Abstract

Details all the artifacts in the gitlab repository

project Repurpost

Read Me Document



Saju Chacko Rajan

Atharva Shantanu Kulkarni

Yashwanth Balan Arumugam

Pragya Avinash Mishra

Shruti Sham Kotwal

Table of Contents

[Introduction 2](#_Toc120800598)

[Web API – Render.com 2](#_Toc120800599)

[API Contracts 3](#_Toc120800600)

[Auto Suggest Tags 3](#_Toc120800601)

[End Points 3](#_Toc120800602)

[API Input 3](#_Toc120800603)

[API Output 3](#_Toc120800604)

[Web Interface - Retool.com 4](#_Toc120800605)

[Home App 4](#_Toc120800606)

[Content Creation App 4](#_Toc120800607)

[Repository Details 5](#_Toc120800608)

[/documents/ 5](#_Toc120800609)

[Repurpose high level solution diagram.pdf 5](#_Toc120800610)

[Repurpost proposed wireframe.pdf 5](#_Toc120800611)

[Repurpost user flows.pdf 5](#_Toc120800612)

[/datasets/ 5](#_Toc120800613)

[/notebooks/ 6](#_Toc120800614)

[Repurpost\_StackOverFlow\_Raw\_Dataset\_Cleanup.ipynb 6](#_Toc120800615)

[Repurpost\_Multi\_Label\_Classification\_(SVC, SGD, Linear Regression) Model\_Building.ipynb 6](#_Toc120800616)

[Repurpost\_StackoverFlow\_CNN.ipynb 7](#_Toc120800617)

[Repurpost\_StackOverflow\_BERTModel.ipynb 7](#_Toc120800618)

[Repurpost\_StackOverflow\_LSTM\_RNNModel.ipynb 7](#_Toc120800619)

[/api/ 7](#_Toc120800620)

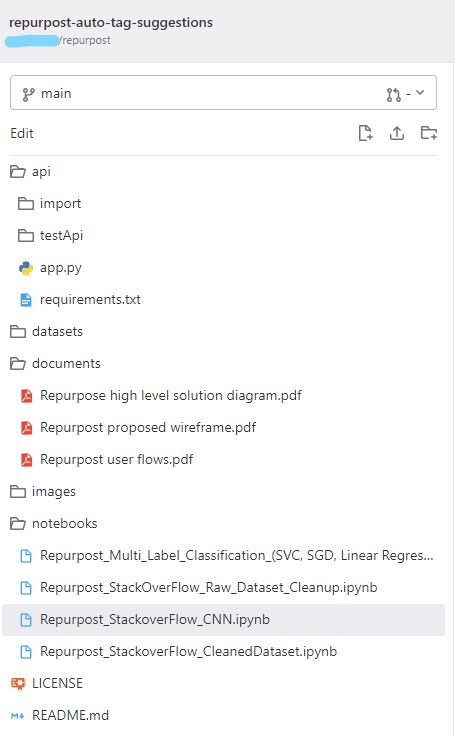
[/import/model\_all.pkl 7](#_Toc120800621)

[requirements.txt 8](#_Toc120800622)

[app.py 8](#_Toc120800623)

Read Me – Auto Suggest Tags

# Introduction

This project is about creating a performing a proof-of-concept web application that can auto suggest tags and content classification based on the content text that is provided by the users on the Repurpost platform.

All the work done by NEU team Repurpost has been uploaded into the GitLab repository available here - [repurpost-auto-tag-suggestions](https://gitlab.com/sajucrajan/repurpost)

The purpose of this document is to logically represent each artifact in the repository for the sponsor to use in the next phase of implementation. The repository structure is available in the image provided.

The repository is divided into folders to logically store certain type of artifacts and also support build and deployment from the same repository. Detailed description of each folder structure and files will be available later in the document.

# Web API – Render.com

Render.com is a unified cloud that allows to create, build and deploy apps, web api’s, and websites.

