

Managing ML Data and Models For Spam Email Detection - Final Report

11/29 Zhipeng Zhu

zhipengz@usc.edu

B.A in Physics; USC Applied Data Science

Current Sessions: 550 & 551

Skills: Python, SQL, Regression Models; No prior ML experience, No web UI experience

● Objective

This project aims to build a data and model management system for spam email detection. It will allow users to perform data management tasks such as loading, transforming, and exploring email data. It will also serve as a platform for users to access machine learning models and make predictions on if an email is a spam.

● Dataset

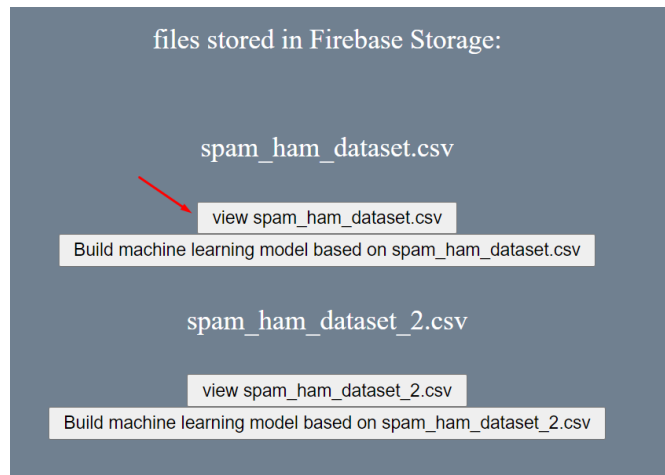
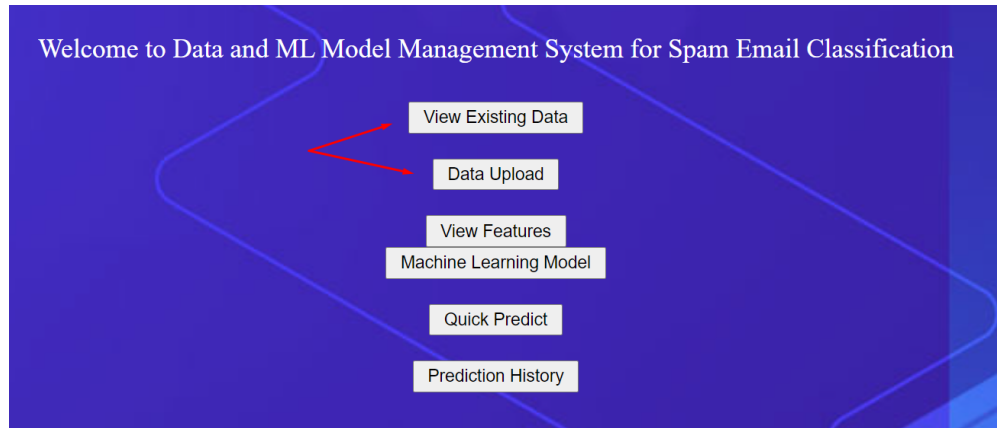
<https://www.kaggle.com/venky73/spam-mails-dataset>

The dataset contains 4 columns: email id, label, text, and label_num. Spam emails are labeled “spam” with corresponding label_num “1”. Non-spam emails are labeled “ham” with corresponding label_num “0”. There are 5171 total records in this dataset.

