PIRAMAL FINANCE HACKATHON - IIT GUWAHATI

UNDERSTANDING THE DATA

1. I initiated a comprehensive review of the dataset to grasp its intricacies.

2. Recognizing the dataset's mixed data types—ranging from strings to ranges—presented a notable challenge.

3. Additionally, the prevalence of null and empty values raised concerns regarding the dataset's reliability.

4. To streamline the analysis process, I utilized a CSV AI tool to compare unique values within each column.

5. Furthermore, as the CV files were initially in DOC format, I converted them to PDF for easier data extraction, facilitating smoother handling and analysis.

DATA CLEANING

• Utilized tools like ChatCSV for efficient data cleaning, effectively addressing null values by employing statistical methods such as mean, median, or 0 where appropriate.

• Converted string data types into simpler numerical representations conducive to model processing, enhancing data compatibility for analysis.

• Leveraged the pandas library to develop Python scripts tailored for data cleaning, implementing specific parameters to ensure data integrity and quality.

• Executed Python scripts within Visual Studio Code (VSCode) and Google Colab, optimizing workflow and leveraging their capabilities for streamlined data processing.

• Employed various techniques for visualizing data, including tables, charts, and other graphical representations, facilitating insightful analysis and interpretation.

• Automated the conversion of document files, such as CVs in test and train sets, into PDF format using Python scripts, enhancing data accessibility and compatibility.

• Implemented data refinement strategies, such as removing common data attributes like designations, to streamline analysis and focus on relevant information.

• Encoding String Data: For string-type categorical variables, we can encode them into numerical values using techniques like one-hot encoding or label encoding.

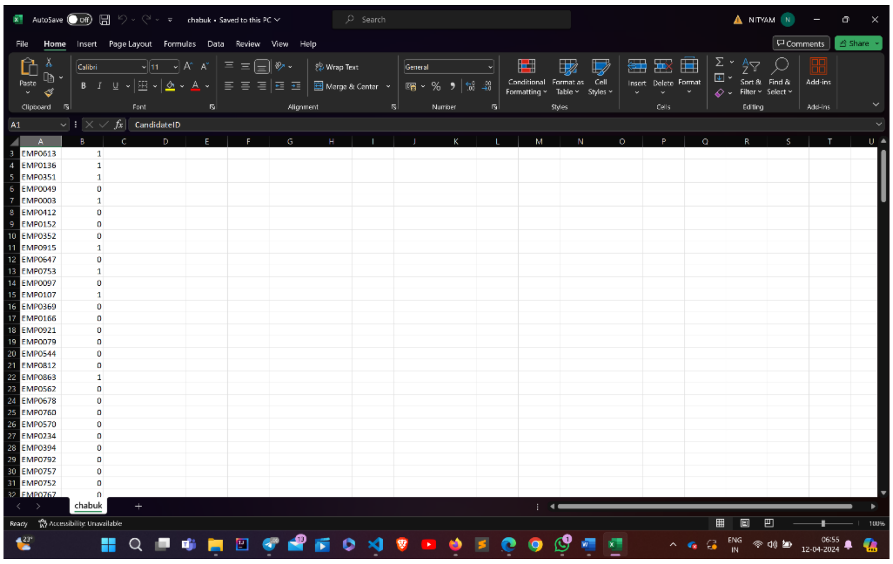
• One-Hot Encoding: This technique converts each category into a binary vector where each category is represented by a binary digit (0 or 1). This is particularly useful when the categories don't have a natural ordering or hierarchy.

• Label Encoding: Label encoding assigns a unique integer to each category. This is suitable when the categories have a natural order or hierarchy.

• Integer Encoding for Range Data: Range data, which consists of numerical intervals, can be encoded into integer values based on predefined ranges or bins.

