DSCI 551 Final Project

A tool to analyze people's personal information and MRI image and predict the chances of getting Alzheimer's

Group 35

Xinyue Yu (Communication Data Science, M.S.)
yuxinyue@usc.edu

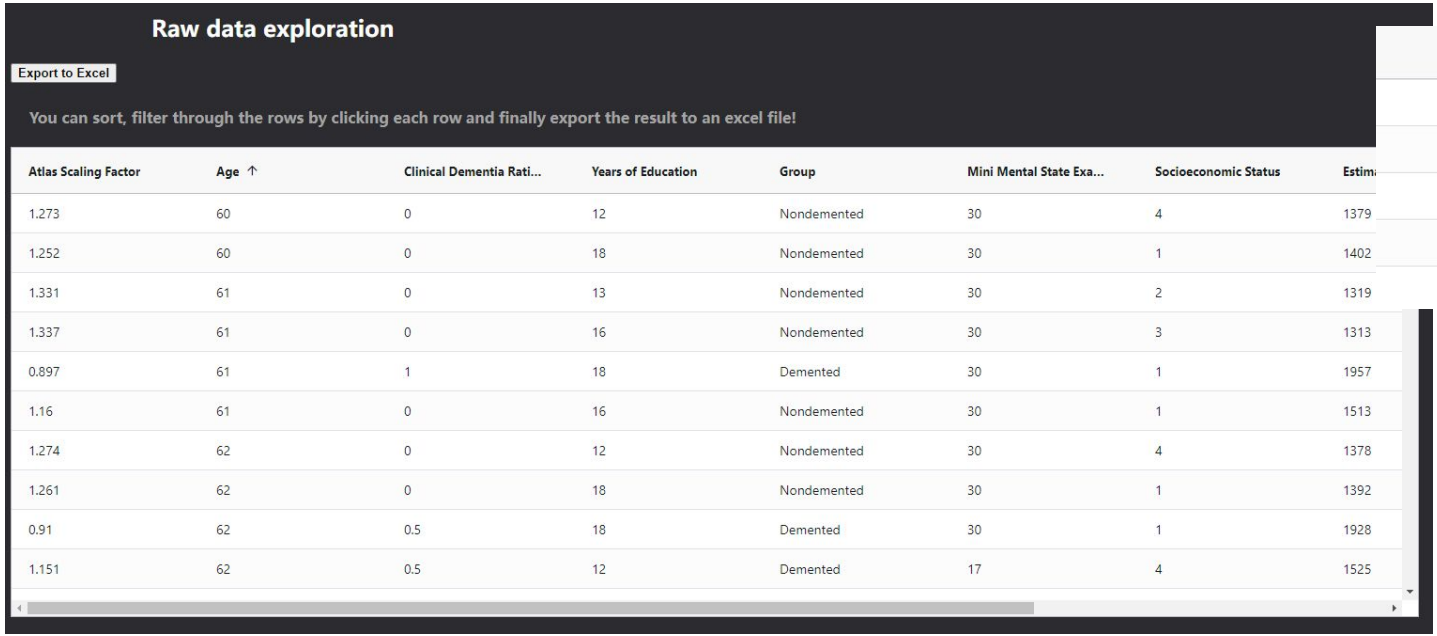
Yishan Li (Applied Data Science, M.S.)
liyishan@usc.edu

Project Introduction

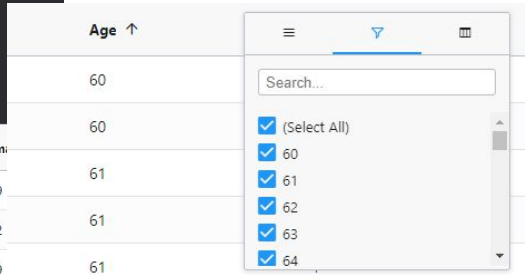
We aim to identify and distinguish features in the MRI image and the text data of doing the MRI procedure that we collect using deep learning to determine the category of a patient’s Alzheimer's disease level.

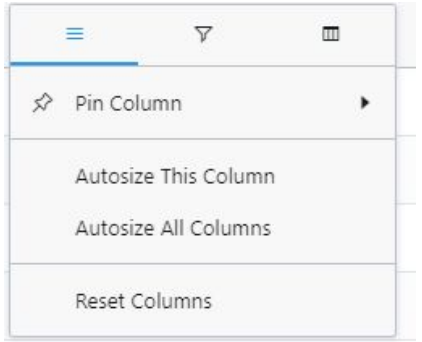
Raw Text Data Exploration

AG-Grid: An interactive grid which allows user to sort, filter through rows.