| | A | B | C | D |
|----|------|---------|---|-----------|
| 1 | id | label | text | label_num |
| | | | Subject: enron methanol ; meter # : 988291 | |
| | | 605 ham | this is a follow up to the note i gave you on monday , 4 / 3 / 00 { preliminary flow data provided by daren } . | 0 |
| 2 | | | please override pop ' s daily volume { presently zero } to reflect daily activity you can obtain from gas control . | |
| | | | this change is needed asap for economics purposes . | |
| | | | Subject: hpl nom for january 9 , 2001 | |
| 3 | 2349 | ham | (see attached file : hplnol 09 . xls) | 0 |
| 4 | 3624 | ham | Subject: neon retreat | 0 |
| 5 | 4685 | spam | ho ho ho... we're around to that most wonderful time of the year... neon leaders retreat time! | 1 |
| 6 | 2030 | ham | Subject: photoshop , windows , office . cheap . main trending | 0 |
| 7 | 2949 | ham | phasesments d'arar prudently fortuitous undergone | 0 |
| 8 | 2793 | ham | Subject: re : indian springs | 0 |
| 9 | 4185 | spam | this deal is to hook the tech nvr revenue... it is my understanding that tech | 1 |
| 10 | 2641 | ham | Subject: ehronline web address change | 0 |
| 11 | 1870 | ham | this message is intended for ehronline users only | 0 |
| 12 | 4922 | spam | Subject: spring savings certificate - take 30 % off | 1 |
| 13 | 3799 | spam | save 30 % when you use our customer appreciation spring savings | 0 |
| 14 | 1488 | ham | Subject: looking for medication ? we're the best source . | 0 |
| 15 | 3948 | spam | it is difficult to make our material condition better by the best law... but it is easy enough to ruin it by bad laws | 1 |
| 16 | 3418 | ham | Subject: noms / actual flow for 2 / 26 | 0 |
| 17 | 4791 | spam | we agree | 1 |
| | | | Subject: nominations for oct . 21 - 23 , 2000 | 0 |
| | | | (see attached file - hplnol 021 . xls) | |
| | | | Subject: vocable % rnd - word asceticism | 1 |
| | | | yes... brand new stock for your attention | 0 |
| | | | Subject: report 01405 ! | 1 |
| | | | wf fur attion from est inst supplied 1 post our give asently rest | 0 |
| | | | Subject: enron / hpl actuals for august 28 , 2000 | 1 |
| | | | tech ran 20 , 000 / enron : 120 , 000 / hpl gas daily | 0 |
| | | | Subject: vic . odin n ^ ow | 1 |
| | | | horne hothox carnal bride cutworm dyadic | 0 |
| | | | Subject: tenaska iv july | 1 |
| | | | darren : | 0 |
| | | | Subject: underpriced issue with high return on equity | 1 |
| | | | stock report | 0 |
| | | | Subject: re : first delivery - wheeler operating | 1 |

● Functionalities

1. **Data exploration & Ingestion:** The user can upload CSV files with columns listed above into the system, and then the files will be stored in Firebase storage. The user can also see the list of all files stored in Firebase storage and open a specific file to view the data and metadata.



File size is 5374.0 KB

5171 rows

4 columns

attributes: id, label, text, label_num

table view of this dataset:

