PGR210-1 23H Machine Learning and Natural Language Processing

Candidate numbers: 2004, 2018, 2020, 2021, 2030

December 2023

SECTION I: For both Machine Learning and Natural Language Processing

INTRODUCTION

This report comprises three distinct sections. Section I encompasses literature reviews relevant to BOTH Machine Learning and Natural Language Processing. Sections II and III encompasses abstract, introduction, methodology, results and discussion to ML and NLP, respectively.

LITTERATURE REVIEW

Machine Learning (ML) and Natural Language Processing (NLP) represent a dynamic subset of artificial intelligence that enables computers to acquire knowledge from data without any explicit programming. ML identifies patterns, and predicts outcomes via supervised, unsupervised, and reinforcement learning methods.

Natural Language Processing (NLP) trains computers to comprehend, interpret and generate human language. NLP leverages ML techniques to perform various tasks, such as sentiment analysis, and language translation. In this case, the emphasis centers on identification of similarity or dissimilarity in texts. Tasks such as text classification are handled by supervised learning, unstructured textual material is arranged by unsupervised learning and interactive language tasks such as training chatbots are handled with reinforcement learning.

The combination of ML and NLP is helpful in automating and understanding natural languages, benefiting industries relying on human language. Their impact continues to drive advancements in automation and human-computer interaction.

Typical steps in Machine Learning and Natural Language Processing.

1. Define the problem.
2. Gather and explore data.
3. Data preprocessing.
4. Data splitting.
5. Model building.
6. Train the model.
7. Evaluate the model.
8. Hyperparameter Tuning
9. Feature Engineering
10. Cross validation.
11. Model Interpretation.
12. Deploy the model.
13. Monitor and maintain the model.

The 13 steps are conventional but their execution varies based on the problem, ML algorithm, and dataset features.

It is important to understand that the success of ML and NLP performance does not only rely on those 13 steps but on meticulous attention to details, methods, techniques, and key variables is crucial. Effectiveness depends on their application and consideration of various elements.

1. Define the Problem: Understand the problem and define the goal of the project. In our case the questions were defined and given in the exam instructions.

2. Gather and Explore Data: Collect and explore relevant data for training and testing. (datasets were provided).

3. Data Preprocessing: Data preprocessing is a crucial step in both ML and NLP workflows. The aim is to enhance the quality and relevance of input data to optimize model performance. In ML, this involves tasks such as handling missing values, normalizing numerical features, and encoding categorical variables. In NLP specific pre-processing steps are applied to text data, encompassing tokenization, stemming or lemmatization, and removal of stop words and punctuation. Additionally, both ML and NLP benefit from addressing issues like data imbalance, noise reduction, and scaling to ensure a balanced and representative dataset. Effective data pre-processing not only mitigates biases and inconsistencies but also facilitates the extraction of meaningful patterns and relationships, ultimately contributing to the robustness and accuracy of machine learning models in various applications.

HANDELING MISSING VALUES

Handling missing values is a critical step of data preprocessing hence most ML algorithms require all the values in a dataset. There are various techniques that can be used to address the missing values such as total removal of missing values, mean/median/mode imputation, forward/backward fil, matrix factorization and even deep learning approaches such as autoencoders, however we have opted for model based imputation techniques such as K-Nearest Neighbor and regression models that use the correlations and patterns in the dataset to predict the missing values.

Dropping Missing Values is a straightforward method that maintains the dataset's structure but may result in significant data loss, particularly when values. When dropping missing values, caution must also be used since eliminating non-random missing values may result in unwanted biases [19]

Imputing with Mean/Mode/Median is an easy-to-implement method that works well for variables with normal distribution. This preserves the number of observations but overlooks relationships between features and it may even distort the data's distribution. [10]

Predictive Model Imputation, using the tool scikit-learn's 'IterativeImputer', incorporates correlations between variables while still preserving data variability and the structure. However, it operates under a linear assumption and may face challenges with small datasets or intricate connections, rendering it susceptible to outliers. [21].

