
MLGround

PROJECTS

Market Data Analysis and Segmentation	1
Market Data EDA	
Python File deployed via streamlit	
Multi Class Classification	2
Data EDA and Processing	
Model Generation	
Model Evalution via metrics	
Product Recommender System	3
EDA and Data Processing	
Different Model Types	
Different Types of Recommender Systems	4
Basic EDA and Data Processing	
Types of Models	
Model Serving	
Deep Learning Model to classify person's actions while driving	5
Data Generation	
Model Creation	
TEAM	6
Guide:	
Intern : P Charith	

Market Data Analysis and Segmentation

Required Libraries and Installation Commands

Numpy [pip install numpy], pandas [pip install pandas], matplotlib [pip install matlpotlib], scipy [pip install scipy] and sklearn [pip install scikit-learn]

Tasks performed under Market Analysis

- **Exploratory Data Analysis using Numpy, Pandas and matplotlib**

Data visualization using matplotlib and data preprocessing using pandas and numpy

- **Observations Based on EDA**

Observations carried out to perform market segmentation based on given Data

- **Streamlit Deployment**

App.py file with streamlit code is run to provide user interactive data visualizations.

How to run the Project?

First install jupyter notebook in your PC using **pip install notebook** or download anaconda [<https://test-jupyter.readthedocs.io/en/latest/install.html>].

Using the following link to get the code and dataset for market data analysis and segmentation :

[https://drive.google.com/drive/folders/1LlcGUs9qeco2AeQPjS_BbCKUPcWk5ULI?usp=sharing]

Run Jupyter notebook by typing **jupyter notebook** on command line. Open the downloaded notebook and run it. Refer the following link to start jupyter notebook [<https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html>]
Install all the specified libraries to get complete result.

Now to run app.py , install streamlit , refer the following link :

[<https://docs.streamlit.io/en/stable/>] . you can use visual studio code/ pycharm etc to run/ modify app.py code.

Multi Class Classification

Required Libraries and Installation Commands

Numpy [pip install numpy], pandas [pip install pandas], matplotlib [pip install matplotlib], scipy [pip install scipy] , sklearn [pip install scikit-learn], tensorflow [pip install tensorflow], wordcloud [pip install wordcloud], imblearn [pip install imblearn], xgboost [pip install xgboost], keras [pip install keras].

Tasks performed

- **Exploratory Data Analysis using Numpy, Pandas and matplotlib**

Data visualization using matplotlib and data preprocessing using pandas and numpy

- **Models considered**

Neural Network, XGB, SVM, OrdinalModel, Bidirectional LSTM, RBFN , Logistic Model, Decision Tree classifier, Random Forest classifier, Extra Trees classifier, AdaBoost , GradientBoost, Bagging Classifier and Convolutional Neural Network.

- **Model Evaluation**

Model Evaluation using Metrics such as Accuracy , Precision , Recall, F1 score, RMSE and AUC score

How to run the Project?

First install jupyter notebook in your PC using **pip install notebook** or download anaconda [<https://test-jupyter.readthedocs.io/en/latest/install.html>].

Using the following link to get the code and dataset for MultiClass Classification of textual data :

[<https://drive.google.com/drive/folders/1nCFZQLGLi46DJEUmGc3ncNK67qEZgoz7?usp=sharing>] . **Sample.xlsx** and **Sample1.xlsx** are two different datasets. **MODELS-1.ipynb** notebook is code with respect to general loss whereas **MODELS_RMSE.ipynb** is code with respect to RMSE loss.

Run Jupyter notebook by typing **jupyter notebook** on command line. Open the downloaded notebook and run it. Refer the following link to start jupyter notebook [<https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html>]
Install all the specified libraries to get complete result.

