10 clusters as the output FAQ.

   (c) We also experimented with **TfidfVectorizer** to compute similarity scores but it gave very poor results. The primary reason for its failure can be attributed to the fact that it computes similarity by giving scores to words based on their "rareness" which didn't quite work out for our case where most queries are based on rare words only.

5.4 Multi-Class Classification

Here, we worked on 2 classification problems-

- Given the query text, classification of queries into different sectors
- Given the query text, classification of queries into different query types

5.4.1 Classification Pipeline

We designed a classification pipeline consisting of the following 3 components-

1. Dimensionality and Numerosity Reduction:
   - Feature Selection
   - Removal of duplicates
   - Removal of inconsistencies
   - Sampling
2. Extraction of features from query text
3. Training and testing of the classification model

5.4.2 Dimensionality and Numerosity Reduction

- Feature Selection
  - The district, state, category and crop fields are not useful in predicting the sector or the query type. Also, in practice, the auto-fill application should ideally take in only the query text. Hence, these features were not considered in this task.
- Removal of duplicates
  - In the dataset of 6 million queries that we analyzed, we observed that the number of unique query texts is about 1.2 million only. Hence, we considered only 1 instance each of these duplicate queries.
- Removal of inconsistencies
  - There were around 35,000 data tuples that had an invalid entry in either the sector field or the query type field. We removed all these data tuples.
- Sampling
  - Since the dataset that we prepared for this task was huge, we took samples from different parts of the dataset in order to train and test our models.
  - For the Sector Classification task, we randomly chose around 4,000 samples of each sector.
  - For the Query Type Classification task, we chose the top 10 most frequently occurring query types in the dataset and chose around 1,000 samples of each query type.

5.4.3 Feature Extraction

For the feature extraction step, we initially explored the scope of simple methods such as keyword searching coupled with rule-based classifiers that predict the query's sector and type based on the presence or absence of certain important words. But, due to the sheer volume of the data and its diversity, we realized that such traditional methods will not be feasible for this problem.

Hence, we explored large language models that are capable of capturing the meaning and the context of the sentences. We used the following 3 transformer models to generate context-aware embeddings that capture the meaning of each sentence-

- BERT Transformer
- Electra Transformer
- Sentence Transformer (a hybrid model that uses BERT, RoBERTa and XLM-RoBERTa transformers and can handle multi-lingual datasets)

While using each of the transformers, the following process was followed-

- Tokenizing the query text using the transformer model's tokenizer
- Generating embeddings for each token using the transformer model
- Taking average of all token embeddings to generate the sentence-level embedding that captures the meaning of the query

5.4.4 Classification Models

The following 7 machine learning classification algorithms were experimented with in order to solve the 2 multi-class classification problems-

1. Decision Tree Classifier
2. Random Forest Classifier
3. Maximal Margin Classifier (SVM)
4. K Nearest Neighbor Classifier
5. Naive Bayes Classifier
6. Logistic Regression
7. Classification using Linear Regression-
    - In this approach, we trained a linear regression model that predicted a dummy embedding for the target class from the sentence embedding of the query text. Then, the target class was chosen as the class whose embedding has the highest cosine similarity with the dummy embedding.

5.5 Government Schemes Queries

The total number of queries that had the type 'government scheme' were around **8 lakh** and hence for being able to tag schemes for the given data number of queries had to be reduced. The main issue was getting the scheme tags from such a large data having many variations of spellings for the same scheme.

5.5.1 General Data Pre-processing

- The original data is of form year→month→statedist→[Sector,Crop type,Crop,query type,query]. This data was then converted to the form "yr→month→state→queries" for all the queries that had query type as "Government Scheme".
- Total number of queries having the query type as "Government Schemes" were around 8 lakh. So first of all the number of distinct string for the given 8 lakh queries, were found which were around 1.25 lakh.

5.5.2 Getting Scheme Tags

Although the number of queries were reduced to 1.25 lakhs there was another problem, there were numerous spellings for the same scheme for eg 'samman nidhi yojana' could be 'sammman nidhi scheme', 'pm kisan yojana', 'nidhi scheme', 'nedhi yojana' and many more.

1. **Try 1: Performing Scheme Tagging**
   - On these 1.25 lakh. We tried to perform tagging of schemes as follow:
     (a) Get the distinct strings having the words yojana(and its variants) or scheme in it
     (b) Try to get key word of a scheme in the list obtained.
     (c) Remove the strings that have the key word (obtained from b) from the list of strings.
     (d) Repeat from step b.
   - Although the number of queries having the words yojana or scheme in it were around 40k however performing the steps mentioned above seems quite hectic. After the number of strings in the list were reduced to around 20k after applying the algorithm mentioned above. This was because of the following reasons(these were affirmed later on):
     (a) Number of queries were in hinglish and there was a huge variety of words for each Hindi words thus tagging had to be done multiple number of times for the same scheme.
     (b) There were huge number of schemes that had queries in quite small numbers(10-100).
     (c) There were many statements of type "tell me which yojanas are going on?" non-specific to any scheme (can't say conclusively but half of 8 lakh queries were of this type).
2. **Try 2: Performing Scheme tagging**
   - In order to avoid the first problem mentioned above, the entire data of 1.25-lakh distinct queries were, first translated to Hindi so hinglish sentences were converted to Hindi and the spelling variations were reduced.
   - There were some English words in the translated text too but finding those, that were useful for scheme tagging were quite easy by filtering out English words and then sorting them.
   - Now to find tags, distinct phrases of 2-3 words before the word "yojana"(in hindi) were considered. Although the sentences having the word "yojana"(in hind) were 40k the number of such distinct phrases were around 4.2k.
   - For there 4.2k phrases their frequency was found in the original 8 lakh queries containing the words "yojana"(in hindi)". (each of the 8 lakh query would have a mapping to a translated Hindi query). And the following observations were made:
     (a) "samman nidhi"(in hind) came in 15% of the queries many of it variants occur ed till end.
     (b) 'fasal bima'(in hind) occurred for 0.7% and many of its variants were there in the 4.2k phrases
     (c) 'man dhan'(in hind) was around 0.5% and the same case as in 2.
   - Then we performed some word transformation to make the the scheme tags uniform like 'man dhan'(in hind) could have been 'mandhan'(in hind) so this was then transformed to 'man dhan'(in hindi) for all possible tags, and it was relatively easy as compared to the previous approach as only 4.2k phrases were there.
   - After applying transformation on the given schemes the frequencies became:
     (a) "samman nidhi"(in hind) came in 58% of the queries many of it variants occur ed till end.
     (b) 'fasal bima'(in hind) occurred for 2% and many of its variants were there in the 4.2k phrases
     (c) 'man dhan'(in hind) was around 1.7% and the same case as in 2.
   - Then for further scheme specific analysis only the schemes have atleast 10k queries were considered so the scheme specific analysis was performed for samman nidhi, mandhan, kcc, fasal bima.
   - Finally Try 2's method was used.

5.5.3 Finding key words for scheme specific analysis

- The aim was to tag the type of query(eg registration related,finance related etc) for the scheme in consideration.
- For this first of all the queries that were of a given scheme like "samman nidhi" were extracted(2.7 lakh queries arround 40k distinct queries), and then there corresponding distinct hind queries which were around 10k (this is an indicator that number of translated Hindi query were even lesser than the 1 lakh).