The type of data, the quantity of missing values, and the research's objectives, must be taken into consideration in order to determine which course of action is optimal. Therefore, data exploration and identification of the root causes should be completed before deciding on the best step for the missing values.

K-Nearest Neighbors imputation

[10] K-Nearest Neighbors (KNN) imputation replaces missing values in the dataset with mean val-

ues from the nearest neighbors of the n_neighbors in the training set. Thereafter it utilizes the Euclidean distance metric to complete the imputation.

$$Distance = \sqrt{\sum_{i=1}^{n}(X_i - Y_j)^2}$$

The variables "Xi" and "Yi" represent the values of the i-th feature data points respectively.

KNN imputation has several benefits and disadvantages. The advantages are that it can identify correlations between variables and maintain the underlying data structure while the disadvantages include sensitivity to outliers and computational cost, especially for big datasets [10].

Regression imputation

The regression imputation uses a regression model to replace the missing values in a variable by creating a mathematical relationship between the dependent (feature with the missing values) and independent variables (other features of the dataset) [1].

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$$

The Y is the dependent variable, the X's are the independent variables, the B's are the coefficients that represent the strength and directions of the relation between the variables and the ϵ represents the differences between the actual and predicted values.

The imputation process is split in two parts. The first one involves training the model with the dependent and independent variables while the second part involves applying the trained model to the missing values instances to predict their values.

However, it requires careful consideration of linearity assumptions and text data handling, impacting the accuracy of imputed values.

Normalization Methods

Data Normalization is a process in ML that standardizes and scales the data ensuring that all the features have an equal impact, preventing one feature from being more dominating than other. Additionally, it improves the workflow by eliminating the redundant data and minimizing the data modification errors. There are various normalization methods such as Min-Max Scaler, StandardScaler, Robust Scaler..

StandardScaler is the most utilized method to scale numerical values in ML. Standard Scaler transforms the data point by subtracting the mean and dividing the standard deviation resulting in a standard deviation of 1 and a mean of 0. The standard scaler uses the following formula and includes a total of 3 steps to achieve the end result. [1]

$$Xstandardized = \frac{x - mean(X)}{std(X)}$$

1. Finding the mean by using the formula below and the result will represent the avg value of all the features in the dataset.
$$\bar{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_{ij}$$
2. Finding the standard deviation (amount of variation in all the values of the dataset).
$$\sigma X_i = \sqrt{\frac{\sum_{j=1}^{n}(X_{ij} - mean(Xi))^2}{n}}$$
3. The final step will be to apply StandardScaler transformation to each and every feature of the dataset by using the following formula:
$$X^i_{standardized} = \frac{x^i - mean(X)}{std(X)}$$

MinMaxScaler is another popular method used for scaling numerical values, however it only scales the data between 0 and 1. To achieve this it subtracts the minimum value and divides it by the range. It uses the formula below and follows three key steps to achieve the end result [32].

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

1. Finds the min and max value for a feature in the dataset.
2. Applies the MinMaxScaler transformation for each feature X_i by using the following formula which results in a new normalized feature X.

$$X_{normalized}\, i = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

3. The final step will be to scale the value to ensure that the min value corresponds to 0 and the max value corresponds to 1. This step ensures that all the features stay in the same range, preventing dominance between the features.

Robust Scaler is an algorithm that scales features by using statistics, resistant to outliers, primarily through the quantile range (IQR: Interquartile Range by default) which is the range between the 25th and 75th quantiles, and centers data by removing median.While it is robust to outliers, it does not scale data to a specific range . In the case of NLP, it is crucial to obtain a 'cleaned' collection of tokens, Bags of Words (BoW) which is a list of qualifying, unordered, un structured words from texts. This can be achieved by implementation of the following steps [33].

$$X_{Scaled} = \frac{x - median(x)}{IQR(X)}$$

Text processing in NLP

In the case of NLP, it is crucial to obtain a 'cleaned' collection of tokens, Bags of Words (BoW) which is a list of qualifying, unordered, un structured words from texts. This can be achieved by implementation of the following steps. (scikit-learn-robust scaler)

Tokenization is the process by which texts are broken down. and documents into smaller words or tokens. [6]

Lowercasing involves converting all capital letters to lowercase for consistency when processing text data.

