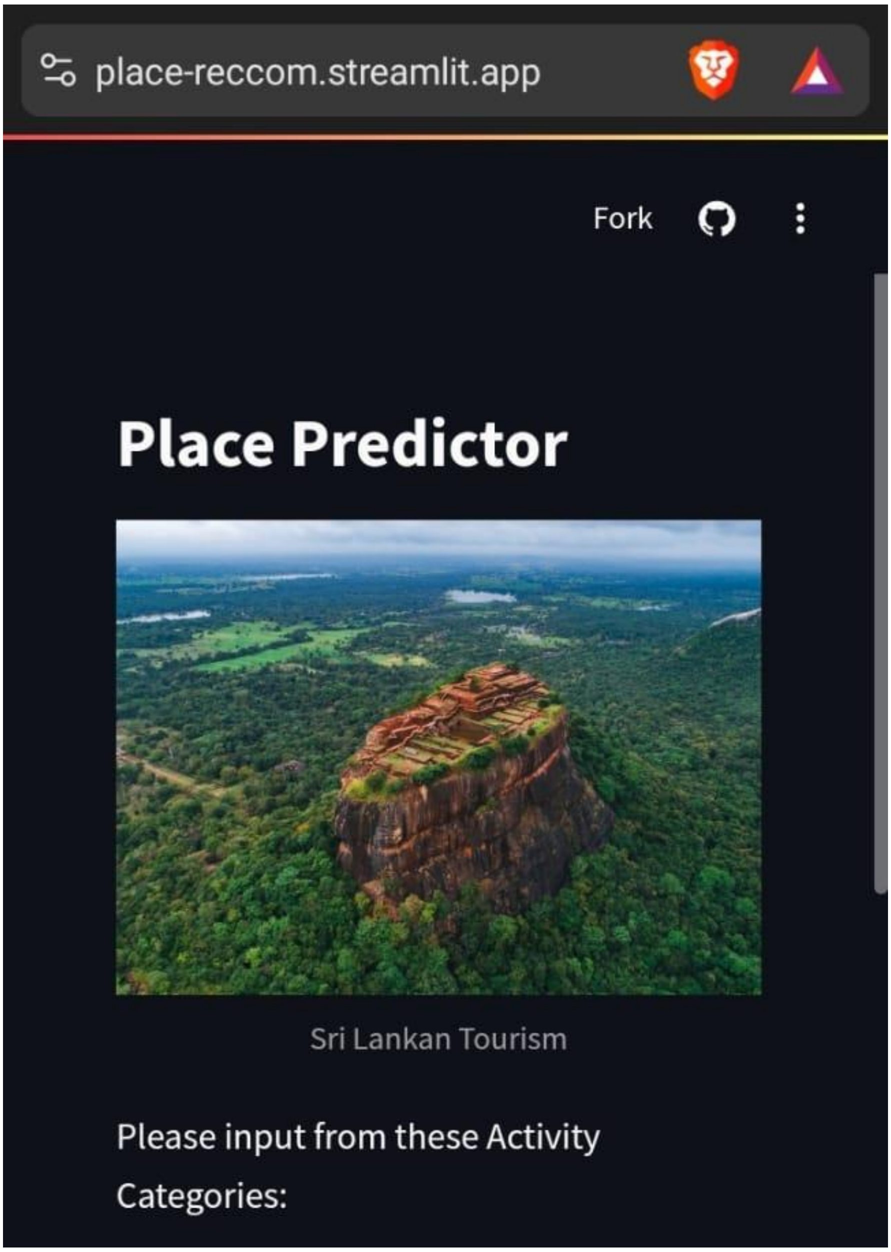
Team Neon Genesis – Rootcode Datathon Official Documentation