- We tried to get key words in following way:
    1. Get the distinct strings having the scheme name in it.
    2. Try to get key word of the type or query(eg helpline,website etc) in the list obtained.
    3. Remove the strings that have the key word (obtained from b) from the list of strings
    4. Repeat from step b.
- This us similar to the algorithm that we followed earlier only this time the best key words were obtained by repeatedly getting the top "tf-idf" words hence the finding of key words was quite easy.
- Such keywords were identified for each schemes the samman nidhi, mandhan, kcc, fasal bima only (only these schemes had more than 10 queries) and then clustered into registration,finance,general information,website,contact info manually(as key words were <100 in all the schemes).

5.6 Plant Protection analysis

The aim was to analyse the queries related to plant protection. The analysis is divided into 4 major categories:

5.6.1 General analysis

This section contains the basic analysis of the data related to plant protection.

- Frequencies of crops in the queries related to plant protection were calculated to find which crops are more popular or more grown.
- Number of queries per state was computed for queries related to plant protection to show the variations in the number of queries based on location.

5.6.2 Per crop disease analysis

Disease analysis per crop was done for the major crops computed in the previous section.

- First all the queries with query type "Plant Protection" were fetched from the dataset. There were total ∼1M queries.
- All the English stopwords present in the python nltk library and some manually picked non-relevant words were removed from all the queries so that the further analysis becomes convenient. Some examples of the non-relevant words are ask, tell, want, need, know, etc.
- This preprocessed data was stored in a json file to later compute major categories/diseases in the top 3 of the major crops: Paddy, Cotton, and Wheat.
- Duplicate data was removed and the original count corresponding to each query was calculated so as to make the clusters easily. There were ∼300K distinct queries initially which reduced to ∼200K after removing stop words and non-relevant words.
- For each of the crops: Paddy, Cotton and Wheat, the distinct queries with their counts were visualized in reverse sorted order w.r.t. their counts.
- Mapping with keywords to count was maintained and the mapping was updated iteratively by visualizing the above sorted order where in each iteration the queries which didn't have any of the computed keywords decreased as new keywords were picked out.
- Different mapping was maintained to keep an equivalence set of the keywords as some of the phrases were different but had the same meaning. E.g. pili and yellow, aphid and insect, keywords with spelling mistakes, kide and insect, etc.
- Term Frequency-Inverse Document Frequency was also used to get important keywords to incorporate them in the mappings.
- Finally, these mappings were used to categorize the query texts to find the major type of categories per crop related to plant protection.

5.6.3 Statewise analysis

This section contains the statewise analysis per crop.

- The original data was used to compute in which states queries which belonged to a particular crop and with type "Plant Protection" are more prevalent.
- The analysis was performed for major crops: Paddy, Cotton and Wheat.
- For better visualization data was plotted on India map.

5.6.4 Monthwise analysis

This section contains the monthwise analysis per crop.

- The original data was used to compute in which months queries of type plant protection are more prevalent per crop.
- The analysis was performed for major crops: Paddy, Cotton and Wheat.
- This analysis was broken down into 2 years: Oct 2018 - Sept 2019 and Oct 2019 - Sept 2020.

5.7 Analysis of weather related queries

- Data Preprocessing
  1. The original data is of form year→month→statedist→[Sector,Crop type,Crop,query type,query]
  2. From this data all queries with query type "Weather" were extracted.This amounted to close to 25 lakh queries.
  3. Also there were 46 thousand queries which did not have "Weather" as query type but the word "weather" or any of its errors like "wheather" were present.Being just 2 percent of all queries and having word "weather" does not necessitate it to be a weather related query,they were ignored.
  4. To analyse 25 lakh query manually is impossible .So the following method was adopted-
     (a) First the distinct queries were found and stored in list
     (b) In these queries, keywords were found manually and queries with these queries were removed from list.Repeat the above steps till no more weather related keyword can be found.
     (c) Analysing the queries it was found that some queries were not weather related.
     (d) The relevant keywords like temperature were used to search queries and queries which did not have any keyword were ignored.1 lakh queries with query type "Weather" were ignored from actual 25 lakh queries.
  5. So now 24 lakh queries were left from which analysis has been done.In these queries, keywords,their possible errors and hindi translation were clubbed together.
  6. Analysing the above weather related queries ,it was found that 99.07 percent of the queries were general queries like asking condition of weather while only 0.93 percent of the queries were specific queries like asking rainfall.So the major part of the analysis is of general queries.
- Analysis
  1. The analysis can be divided into broadly 2 parts -
     (a) **General Analysis:**This comprised of 99.07 percent of data.The analysis can be further divided into
        i. Temporal analysis: Variation in queries across months or seasons.
        ii. Spatial analysis: Variation across states.
     (b) **Specific Analysis:**This pertains to specific cases like cyclones.

6 Results/Discussion

6.1 FAQ Generation Results

6.1.1 FAQ Generator

We created a FAQ generator which can help you generate FAQs for all data or a subset of data. One can ask for generation of FAQs of data that satisfies all of the given list of meta-data constraints.

Possible Constraints

1. *Year* - eg: you can ask for FAQs for the year 2019
2. *Month* - eg: you can ask for FAQs for year 2019, month - 9 (September)
3. *State* - eg: you can ask for FAQs in state Maharashtra
4. *District* - eg: you can ask for FAQs in Mumbai
5. *Crop* - eg: you can ask for FAQs regarding wheat

One can also give multiple constraints together from above like ask for FAQ regarding wheat in Mumbai in year-2020

6.1.2 Key Insights from FAQs

1. **Major Cultivators, Major Problems**
   (a) Weather fluctuations and weed control in wheat are the major concerns of farmers in Uttar Pradesh which is the major cultivator of Wheat in India
   (b) Leaf brown spot, stem borer and sheath rot in paddy are the major concerns of farmers in West Bengal which is the major cultivator of Paddy in India
   (c) Fertiliser Management and sucking pests problem in cotton are the major problems of farmers in Gujrat which is the major cultivator of Cotton in India
   (d) Market rate, fertiliser dose and wilt attack on onion are the major concerns of farmers in Maharashtra which is the major cultivator of Onion in India
2. **Other Insights**
   (a) In Kerala, most of the farmers have a lot of query regarding the "PM Kisan Samman Nidhi Yojna"
   (b) The most crucial concerns among farmers regarding Mango cultivation is about its disease control
   (c) In Delhi, farmers are more concerned about weather fluctuations and soil testing related information
   (d) Farmers are facing a lot of issues in controlling insects in Bajra cultivation and are asking for remedy

6.2 Multi-Class Classification

6.2.1 Baseline System

In order to analyze the performance of our classification models, we considered the human annotation as the baseline system. We manually verified 10 samples of 100 data tuples each and observed that in each each of these 10 samples, around 15 data tuples have been classified incorrectly. Thus, we considered the baseline classification accuracy to be 85%.

6.2.2 Results

The classification accuracies obtained using different methods are reported in Table 1. The results can be summarized as-

- Best Accuracy obtained for Sector Classification is 0.89, when Sentence Transformer model is used with logistic regression. This model beats the human baseline of 0.85.
- Best Accuracy obtained for Query Type Classification is 0.67, when Sentence Transformer model is used with logistic regression. This model beats the human baseline of 0.85.

