**Low Level Design (LLD)**

**NEWS ARTICLE SORT USING NLP**

**Revision Number: 1.0**

**Last Date of Revision: 29/01/2023**

# **Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 29/01/2023 | **1** | **Initial LLD – V1.0** | **Yash Jain, Vinod Kumar Jodu** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Document Version Control 2](#_Toc126486804)

[Abstract 4](#_Toc126486805)

[1. Introduction 4](#_Toc126486806)

[Why this Low-Level Design Document? 4](#_Toc126486807)

[2. Architecture 5](#_Toc126486808)

[Training: 5](#_Toc126486809)

[Working in Production: 5](#_Toc126486810)

[3. Deployment 6](#_Toc126486811)

[4. Conclusion 7](#_Toc126486812)

# **Abstract**

In today’s world, data is power. With News companies having terabytes of data stored in Servers, everyone is in the quest to discover insights that add value to the organization. With various examples to quote in which analytics is being used to drive actions, one that stands out is news article classification. Nowadays on the Internet there are a lot of sources that generate immense amounts of daily news. In addition, the demand for information by users has been growing continuously, so it is crucial that the news is classified to allow users to access the information of interest quickly and effectively. This way, the machine learning model for automated news classification could be used to identify topics of untracked news and/or make individual suggestions based on the user’s prior interests.

# **1. Introduction**

# **Why this Low-Level Design Document?**

This document describes the low-level design for a News Article Sorting NLP assignment, which will sort news articles based on relevance and category.

# **2. Architecture**

## Training:

* **Data Ingestion**: Ingest training data and load it into Pandas Dataframe
* **EDA**: Explore columns using WordCloud and TFIDf.
* **Data Cleaning**: Removed rows with missing data. Removed stop words from Article columns. Tokenize the article rows
* **Data Transformation**: Transform Article column using TFIDF. Transform Category column using LabelEncoder
* **Choose Algorithm**: Try different algorithms for maximum accuracy on 20% test data. Eg., Logistic Regression, RandomForest. Once the model is final, save it into a pickle file to be used later.

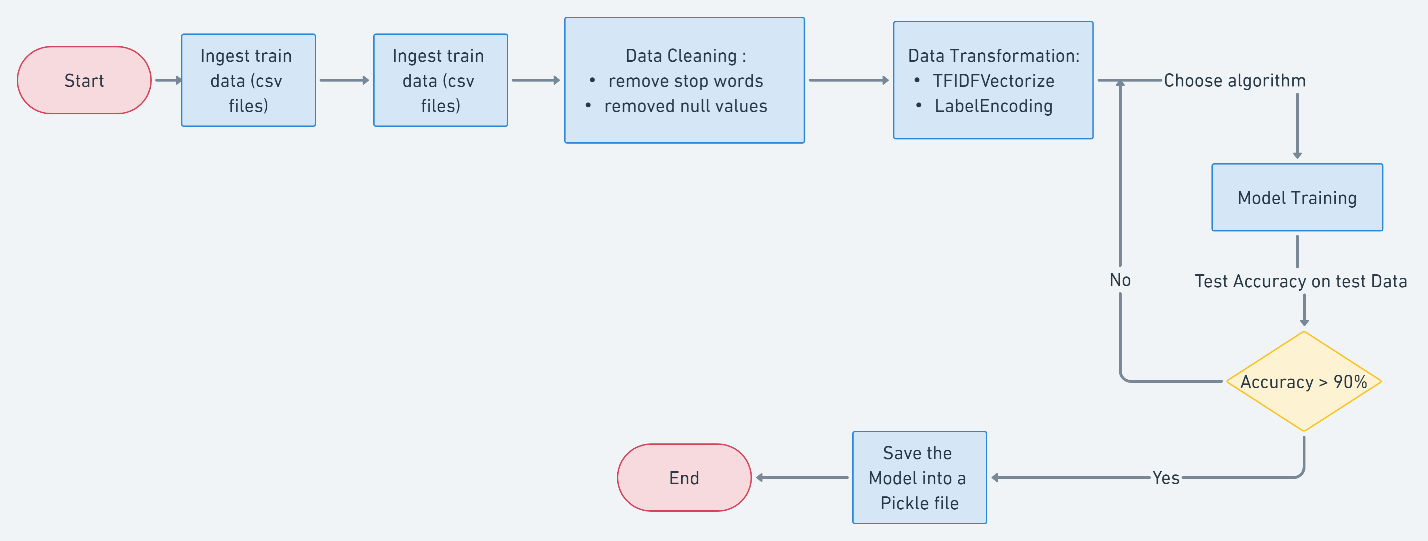


Figure 1 Architecture Flow chart (please zoom in for better clarity)

