OUTSTANDING PROJECTS

NAME: D KARTHIK

BATCH: ML-MAY

Parkinson-detection

Description and Problem statement:

A machine learning based approach to detecting the presence of Parkinson's disease from spiral tests of patients. T

The dataset contains tests of 15 people from the control group and 62 tests of people suffering from Parkinson's disease.

Results

We tested several classification algorithms such as Logistic Regression, Random Forest, SVM etc. The best results were obtained using SVM

Accuracy	100 %
F1	0.66
Precision	0.5
Recall	1

Future work

As indicated in Results, not all handwriting tasks provide the same level of discrimination power. After evaluating our results, it is evident that some features are more useful for diagnosis than others. We can use actual handwriting to improve on our results.

Decision support tools are gaining significant research interest due to their potential to improve health-care provision. Among many possible approaches, those that provide noninvasive monitoring and diagnosis of diseases are of increased interest to clinicians and biomedical engineers. We aim to provide this diagnosis to people in remote areas where healthcare is not just lacking but extremely inadequate

Stroke Prediction

Problem statement:

According to the World Health Organization (WHO) stroke is

the 2nd leading cause of death globally, responsible for approximately 11% of

total deaths.This dataset is used to predict whether a patient is likely to get a

stroke based on the input parameters like gender, age, various diseases, and

smoking status. Each row in the data provides relevant information about the

patient.

Task: Build a model that can be used to predict stroke.

Summary

In summary, two highest stroke prediction performance were achieved by XGBoost and random forest; three most important features (in descending order) for stroke prediction were 'age', 'avg_glucose_level', and 'bmi'.

Netflix Movies and TV Shows

Problem Description

Netflix is all about connecting people to the movies they love. To help customers find those movies, they developed world-class movie recommendation system: CinematchSM. Its job is to predict whether someone will enjoy a movie based on how much they liked or disliked other movies. Netflix use those predictions to make personal movie recommendations based on each customer's unique tastes. And while **Cinematch** is doing pretty well, it can always be made better.

Now there are a lot of interesting alternative approaches to how Cinematch works that netflix haven't tried. Some are described in the literature, some aren't. We're curious whether any of these can beat Cinematch by making better predictions. Because, frankly, if there is a much better approach it could make a big difference to our customers and our business.

Problem Statement

Build a model to predict best rated movies and TV shows

Conclusion

- According to our ploblem statment Netflix is all about connecting people to the movies they love. To help customers find those movies, they developed world-class movie recommendation system: CinematchSM. Its job is to predict whether someone will enjoy a movie based on how much they liked or disliked other movies. Netflix use those predictions to make personal movie recommendations based on each customer's unique tastes.

1. As we know we have dataset which contains MovieIDs range from 1 to 17770 sequentially. CustomerIDs range from 1 to 2649429, with gaps. There are 480189 users.Ratings are on a five star (integral) scale from 1 to 5. Dates have the format YYYY-MM-DD. And as we can see that we have data are in different formate and we need to make it in a format so that we are able apply models on it. And for that what we are doing as we are puting it all the file and merging movies with users and their rating in single dataframe.
2. So after doing all this we will do some EDA on whole dataset, so that we will able to visualise our dataset like distribition of the ratings,what is the avg rating of the movie or avg rating given by the users to the movie and lot more
3. After that we we split our data in train and test which is in ratio of 80:20 and try to to EDA on it. And then we are creating MF of user and movies and make it sparse as we can see our data frame is more than 90% sparse which means very less non zero value in the matrix. and we will do this for our both train and test data set.
4. And then we try Computing Similarity matrices for both user-user similarity and movie-movie similarity but as we can see calculating Similarity_Matrix is not very easy(unless we have huge Computing Power and lots of time) because of number of. users and movies being large.
5. In above points as we have true to compute similarity but it doest works and after we try some other methods like dim reductions and try to compute but unfortunatly it also doest works and as we can see it taking more time and memory than our above method amd the ple is due to dense matrix. so at last what we do we will try to compute similar users for a particular user, whenenver required (ie., Run time) so that at one time we are not going to compte similarity for the whole users/ movies we will do it at run time when ever required for that pertifular user/ movie. And after that we just try to see that it really works or not and we jut got a awsome result. As we can see we have provied a movie id that with

movie name Vampire Journals and we got a good result which is similar type movie which we have provied as input.

6. After doing lots of stuff now we will work with different machine learning models and try to compare results of all that and but before that lets first sample our data set because we have lots of data and if we work with all data it will take lots of time so first we will samle our data and then we will introduce with some feature engineering which we are going to use it as a feature on our machine learning models.
7. As we can see in given diagream its shown that in this case study we are using a need lib that is surpise lib with paralell to xgboost models with perform matrix RMSE and MAPE with some hyperparameter tuning on xgboost.

THANK YOU