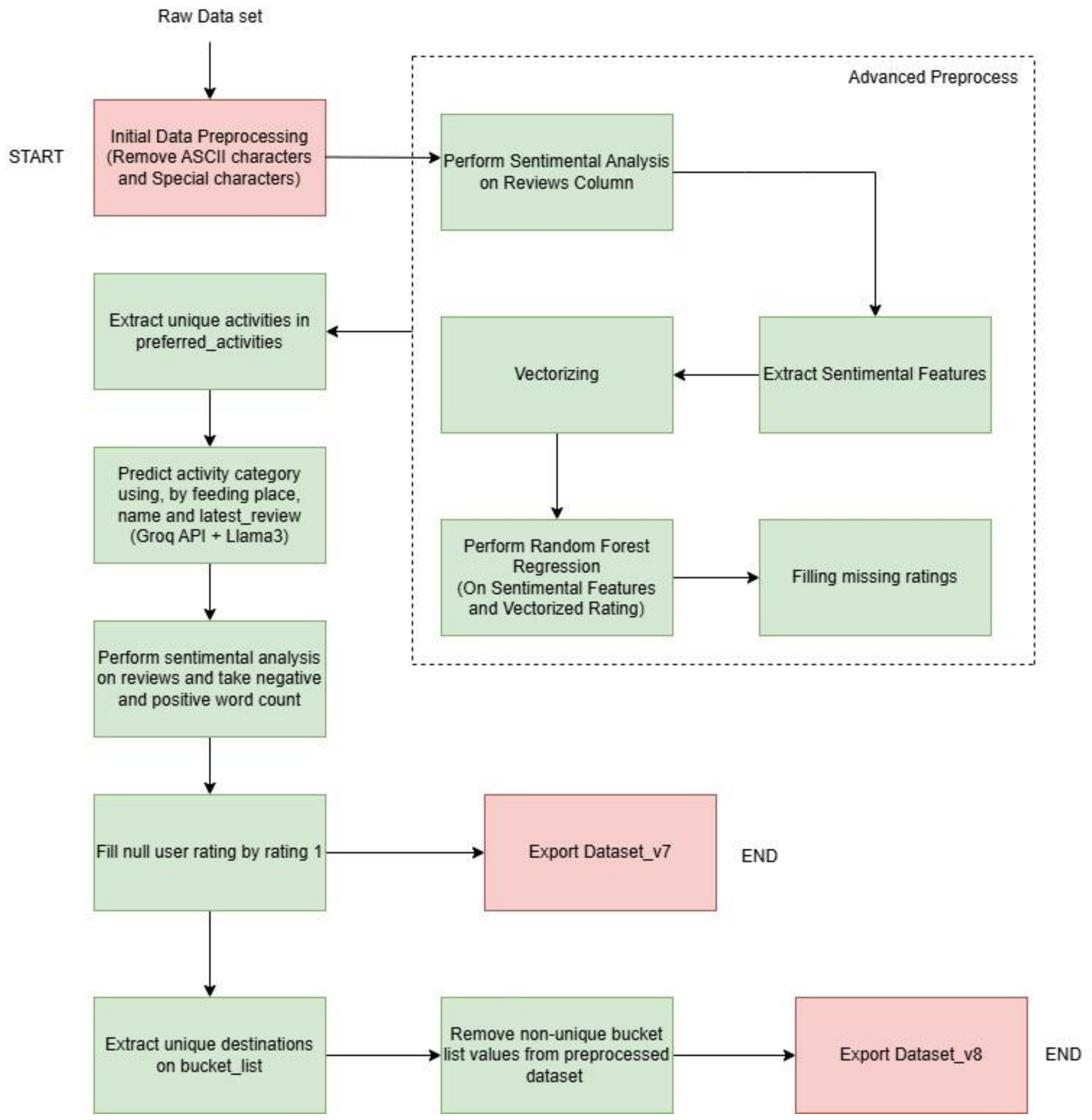


Place Predictor

Sri Lankan Tourism

Please input from these Activity Categories:

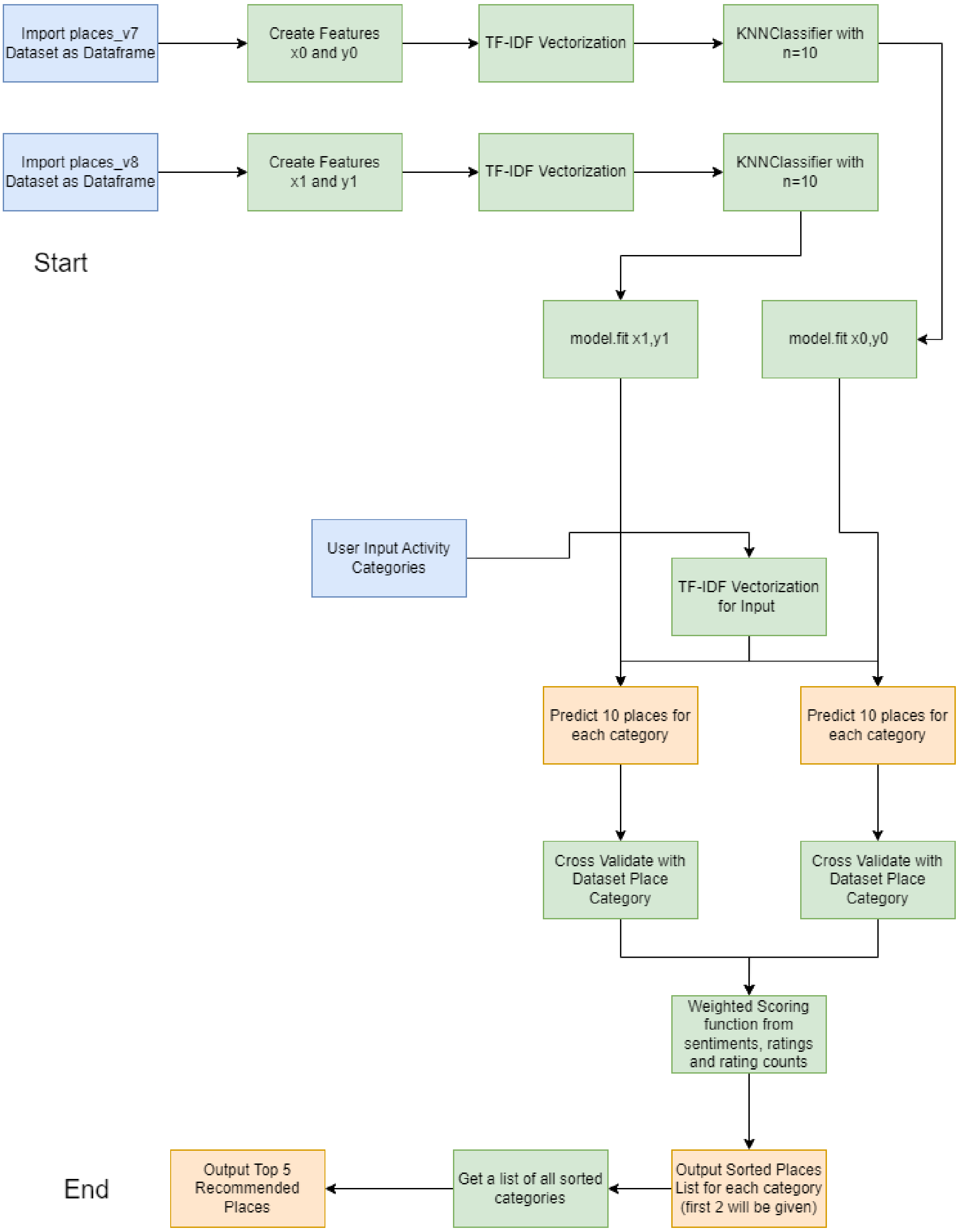
Data Preprocessing and Enrichment


Raw Data set
START
Initial Data Preprocessing (Remove ASCII characters and Special characters)
Advanced Preprocess
Perform Sentimental Analysis on Reviews Column
Extract Sentimental Features
Vectorizing
Perform Random Forest Regression (On Sentimental Features and Vectorized Rating)
Filling missing ratings
Extract unique activities in preferred_activities
Predict activity category using, by feeding place, name and latest_review (Groq API + Llama3)
Perform sentimental analysis on reviews and take negative and positive word count
Fill null user rating by rating 1
Export Dataset_v7
END
Extract unique destinations on bucket_list
Remove non-unique bucket list values from preprocessed dataset
Export Dataset_v8
END