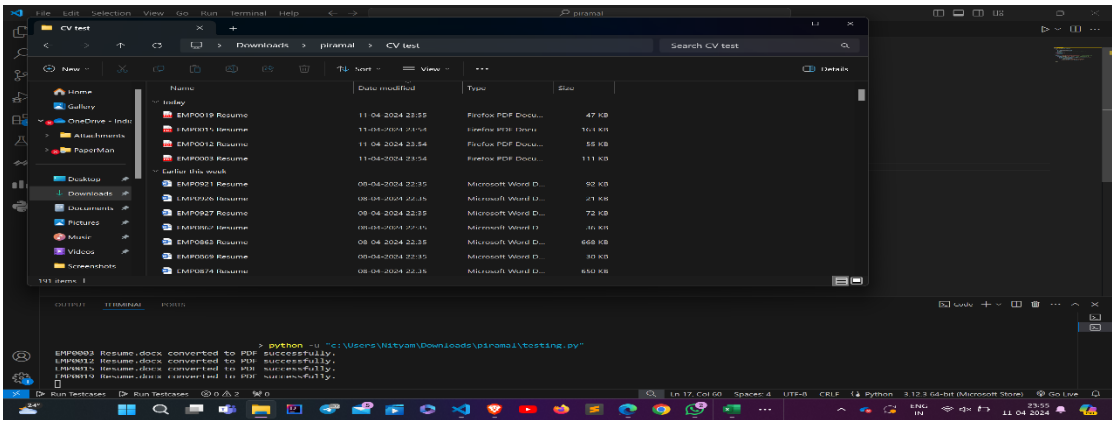
• Binning or Discretization: This involves dividing the range of numerical values into discrete intervals or bins and assigning an integer value to each bin. This approach is useful for converting continuous data into categorical or ordinal data.

• Custom Encoding: Depending on the specific requirements and characteristics of the range data, custom encoding schemes can be devised. For example, assigning integer values based on predefined thresholds or business rules.



A1 = CandidateID

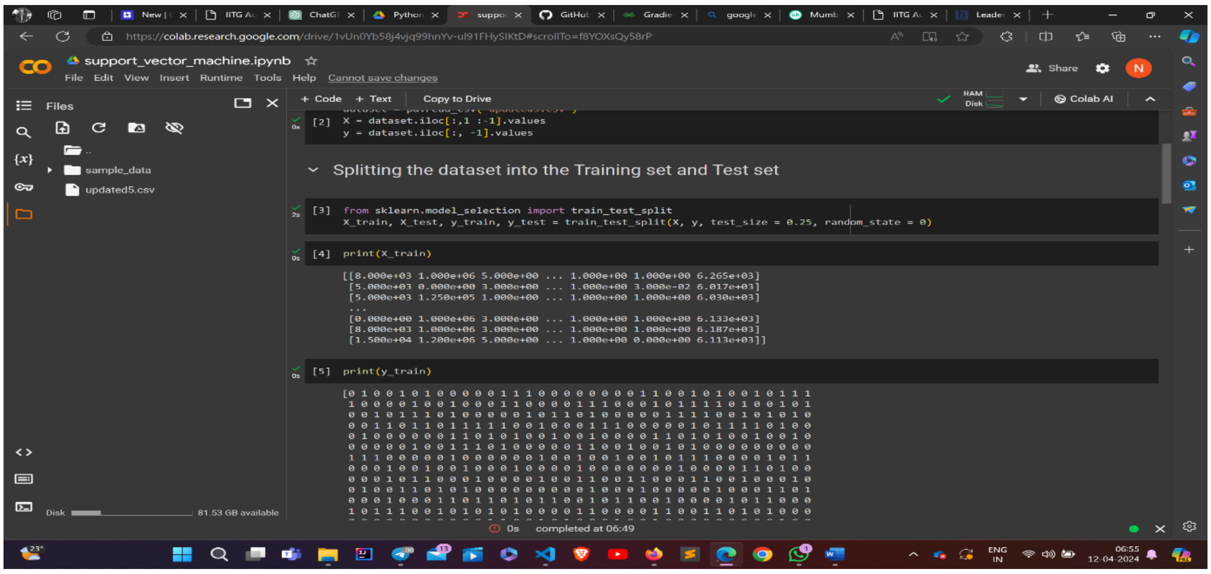
	A	B
3	EMP0613	1
4	EMP0136	1
5	EMP0351	1
6	EMP0049	0
7	EMP0003	1
8	EMP0412	0
9	EMP0152	0
10	EMP0352	0
11	EMP0915	1
12	EMP0647	0
13	EMP0753	1
14	EMP0097	0
15	EMP0107	1
16	EMP0369	0
17	EMP0166	0
18	EMP0921	0
19	EMP0079	0
20	EMP0544	0
21	EMP0812	0
22	EMP0863	1
23	EMP0562	0
24	EMP0678	0
25	EMP0760	0
26	EMP0570	0
27	EMP0234	0
28	EMP0394	0
29	EMP0792	0
30	EMP0757	0
31	EMP0752	0
32	EMP0767	0