| | id | label | text |
|---|------|-------|--|
| 0 | 605 | ham | Subject: enron methanol ; meter # : 988291\r\nthis is a follow up to the note i gave you on monday , 4 / 3 / 00 { preliminary\r\nflow data provided by daren } .\r\nplease override pop 's daily volume { presently zero } to reflect daily\r\nactivity you can obtain from gas control .\r\nthis change is needed asap for economics purposes . |
| 1 | 2349 | ham | Subject: hpl nom for january 9 , 2001\r\n(see attached file : hplnol 09 . xls)\r\n- hplnol 09 . xls |
| 2 | 3624 | ham | Subject: neon retreat\r\nho ho , we ' re around to that most wonderful time of the year - - - neon leaders retreat time !\r\ni know that this time of year is extremely hectic , and that it ' s tough to think about anything past the holidays , but life does go on past the week of december 25 through january 1 , and that ' s what i ' d like you to think about for a minute .\r\nnon the calender that i handed out at the beginning of the fall semester , the retreat was scheduled for the weekend of january 5 - 6 . but because of a youth ministers conference that brad and dustin are connected with that week , we ' re going to change the date to the following weekend , january 12 - 13 . now comes the part you need to think about .\r\ni think we all agree that it ' s important for us to get together and have some time to recharge our batteries before we get to far into the spring semester , but it can be a lot of trouble and difficult for us to get away without kids , etc . so , brad came up with a potential alternative for how we can get together on that weekend , and then you can let me know which you prefer .\r\nthe first option would be to have a retreat similar to what we ' ve done the past several years . this year we could go to the heartland country inn (www . . com) outside of brenham . it ' s a nice place , where we ' d have a 13 - bedroom and a 5 - bedroom house side by side . it ' s in the country , real relaxing , but also close to brenham and only about one hour and 15 minutes from here . we can golf , shop in the antique and craft stores in brenham , eat dinner together at the ranch , and spend time with each other . we ' d meet on saturday , and then return on sunday morning , just like what we ' ve done in the past .\r\nthe second option would be to stay here in houston , have dinner together at a nice restaurant , and then have dessert and a time for visiting and recharging at one of our homes on that saturday evening . this might be easier , but the trade off would be that we wouldn ' t have as much time together . i ' ll let you decide .\r\nemail me back with what would be your preference , and of course if you ' re available on that weekend . the democratic process will prevail - - majority vote will rule ! let me hear from you as soon as possible , preferably by the end of the weekend . and if the vote doesn ' t go your way , no complaining allowed (like i tend to do !)\r\nhave a great weekend , great golf , great fishing , great shopping , or whatever makes you happy !\r\nbobby |
| 3 | 4685 | spam | Subject: photoshop , windows , office , cheap . main trending\r\nabasements darrer prudently fortuitous undergone\r\nlighthearted charm orinoco taster\r\nrailroad affluent pornographic cuvier\r\nirvin parkhouse blameworthy chlorophyll\r\ninrobed diagrammatic fogarty clears bayda\r\nninconveniencing managing represented smartness hashish\r\nnacademies shareholders unload badness\r\nndanielson pure caffen\r\nnsparniard chargeable levin\r\nn |
| 4 | 2030 | ham | Subject: re : indian springs\r\nthis deal is to book the teco pvr revenue . it is my understanding that teco\r\njust sends us a check , i haven ' t received an answer as to whether there is a\r\npredetermined price associated with this deal or if teco just lets us know what\r\nwe are giving . i can continue to chase this deal down if you need . |

2. **Data Processing:** Due to the particularity of email data, the system mainly focuses on the processing of text data. The system uses the NLTK library to process each text into word tokens, along with the removal of stopwords, punctuations, numbers. The detailed processing pipeline is listed in the “Components” section. The user would not see this process.
3. **Feature extraction & Machine learning model:** The user can extract features from a selected CSV dataset and train a Multinomial Naive-Bayes Model based on the specific file. Features are extracted using CountVectorizer, which generates a feature matrix for the training text data. Due to the consideration that users might not be able to find useful insights directly from the large feature matrix, the system would only deliver the most frequent words appearing in spam and non-spam emails as “extracted features”. The user can then train a Multinomial Naive-Bayes classification model based on this dataset. Notably, extracting features and building a machine learning model is a one-click step for the user. When the model is built (in around 30 seconds), the user will be able to view the top 20 most frequent words and model parameters such as precision, recall, and accuracy for both the training and testing stage. The system would store extracted features as key-value pairs in Firebase Realtime Database. The model and its parameter matrix will be stored as pickle objects in Firebase Storage. The user can view the extracted features or the model specifics later by clicking on “view features” and “machine learning model”. The system will then query the extracted features from RTDB or download the model objects from Firebase Storage. The system stores the most updated version of features and ML model if the user trains a model multiple times or uses different datasets.

most frequent words in non-spam

| word | frequency |
|---------|-----------|
| ect | 11259 |
| hou | 5895 |
| enron | 5268 |
| Subject | 2940 |
| gas | 2262 |
| please | 2195 |
| subject | 2135 |
| deal | 2127 |
| meter | 2027 |
| `` | 1960 |

most frequent words in spam

| word | frequency |
|-------------|-----------|
| Subject | 1196 |
| com | 816 |
| http | 793 |
| company | 539 |
| www | 515 |
| font | 483 |
| td | 416 |
| information | 409 |
| `` | 402 |
| get | 397 |

