High Level Design (HLD)

Facebook Post Status Prediction

Document Version Control

Date Issued	Version	Description	Author
09/09/2022	1	HLD – V1.0	Diana Laveena DSouza

Contents

Abstract

Many teenagers go through a difficult adolescence period, which can lead to depression. It's usually linked to the emotional and socioeconomic pressures they're under. This can lead to dangerous and violent conduct, substance misuse and self-harm. Young people use social media to stay in touch with friends, teachers, family, and other members of their peer group. As a result, it can be utilized as an efficient instrument for disseminating information to those young people who are affected. So, here we need to identify the emotional quotient of young people through their Facebook posts.

1   Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions before coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:
- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the system's structure, including the database architecture, application architecture, application flow and technology architecture. The HLD uses non-technical to mildly-technical terms, which should be understandable to the system's administrators.

1.3 Definitions

Term	*Description*
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment

Heroku	Google Cloud Platform

2   General Description

2.1 Product Perspective

The Facebook Post Status Prediction is a machine learning model to detect a person's emotional intelligence.

2.2 Problem Statement

To create a classification model to detect the emotional intelligence of the person using Facebook data and implement the following use cases.

- To detect the emotional intelligence of the person (abnormalities) and gives details to the concerned person for treatment.

2.3 Proposed Solution

The solution proposed here is to build a classification model to detect the abnormal behaviour of a teenager. In the above case, if the model detects abnormalities in the person, it gives details to the concerned authorities for treatment.

2.4 Data Requirements

Data requirements depend entirely on our problem statement.

- We need a large amount of data, nearly 16000 posts.

We use the Cassandra database by inserting records into it and then exporting them to a CSV file for further use.

