**Machine Learning Document.**

**User Recommendation Task**

This task involves recommending Users based on two categories:

* How popular they are. This is measured by the number of followers they have
* Their similarity with other users.

Recommendation made on how popular they are. (Popularity based recommender system):

This includes 2 python modules;

* Rec.py:
* This module is introduced with importation of the following modules:

Math

Copy

* It includes a class that includes the following functions:
* init:

This function initializes user id for the model

* Smooth user preference:

This function defines user preference

* Create:

This function creates a popularity model based recommender, trains it and it returns the top 10 recommendations.

* Recommend:

This function recommends followers based on the popularity model and returns popular users.

* Popularity.py:
* This recommends users with the most followers to be followed. It starts by importing the needed modules that include:
* Pandas
* Joblib
* Sklearn.model\_selection
* Rec
* After importation of the needed modules, It implements the following functions:
* Get data:

This function is the first step to recommending the most followed users. it reads the data from the data file and returns what has been read.

* Save model:

This function saves the recommended output to a file and returns the list of file names in which the data is stored.

* Main:

The Function classifies and defines train data and test data

Recommendations based on User’s similarity with other users.

This includes three python modules;

* Sorter.py:

This module sorts users bio and titles of articles of a user. To accomplish this task, it includes the following steps:

* Import of needed modules:
* Pandas
* Create a function that defines users preference and returns the title of articles by a user, as well as their bios in an output file.

* User Recommender System.py:

This module includes the following steps to build the model needed for the recommendation:

* Import of needed modules:
* Pickle
* Pandas
* Sqlalchemy
* Sklearn.metrics.pairwise
* Sklearn.feature extraction.text
* Mysql.connector
* Get the data from the data base and showing the tables
* Read the database into a variable
* Remove irrelevant tables from the data
* Check the other tables shown that id means user id , so id column is renamed to user id
* Fill the missing values with empty string and dropping rows with empty bio
* Remove special characters
* Check tf-idf matrix required for calculating the cosine similarity
* Declare the function that uses the model to get similar users based on user id
* Define a function that recommends top ten people to follow based on their similar bio
* Save the model to disk
* Load the model
* Read the posts from database
* Fill missing values with empty strings
* Take out special characters
* Remove html tags
* Remove white spaces including new lines
* Remove image files, brackets
* Compute tf-idf matrix required for calculating cosine similarity
* Run the function that would use cosine model to recommend articles to users
* Load the model
* Save the model to disk
* Load the model from the disk
* User Recommender.py:

This module takes the following steps to make recommendations of similar users:

* Import the needed modules:
* Argparse
* Pandas
* Mysql.connector
* Sqlalchemy
* Sklearn.metrics.pairwise
* Sklearn.feature extraction
* Read from database
* Extract relevant tables with users and their bio
* Read users table
* Remove irrelevant columns
* Construct a reverse map of indices and user names
* Fill empty string
* Remove users without bio
* Clean user bio data
* Create tfid vectorizer object
* Computing tf-idf matrix requires for calculating cosine similarity
* Defining a function that uses the cosine similarity model to get users similar to the user based on their bios and then returns a series of similar users based on their bios
* Defining a function that returns the recommendations