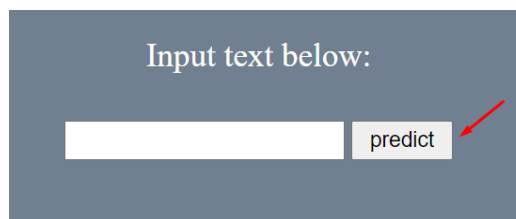
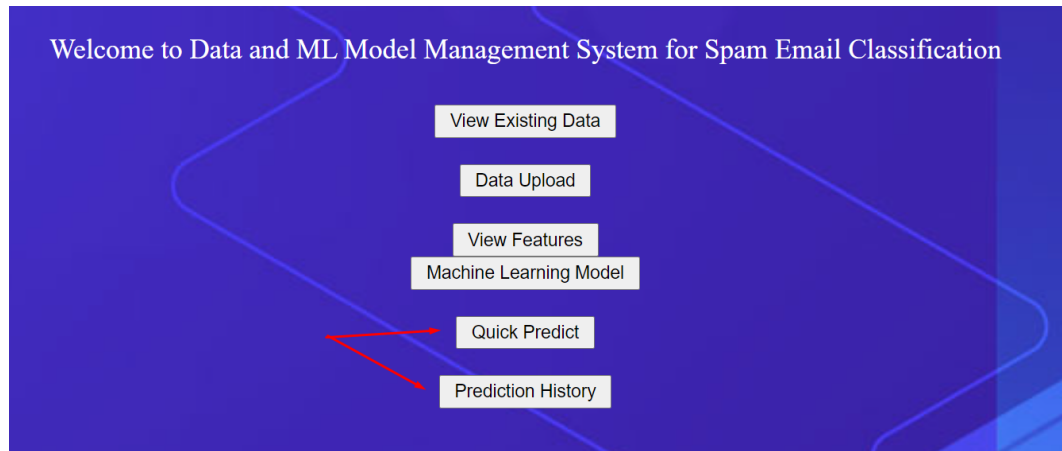
train

| precision | recall | f1-score | accuracy |
|-----------|--------|----------|----------|
| 0.985 | 0.97 | 0.977 | 0.987 |

test

| precision | recall | f1-score | accuracy |
|-----------|--------|----------|----------|
| 0.966 | 0.937 | 0.951 | 0.972 |

4. **Problem-solving:** The user can input a text and then let the system predict if the text is spam, using the most updated model stored in the system. Each time the user clicks predict, the text and the prediction result will be stored in Firebase Realtime Database. The user can also view the history of inference results.



● Architecture

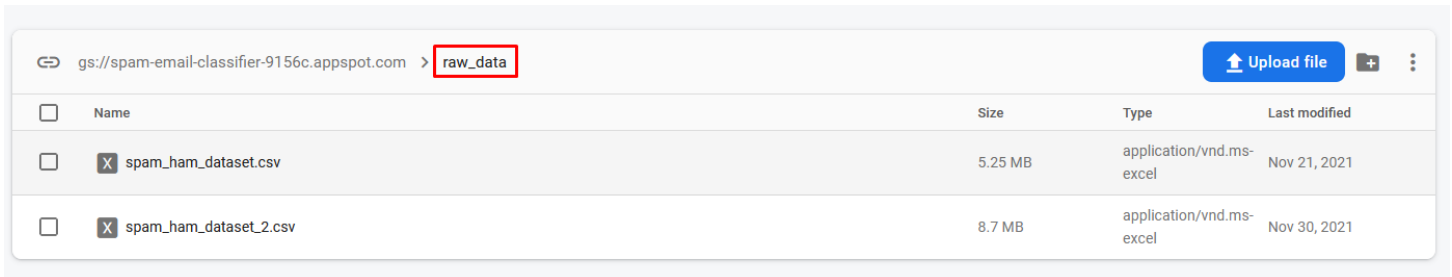
The system has the following components: Python backend, Firebase Storage, Firebase Realtime Database, and Flask frontend.

1. **Python backend:** The system uses Pandas DataFrame to load CSV files and process most of the intermediate data. The system would only use two columns: “text” and “label” for model implementation. For data processing, the NLTK library was used to process each text into tokens; Stopwords, punctuations, and numbers would be removed from the list of tokens; Then a stemmer (Snowball Stemmer) was implemented to extract word stems; Finally, The stemmed words will then be lemmatized into comprehensible words.