2.5 Tools used

Python programming language and frameworks such as NumPy, Pandas, and Scikit-learn are used to build the whole model.
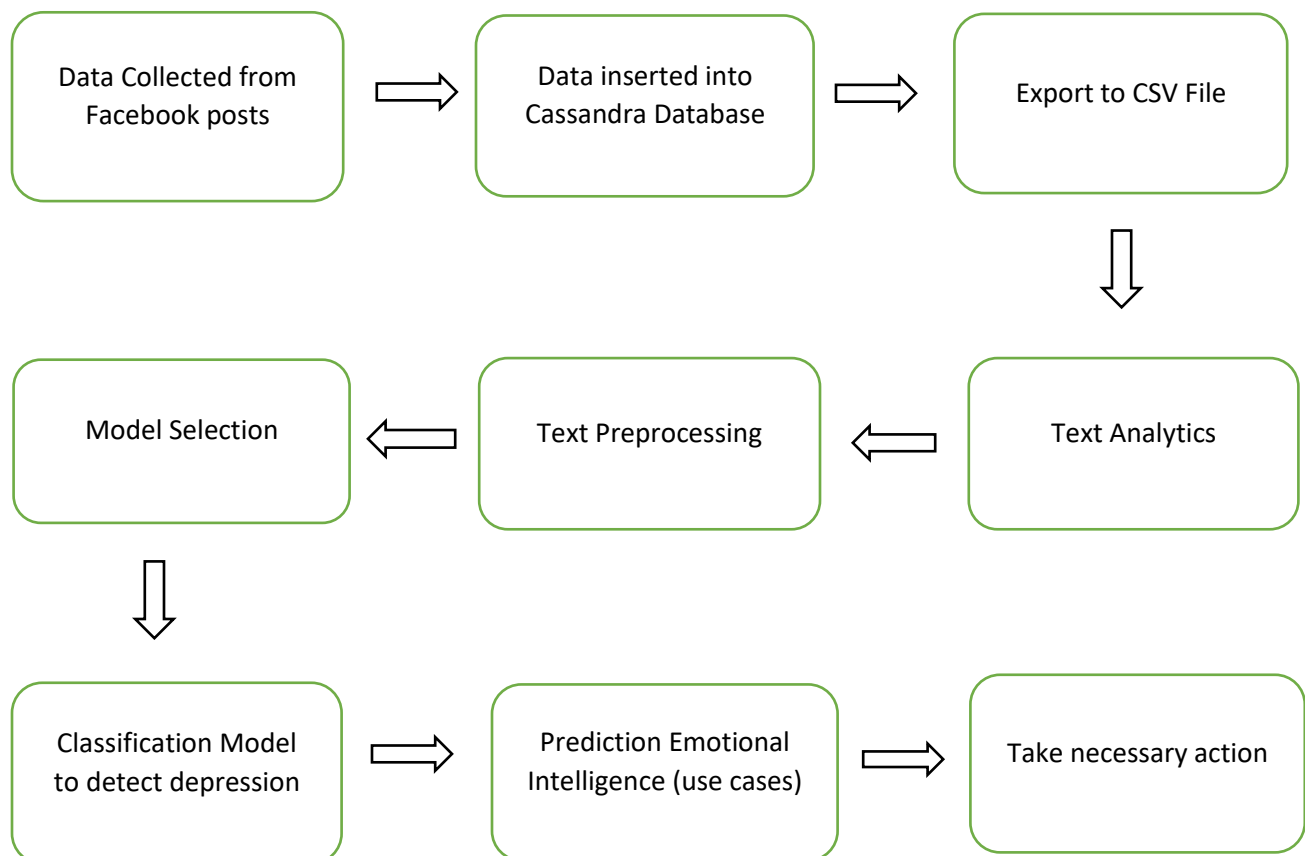
- PyCharm is used as IDE.
- For Text Analytics, we used, Matplotlib and Wordcloud are used.
- Heroku is used for the deployment of the model.
- Cassandra is used to retrieve, insert, delete and update the database.
- Front-end development is done using HTML.
- Python Flask is used for backend development.
- GitHub is used as a version control system.
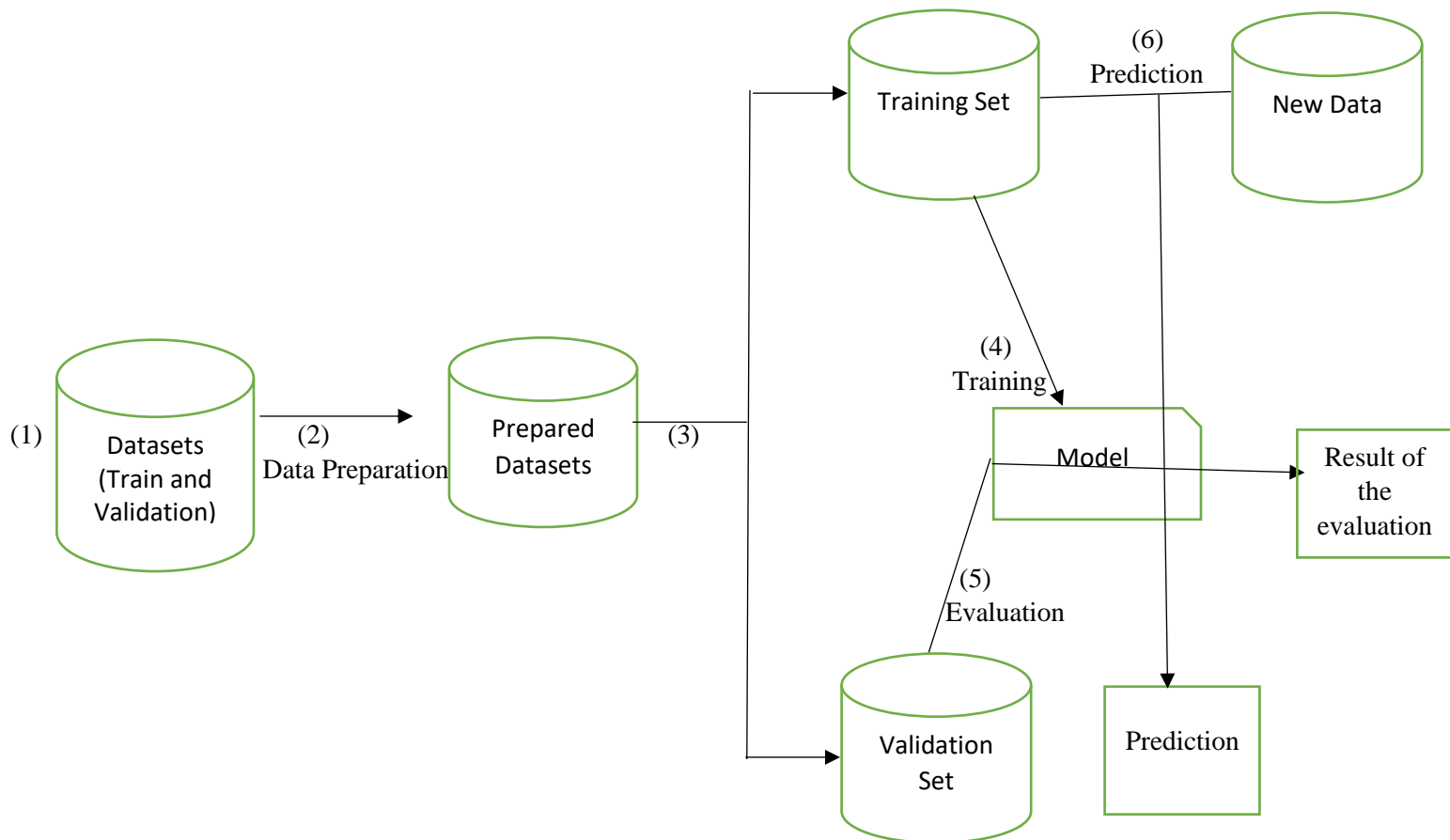
3   Design Details

3.1   Process Flow

To identify the different types of Emotional intelligence, we will use the classification model. Below is the process flow diagram is as shown below.