Feature Extraction Model	Classification Algorithm	Sector Classification Accuracy	Query Type Classification Accuracy
-	Human Baseline	0.85	0.85
BERT Transformer	Decision Tree	0.64	0.36
BERT Transformer	Random Forest	0.79	0.58
BERT Transformer	Support Vector Machine	0.85	0.61
BERT Transformer	K Nearest Neighbors	0.77	0.57
BERT Transformer	Naive Bayes	0.72	0.52
BERT Transformer	Logistic Regression	0.86	0.63
BERT Transformer	Classification using Linear Regression	0.85	0.62
Electra Transformer	Decision Tree	0.44	0.25
Electra Transformer	Random Forest	0.59	0.43
Electra Transformer	Support Vector Machine	0.68	0.53
Electra Transformer	K Nearest Neighbors	0.60	0.40
Electra Transformer	Naive Bayes	0.46	0.25
Electra Transformer	Logistic Regression	0.67	0.54
Electra Transformer	Classification using Linear Regression	0.64	0.48
Sentence Transformer	Decision Tree	0.78	0.44
Sentence Transformer	Random Forest	0.88	0.63
Sentence Transformer	Support Vector Machine	0.87	0.63
Sentence Transformer	K Nearest Neighbors	0.89	0.60
Sentence Transformer	Naive Bayes	0.80	0.53
Sentence Transformer	Logistic Regression	0.89	0.66
Sentence Transformer	Classification using Linear Regression	0.88	0.67

Table 1: Multi-class Classification Results

- Sector Classification, having 4 classes gave better results as compared to Query Type Classification having 10 classes.
- Sentence Transformer performed the best among the 3 language models.
- The method of treating classification as a regression task and logistic regression were the 2 algorithms that showed the most promising results across all 3 feature extraction models.

The best results for each class are summarized in the 2 confusion matrices in Figure 1.

Some key factors affecting the classification results are as follows-

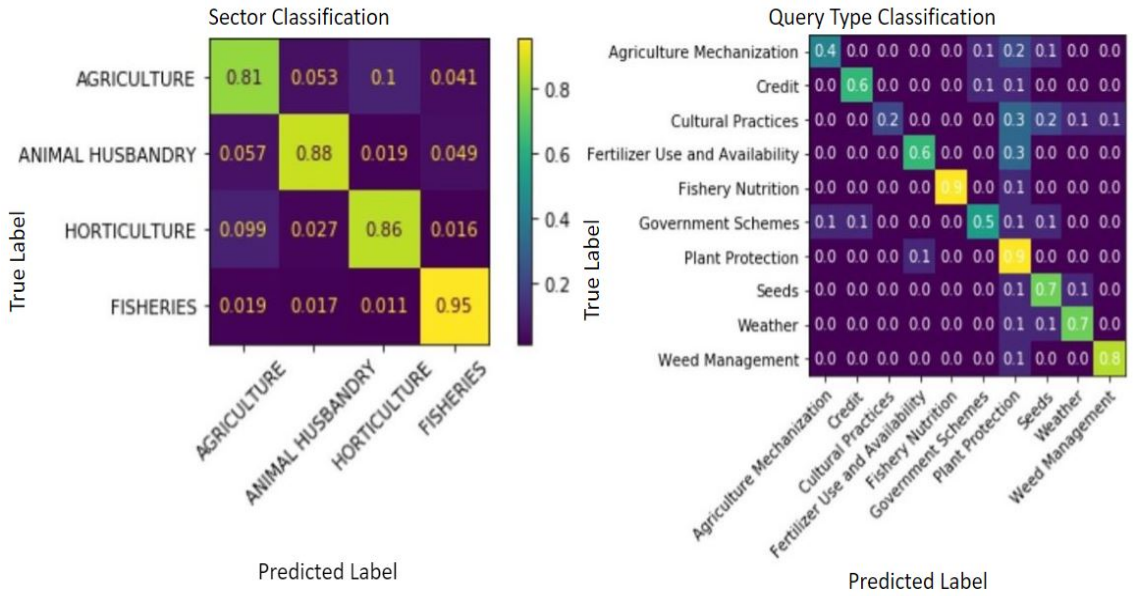


Figure 1: Best Results: Confusion Matrices

- **Errors in the training dataset:** As mentioned earlier, the accuracy with which the KCC executives have manually assigned sectors and types to queries is around 85%. Although we manually verified and corrected these errors to some extent, it was not possible to do it for the entire train and test sets.
- **Uncorrelated classes:** For query type classification, we used training and testing datasets having 10 classes. But these classes are not mutually exclusive. For example, a query about the government schemes related to agricultural mechanization (eg. government subsidy on tractors) can be classified under "Government Schemes" as well as "Agricultural Mechanization".
- **Multi-lingual Dataset:** BERT Transformer and Electra Transformer models are pre-trained only on english corpora, whereas the Sentence Transformer can be used to tackle multi-lingual tasks as well. Since our dataset has "Hinglish" query text as well, Sentence Transformer model performs better than its counter-parts.

6.3 Top level Analysis

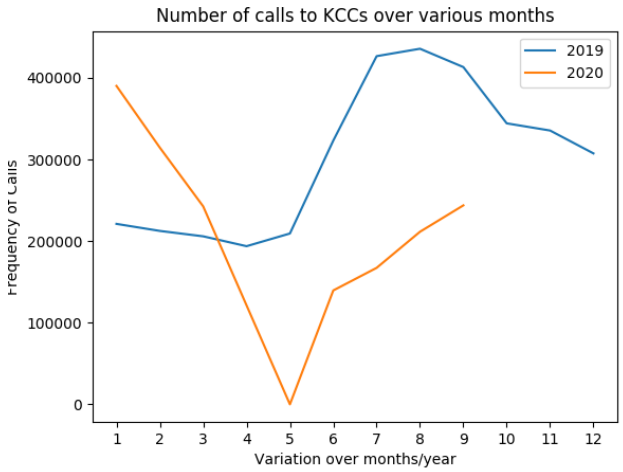


Figure 2: Variation of total queries over the year

The graph on the left shows the variation of number of queries over the months for years 2019 and 2020

Peak Months - July - November
Off-Peak Months - March - May

The KCC authorities should actively manage their working staff taking into consideration the peak/off-peak months to manage calls efficiently

Year-2020 graph shows a sudden downfall during months from Feb-June which were the lockdown period in India, justifying the same.

The graph on the right shows the spread of overall queries in different states of India

Most Active States - Uttar Pradesh, Rajasthan, Maharashtra

More effective resource management by KCC is needed in these locations primarily.

Least Active Regions - North-Eastern, Northern region, Chhattisgarh, Jharkhand

The above KCCs are **critical**, either the farmers don't require KCC or their is poor communication. Effective measures should be taken to spread information about KCC.