Removing white space eliminates white spaces from each word or token to standardize format.

Stop words removal involves removing all the words that have little meaning such as "and"", "the ","in ","is", for efficiency and reduced dimensionality. [15]

Removing punctuation and numbers involves removing all the non-alphabetic characters to focus on textual content.

Stemming and Lemmatization: involves reducing words to their stem by removing suffixes or prefixes while Lemmatization involves reducing words to a meaningful base form, or "lemma" considering morphological analysis. Both methods enhance normalization and model performance, making the process of analysis efficient. [36]

Removal of duplicates involves eliminating all the duplicates data, ensuring distinctiveness and the lightness of the dataset's overall weight.

Creating TF and TF-IDF vectors.

Term Frequency (TF) and Inverse Document Frequency (IDF) are fundamental concepts in NLP for text representation.

Term Frequency (TF) measures how frequently a term appears in a document. E.g., in document d, the term frequency of a given word t is calculated as [34]

$$tf(t,d) = \frac{\text{count of } t \text{in } d}{\text{total number of words in } d}$$

TF indicates a word's relevance in a document, allowing representation of the document as a vector with entries for each term's frequency. Document Frequency (DF) measures term importance in a document collection by counting how many documents contain that term. It's represented as df(t), and higher values indicate greater term presence.

Inverse Document Frequency (IDF): quantifies how relevant a word is globally (to the documents in the corpus.) It balances out the impact of common words.

$$idf(t) = log(\frac{N}{df(t)})$$

N is the total number of documents.

The primary benefit of the IDF algorithm is that it ensures that frequent terms don't dominate.

TF-IDF: combines both TF and IDF. It assigns words weights according to how important they are in a document in relation to the total corpus. The formula for TF-IDF is [7]

$$tf - idf(t, d) = tf(t, d) \cdot idf(t)$$

Words with higher TF-IDF scores are considered more significant than the others..

In practice, libraries like scikit-learn provide efficient implementations of TF-IDF vectorization for NLP tasks. Researchers and practitioners use it extensively for text classification, information retrieval, and clustering [35]

Dimension Reduction in NLP

In natural language processing, dimensional reduction seeks to minimize the features or dimensions in a dataset while maintaining critical information, addressing issues like computational complexity, and risk of overfitting. Strategies include:

Feature Selection: Working with a dataset containing class labels, one can filter out low-information features using metrics like Information Gain or Chi-Squared. By keeping filtering out and keeping top-scoring terms (e.g., 1-10% based on Information Gain) can improve classification accuracy.

In unsupervised tasks (e.g., text retrieval or clustering), focus on terms occurring between 1% and 10% of documents, as they are most discriminative [14].

Cosine Similarity: Compute similarity between documents using cosine similarity.

It scales well for documents of varying lengths and provides values in the [0, 1] range, indicating similarity percentages.

Feature Selection: Linear SVMs work well with high-dimensional features. Consider random forests and deep learning models, which excel in NLP tasks and machine learning in general [14]

4. Data Splitting: This step involves dividing the dataset into a minimum of two (training and testing sets). Ensuring the model performs well on the unseen data is the main goal. Splitting the data also makes model evaluation using measures such as R^2, MAE, and MSE possible. Random splitting avoids biases and ensures replicability by specifying the random state. To this end, a proper use of stratification is critical.

Splitting the data also makes model evaluation using measures such as R^2, MAE, and MSE possible. Random splitting avoids biases and ensures replicability by specifying the random state. To this end, a proper use of stratification is critical.