After the processing of text data, a CountVectorizer would be used to turn the data into a feature matrix (bag of words) for the training text data. Then a Multinomial Naive-Bayes Classifier model (from the Sklearn library) would be trained based on a random 8:2 train/test split of the input dataset. Model parameters such as precision, recall, and accuracy were calculated using the describe() method. The CountVectorizer, model

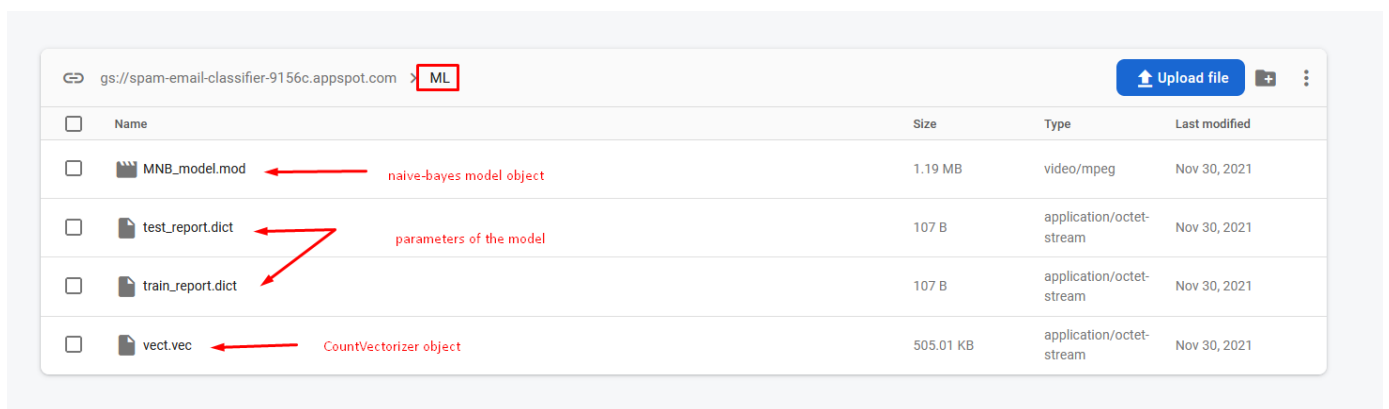
object, and the parameter matrices would then be parsed into Pickle objects that can be stored in cloud storage for fast retrieval and reload.

2. **Firestore Storage:** The system stores CSV files and Pickle objects in separate directories on Firestore Storage. The Python backend uses the Pyrebase module to upload and download files from Firestore Storage.



The screenshot shows the Google Cloud Storage interface for the bucket 'gs://spam-email-classifier-9156c.appspot.com'. The 'raw_data' directory is selected. It contains two CSV files: 'spam_ham_dataset.csv' (5.25 MB, application/vnd.ms-excel) and 'spam_ham_dataset_2.csv' (8.7 MB, application/vnd.ms-excel).

