DESIGN DOCUMENT

### CLOUD EVENTS

EVENT RECOMMENDATION SYSTEM

LIVE PROJECT - CLOUD COUNSELAGE PVT. LTD. IP PROGRAM

# 

# Introduction

Cloud Events is a recommendation system program that gives recommendations to employees of the company based on cognitive filtering. We are provided with an employee dataset which contains the name of the employees, their domain and 2 preferences for event types. Our main goal by our problem statement is to extract the domain and event type from an event heading using an ML approach. After their extraction a simple string matching algorithm is used to find employee names and those who match with extracted domain and event type are recommended.

# Real-Life-Scenario

Here we will visualize our Cloud Events system a bit as of how it will operate in a real life situation. Once the designated company employee receives the names of events that are going to be held within a few days, he/she will convert them into a CSV format. Our model is to read the CSV file and run the model for the inputs in the file. The employees to be recommended for a specific event are to be displayed on an excel sheet. This sheet will give us proper content the job of recommendation will get completed. This is the basic idea behind the problem statement and the constraints of our project.

# Techniques

For implementation of this project we are using a Natural Language Processing model with the help of the Bag-of-Words model. Though quite simple, it is a really powerful method for NLP models and has proved to be quite reliable in real life projects. We will use this model for extraction of domains and events. BoW represents text as a bag of words, disregarding the grammar and word order of the text. We will create a small text for each of our domains. The text will include description or any of the matter in relation with domain. The texts should define our domain so that BoW will contain words which become keywords for our domain. This technique requires a lot of time though it is quite useful if executed in the proper way.

# Program Functionality

# As mentioned in the techniques criteria we will implement a BoW representation in our NLP model. We will make use of the text stored for respective domains and create a BoW model. We will use CountVectorizer or TF-IDF for tokenization of keywords. Then we will use a classification model to train our domains and tokens. However we first filter out stop words like articles, prepositions and punctuations. This can be dealt with the help of the “re” library. We can create a diagrammatic representation of our functioning of the program.



After this we can make use of the tokenized text to train them with respect to their feature domain with the help of ML models. However in real life situations its not practical to make the train the model every time when someone accesses the recommendation system. Hence we can store the machine learning model inside a file and use them for prediction purposes only. We can store these models in pickle files. This is termed as serialization.



This ML model can be stored in a pickle file after training and later can be used to predict the tokenized input event. The output i.e. list of domains and events can be then used to find employees who are affiliated with them.

We can save these two lists in a dataframe to convert them into excel sheets.This is our basic program functionality. 

Our NLP implementation will however differ from the regular NLP solutions. We will use three ML models as our classification models.

# Program Constraints

The techniques implemented in the program are good though there can be many shortcomings in the project.

* For starters in the input event CSV file, the number of domains and events per instance has to be limited. The NLP model algorithm will work on the BoW tokens which are generated by using small text documents and average filtering techniques. So an event affiliated to five or more domains/ event types in a single instance will disrupt the algorithm.
* Following the above risk, there arises a problem of classification models failing so three base liner models are implemented for our project for extraction of domains and events.
* There is a strict constraint on recommendation by the problem statement of our project. If there are no employees for a certain domain and event type, the recommendation system should not recommend the employees of the same domain but different event types. E.g. If there are no employees to be recommended for Java Seminars then not even the employees who prefer Java Webinars or Talks should be recommended.
* The BoW model however may have some keywords similar to each other. For instance Object Oriented Programming Language goes for Java as well as Python, or Deep Learning may be allocated for Machine Learning or Artificial Intelligence. So domains with similar tokens have a chance of being recommended. This may decrease the accuracy of the model.
* The project doesn’t deal with the UI of the program nor as an executable file. Hence the code needs to be executed as a python code. Hence the program user should have the basic knowledge of the python language, code files and CSV input files, and other program directories.

# Performance

The project time complexity performance is one of the things that will not be easily comprehended. To tackle the quality of the output of the models, complex data structures must be implemented. The time complexity increases with employee data, text documents apprehend for tokenization for keywords, etc. Though there are no ML or DL algorithms that may require an accelerator or so, hence an average mid grade computer/ laptop may be able to handle the program quite easily.

# Security

This is an out-of-scope characteristic of the code that should be given priority towards, The employees might want to keep their preferences private to themselves and hence database encryption is necessary.

# Criteria for Project Completion

Cloud Events must be a reliable recommendation system that needs to satisfy all the problem statement’s constraints and goals. The ML algorithms applied for the project need to perform their tasks of predicting accurate results. The model should be flexible to be engaged for real life problems. It should make a positive difference in the company in its duration.