## API Contracts

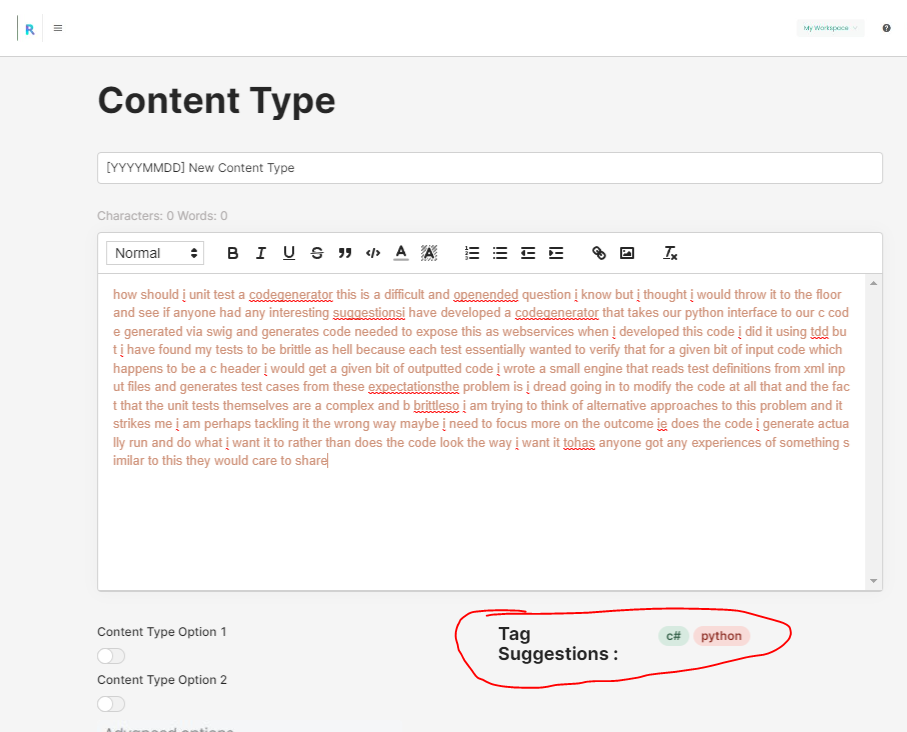
|  |  |  |
| --- | --- | --- |
| Auto Suggest Tags When the user is creating content on the Repurpost platform, it is required to suggest and create tags based on the content. The endpoint provides the facility to pass in the content and receive tag suggestions. More information is available in the sections below | | End Points / tagSuggestsAPI |
|  | |  |
| API Input | |  |
| content – string |  | |
| The text data from the content description. The maximum length of the content is not restricted currently, but based on further testing, the content length will need to be restricted. | **{**  **“content”: “This is the python and net”**  **}** | |
|  |  | |
| API Output |  | |
| [list of strings] |  | |
| The output contains the list of tags that the ML model predicts. The list might be empty or have one or more tags in the return list object. | [[  ‘python’,  ‘.net’  ]] | |

# Web Interface - Retool.com

Retool is a low-code platform that allows rapid and easy to build internal tools. Retool platform provides building blocks that can be easily used to create web interfaces. Retool provides creation of pages and provide easy integration with web apis.

Retool supports has mainly divided the components into two sections Apps and Resources. All the web interfaces are categorized as Apps and Web API’s and Databases are categorized as Resources. For this project, we have created two Apps (web interfaces) and one Resources (web API integration).

## Home App

Home page is very similar to Repurpost’s home page where it includes a dashboard and a create section. The create section event will navigate the web interface to the Content Creation App.

## Content Creation App

This is the main page for providing textual content for the post. When the user provides the textual content, the web API is invoked that generates tags based on the content.

# Repository Details

## /documents/

Contains all the documents that has been created for this project.

### Repurpose high level solution diagram.pdf

This solution's design and architecture follow a typical pattern for web applications and web APIs. This architecture's primary distinction is that the web API employs a multi-label classification model that has been pretrained to make tag suggestions based on user-provided text content in the web interface.

### Repurpost proposed wireframe.pdf

A quick and simple way to visualize a user interface, detect any problems, and get immediate input from sponsors and stakeholders. sketch skeletons of an interface.

### Repurpost user flows.pdf

The Repurpost team investigated the platform and determined the present user flow. The tag suggestions will appear on the main page itself based on the current user flow.

### Repurpost Project Reflection Paper.pdf

The reflection paper consists of the entire journey of the project that the NEU team went through from project initiation to completion. It takes the reader through the tool selections, datasets search, reference documents and links to model creation, evaluation and deployment.

