Low-Level Design

Facebook status prediction

s.nitish raj

- LOW LEVEL DESIGN(LLD)

Contents

1. Introduction.....1

1.1. What is a Low-Level design document? 1

1.2. Scope.....1

2. Architecture.....2

3. Architecture Description.....3

3.1. Data Description.....3

3.2. Web Scrapping.....3

3.3. Data Transformation.....3

3.4. Data Insertion into Database.....3

3.5. Export Data from Database.....3

3.6. Data Pre-processing 3

3.7. Model Building 4

3.8. Data from User.....4

3.9. Data Validation.....4

3.10. User Data Inserting into Database.....4

3.11. model prediction.....

3.15. Facebook prediction & Saving Output in Database 4

3.16. Deployment.....4

4. Unit Test Cases 5

1. Introduction

1.1. What is a Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for the Facebook status prediction System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

2. Architecture

1. start	**2. data for Facebook status**	**3. web scrapping**	**4. data transformation**	**5. data insertion in databases**
6. export data from the database	**7. data preprocessing**	**8.nlp techniques**	**9. data vectorization**	**10. model building**
11. cloud setup	**12. pushing the app to the cloud**	**13. application start**	**14. data from user**	**15. data validation**
16. data insertion into the database	**17. bring data into prediction**	**18. model call from a prediction**	**19. Facebook status prediction**	**20. saving output at the database**

21. end				

3.1. Data Description

Facebook status prediction dataset is not a publicly available dataset. in order to get the data we have to use web scraping techniques and also we can take data present in some websites and add labels to it for this dataset we have nine ranges of emotions(fear, joy, sadness, surprise, anger, disgust, love, neutral) and this data contain two columns one is text and the second one is emotion each columns having 45,000 rows

3.2. Web Scrapping

In order to create a more great Facebook status prediction system emotion text collection we will need some more datasets which will contain emotion labels and text for the emotion.

3.3. Data Transformation

In the Transformation Process, we will convert our original data to CSV Format. and we add labels to text data.

3.4. Data Insertion into Database

a. Database Creation and connection - Create a database with the name passed. If the database is already created, open the connection to the database.

b. Table creation in the database.

c. Insertion of files in the table

3.5. Export Data from the Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

3.6. Data Pre-processing

Data Pre-processing steps we could use are Null value handling, stop words removal, punctuation removal, Tokenization, Lemmatization, counter vectorizer, label encoder, TFIDF, Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

3.7 model training

some of the algorithms which are well suited for a classification problem like (naive bais,logistic regression, random forest, and decision tree)are applied to the preprocessing data after trying a bunch of different algorithms we have taken the algorithm which gives the highest score after that we converted model into pkl file and used in the further part of ml lifecycle like prediction pipeline.

3.8. Data from User

Here we will collect some of the emotion statuses of people from facebook and other social networking apps and do some preprocessing of data.

3.9. Data Validation

Here Data Validation will be done, given by the user

3.10. User Data Inserting into Database

Collecting the data from the user and storing it in the database. The database can be either MySQL or Mongo DB.

3.11.model prediction

The model created during training will be loaded, and the user data will be predicted.

3.12. status prediction& Saving Output in Database

After calling the model emotion will be predicted, this output will be saved in Database and it will be used to show the same Output if other users provide the same data.

3.14. Deployment

We will be deploying the model to google cloud Kubernetes using docker

This is a workflow diagram for the Facebook status prediction...

Test Cases:

1. text case description	2. pre-requisites	3. expected result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether the User is able to sign up in the application	1. Application is accessible	The User should be able to sign up in the application
Verify whether user is able to successfully login to the application	1. Application is accessible 2. User is signed up to the application	User should be able to successfully login to the application

Verify whether user is able to see input fields on logging in	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to edit all input fields
Verify whether user gets Submit button to submit the inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get Submit button to submit the inputs
Verify whether user is presented with recommended results on clicking submit	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The recommended results should be in accordance to the selections user made
Verify whether user has options to filter the recommended results as well	1. Application is accessible 2. User is signed up	User should have options to filter the recommended results as well