Downloads > piramal > CV test

Name	Date modified	Type	Size
Today			
EMP0019 Resume	11-04-2024 23:55	Firefox PDF Docu...	47 KB
EMP0015 Resume	11-04-2024 23:54	Firefox PDF Docu...	163 KB
EMP0012 Resume	11-04-2024 23:54	Firefox PDF Docu...	55 KB
EMP0003 Resume	11-04-2024 23:54	Firefox PDF Docu...	111 KB
Earlier this week			
EMP0921 Resume	08-04-2024 22:35	Microsoft Word D...	92 KB
EMP0926 Resume	08-04-2024 22:35	Microsoft Word D...	21 KB
EMP0927 Resume	08-04-2024 22:35	Microsoft Word D...	72 KB
EMP0862 Resume	08-04-2024 22:35	Microsoft Word D...	46 KB
EMP0863 Resume	08-04-2024 22:35	Microsoft Word D...	668 KB
EMP0869 Resume	08-04-2024 22:35	Microsoft Word D...	30 KB
EMP0874 Resume	08-04-2024 22:35	Microsoft Word D...	650 KB

```
                     > python -u "c:\Users\Nityam\Downloads\piramal\testing.py"
EMP0003 Resume.docx converted to PDF successfully.
EMP0012 Resume.docx converted to PDF successfully.
EMP0015 Resume.docx converted to PDF successfully.
EMP0019 Resume.docx converted to PDF successfully.
```


support_vector_machine.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Files
sample_data
updated5.csv
Disk 81.53 GB available
Share
Colab AI

```
[2] X = dataset.iloc[:,1 :-1].values
    y = dataset.iloc[:, -1].values
```

Splitting the dataset into the Training set and Test set

```
[3] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

```
[4] print(X_train)

[[8.000e+03 1.000e+06 5.000e+00 ... 1.000e+00 1.000e+00 6.265e+03]
 [5.000e+03 0.000e+00 3.000e+00 ... 1.000e+00 3.000e-02 6.017e+03]
 [5.000e+03 1.250e+05 1.000e+00 ... 1.000e+00 1.000e+00 6.030e+03]
 ...
 [0.000e+00 1.000e+06 3.000e+00 ... 1.000e+00 1.000e+00 6.133e+03]
 [8.000e+03 1.000e+06 3.000e+00 ... 1.000e+00 1.000e+00 6.187e+03]
 [1.500e+04 1.200e+06 5.000e+00 ... 1.000e+00 0.000e+00 6.113e+03]]
```

```
[5] print(y_train)