- For Data Preprocessing first we did an analysis of the 2 datasets and foundout that these are some missing values, duplicates and ascii characters in the places dataset.
- Then we have started to perform a sentiment analysis on the latest review column to extract sentimental features by vectorizing using TF-IDF vectorization and using a Random Forest to fill missing rating values .

- After the sentiment analysis to extract unique activies from preferred activies in visitors info dataset and found that there are 68 unique activity types in this column.
- After that we used Generative AI to predict the Activity category for each destination in the places dataset which used Llama 3 70B model in Groq API.
- Then we Did a sentiment analysis count to get a positive and negative value count.
- Then we filled the null values in rating count column in places with 1's because most of the missing value destinations are cities or low rated places.
- Exported 1st dataset as places_v7.csv.
- Then removed non-unique bucket values form the places dataset and exported places_v8.csv because we saw that only less than half of the destinations mention in the bucket list destinations because those are the most value and famous tourist destinations in Sri Lanka.

Model Training, Evaluation & Deployment


Import places_v7 Dataset as Dataframe
Create Features x0 and y0
TF-IDF Vectorization
KNNClassifier with n=10
Import places_v8 Dataset as Dataframe
Create Features x1 and y1
TF-IDF Vectorization
KNNClassifier with n=10
Start
model.fit x1,y1
model.fit x0,y0
User Input Activity Categories
TF-IDF Vectorization for Input
Predict 10 places for each category
Predict 10 places for each category
Cross Validate with Dataset Place Category
Cross Validate with Dataset Place Category
Weighted Scoring function from sentiments, ratings and rating counts
End
Output Top 5 Recommended Places
Get a list of all sorted categories
Output Sorted Places List for each category (first 2 will be given)

Deployed App URL - https://place-reccom.streamlit.app/

- For the model training we used an approach in Ensemble Learning with 2 KNN classifiers to ensure the quality and performance of the predictions.
- And for the quality of the data we have took predictions for each individual input category and got the predictions from the 2 models. Then we have considered the

weighted score of the destination and prioritized the similar results of the model then predicted the best 5 destinations to the given user input

- As for the model deployment, we have used streamlit as our both front & backend to demonstrate the usability to the user.

Model Limitations

- For the best performance of the model we are recommending you to input 3 activity categories.
- And activity input keywords must be bound to the 68 unique categories given in the input description.

Problems We Faced

Choosing the exact model architecture for our project has been challenging, prompting us to explore various approaches including K-Nearest Neighbors (KNN), Ensemble Learning, Reinforcement Learning, and Ontology-Based Systems. We ultimately developed two models: a KNN model and an additional Q-Learning based agent system. We chose KNN for its simple and flexible nature, which makes it well-suited for recommendation systems. KNN easily integrates new data without retraining, handles nonlinear relationships effectively, and is ideal for dynamic environments with constantly updating data. The additional Reinforcement Learning model addresses continuously changing user reviews, allowing the model to learn from users on an ongoing basis. In managing user experience, we faced challenges with discrepancies between current recommendations and past experiences. To address this, we implemented a solution using Sentiment Analysis, developing a Random Forest model as a pattern matching system between "Rating" and "Users Review" columns to predict ratings for cells with null values. We also encountered evaluation challenges, as ratings are inherently subjective and based on human opinion. Assigning ratings to text with zero or null values proved particularly difficult. To overcome this, we utilized the aforementioned pattern matching system for more accurate predictions.