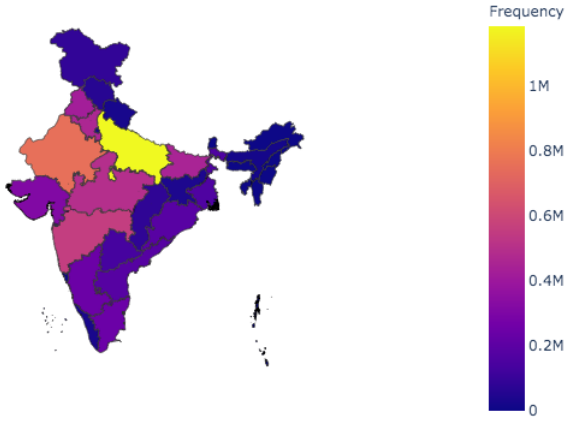


Figure 3: State-wise density plot for overall queries

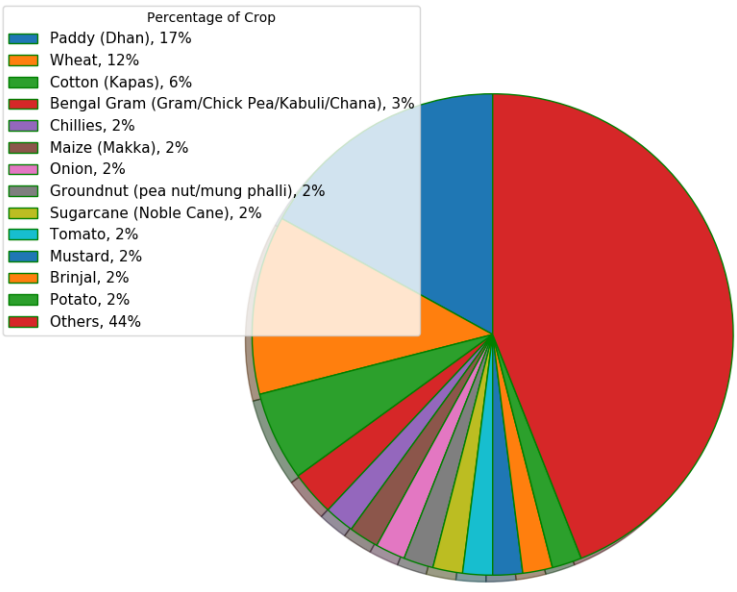


Figure 4: Percentage domination of different crop queries

The graph on the left shows the spread of crop related queries into the various types of crops

Major contributors -

- Paddy
- Wheat
- Cotton

One of the reasons for this it that India is a major cultivator of the above. But India is also a major cultivator of other crops, the queries of which are very low. The KCC should conduct a survey to see if it is due to the good knowledge of farmers regarding those crops or because of unawareness.

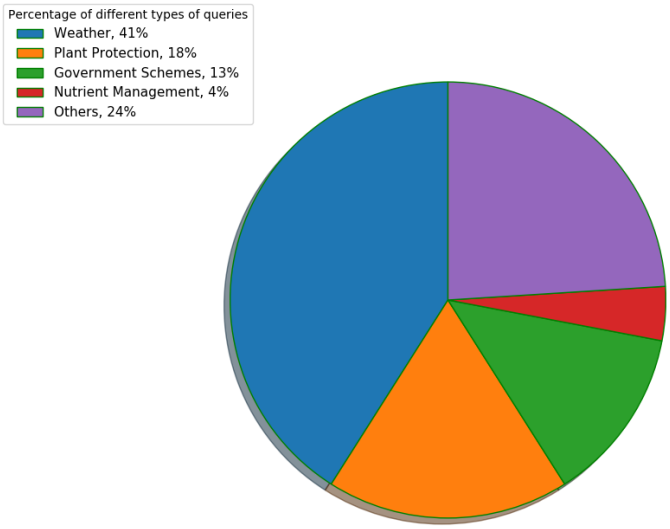


Figure 5: Percentage domination of different types of queries

6.4 Government Scheme Analysis

Note 1: Holding is basically an area of land under a farmer. A given farmer can have more than one holding but since the population of farmers per state was not available number of holdings of each state was considered from "https://www.prsindia.org/policy/discussion-papers/state-agriculture-india". Holdings of Telangana and Andhra-Pradesh were searched separately.
Note 2: The per-holding graphs are actually obtained by multiplyting the actual count with 10,000 and removing (1-2) top level entries for better contrast.

6.4.1 PM Samman Nidhi Scheme

Number of queries

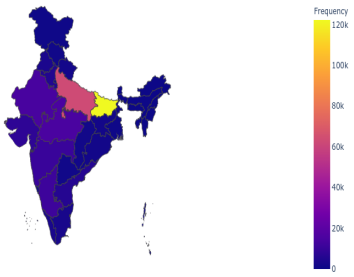


Figure 6: State wise for Samman Nidhi scheme

Number of queries per holding

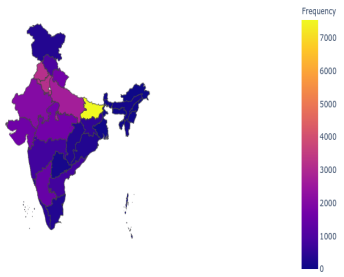


Figure 7: State wise for Samman Nidhi scheme

- Total number of queries of PM Samman Nidhi scheme were around 2.7 lakhs(of 8 lakh) implying it is a quite popular scheme hence considerable number of resources should be allocated to the given scheme.
- The top 3 states where most number of queries were asked are Bihar(1.2 lakhs), Uttar-pradesh(63K), and Madhya Pradesh(14k),Whereas the bottom 3 states/UT where Arunachal Pradesh(2), Lakshwadeep(0), A N Islands(0). Considering the fact that these UTs are quite small an not agriculturally involved its quite appropriate, however efforts must be made to advertise the scheme where the number of queries are below 100. Total number of such states/UTs are 14(below 100 queries).The entire list comes as output of the code part.

– The number of queries per holding is a metric that shows awareness of the scheme, it is quite high in the northern states/UT the top ones being Chandigarh, Delhi, Bihar, Punjab, Haryana. Whereas states like Nagaland, Manipur, Tripura, Arunachal Pradesh are below 0.005 indicating lack of popularity of the scheme.

– Based on key-words finding, and clustering of these keywords in various categories for the queries of the scheme, in the entire country there were 91768 registration related queries, 68966 finance related queries, 24427 contact information related queries, 1840 website related queries, 108514 queries regarding general information of the schemes, and 10142 queries that were not tagged.

– The 108514 general information related queries indicates that the scheme is quite unknown but fast spreading whereas the procedures regarding the registration and finance of the schemes must be made more simpler for better access and resources must be spent in these arena.

– Stat-wise distribution of the query types(registration,finance etc) is also given which can be used by the authorities for identifying key areas on which work has to be done in the scheme for a given state.

6.4.2 PM Man Dhan Scheme

Number of queries

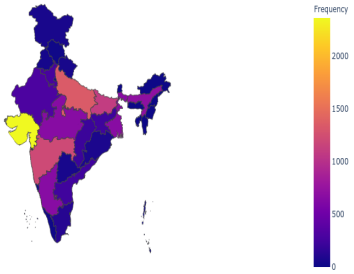


Figure 8: State for man dhan yojana

Number of queries per holding

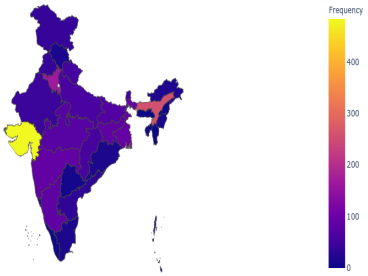


Figure 9: State for man dhan yojana

– The total number of queries asked for the mandhan yojana were around 10k, which means it is relatively lesser known than samman nidhi yojana. But all the schemes have lesser number of queries as comapared to samman nidhi yojana by a considerable margin(10 to 1000 times lesser number of queries).