[0 1 0 0 1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 1 0 1 1 1
 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 1 0 1 1 1 1 0 1 0 0 1 0 1
 0 0 1 0 1 1 1 0 1 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0 1 0
 0 0 1 1 0 1 1 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 0 1 1 1 1 0 1 0 0
 0 1 0 0 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 1 0
 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0
 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 1 1
 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0
 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 0 0 0 1 0
 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1
 0 0 0 1 0 0 0 1 1 0 1 1 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 0 0 0
 1 0 1 1 1 0 0 1 0 1 0 1 0 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 0 1 0 0 0
```

0s completed at 06:49

FEATURE SELECTION

• Statistical Analysis on Google Colab: Initially, I performed a detailed

statistical analysis using Google Colab, which allowed me to sift through

the multitude of parameters and discern their impact on performance.

This step was crucial in understanding which variables significantly

influenced the outcome.

• Linear Regression for Parameter Importance: Following the initial

analysis, I employed linear regression techniques in Python to further

investigate the importance of various parameters. By running regression models, I could quantify the influence of each parameter and identify

the ones that were most critical for performance.

• Conversion of Parameters into Usable Form: Once the important

parameters were identified, I converted them into a usable format

suitable for further analysis and modeling. This step involved

transforming the data into a structured and manageable form for

subsequent processing.

• Optimization with Singular Value Decomposition (SVD): To enhance

the efficiency of the testing data, I utilized Singular Value

Decomposition (SVD) techniques. By decomposing the data matrix, SVD

helped optimize the test dataset, improving the accuracy and reliability

of subsequent analyses.

• Utilization of scikit-learn for ML Classification Models: I employed