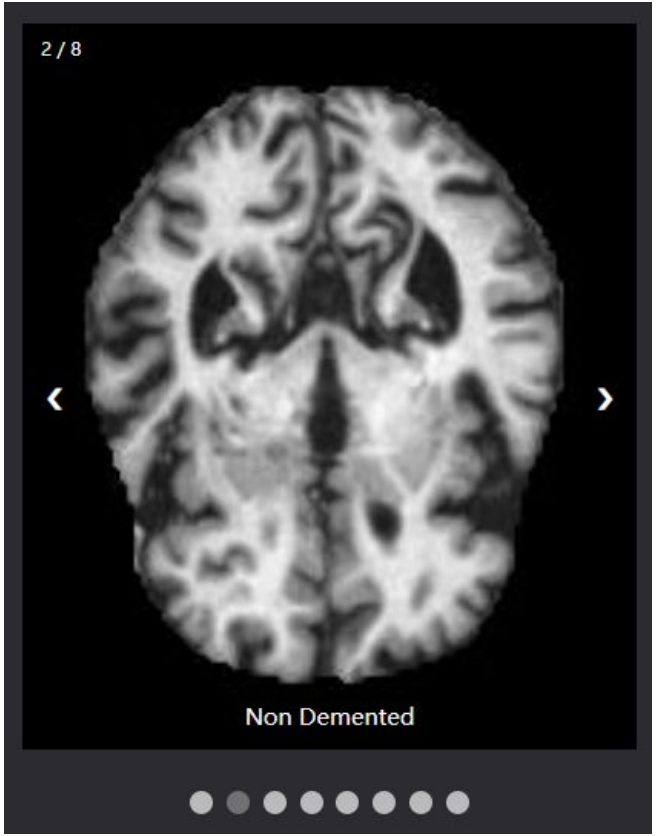


Atlas Scaling Factor	Age ↑	Clinical Dementia Rati...	Years of Education	Group	Mini Mental State Exa...	Socioeconomic Status	Estim
1.273	60	0	12	Nondemented	30	4	1379
1.252	60	0	18	Nondemented	30	1	1402
1.331	61	0	13	Nondemented	30	2	1319
1.337	61	0	16	Nondemented	30	3	1313
0.897	61	1	18	Demented	30	1	1957
1.16	61	0	16	Nondemented	30	1	1513
1.274	62	0	12	Nondemented	30	4	1378
1.261	62	0	18	Nondemented	30	1	1392
0.91	62	0.5	18	Demented	30	1	1928
1.151	62	0.5	12	Demented	17	4	1525





Raw Image Data Exploration



Data processing and Data storage

Data Source:

- Text data (Open Access Series of Imaging Studies (OASIS))
- Non-text data (https://www.kaggle.com/legendahmed/alzheimermridataset)

Data Processing and Feature Extraction for Text Data

Libraries we used: pyspark, pandas, sklearn, json, requests

Use Pyspark Dataframe to drop columns, rename columns, and change value of gender column from ‘M/F’ to ‘1/0’

Change to pandas dataframe and use Iterative Imputer from sklearn to fill in the missing values of columns

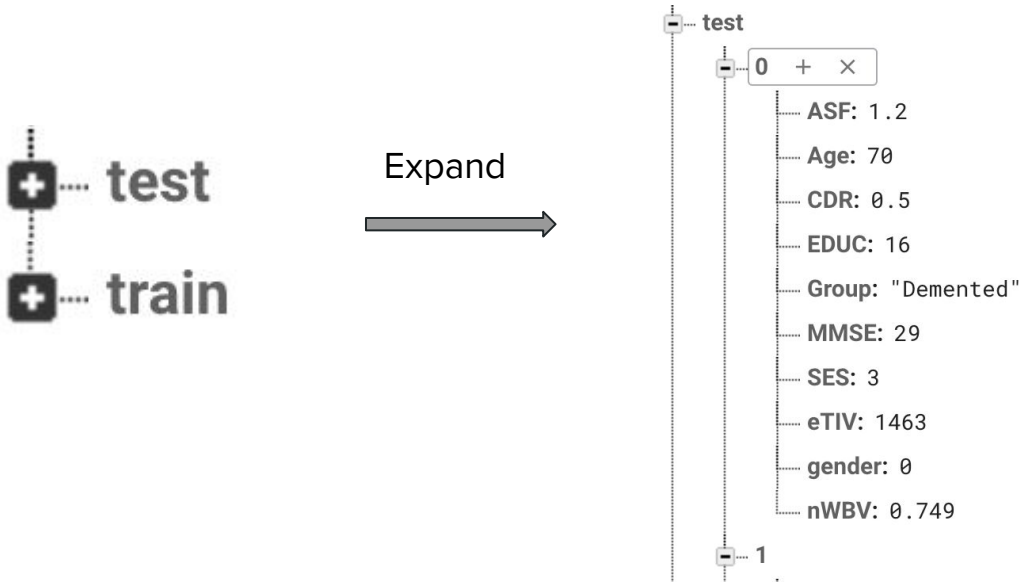
Split data in training and testing set. Convert data to json format and upload it to Firebase.

```
sc = SparkContext.getOrCreate()
spark = SparkSession(sc)
#read the file
oslong_df = spark.read.option("header","true").csv("oasis_longitudinal.csv")
#drop the column
oslong_df = oslong_df.drop('Subject ID', 'MRI ID', 'Visit', 'MR Delay', 'Hand')
#rename the column
oslong_df = oslong_df.withColumnRenamed("M/F","gender")
#change gender when male to 0, when female to 1
oslong_df = oslong_df.withColumn('gender', when(col('gender') == 'M', '0').otherwise('1'))
```

```
imp = IterativeImputer(max_iter=10, random_state=57)
imp_data = imp.fit_transform(impt)
impt_data = pd.DataFrame(imp_data, columns = impt.columns)
impt_data['Group'] = data['Group']
```