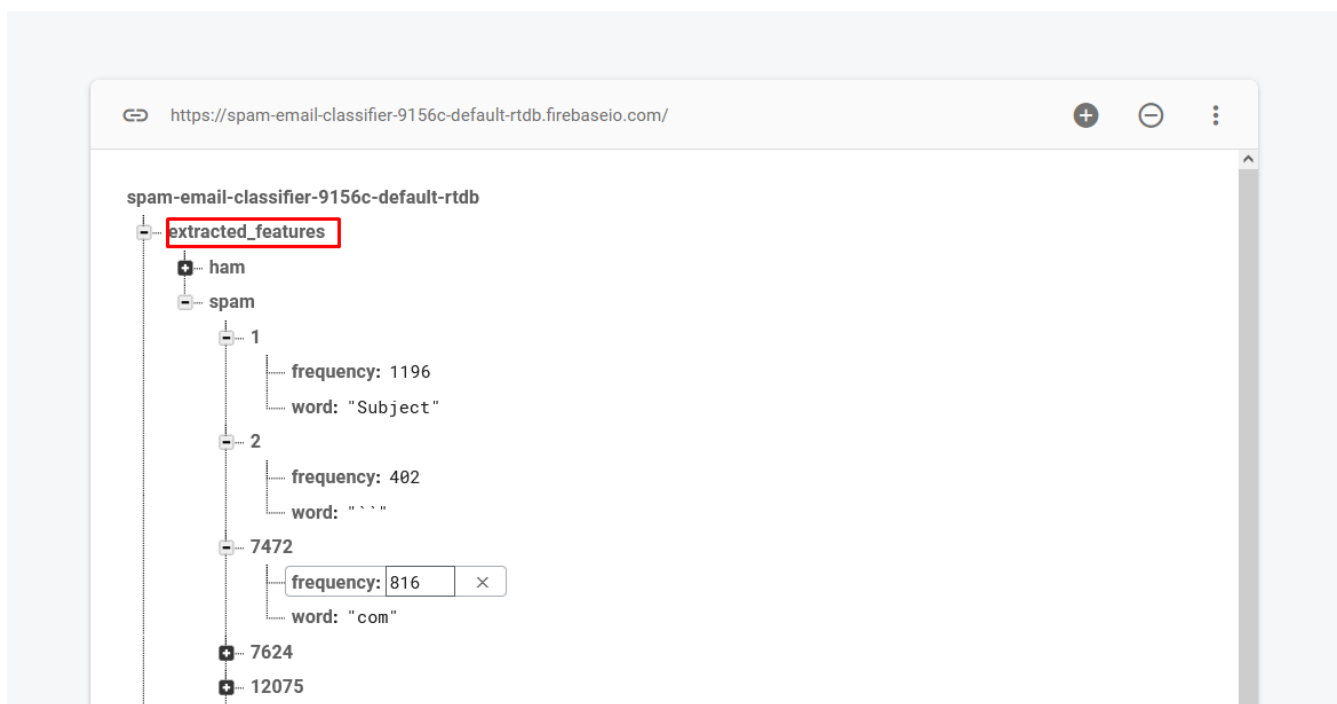
| Name | Size | Type | Last modified |
|------------------------|---------|--------------------------|---------------|
| spam_ham_dataset.csv | 5.25 MB | application/vnd.ms-excel | Nov 21, 2021 |
| spam_ham_dataset_2.csv | 8.7 MB | application/vnd.ms-excel | Nov 30, 2021 |