scikit-learn, a powerful machine learning library in Python, to

experiment with various classification models. By employing different algorithms, I aimed to ascertain which parameters were indispensable

for accurate predictions and classification tasks.

• PDF CV Parsing with PyPDF2: In order to extract relevant details such

as languages spoken and skills possessed by candidates, I utilized

PyPDF2's PdfReader module. This enabled the automated extraction of

information from PDF-based resumes, streamlining the data collection

process for further analysis.

FEATURE TRANSFORMATION

• Employed encoding techniques to transform string-type categorical variables into numerical values, such as:

• Utilized one-hot encoding for non-ordinal categories, converting each category into a binary vector representation.

• Implemented label encoding for categories with a natural order or hierarchy, assigning unique integer values to each category.

• Applied integer encoding methods to handle range data, including:

• Utilizing binning or discretization to divide numerical intervals into predefined bins or categories.

• Assigned integer values to each bin, facilitating the conversion of continuous variables into categorical or ordinal ones.

• By employing these feature transformation techniques, ensured all data types, including strings and range data, were converted into numerical formats suitable for machine learning algorithms.

• Prepared the dataset for integration into various machine learning models, enabling robust analysis and accurate predictions.

Integrated normalization for selected parameters to ensure consistent scaling across features:

• Identified specific parameters where normalization would be

beneficial, such as those with varying scales or units.

• Applied normalization techniques to rescale these parameters

to a standard range, typically between 0 and 1

• Experimented with a combination of classification models including SGBoost, CatBoost, Random Forest Classification, and K-Nearest Neighbors (KNN).

• Implemented ensemble techniques to leverage the strengths of each model and mitigate their individual weaknesses.