– The top three states were the most number of queries were asked for this scheme were Gujarat(2.3k), Uttar-Pradesh(1.3k) Maharashtra(1.2k), where as there were 7 states/UTs were no queries were asked for the given scheme. The states were A N islands, Chandigarh, Daman and Diu, Lakshwadeep, Meghalaya, Mizoram, Nagaland.

– The number of queries per holding were high in Delhi, Gujarat, Assam however the per holding queries are all lesser than 0.1(except in Delhi) indicating this scheme has to more advertised.

– Overall the states lagging in both of the metrics mentioned above need considerable investment along with an comprehensive effort to make the scheme more wide-spread.

– The queries of the scheme that were asked comprised of 3037 registration related queries, 817 finance related queries, 133 contact information related, 1 website related query, 5951 general information related query, and 888 queries that were not tagged.

– The 5951 general information related queries indicate the farmers are indeed want the queries but the 1 website related query, 817 finance related queries, 133 contact information queries may be an indication that the base infrastructure to run the scheme isn't quite intact.

6.4.3 Kissan Credit Card

– The number of queries that were asked for the Kissan Credit card were around 33k. The top 3 states where the most number of queries were asked are Bihar(7.5k), U.P(6.3k) and Rajasthan(2.6k) the bottom 3 were A N islands, Lakshwadeep, Chandigarh.

Number of queries

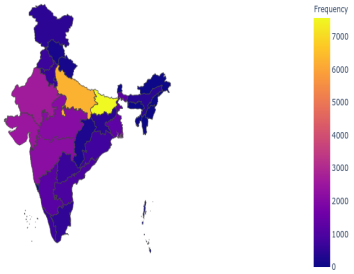


Figure 10: for kissan credit card

Number of queries per holding

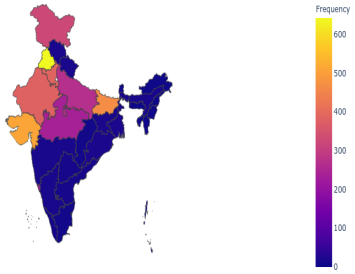


Figure 11: for kissan credit card

- The highest per holding queries were in Delhi,Chandigarh(an anomaly because it had just 2 queries but because of very less number of holdings this metric for it is quite high), Punjab and Gujarat, the bottom ones for this metric were Tripura, Mizoram, Meghalya.
- Out of the queries asked 13205 were registration related, 2861 were finance related, 69 web-site related, 14926 general information related, 2433 queries were un-tagged.
- The 13205 registration related queries related queries indicate a comparatively good infrastructure, but the 2861 finance related queries indicate not many people use it once registered which is not a good sign considering the fact that this is a finance related scheme. So better financial and user friendly schemes should be constructed under Kissan Credit Card yojana to make it more viable.

6.4.4 Fasal Bima Yojana

Number of queries

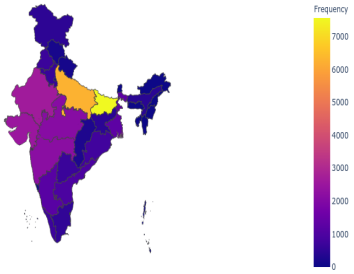


Figure 12: State wise for fasal bima yojana

Number of queries per holding

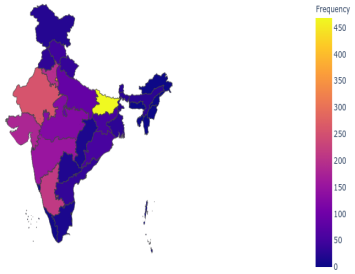


Figure 13: State wise for fasal bima yojana

- The total number of queries asked for this scheme were close to 19k.The top 3 states where most number of queries were asked were Bihar, Maharashtra, UP and there were 12 states/UTs (mostly UTs) where 0 queries were asked regarding this scheme which quite drastic considering the fact that the previous schemes had considerably more spread out spatial distribution as compared to this one.
- The per holding query distribution in high in Bihar,Delhi,Rajasthan and Karnataka.
- There wer 4261 registration related queries, 1895 finance related queries, 3227 contact information related queries, 41 web-site related queries, 9277 general information queries, 301 crop related queries(as this is an insurance policy so crop specific queries were there), 1112 that were not tagged.

6.5 Plant Protection

This section contains the results obtained after doing analysis on the plant protection queries. The analysis is divided into 4 major categories:

6.5.1 General analysis

This section contains the basic analysis of the data related to plant protection.

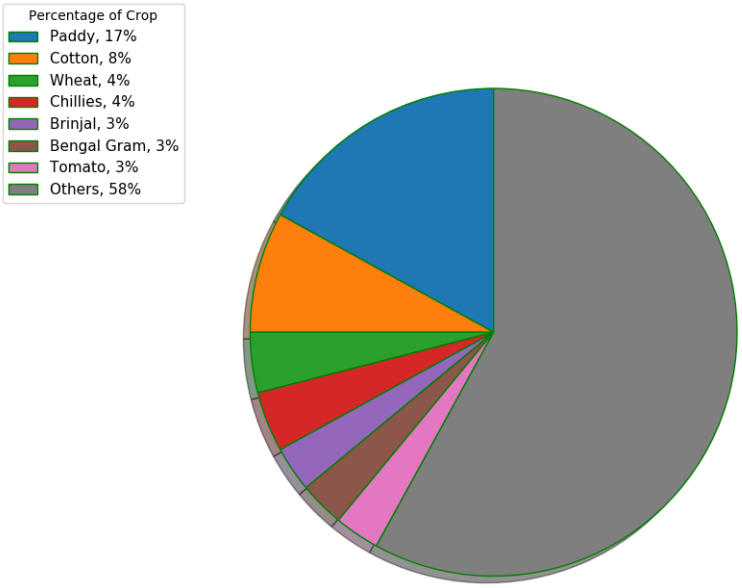


Figure 14: Percentage of crops in the plant protection queries

- **Major Crops**
  - There are total of 272 crops and just 7 crops constitute 42% of the total queries. Paddy has the most number of queries on plant protection with almost $1/6^{th}$ of the queries i.e. ∼200K from ∼1.1M queries.
  - According to 2018 data available on the internet[4], sugarcane is the most produced crop in India. So, either paddy has moved to top in these 2 years which is very less probable. Another reason can be that sugarcane doesn't have much issues regarding protection due to which more calls of it are not observed.
  - So, government should focus mostly on these major crops on spreading awareness regarding protection from insects and various diseases.