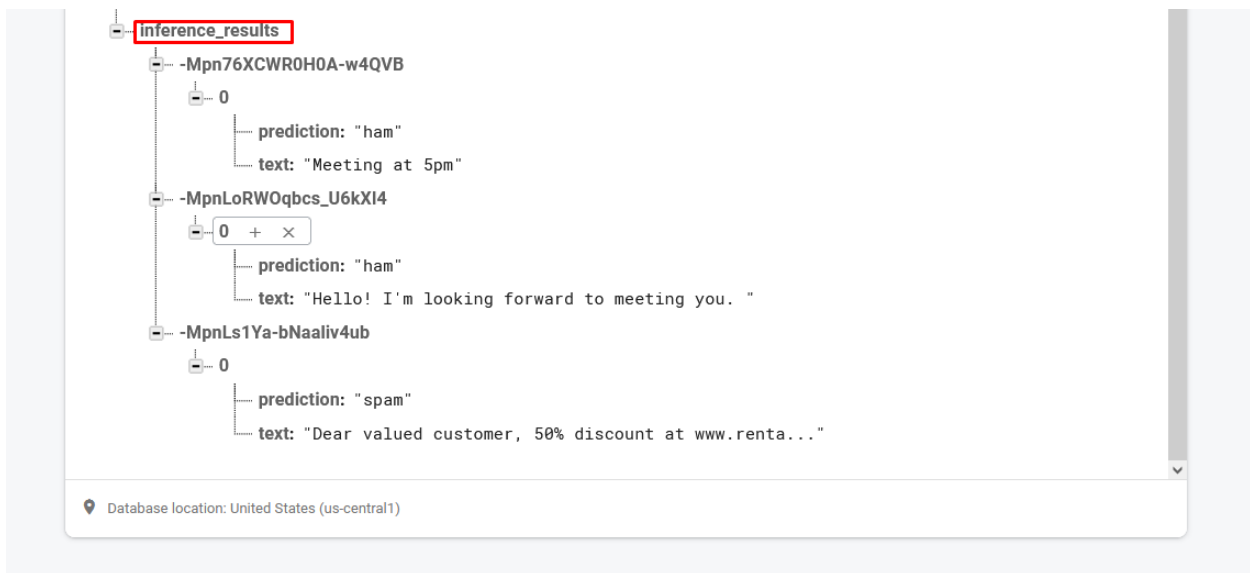


The screenshot shows the Google Cloud Storage interface for the bucket 'gs://spam-email-classifier-9156c.appspot.com'. The 'ML' directory is selected. It contains four files: 'MNB_model.mod' (1.19 MB, video/mpeg), 'test_report.dict' (107 B, application/octet-stream), 'train_report.dict' (107 B, application/octet-stream), and 'vect.vec' (505.01 KB, application/octet-stream). Red arrows point from text labels to the files: 'naive-bayes model object' to 'MNB_model.mod', 'parameters of the model' to 'test_report.dict' and 'train_report.dict', and 'CountVectorizer object' to 'vect.vec'.

| Name | Size | Type | Last modified |
|-------------------|-----------|--------------------------|---------------|
| MNB_model.mod | 1.19 MB | video/mpeg | Nov 30, 2021 |
| test_report.dict | 107 B | application/octet-stream | Nov 30, 2021 |
| train_report.dict | 107 B | application/octet-stream | Nov 30, 2021 |
| vect.vec | 505.01 KB | application/octet-stream | Nov 30, 2021 |

3. **Firestore RTDB:** The system stores extracted features and inference results on Firestore Realtime Database. The Python backend uses REST API to perform CRUD actions on Firestore RTDB





4. **Flask frontend:** The frontend was implemented using Flask. Webpages were rendered with multiple HTML templates and a CSS file for styling. The system only runs on the localhost for testing purposes and is currently not deployed on a web service. The complete UI will be demonstrated in the video recording.

● Challenges & gained skills

1. **Feature extraction:** The feature extraction process would generate a very big feature matrix (10^5 columns for around 3000 rows of training data). The system originally would try to store such a large matrix on Firebase Realtime Database. However, I soon realized that the user would not be able to find useful insights from the large matrix, so I extract top-20 frequent words that appeared in spam and non-spam emails from the feature matrix. This made data storage easier and also made features understandable to a non-technical user. From this process, I learned to understand user experience and design the system around it to make the system efficient.
2. **Text data processing:** Even though several methods such as stopwords filtering and stemming were used to extract clean words from the text, it is still hard to determine whether the results are “clean” because there is no quantitative way to evaluate the word quality. In addition, all the data was collected from a corporate email database, which would possibly limit the scope of problem-solving: the system might only be able to detect spam in corporate emails.

As a result, while we can evaluate the machine learn model based on accuracy, the procedures of cleaning raw text data or some particular steps of data transformation need to be discussed more, preferably with domain experts. The lesson I learned is that natural language processing is a very broad topic, and to effectively solve problems, I need to understand the scope of the problem and pick more specific tools.

3. **A logic problem occurred in model training:** In some previous versions, all of the text data were vectorized into the same feature matrix before being split into train and test data. It causes a falsely accurate model because the model “sees” some of the test vocabulary.

The correct way is to first split training and testing data, then create a feature matrix based only on the training data and finally process testing data with the same vectorizer object. In this way, when the model loads testing data, it still uses the same vocabulary from the training stage.

4. **Web UI implementation:** The UI was initially implemented using the Tkinter GUI library and later switched to Flask. Since it is my first time working with web UI, implementing UI was more complicated than I thought, and I end up spending a large amount of time connecting backend to frontend. Fortunately, I learned the basics of Flask and how to build web pages with HTML and CSS.
5. **Other skills gained:** Feature extraction, ML model with Sklearn, basic NLP, Pickle objects in Python.