**Developer Manual**

# Introduction to Travel Service chatbot

Travelling is an act of movement from one place to another for various purposes. It can passion for someone, whereas it can be work, or need for another. It is an enviable part of our life. One should travel from one place to another for various country. These travels can be within a country or outside one. So, travelling is inevitable.

Travelling seems like one task but is associated with multiple task. One who travels needs to find a place to sleep, place to visit, place to eat and places to do various other day to day activities required for living. Traveller need to manage these things for all these purposes. Hence to help travellers to manage all their requirements, Travel service chatbot provide one stop solution at your fingertips.

Travel service chat bot help user to find their location, find nearby location, find tourist attraction, find gyms nearby, find hotels nearby, find restaurants in user location, predict user eating habit, etc. It is a system that is capable to incorporate user inputs through speech recognition. This travel service chat bot is capable to predict user mood and do language processing on their inputs. The chatbot used google map extension to facilitate user for location-based response about various facilities such as hotels, gyms, restaurant, travel places, etc.

# Web Service (API) incorporated

The travel service chat bot uses various API to fulfil the user requirement like location, places speech to text conversion, etc. The First API it uses is Web Speech API

## Web Speech API

In the travel service chat bot, the Web Speech API is utilised in the application to convert speech input of user into the text input. This API converts the voice command from the user and convert into text format that can be analyses by the system. It is useful API to provide audio-based interaction with the user of the Travel service chat bot.

## Text Analysis API

In the Travel service chat bot application, we use text analysis API to analyse the text and do the opinion mining and sentiment analysis. In travel service chat bot, we use this API to analyse the feedback from the user and mine their current mood based upon their feedback.

## Geo location API

In travel service chat bot application, the geolocation API is utilised to get the current location of user based on the google GPS services. When a user wants to know his location the geo location API is used to provide google map integration showing user location.

## Google Place API

In travel service chat bot application, the google place API is utilised to get the places near by the user current location. For example, if the user wants to know about hotels, restaurant, gyms, tourist attraction, etc. then the place API is utilised to set markers on the google map near the user location within the radius of 5000 meter from the user current location.

# Dialog Script for the Chatbot system.

The following in the dialog script for the travel service chat bot system

[When user starts the application and press the microphone button]

* User: Hello
* System: Hello Wenli
* User: How are you
* System: Good Thanks, how about you?
* User: show my location
* System: I am checking your location, wait a minute
* User: Show me where I am
* System: I am checking your location, wait a minute
* User: I want to have dinner
* System: Looking for places, where you can have delicious food. Yummmmmm…..
* If User says” wow nice result”
  + System: Thank you for your positive response, you look in a good mood.
* If User says” very bad result”
  + System: Thank you for your positive response, you look in a good mood.
* User: \*\*\*\*\*\*\*\*\* restaurant \*\*\*\*\*\*\*\*
* System: do you want to search restaurant?
* If User says “\*\*\*\* yes, \*\*\*\*\*\*\*”
  + System: Loading nearby restaurant.
  + If User says” wow nice result”
    - System: Thank you for your positive response, you look in a good mood.
  + If User says” very bad result”
    - System: Thank you for your positive response, you look in a good mood.
* If user says “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”
  + System: sorry I did not understand.
* User: \*\*\*\*\*\*\*\*\* Hotel \*\*\*\*\*\*\*\*
* System: do you want to search Hotel and lodge?
* If User says “\*\*\*\* yes, \*\*\*\*\*\*\*”
  + System: Loading nearby Hotel and lodge.
  + If User says” wow nice result”
    - System: Thank you for your positive response, you look in a good mood.
  + If User says” very bad result”
    - System: Thank you for your positive response, you look in a good mood.
* If user says “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”
  + System: sorry I did not understand.
* User: \*\*\*\*\*\*\*\*\* Lodge \*\*\*\*\*\*\*\*
* System: do you want to search Hotel and lodge?
* If User says “\*\*\*\* yes, \*\*\*\*\*\*\*”
  + System: Loading nearby Hotel and lodge.
  + If User says” wow nice result”
    - System: Thank you for your positive response, you look in a good mood.
  + If User says” very bad result”
    - System: Thank you for your positive response, you look in a good mood.
* If user says “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”
  + System: sorry I did not understand
* User: \*\*\*\*\*\*\*\*\* Gym \*\*\*\*\*\*\*\*
* System: do you want to search Gym?
* If User says “\*\*\*\* yes, \*\*\*\*\*\*\*”
  + System: Loading nearby Gyms.
  + If User says” wow nice result”
    - System: Thank you for your positive response, you look in a good mood.
  + If User says” very bad result”
    - System: Thank you for your positive response, you look in a good mood.
* If user says “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”
* System: sorry I did not understand
* User: \*\*\*\*\*\*\*\*\* tourist attraction \*\*\*\*\*\*\*\*
* System: do you want to search tourist attraction?
* If User says “\*\*\*\* yes, \*\*\*\*\*\*\*”
  + System: Loading nearby tourist attraction.
  + If User says” wow nice result”
    - System: Thank you for your positive response, you look in a good mood.
  + If User says” very bad result”
    - System: Thank you for your positive response, you look in a good mood.
* If user says “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”
* System: sorry I did not understand
* User: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* System: Sorry Wenli I don’t understand.