• Combined predictions from SGBoost, CatBoost, Random Forest, and KNN to generate a consolidated output, aiming to achieve higher accuracy and robustness.

• Fine-tuned the ensemble approach to optimize performance based on validation metrics and domain-specific requirements.

- Evaluated the effectiveness of the ensemble method compared to individual models through rigorous testing and analysis.

DATA SPLITTING

• Data Preparation:

• Created separate training and testing files from the provided

train.csv and test.csv datasets.

• Preserved approximately 20% of the data from train.csv for

validation purposes, ensuring a sufficient amount for training the

model.

• Designated test.csv exclusively for prediction and final model

evaluation.

• Data Splitting Strategy:

• Utilized a 80-20 data splitting strategy, allocating 80% of train.csv

for model training and the remaining 20% for validation.

• Ensured that the splitting process preserved the integrity of the

dataset and maintained representative samples across the

subsets.

• Random Shuffling:

• Randomly shuffled the data to eliminate any inherent ordering or

biases that may exist.

• This step was crucial for enhancing model accuracy and

preventing overfitting by exposing the model to diverse samples

during training.

• Validation and Evaluation:

• Reserved the validation set for fine-tuning model

hyperparameters and assessing performance during training.

• Kept the test set separate for final evaluation, enabling unbiased

assessment of the model's generalization capabilities on unseen

data.

• Data Integrity Check:

• Ensured that both the output files and the original datasets

(train.csv and test.csv) had the correct number of rows and

columns.

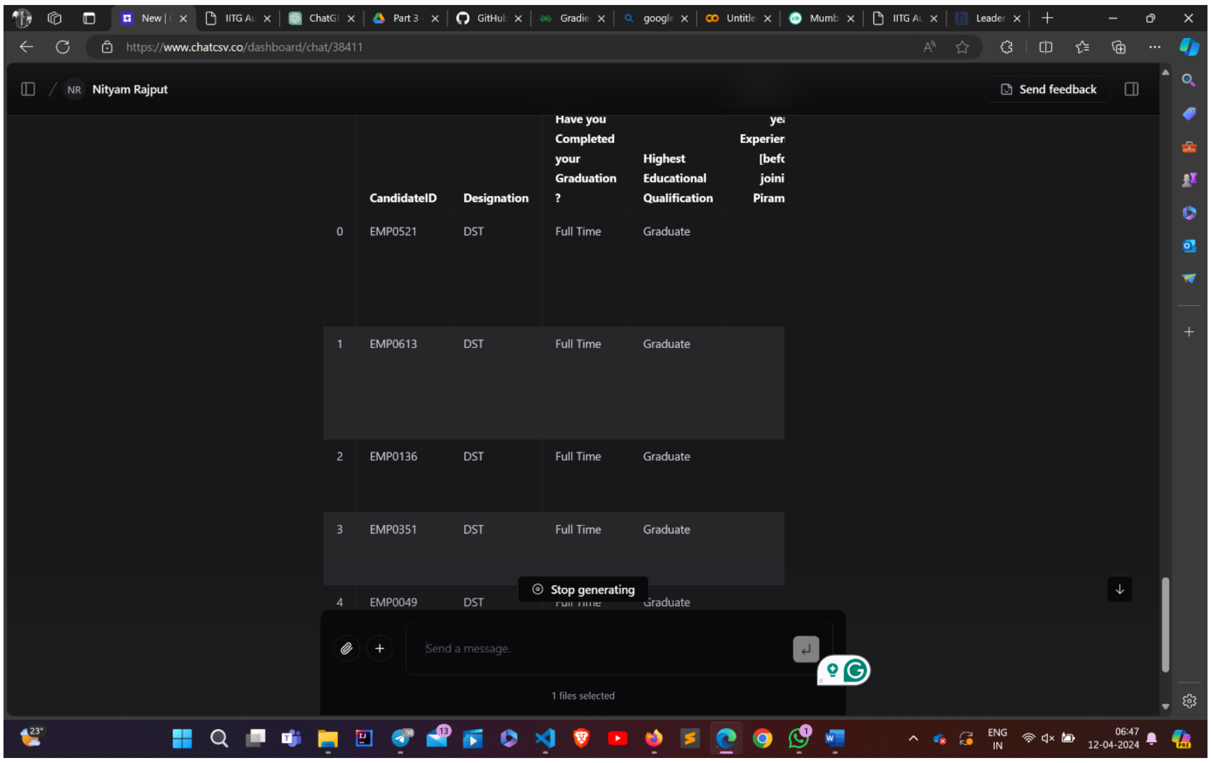
• Maintained consistency in data dimensions to prevent errors

during training and evaluation processes.By meticulously organizing and preparing the data in this manner, facilitated the

development of robust and generalizable machine learning

models capable of delivering accurate predictions on new, unseen

data.



	CandidateID	Designation	Have you Completed your Graduation ?	Highest Educational Qualification	yea Experien [befo joini Piram
