AI-ML Assignment Solution

1. Genre Prediction :

- Firstly, import all required libraries.
- Then import the data (Kitaab)
- Creating the dataframe of necessary features that we needed.

- Preprocessing the synopsis :

  - Cleaning the data - only the alphabets are kept while filtering out everything else and then the alphabets are converted into lowercase.

  - Removal of stop words (most common words)- Another part of data cleaning is the removal of stop words – that is, common words like "the", "a", "an". They are assumed to have no consequence over the classification process.
  - Lemmatization is performed- Greater the degree of randomness, greater would be the computation time, and lesser would be the efficiency of the detection of patterns in the textual corpora. We introduced the module which performs lemmatization on words, that is, it groups different versions of the same word into one – for example, "do/doing/does/did", "go/going/goes/went" – so as to not let the algorithm treat similar words as different, and hence, make the analysis stronger.

  - Stemming is performed- Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words "chocolates", "chocolatey", "choco" to the root word, "chocolate" and "retrieval", "retrieved", "retrieves" reduce to the stem "retrieve"

- Converting Text in Books dataset to Features

  Now, we denote an unique number to each of the 10 genres from 0 – 9

- **Fitting the models**
- At first a **80-20% split** was performed on the dataset
- tf-idf was performed on "synopsis" for both train(i.e xtrain) and test(i.e xval).

  tf–idf or TFIDF, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

  tf(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

  idf(t) = log_e(Total number of documents / Number of documents with term t in it).

- **Using Logistic Regression : 61% accuracy**

- **Using SVM : 74% accuracy**

2. Rating Prediction :

- Then import the data (Kitaab)
- Creating the dataframe of necessary features that we needed.

- Preprocessing the synopsis :
  - Cleaning the data remove comma
    - replace k with 000
    - remove decimal
    - remove all characters only
  - creating new features
- **Fitting the models**
- At first a **80-20% split** was performed on the dataset
- **Using Linear Regression : Score-42.7%**
- **Random Forest : Score- 37.2%**