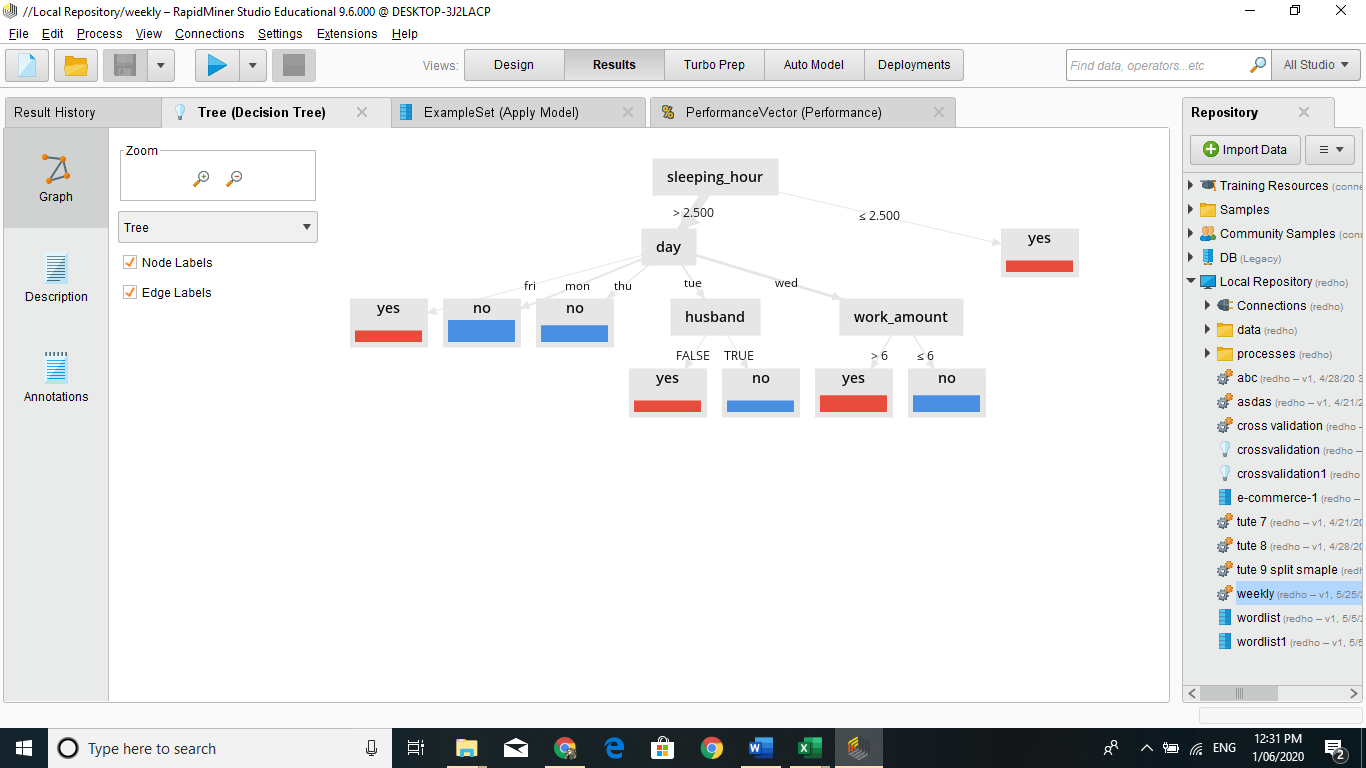
# AI Technique Utilised

## Opinion mining and Sentiment analysis

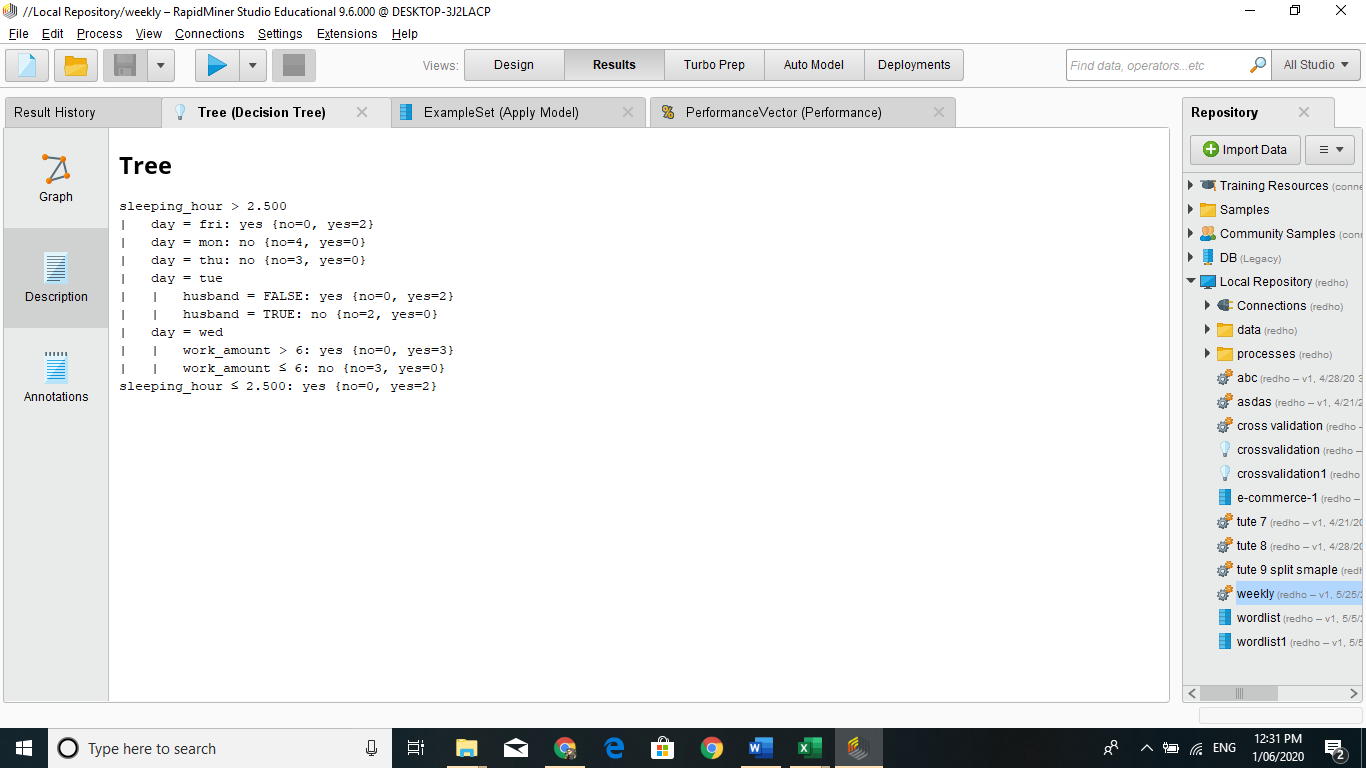
For opinion mining and sentiment analysis, this application has utilised the text analysis API to analyse the sentiment for the feedback of user. And based upon it, predicts the user mood, and displays upon the Application. The text analysis API does the semantic analysis of the user input data and replies with the proportion of negative, positive, and neutral response of the sentence. Based upon it the mood and sentiment of the user is described.