Models used and References:

- Neural Network [https://medium.com/@gk_/text-classification-using-neural-networks-f5cd7b8765c6]
- XGB [https://suatatan.com/posts/sklearn_xgboost_tc/]

-
- SVM [<https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34>]
 - OrdinalModel [Refer Notebook]
 - Bidirectional LSTM [<https://towardsdatascience.com/sentence-classification-using-bi-lstm-b74151ffa565>]
 - RBFN [<https://towardsdatascience.com/radial-basis-functions-neural-networks-all-we-need-to-know-9a88cc053448>]
 - Logistic Model [<https://medium.com/analytics-vidhya/applying-text-classification-using-logistic-regression-a-comparison-between-bow-and-tf-idf-1f1ed1b83640>]
 - Decision Tree classifier [<https://pythonmachinelearning.pro/supervised-learning-using-decision-trees-to-classify-data/>]
 - Random Forest classifier [https://medium.com/@tenzin_ngodup/simple-text-classification-using-random-forest-fe230be1e857]
 - Extra Trees classifier [<https://machinelearningmastery.com/extra-trees-ensemble-with-python/>]
 - AdaBoost [<https://www.kaggle.com/mehmetlaudekman/getting-started-tfidf-adaboost-text-classification>]
 - GradientBoost [<https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/>]
 - Bagging Classifier [<https://www.geeksforgeeks.org/ml-bagging-classifier/>]
 - Convolutional Neural Network [<https://medium.com/jatana/report-on-text-classification-using-cnn-rnn-han-f0e887214d5f>] ,
[<https://machinelearningmastery.com/develop-n-gram-multichannel-convolutional-neural-network-sentiment-analysis/>]

Product Recommender System

Required Libraries and Installation Commands

Numpy [pip install numpy], pandas [pip install pandas], matplotlib [pip install matplotlib], sklearn [pip install scikit-learn], BentoML [pip install bentoml]

Tasks performed

- **Exploratory Data Analysis using Numpy, Pandas and matplotlib**

Data visualization using matplotlib and data preprocessing using pandas and numpy

- **Models used**

Popularity Based , Utility Based and item-item based Models.

- **Model Serving**

Model Serving using BentoML , Refer the following link to start with bentoML [<https://docs.bentoml.org/en/latest/quickstart.html>]

How to run the Project?

First install jupyter notebook in your PC using **pip install notebook** or download anaconda [<https://test-jupyter.readthedocs.io/en/latest/install.html>].

Using the following link to get the code:

[<https://drive.google.com/drive/folders/1wOVSGIzwwu1E-s7H8v7LzT4LRvNI9azO?usp=sharing>]

and dataset for Product recommender System:

[https://drive.google.com/drive/folders/1_70AfGZH3WFv9AHzEOahB_Wvddrzc8L0?usp=sharing] . **product_recommendation_system_for_e_commerce_bentoml.ipynb** has BentoML code.

Run Jupyter notebook by typing **jupyter notebook** on command line. Open the downloaded notebook and run it. Refer the following link to start jupyter notebook [<https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html>]
Install all the specified libraries to get complete result.

Models used and References:

- Popularity Based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]
- Utility Based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]
- item-item based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]

[recommendation-systems-machine-learning](#)]

Different Types of Recommender System

Required Libraries and Installation Commands

Numpy [pip install numpy], pandas [pip install pandas], matplotlib [pip install matplotlib], sklearn [pip install scikit-learn], BentoML [pip install bentoml], lightfm [pip install lightfm], scikit-optimize [pip install scikit-optimize], scipy [pip install scipy], Pandas-profiling [pip install pandas-profiling], surprise [pip install surprise]

Tasks performed

- **Exploratory Data Analysis using Numpy, Pandas and matplotlib**

Data visualization using matplotlib and data preprocessing using pandas and numpy

- **Models used**

Popularity based, Utility Based, item-item based , Content Based, Knowledge Based, Demo Filtering, Meta Based , Hybrid and Model Based Recommender System Models.

- **Model Serving**

Model Serving using BentoML , Refer the following link to start with bentoML [<https://docs.bentoml.org/en/latest/quickstart.html>]

How to run the Project?

First install jupyter notebook in your PC using **pip install notebook** or download anaconda [<https://test-jupyter.readthedocs.io/en/latest/install.html>].

Using the following link to get the code:

[https://drive.google.com/drive/folders/1P6YTyJ9qPzJNbCK7_shb_9QkeXrdun90?usp=sharing]

and dataset for Types of Recommender Systems:

[https://drive.google.com/drive/folders/1_70AfGZH3WFv9AHzEOahB_Wvddrzc8L0?usp=sharing] .