## Working in Production:

* Our utility which is deployed as Flask application has a Welcome screen where we can input the article to be sorted.
* On Hitting the ‘Predict’ button, it will call the /predict method at Backend API.
* This API would be internally, pre-process the input article and predict the category using trained model.
* Predicted category originally comes as LabelEncoded, so Inverse-Transformation would be done to get the category name as output.
* Output would be displayed on UI

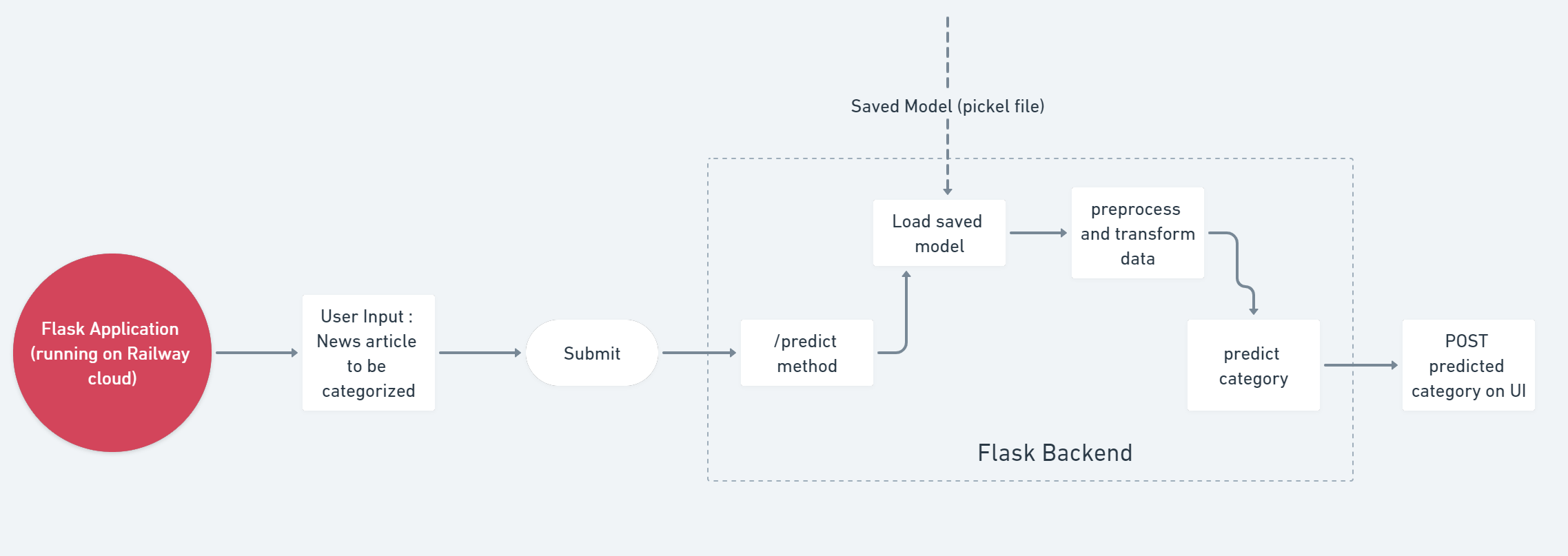


Figure 2 Working of Article sorting at production

* Top of Form

# **3. Deployment**

Deployment is done to **Railway.app** cloud service. Steps are here as follows:

* The code for Flask Application is pushed to a git repo.
* New project is created on Railway.app
* Choose “Deploy from Github repo” option.
* Choose your project repo and branch.
* Deployment would start automatically.
* Go to settings and choose “Generate DNS”

# **4. Conclusion**

The News Article Sorting NLP assignment will use NLP techniques and machine learning algorithms to sort news articles based on category. The system will provide a user-friendly interface for searching articles, making it easy for users to find relevant information.