

**Low Level Design**

News Article Sorting

|  |  |
| --- | --- |
| Created By | Shraddha Manoj Gujar |

Index

1 . Introduction

* 1. What is a Low-Level design document?
  2. Applications

2. Architecture

3. Architecture Description

3.1 Data Description

3.2 Data Pre-processing

3.3 Data Modelling And Training

3.4 Best Fit Model

3.5 Saving Model

3.6 Front-end Website For project

3.7 Form Data from the website

3.8 Fed Data To ML model

3.9 Prediction

3.10 Deployment

4. Unit Test Cases

1. Introduction

1.1 What is a Low-Level design document?

* The LLD stands for Low-Level Design, in which the designer will focus on the components like a User interface (UI).
* LLD describes detailed description of each and every module means it includes actual logic for every system component and it goes deep into each modules specification.
* The goal a low-level design document is to give the internal logical design of the future state application architecture.
* Structure the design and start with the table of contents with the most important chapters of the document.
* LLD describes the class diagrams with the methods and relations between classes and program specs
* Low-level design is created based on the high-level design.
* It is also known as micro level/detailed design. It is created by designers and developers. It converts the High Level Solution into Detailed solution. It is created second means after High Level Design.

1.2 Applications

* LLD can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms.
* Designer and developer prepare the Low-Level Design.
* The output bases in the low-level design are the unit test plan and program specification.

1. Architecture



Data Validation

App Starts

Data from user

End

Display Output

Prediction

3.Architecture Description

33.1 Data Description

Dataset File Descriptions:

BBC News Train.csv - the training set of 1490 records

BBC News Test.csv - the test set of 736 records.

Initially raw data is stored as csv file for the project which was provided by iNeuron

Columns:

▪ ArticleId: Article id unique.

▪ Text: Text of the header and article.

▪ Category : Category of the article (tech,business,sport,entertainment,politics)

3.2 Data Pre-processing

Performing EDA to remove special character, remove stopwords, Lemmatization etc.

Check for null values in the columns. If present impute the null values.

Data cleaning: this step involves identifying and removing missing, inconsistent, or irrelevant data. This can include removing duplicate records, filling in missing values, and handling outliers.

Data integration: this step involves combining data from multiple sources, such as databases, spreadsheets, and text files. The goal of integration is to create a single, consistent view of the data.

Data transformation: this step involves converting the data into a format that is more suitable for the data mining task. This can include normalizing numerical data, creating dummy variables, and encoding categorical data.

Data reduction: this step is used to select a subset of the data that is relevant to the data mining task. This can include feature selection (selecting a subset of the variables) or feature extraction (extracting new variables from the data).

Steps Involved in Data Pre-processing:

1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

(a). Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

Ignore the tuples:

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

Fill the Missing values:

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

3.3 Data Modelling And Training:-

For feature extraction of text we have used CountVectorizer

Created many ML model (Logistic regression, Gradient boosting classifier, Random forest classifier, etc.)

We splitted our datset in training and testing set.

Before diving the data in training and validation set we performed preprocessing.

We splitted our datset in training and testing set.

Algorithms like KNN, Logistic Regression, SVM , Random Forest, Decision Tree Classifier etc were used.

Calculated Accuracy, Precision, Recall, F1-Score for each model.

Selected best fit final model and we saved that as Model.pkl.

Model selection :- We find the best model. After comparing Accuracy, Precision, Recall, F1-Score for each model we have selected best performing model.

3.4 Best Fit Model

We have found “Random forest” best fit for our data.

3.5 Saving Model:-

We saved our model as Model.pkl.

3.6 Front-end Website For project :-

Created “index.html”

Created Form for our features (independent variables) and used “Post” method.

We feed all that data to our saved best fit model ( Random forest model).

If our output is “0”, then the news is of business type, if our output is "1", then the news is of tech news, if our output is “2” then our the news is of politics type , if our output is “3” then our the news is of sport type , if our output is “4” then our the news is of entertainment type

Display this result on screen.

3.7 Form Data from the website:-

Form data is collected from user interface.

3.8 Fed Data To ML model:

Data filled in form is fed to ML model so that it will generate the result.

3.9 Prediction:

ML model will predict category of news and it will be displayed.

3.10 Deployment

Deployment is done on render.com

4. Unit Test Cases

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Prerequisite** | **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 |