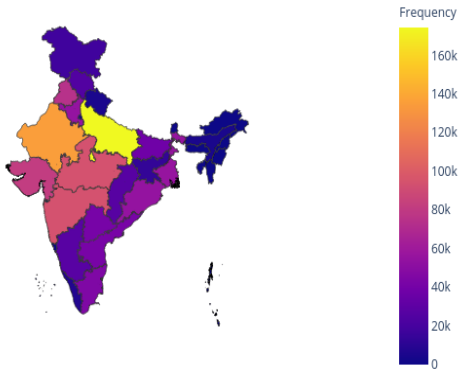


Figure 15: Plant protection queries in each state

- **Queries per state**

– Around 50% of the total plant protection related calls were from these 5 neighbouring states:

1. Uttar Pradesh
2. Rajasthan
3. Madhya Pradesh
4. Maharasthtra
5. Gujarat

The possible reason can be that either these states are producing more crops than other states due to which they have more queries or the KCCs in these states are more active than the ones in other states. So, government should mostly emphasize on these major states for better crop production in India.

– Assam is in the top 10 crop producing states in India according to data available on internet[1] but on the basis of plant protection queries it is on 21^{st} place. Possible reason can be that the farmers of Assam are well aware about these issues or the crop conditions are favourable or worse that the KCCs there are not active. If the last one is the case then government should improve the KCC conditions there to protect the crops of one of the top 10 crop producing states.

6.5.2 Per crop disease analysis

Disease analysis per crop was done for the major crops computed in the previous section.

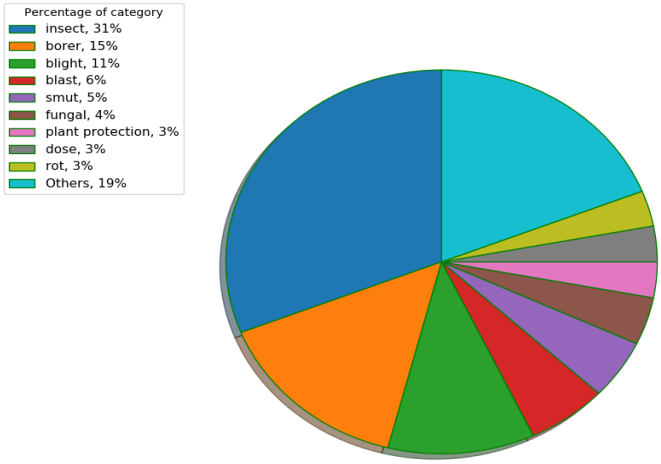


Figure 16: Percent of categories in paddy queries

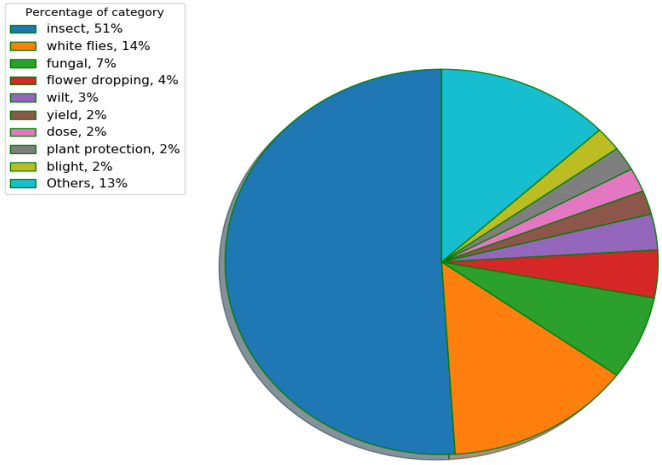


Figure 17: Percent of categories in cotton queries

- **Paddy**
  - Insects:
    - Around 46%(31% of insect and 15% of borer) queries of plant protection in paddy were based on attack of insects like termite, grasshoppers, mite, thrip.
    - 15% were of specific insect - borer. There were 2 major borers: stem borer and root borer.
    - As almost half of the queries are related to insects, farmers should be made more aware about respective insecticides and pesticides to prevent damage to crops.
  - Fungus[5]:
    - Around 11% of the queries were related to a bacterial and fungal disease - blight. The 2 major types of blight in the queries were: leaf blight and sheath blight.
    - 6% of the queries were on a disease named blast.
    - 5% queries were on a disease smut. There were major 2 categories of smut: false smut and kernel smut.
    - As all of these are fungal diseases, total percent of fungal diseases correspond to 26%(11% of blight, 6% of blast, 5% of smut and 4% of general fungal diseases from the plot).

    - Hence, almost $1/4^{th}$ of the queries are related to fungal diseases, appropriate fungicides should be provided to Paddy farmers to stop blight, blast, smut and other general fungal diseases.
  - 3% queries were on general plant protection of paddy. These queries didn't ask about any specific category.
  - 3% were on doses of fertilizers, pesticides, weedicides, etc.
  - Rest of the queries had those categories which are less than these percentages. Some of the examples include yellowing of leaves, spot disease in paddy, steps for better yield, etc.
  - These percentages show that the farmers are not much aware of the issues like attack of insects, fungal diseases, etc. As more than 70% of the paddy plant protection queries are based on getting the information for the attack or disease and very less ask about the corresponding insecticides, pesticides, etc.
  - So, government should increase awareness by creating some camps in the more paddy producing states or states with more queries on plant protection of paddy which is discussed in the next section. In those camps major focus should be on insects and fungus related issues.

- **Cotton**
  - Insects:
    - General insects like caterpillars, aphids, thrips constitute 51% of all the plant protection queries for cotton.
    - Specific insect white fly constitute 14% of the queries.
    - So, in total the queries about insect attack were 65% which is a much great proportion for a single category. So, cotton farmers should be made more aware of the insect repellent technologies and insecticides, pesticides, etc.
  - Fungus:
    - General queries on fungal diseases were around 7% and there were 2% queries on sheath blight which is also a fungal disease.
    - So, fungal diseases comprise 9% of all the queries. Farmers should be made aware about relevant fungicides to prevent damage to their crops.
  - Flower dropping queries comprise about 4% of the total queries. Insufficient soil moisture during reproductive growth in cotton promotes bolls shedding.
  - Rest of the queries had those categories which are less than these percentages. Some of the examples include wilting of leaves, reddning of leaves, leaf spot in cotton, about fertilizers, etc.
  - These percentages show that the farmers are not much aware of the issues like attack of insects, fungal diseases, etc. As more than 70% of the paddy plant protection queries are based on getting the information for the attack or disease and very less ask about the corresponding insecticides, pesticides, etc.
  - So, government should increase awareness by creating some camps in the more cotton producing states or states with more queries on plant protection of cotton which is discussed in the next section. In those camps major focus should be on insects and fungus related issues.

6.5.3 Statewise analysis

This section contains the statewise analysis per crop.

- **Paddy**
  - Most of the plant protection queries for Paddy were asked in the states Uttar Pradesh, Punjab, Odisha, West Bengal in the decreasing order. These 4 states comprise more than 50% of the total queries.
  - According to the rice production per state data of India of 2014-15[3], West Bengal is the leading state and its followers are Uttar Pradesh, Punjab and Odisha. So, either West Bengal has moved a bit down in these years, or the farmers there have better knowledge of plant protection, or it may be the case that the KCCs there are not very active. If the last one is the case then government should take appropriate actions to improve the conditions.

- **Cotton**
  - Most of the plant protection queries for Cotton were asked in the states Maharashtra, Haryana, Gujarat in the decreasing order. These 3 states comprise more than 50% of the total queries.