0	EMP0521	DST	Full Time	Graduate	
1	EMP0613	DST	Full Time	Graduate	
2	EMP0136	DST	Full Time	Graduate	
3	EMP0351	DST	Full Time	Graduate	
4	EMP0049	DST	Full Time	Graduate	

Chatcsv for getting details of data

DOCUMENTATION

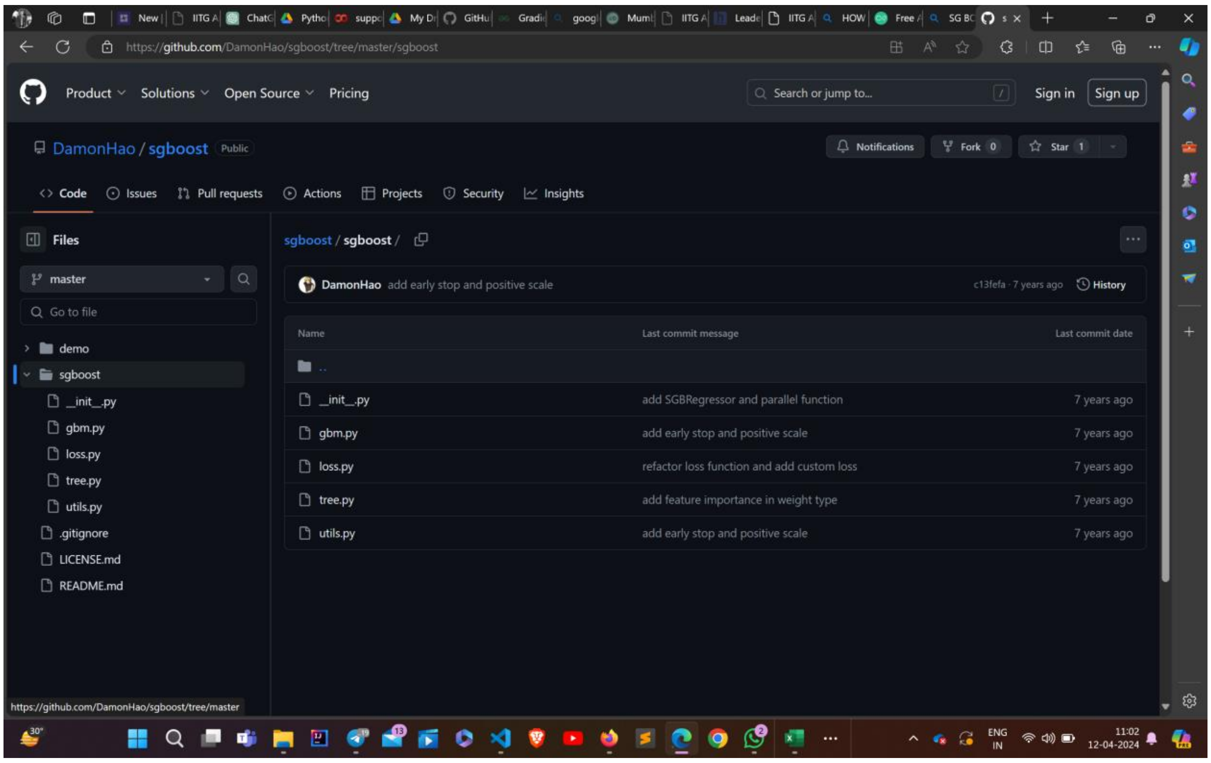
• Utilized cepstrum IITG’S classification ML models for initial classification tasks.

• Incorporated SGBoost and CatBoost software from GitHub repositories to enhance model performance.

• Referenced documentation from GeeksforGeeks (GFG) on SGBoost for implementation guidance and best practices.

• Leveraged ChatGPT to amalgamate results from multiple models, aiming to improve overall accuracy.

• Employed a combination of techniques from various sources to create a robust and accurate classification system.



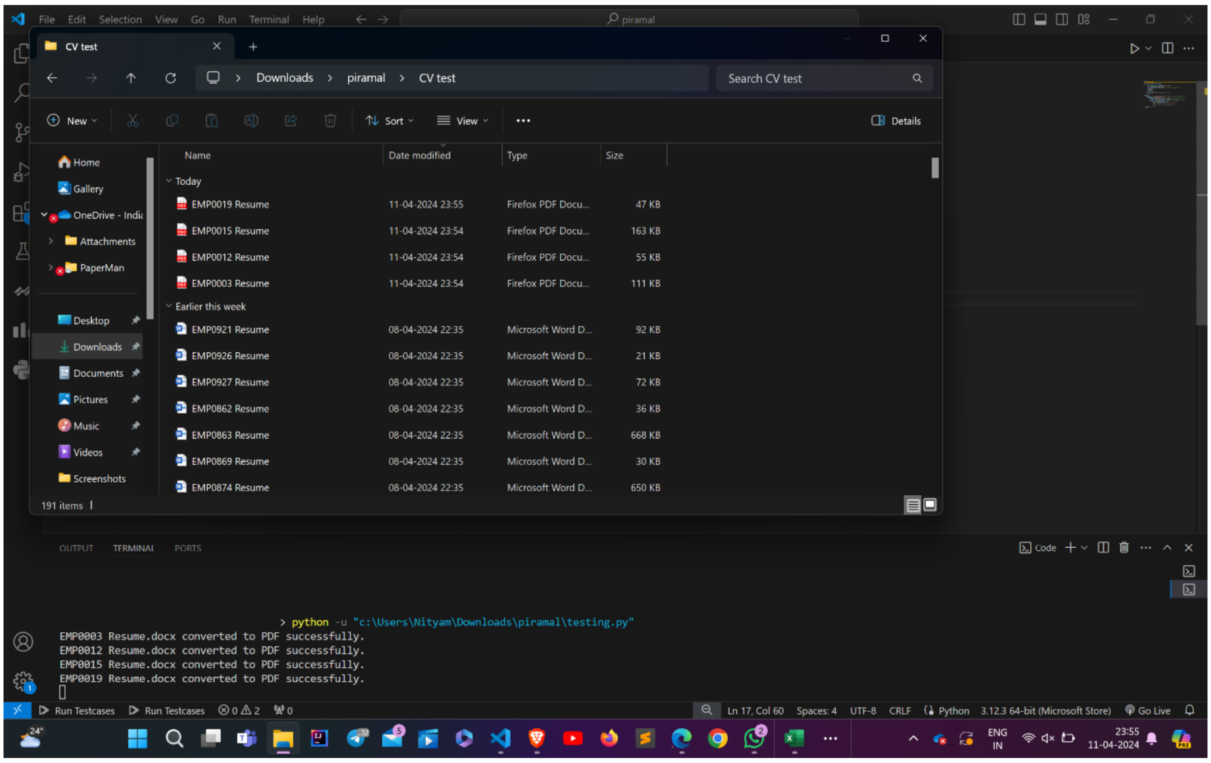
https://github.com/DamonHao/sgboost/tree/master/sgboost

DamonHao / sgboost Public

sgboost / sgboost /

DamonHao add early stop and positive scale c13fefa · 7 years ago History

Name	Last commit message	Last commit date
..		
__init__.py	add SGBRegressor and parallel function	7 years ago
gbm.py	add early stop and positive scale	7 years ago
loss.py	refactor loss function and add custom loss	7 years ago
tree.py	add feature importance in weight type	7 years ago
utils.py	add early stop and positive scale	7 years ago



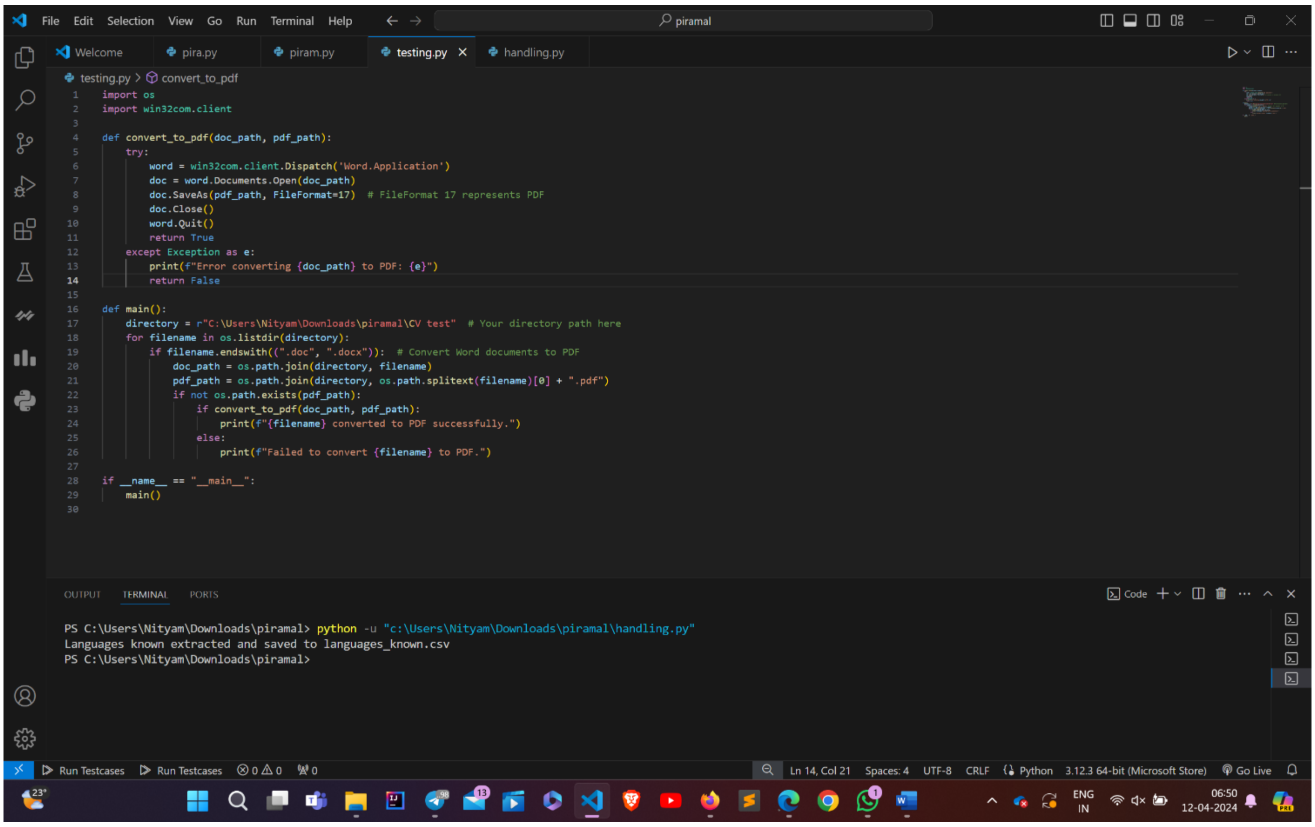
CV test — Downloads > piramal > CV test

Name	Date modified	Type	Size
Today			
EMP0019 Resume	11-04-2024 23:55	Firefox PDF Docu...	47 KB
EMP0015 Resume	11-04-2024 23:54	Firefox PDF Docu...	163 KB
EMP0012 Resume	11-04-2024 23:54	Firefox PDF Docu...	55 KB
EMP0003 Resume	11-04-2024 23:54	Firefox PDF Docu...	111 KB
Earlier this week			
EMP0921 Resume	08-04-2024 22:35	Microsoft Word D...	92 KB
EMP0926 Resume	08-04-2024 22:35	Microsoft Word D...	21 KB
EMP0927 Resume	08-04-2024 22:35	Microsoft Word D...	72 KB
EMP0862 Resume	08-04-2024 22:35	Microsoft Word D...	36 KB
EMP0863 Resume	08-04-2024 22:35	Microsoft Word D...	668 KB
EMP0869 Resume	08-04-2024 22:35	Microsoft Word D...	30 KB
EMP0874 Resume	08-04-2024 22:35	Microsoft Word D...	650 KB

191 items