Proposed Methodology

```
┌──────────────────┐      ┌──────────────────┐      ┌──────────────────┐
│ Data Collected   │ ⟹   │ Data inserted    │ ⟹   │ Export to CSV    │
│ from Facebook    │      │ into Cassandra   │      │ File             │
│ posts            │      │ Database         │      │                  │
└──────────────────┘      └──────────────────┘      └──────────────────┘
                                                              ⇓
┌──────────────────┐      ┌──────────────────┐      ┌──────────────────┐
│ Model Selection  │ ⟸   │ Text             │ ⟸   │ Text Analytics   │
│                  │      │ Preprocessing    │      │                  │
└──────────────────┘      └──────────────────┘      └──────────────────┘
        ⇓
┌──────────────────┐      ┌──────────────────┐      ┌──────────────────┐
│ Classification   │ ⟹   │ Prediction       │ ⟹   │ Take necessary   │
│ Model to detect  │      │ Emotional        │      │ action           │
│ depression       │      │ Intelligence     │      │                  │
│                  │      │ (use cases)      │      │                  │
└──────────────────┘      └──────────────────┘      └──────────────────┘
```

3.1.1 Model Training and Evaluation

3.1.2 Deployment Process

```
   ┌─────────┐        ┌─────────────┐        ┌─────────────┐
   │  Start  │ ═════> │ Load Model  │ ═════> │ Input Data  │
   └─────────┘        └─────────────┘        └─────────────┘
                                                    ║
                                                    ▼
┌───────────────┐    ┌─────────────────┐    ┌──────────────────┐
│Predicted Result│<══ │ Make Prediction │<══ │ Preprocess Data  │
└───────────────┘    └─────────────────┘    └──────────────────┘
```
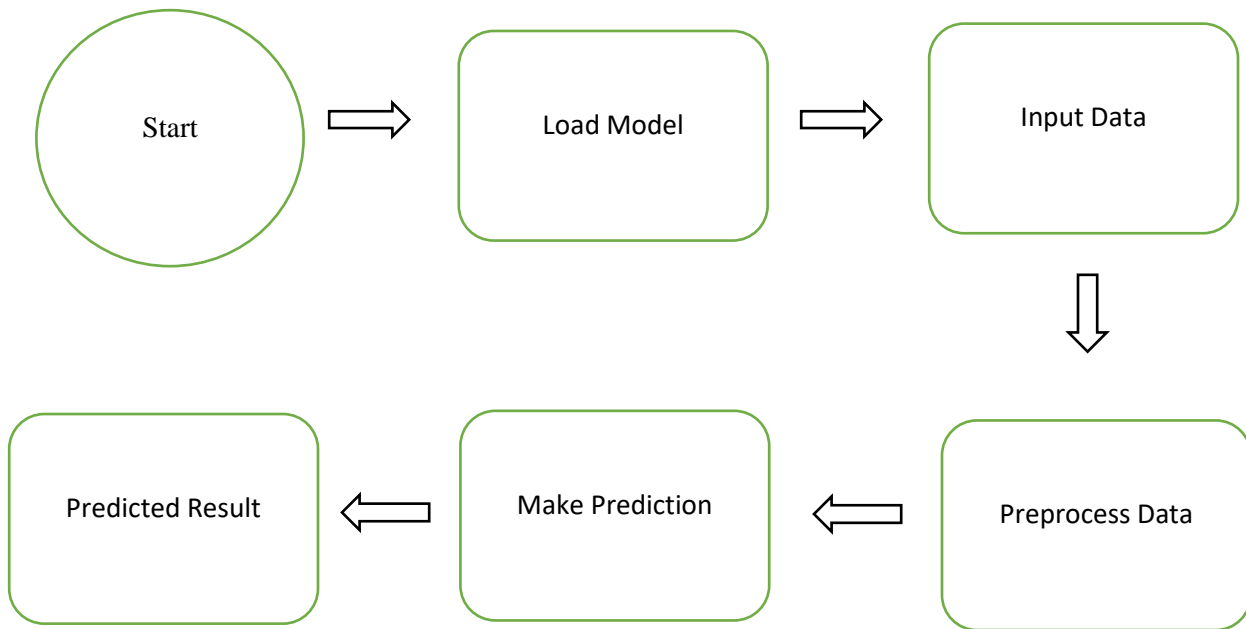
3.2 Event Log

The system should log every event, so the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging is required.
2. The System should be able to log each system flow.
3. Developer can choose the logging method. You can choose database logging/File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues, so logging is mandatory too.

3.3   Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

4   Performance

The Classification Model (SVC) is used to detect depression levels in teenagers. It will inform concerned authorities and takes necessary action, so it should be as accurate as possible so that it will not mislead the concerned authorities like doctors. Also, model retraining is very important to improve performance.

4.1   Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2   Application Compatibility

The different components for this project will use Python as an interface between them. Each component will have its task to perform, and it is the job of Python to ensure the proper transfer of information.

4.3   Resource Utilization

When any task is performed, it will likely use all the preprocessing power available until that function is finished.

4.4   Deployment

5 Conclusion

The Classification model will detect depression in teenagers based on Facebook data used to train our algorithm, so we can identify the abnormalities in the person's emotions and take necessary steps to reduce the person's abnormal behaviour.

6 References

1. https://www.hindawi.com/journals/cin/2022/4395358/