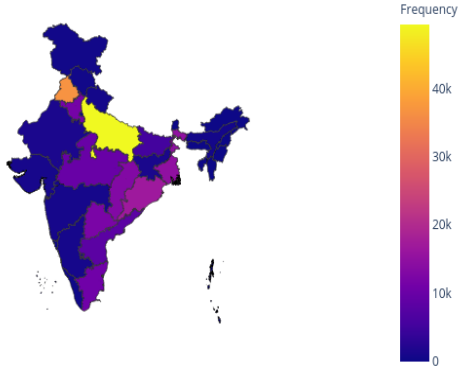


Figure 18: State wise Paddy queries

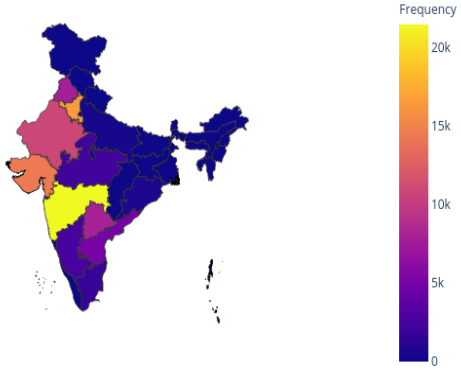


Figure 19: State wise Cotton queries

– According to the cotton production per state data of India of 2015-16[2], Gujarat is the leading state, Maharashtra is second and Haryana was 7^{th}. So, either Haryana is now a bigger producer of cotton relative to earlier times or there might be many plant protection issues in Haryana which is a big problem that the government should solve by conducting some camps to spread awareness about different methods for protection from insects and diseases.

6.5.4 Monthwise analysis

This section contains the monthwise analysis per crop.

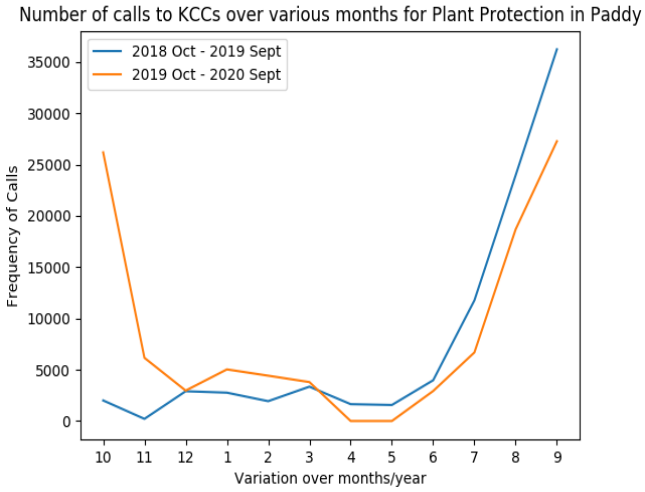


Figure 20: Month wise Paddy queries

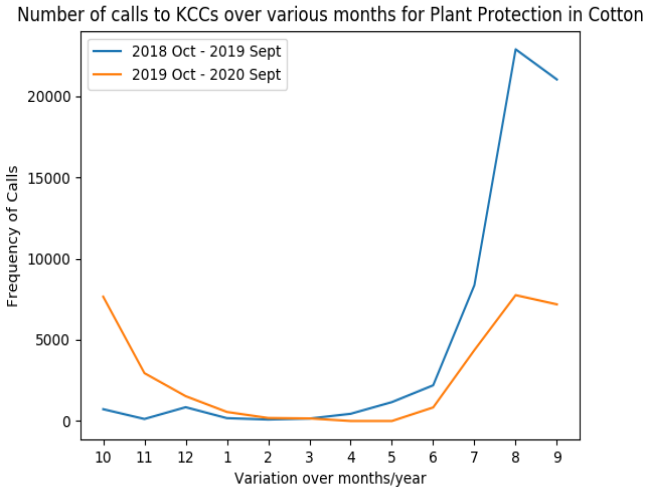


Figure 21: Month wise Cotton queries

- The graphs are made taking two sessions. First session is from 2018 Oct to 2019 Sept and the second session is from 2019 Oct to 2020 Sept. So, basically every point of second session is one year ahead of the first session.
- Both paddy and cotton are kharif crops and the kharif season starts in June and ends in October. It is evident from the plots that most of the queries are asked in the crop's season and the plots are almost concave with the lower part from November to June of next year.
- Also, in both the plots the second session has more queries before March which means that farmers are getting more aware of KCCs with time. But the second session in both the plots has always less queries than first

session after March and the most probable reason is that KCCs were very less active in the lockdown. Even, in April 2020 there are overall no queries because of lockdown.

- More awareness should be spread by government so that the farmers can ask plant protection queries before the crop's season to prepare in advance so that they will incur less loss and the nation would prosper more.

6.6 Weather related queries

6.6.1 Weather queries vis a vis total Queries

1. Weather related queries comprised of nearly 40 percent of all queries.A monsoon dependent agriculture leads to such high weather related queries.
2. The distribution of weather related queries vis a vis total queries is not uniform across states-
   (a) More than 10 states received more than 40 percent weather related queries.
   (b) On the other hand 5 states received less than 10 percent weather related queries.
3. The major reason for this is the variation in land under cultivation(Haryana has 59 percent weather related queries and Delhi having just 12 percent).

6.6.2 Temporal Distribution of Queries

- **Entire Country**

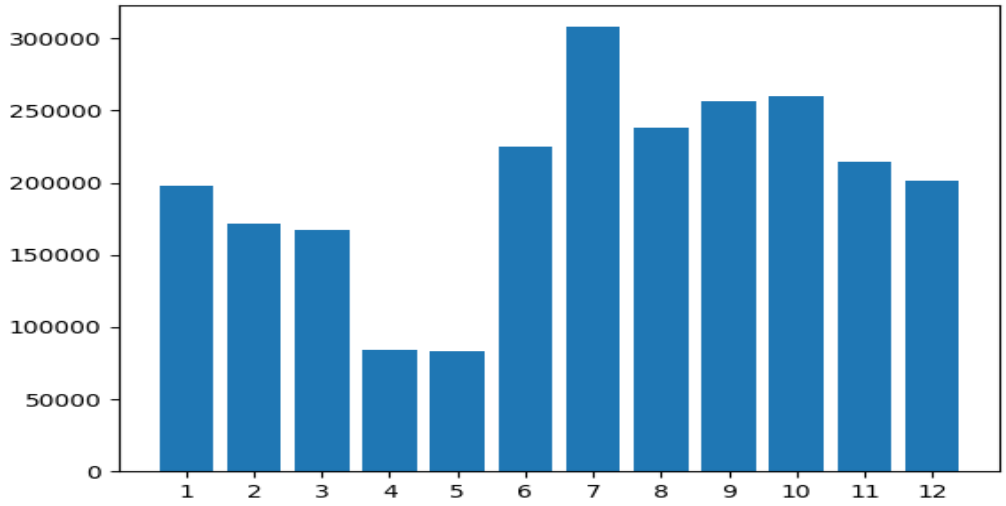


Figure 22: Month wise queries

1. **Summer Months** of April and May got fewer number of queries as weather is more or less stable and few crops are sown .
2. **Monsoon Months** of July got most number of queries due to advent of monsoon.
3. **Winter Months** of October got large number of queries due to withdrawing monsoon.