Run Jupyter notebook by typing **jupyter notebook** on command line. Open the downloaded notebook and run it. Refer the following link to start jupyter notebook [<https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html>]
Install all the specified libraries to get complete result.

Models used and References:

- Popularity based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]
- Utility Based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]
- item-item based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]
- Content Based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]
- Knowledge Based Model [<https://www.fi.muni.cz/~xpelane/PV254/slides/other-techniques.pdf>]
- Demo Filtering Model [<https://www.bluepiit.com/blog/classifying-recommender-systems/>] [<https://thehomeinspectorsgroup.com/wp-content/uploads/formidable/5/demographic-based-recommender-system.pdf>]
- Meta Based Model [<https://medium.com/analytics-vidhya/metadata-based-recommender-systems-in-python-c6aae213b25c>]
- Hybrid Model [<https://www.bluepiit.com/blog/classifying-recommender-systems/>]
- Model Based Model [<https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>]

REFERENCES:

- [<https://towardsdatascience.com/brief-on-recommender-systems-b86a1068a4dd>]
- [<https://blog.cambridgespark.com/nowadays-recommender-systems-are-used-to-personalize-your-experience-on-the-web-telling-you-what-120f39b89c3c>]
- [<https://towardsdatascience.com/3-approaches-to-build-a-recommendation-system-ce6a7a404576>]
- [<https://github.com/YuzhouPeng/Linedin-User-profile-Hybrid-Recommendation>]
- [https://github.com/vjvishaljha/Recommender_Sys]
- [https://github.com/charithcherry/Innomatics_Internship_APR_21/tree/main/Recommender%20System]
- [<https://paperswithcode.com/task/recommendation-systems>]

-
- [<https://ilmoirfan.com/developing-knowledge-based-recommender-system/>]
 - [<https://github.com/Invince11/IMBD-Movie-Recommendation-System>]
 - [<https://towardsdatascience.com/pytorch-bentoml-heroku-the-simple-stack-65196aad181c>]

Deep Learning Model to classify person's actions while driving

Required Libraries and Installation Commands

Numpy [pip install numpy], OpenCV [pip install opencv-python], uuid [pip install uuid], pandas [pip install pandas], matplotlib [pip install matplotlib], sklearn [pip install scikit-learn]

Tasks performed

- **Custom Image Data Generation using Webcam**

Using OpenCV to collect significant amount of images for each class. Classes considered here are : Phone , PhoneEar and awake

- **Models used**

Custom VGG Model Refer the following link to start with VGG [

<https://www.analyticsvidhya.com/blog/2021/06/build-vgg-net-from-scratch-with-python/>]

What is VGG?

VGG- Network is a convolutional neural network model proposed by K. Simonyan and A. Zisserman in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" [1]. This architecture achieved top-5 test accuracy of 92.7% in ImageNet, which has over 14 million images belonging to 1000 classes.

It is one of the famous architectures in the deep learning field. Replacing large kernel-sized filters with 11 and 5 in the first and second layer respectively showed the improvement over AlexNet architecture, with multiple 3×3 kernel-sized filters one after another. It was trained for weeks and was using NVIDIA Titan Black GPU's.

- **Model Evaluation**

Custom VGG Model is evaluated using classification report .A classification report is a performance evaluation metric in machine learning. It is used to show the precision, recall, F1 Score, and support of your trained classification model.

How to run the Project?

First install jupyter notebook in your PC using **pip install notebook** or download anaconda [<https://test-jupyter.readthedocs.io/en/latest/install.html>].

Using the following link to get the code and models:

[<https://drive.google.com/drive/folders/1ksWyE08EFmqvVgg2tseHNAJIAttjGMBR?usp=sharing>].

To create custom dataset run **DATA PREP.ipynb**. Change labels, set number_imgs and provide path to store data. **custom_VGG_model.ipynb** contains code to train VGG model.

Once data is created in class folders, split this data as train and test data.create train and test folder each having the class folders with split data.

Run Jupyter notebook by typing **jupyter notebook** on command line. Open the downloaded notebook and run it. Refer the following link to start jupyter notebook [<https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html>]
Install all the specified libraries to get complete result.