```
> python -u "c:\Users\Nityam\Downloads\piramal\testing.py"
EMP0003 Resume.docx converted to PDF successfully.
EMP0012 Resume.docx converted to PDF successfully.
EMP0015 Resume.docx converted to PDF successfully.
EMP0019 Resume.docx converted to PDF successfully.
```

CONVERTING THE DOC FILES TO PDF


File Edit Selection View Go Run Terminal Help
piramal
Welcome
pira.py
piram.py
testing.py
handling.py
testing.py > convert_to_pdf

```python
import os
import win32com.client

def convert_to_pdf(doc_path, pdf_path):
    try:
        word = win32com.client.Dispatch('Word.Application')
        doc = word.Documents.Open(doc_path)
        doc.SaveAs(pdf_path, FileFormat=17)  # FileFormat 17 represents PDF
        doc.Close()
        word.Quit()
        return True
    except Exception as e:
        print(f"Error converting {doc_path} to PDF: {e}")
        return False

def main():
    directory = r"C:\Users\Nityam\Downloads\piramal\CV test"  # Your directory path here
    for filename in os.listdir(directory):
        if filename.endswith((".doc", ".docx")):  # Convert Word documents to PDF
            doc_path = os.path.join(directory, filename)
            pdf_path = os.path.join(directory, os.path.splitext(filename)[0] + ".pdf")
            if not os.path.exists(pdf_path):
                if convert_to_pdf(doc_path, pdf_path):
                    print(f"{filename} converted to PDF successfully.")
                else:
                    print(f"Failed to convert {filename} to PDF.")

if __name__ == "__main__":
    main()
```

```
PS C:\Users\Nityam\Downloads\piramal> python -u "c:\Users\Nityam\Downloads\piramal\handling.py"
Languages known extracted and saved to languages_known.csv
PS C:\Users\Nityam\Downloads\piramal>
```

Run Testcases
Run Testcases
Ln 14, Col 21 Spaces: 4 UTF-8 CRLF Python 3.12.3 64-bit (Microsoft Store) Go Live
ENG IN
06:50 12-04-2024

EXTRACTING LANGUAGE DATA FROM CV IN TRAIN

NITYAM

ROLL NUMBER – 220107059

CONTACT – n.nityam@iitg.ac.in