## Data classification

The data classification of the use past data is performed in the travel service chat bot. For the data classification, decision tree is developed, and the data of the past user is classified In, the dinner.csv file there are 4 columns, day, husband, work\_amount, sleeping\_hour and eatout using the data set and training data .This decision tree is developed:



*Figure 3.1 decision tree for past user eating habit*



*Figure 3.2 Decision rule model for decision tree*

# Past User Data Utilised

For the Past user data utilization, Dinner.csv file provided in the weekly quiz 1 is utilised. This dataset is used to predict the past user habit of eating outside or not. Various parameters like is husband or not, which day, sleep hours, work hours are utilised to determine the decision tree. Based on the decision tree the eating out pattern of the user is predicted.

The data set of the Dinner.csv is listed below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| day | husband | work\_amount | sleeping\_hour | eatout |
| mon | TRUE | 8 | 4 | no |
| tue | FALSE | 8 | 4 | yes |
| wed | TRUE | 8 | 5 | yes |
| thu | TRUE | 8 | 5 | no |
| fri | TRUE | 4 | 7 | yes |
| mon | FALSE | 5 | 3 | no |
| tue | FALSE | 5 | 3 | yes |
| wed | FALSE | 7 | 3 | yes |
| thu | FALSE | 7 | 6 | no |
| fri | FALSE | 8 | 2 | yes |
| mon | TRUE | 7 | 4 | no |
| tue | TRUE | 7 | 4 | no |
| wed | TRUE | 7 | 4 | yes |
| thu | FALSE | 8 | 3 | no |
| fri | TRUE | 6 | 5 | yes |
| mon | TRUE | 5 | 6 | no |
| tue | TRUE | 5 | 6 | no |
| wed | TRUE | 4 | 7 | no |
| thu | TRUE | 8 | 2 | yes |
| wed | TRUE | 5 | 6 | no |
| wed | FALSE | 4 | 7 | no |

For the creating the decision tree, we utilise the dinner test.csv

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| day | husband | work\_amount | sleeping\_hour | eatout |
| mon | TRUE | 8 | 4 | no |
| mon | FALSE | 8 | 4 | no |
| tue | FALSE | 5 | 3 | yes |
| tue | TRUE | 5 | 3 | no |
| wed | FALSE | 8 | 2 | yes |
| wed | TRUE | 7 | 4 | yes |
| wed | TRUE | 4 | 4 | no |
| thu | FALSE | 6 | 3 | no |
| thu | TRUE | 6 | 5 | no |
| fri | TRUE | 5 | 6 | yes |
| fri | FALSE | 5 | 6 | yes |

Using the Dinner pattern of user, the decision tree is developed and then it is applied with the dinner training set model. Using it we get the decision tree for our travel chatbot application.

# Any other issues

No major issues are found. The chat bot runs properly. People with non-English ascent may find it hard to make the chatbot understand what they want to express.

# User guide video link

Here is a short video link for how to use Travel service chatbot.

https://universitytasmania-my.sharepoint.com/:v:/g/personal/rkhadka\_utas\_edu\_au/EVK\_4skckrxHj8g4FlgHZBkB783u0jein91E3eSjDF97Kw?e=wKvPWq