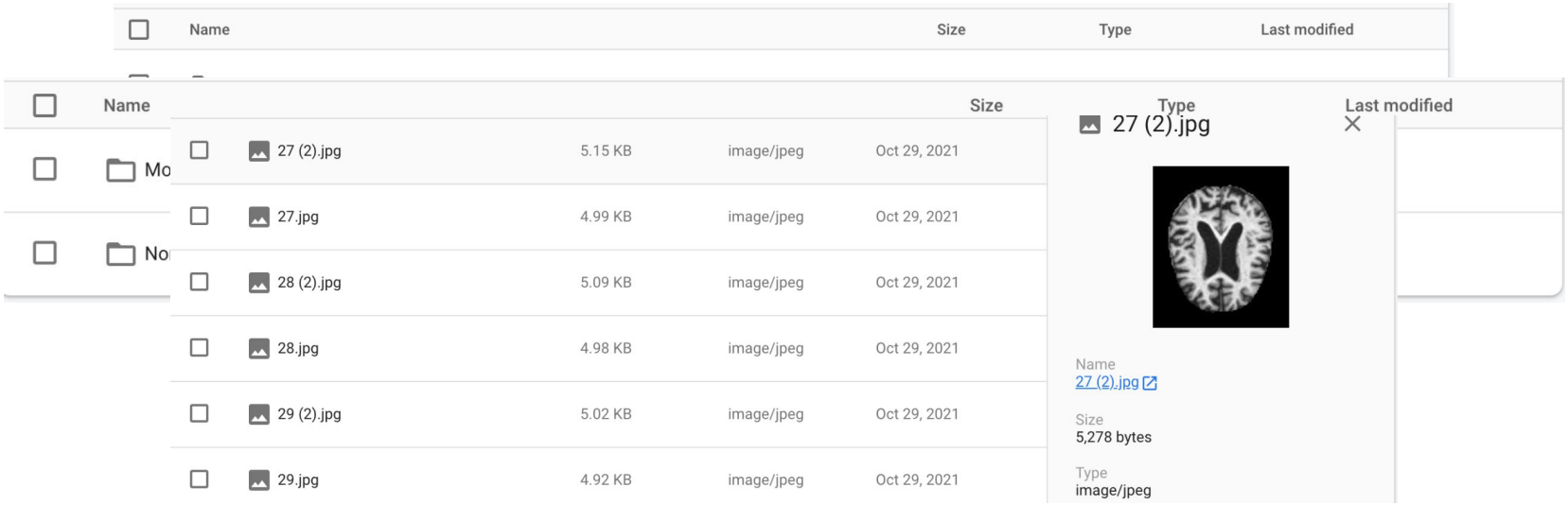
Data Storage for Text Data - Firebase Realtime Database

The text data are converted to json format, splitted into train and test set, and uploaded to Firebase Realtime Database



Data Storage for Non Text data - Firebase Storage (cloud data storage)

The image data are divided into train and test set and then stored in separate folder based on their class.



Retrieve Data from Firebase

Text data

Library: requests

```
response = requests.get(test_url)
text_test = response.json()
test_t = pd.DataFrame.from_dict(text_test, orient='columns')
```

Non-text data

Library: pyrebase

```
base.initialize_app(config)
ase.storage()
```

```
storage.child(train_md_path_ls[i]).download(train_mddl_path_ls[i])
storage.child(train_nd_path_ls[i]).download(train_nddl_path_ls[i])
```

Machine Learning and The Results

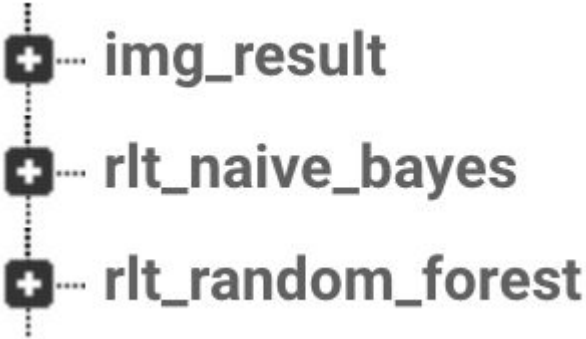
Text data

Classification

- Random Forest Classification
- Naive Bayes Classification

Results

- Result are pushed to Firebase Realtime Database on real time



Non-Text data

Classification

- Library: Tensorflow, Sequential model

Results

- Result are pushed to Firebase Realtime

Thanks for watching

Move on to our website