### Repurpost - Project Plan.pdf

The project plan consists of the plan that was put in place during the initial phase of the project. It also was updated to keep pace with the changes in the plan and delivery timelines.

## /datasets/

***stackoverflow.csv  
stackoverflow\_final.csv***

Since it is not possible to work on real client data from Repurpost, another dataset was identified that has a title and body as separate fields along with multi-label tags which will allow the ML model to trained in close alignment with Repurpost content data. The dataset is 1.7 GB is size and contains 1.25 million features. This dataset contains 3 main parts from a stack overflow site whereby these 176 communities are part of a network that was founded and is managed by professionals and enthusiasts who are passionate about a certain subject. The website primarily acts as a forum for users to post queries and receive responses. For the purpose of the project, we have decided to choose the Answers and the Tags dataset as we are focusing on multilabel tag classification problem in the part of module. Every answer has a tag identifying its subject. Since a answer could be related to numerous topics, each might have up to five tags. Tags make it simple to identify answer/textual content related to your interests so you may either learn from the answers offered or, if the question is relevant to your field of expertise, be able to respond to it. The user may enter the tags when submitting the question, or network must guess them based on the question. The dataset does require pre-processing and some clean-ups before it can be used for applying NLP algorithms.

## /notebooks/

Contains the turning wheels of this project with Code implementation of the ML Model Design and implementation of multi label classification models which include - SVC, SGD, Linear Regression |-| CNN |-| BERT classifier.

### Repurpost\_StackOverFlow\_Raw\_Dataset\_Cleanup.ipynb

This notebook contains code for cleaning the dataset, this will guide implementation of BeautifulSoup package for use of tokenizer and in removing punctuations, white words, and removal of hyphens from the dataset.

### Repurpost\_Multi\_Label\_Classification\_(SVC, SGD, Linear Regression) Model\_Building.ipynb

This notebook will guide on developing model that examines a text remark and forecasts a number of labels connected to the questions Multi-Label Text Classification, a subset of the multiple output model, will be used to construct a tag suggestion system. Additionally, it will assist in the implementation of text vectorization on text data, the encoding of tag labels using MultilabelBinarizer, and the modeling of classical classifiers (Linear SVC, SGD and Linear regression) for comparison of results. The accuracy and loss of the model is measured using fucntions for jaccard score and hamming loss and code is available in this notebook. This notebook also contains the pickle code for implementing all\_model.pkl used in WebAPI and mentioned later in the document.

### Repurpost\_StackoverFlow\_CNN.ipynb

This notebook guides on implementation of CNN model for suggestion of tags, model involves embedding layers, Conv1D layer, Global Max pool and dense layers of the model for predicting tags.

### Repurpost\_StackOverflow\_BERTModel.ipynb

This notebook will help you in the implementation of BERT model which is a bidirectional ML model, this will also help in implementing Huggingface package of BERT model which involves pretrained models, tenserflow and pytorch which will help in predicting tags.

### Repurpost\_StackOverflow\_LSTM\_RNNModel.ipynb

This notebook will be a guide to implement the LSTM\_RNN model for tag prediction for dataset which involves methods of tokenization of the cleaned textual data, Multilabel binarization of tags, Tf-IDF vectorization. After vectorization, this model defines LSTM where different layers like embedding, and LSTM layers are implemented after reshaping the hidden states generated between the input and output states of the model. Training and Validation losses are calculated after epochs are generated. Finally, the predictions are made.

## /api/

Contains all the code and model that is required to build and deploy the web API using Flask and render.com.

### /import/model\_all.pkl

This is serialized export of the models that are required to make the web API functional. This is a pickle of the model which will be used in the ML model to get the desired output of tags. On deserialization a list of models will be available.

At index 0 - the OneVsRestclassifier using the Linear SVC classifier model.  
At index 1 - the TfidVectorizer to convert input text to vectors  
At index 2 - the Multilabel Binarizer to inverse transform the predicted tag arrays.

### requirements.txt

Includes the list of packages and modules that are necessary to be installed for build and deployment of the API.

### app.py

Contains the python code that implements the flask API request and the use of model\_all.pkl to use ML models used for implementing the logic to provide tags suggestions. There are two endpoints in the code. The POST method API is used for the tag suggestions and the GET method API is used for deployment testing.