- **Intra-Country**

1. **July vs October :** States like UP,Punjab, Haryana have far higher queries in July compared to October (as they grow water intensive crops) while states like Maharashtra, Andhra Pradesh have higher queries in October (due to growth of rabi crops)
2. **Max Queries vis a vis states :**8 states have maximum number of queries in November.This number is higher than that of September and October despite November has lower number of total queries.This is because the states which have highest number of queries in November had lower total queries.

6.6.3 Spatial Distribution of Queries

- **Whole Year**

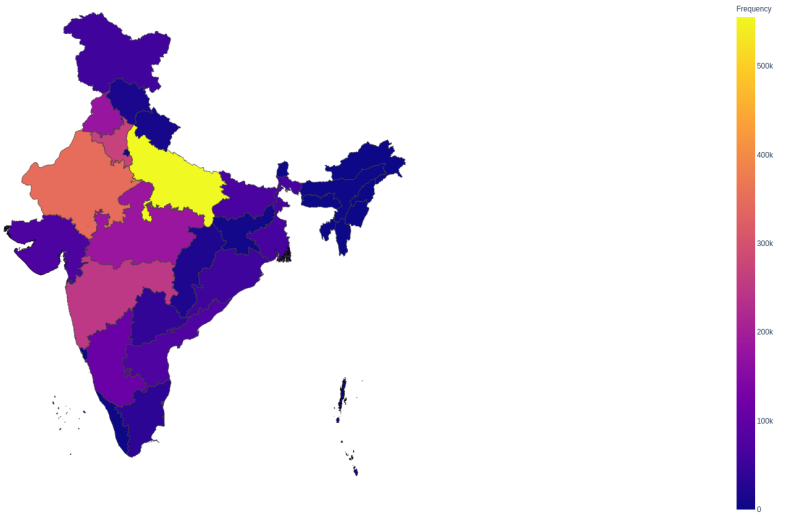


Figure 23: State wise queries

Dominating States

State	Percentage of queries in a state
Uttar Pradesh	23%
Rajasthan	14%
Haryana	11%
Maharashtra	10.5%
Madhya Pradesh	7.5%
Punjab	7.5%

These 6 states comprise nearly 75 percent of all weather related queries.The major reason could be that these states have higher agricultural production.Also these states have in general higher total queries. Other reasons for the same could be awareness among farmers,highly active KCC.

On the other side of the spectrum are those states which have got very few queries.These states have low agricultural production and in general lower queries.

Low Queries States

S.No.	State
1	North-East
2	East coast
3	Himalayan States

1. **Future Prospect**
   (a) Recognizing the poorly performing states
   (b) Adopting successful models of better performing states while assimilating to local conditions

- **Among months**

Summer Months

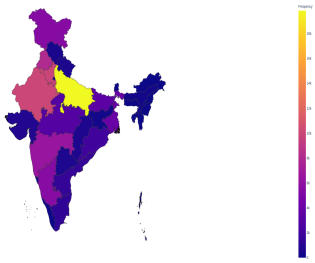


Figure 24: State wise queries

Winter Months

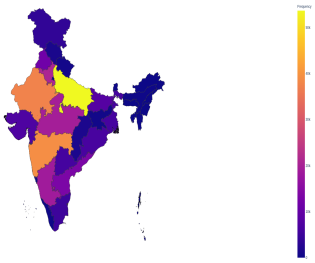


Figure 25: State wise queries

Summer Months

State	Number of queries in a state
Uttar Pradesh	19.9k
Rajasthan	9.9k
Haryana	9.8k
Punjab	7.9k
Maharashtra	6.5k

Winter Months

State	Number of queries in a State
Uttar Pradesh	53.6k
Maharashtra	39k
Rajasthan	37.3k
Haryana	20.6k
Madhya Pradesh	19k

- In general the number of queries in winter months are higher for individual states.
- The states with higher queries in winter months also tend to have higher queries in summer.This can be better appreciated using correlation analysis ahead.

Correlation with all query graph

1. The correlation of every month with total query was found which revealed-
   (a) **MAX :** was 0.986 and it was from August
   (b) **MIN :** was 0.903 and it was from November
2. From this it can be said that -
   (a) Even the minimum value of correlation was high.So the distribution of queries across states does not vary lot from total queries across states.
   (b) The month of August is closest to complete data and could be used as sampling for year long data.

6.6.4 Analysis From Specific Queries

1. **Frequency:**Only 1 percent of data had specific queries. Other types of queries did not have this low specific queries.This shows that farmers wanted general information.
2. **Rainfall:**
   (a) Around 71.5 percent of queries were from Odisha due to robust data collection by KCC.
   (b) Most number of rainfall related queries in August due to advent of monsoon.
3. **Cyclone:**Gujarat in June accounted for 76.92 percent of all cyclone related queries (due to cyclone Vayu and Nisarga).Lower queries from other states might be due to lack of awareness or poor data collection.
4. **Indian Meteorological Department (IMD) :** Queries regarding contact number of IMD was asked.This shows a progressive trend as the farmers wanted information from apex data collection body.Though 54 percent of all IMD related queries were from Gujarat and Uttar Pradesh.

7 Future Direction

7.1 Multi-Class Classification

- With the classification accuracy surpassing the baseline human accuracy for one task, there is scope for improvement in the other task.
- Combining these machine learning modules with speech processing and text summarization modules, we can develop a system wherein farmers' queries are classified in real time. After this online classification, the farmer's call can be answered by a computer or can be transferred to the relevant KCC executive. Thus, the need for difficult=to-use digital assistants. This would smoothen the process not only for the KCC executives but also for farmers.

7.2 FAQ Generator

- Current tool is unable to cluster similar sentences in different languages together like "yellowing of leaves" "pattiya pilli hori hai" together.
- It is able to cluster them into 2 separate clusters (i.e. similar sentences in each language together) but don't merge them since the mapped vectors are far enough
- Future development could be to solve this issue and increase the precision of the FAQs so obtained

7.3 Plant Protection

- The per crop statewise analysis can be done for each of the major diseases. This will help the government to identify that in which states the major diseases are spreading and then it can initially solve problems for those major states.
- Similarly, the monthwise analysis can be done for the major diseases instead of per crop. This will help to figure out that in which season there is more spread of a particular disease.

7.4 Government Scheme

- Number of queries for various schemes vs time can be used by the kcc to plan resources' allocation in future for that particular scheme.
- There are figures related to registration, finance, contact info for 4 schemes in each state this can be utilized by the authorities to ensure appropriate implementation logistics.

References

[1] https://www.tractorjunction.com/blog/top-10-agriculture-states-in-india/

[2] https://www.mapsofindia.com/top-ten/india-crops/cotton.html

[3] https://en.wikipedia.org/wiki/Rice_production_in_India

[4] https://beef2live.com/story-top-50-produced-foods-india-89-120768#:~:text=Sugar%20cane%20is%20the%20most,followed%20by%20rice%20and%20wheat.&text=Foods%20In%20India-,Sugar%20cane%20is%20the%20most%20produced%20food%20commodity,followed%20by%20rice%20and%20wheat.

[5] https://en.wikipedia.org/wiki/List_of_rice_diseases

[6] https://www.gkduniya.com/top-agricultural-producing-states-india

[7] https://github.com/UKPLab/sentence-transformers