Stratification reduces sampling error with distinct subgroups. It is a useful tool for ensuring unbiased analysis especially in tasks involving classification and model evaluation. Populations are divided into homogeneous subgroups, preventing dominance of any subgroup resulting in impartial and accurate analysis for model evaluation and classification. Consider dataset size and subgroup clarity before using stratification, as these factors can impact its effectiveness and results. [9]

5. Choose a Model: Choose a machine learning algorithm in accordance with the present problem. The following are some of the common models

Classification models

Decision Trees models are used for classification and regression in ML. It functions similarly to a tree, with each node standing for a decision or an investigation, while each branch represents an outcome and lastly each leaf note represents a class label or a numerical value. The main goal is to forecast the target variable by learning hierarchical decision rules from the input information, thereafter dividing the data into smaller groups until reaching the final outcome. [2]

Random Forest works by building multiple decision trees models during training. It produces predictions based on the mean or mode of the individual trees ensuring to create a prediction with higher accuracy. [28].

Support Vector Machines (SVM) is a model for solving complex classification, regression and outlier detection problems. Support vectors and kernel functions were used to determine which hyperplane in the feature space best maximizes the margin between different classes in order to capture complex relationships in the data. [22]

K-Nearest Neighbours (KNN) uses the Euclidean distance metric to replace the missing values with the mean values for the n_neighbors' closest neighbors in the training set. [29]

Naïve Bayes is widely used for classification tasks and it is a probabilistic ML algortihm based on Bayes Theorem. The class with the highest probability is selected as the final result once each class's probability is calculated. [3]

Clustering Models

K-Means is a popular data clustering algorithm that iteratively assigns each data point to its closest centroid, grouping the similar data into patterns. The main objective is to reduce the overall distance between each data point and its average group to identify K groups. [25]

Hierarchical Clustering works similar to a tree-like structure. It groups the objects into two categories the ones that are similar to each other and those that aren't, visualizing clusters in a dendrogram which is a hierarchical tree. For this a bottom-up (agglomerative) or top-down (divisive) strategy is used. An advantage of the model is that it does not require a predefined cluster count, but a notable drawback is high computational cost with large datasets. [8]

DBSCAN (Density-Based Spatial Clustering of Application with Noise) is a density-based clustering algorithm that identifies and groups the data points in the dense regions filled with clusters in a dataset . Thereafter it divides them according to their density within the data space into points called core, border or noise. One advantage is its ability to find clusters of arbitrary shapes, robust to outliers while one of its disadvantages is its sensitivity regarding the parameters.[24]

Mean Shift is a non-parametric algorithm for centroid-based clustering whose primary goal is to identify clusters of different sizes within a smooth density of samples excellent for various cluster shapes without specifying the count. However, it can be computationally expensive. (scikit-learn-MeanShift. [26]

Gaussian Mixture Model (GMM) the data has to originate from a composite of several different Gaussian distributions. Its advantages would include the flexibility with relation to cluster covariance and the capacity to provide probability of belonging to each cluster while one of its disadvantages would be its sensitivity to initialization. [5]

OPTICS (Ordering Points To Identify the Clustering Structure) is an extension of DBSCAN. It is successful in finding clusters at different densities. Its advantages would include the robustness to the noise and outlier while one of its disadvantages would be its sensitivity to the parameters. [27]

Self-Organizing Maps (SOM): is a neural network-based method that reduces dimensionality and projects data onto a 2D grid. It is useful for visualization hence it captures non-linear relationships. One disadvantage would be the requirement of hyperparameters. [23]

NLP models:

Latent Dirichlet Allocation (LDA) algorithm in NLP is a topic modeling technique that finds documents by searching for a given list of keywords. This method can be used to discover publications or books using a text summary, and create semantic associations between words, among many other things. Its simplicity and speed make it ideal, but its disadvantages include its inability to handle regular distribution and its unsuitability for texts or documents with few categories. [17]

Truncated Singular Value Decomposition SVD is a matrix factorization technique where matrix M is factored into three matrices, U, $\sum$, and V. Unlike Principal Component Analysis (PCA), SVD works on the data matrix, enhancing model performance and training speed by reducing features while preserving essential information. [4]

Non-negative matrix factorization NMF is a matrix factorization approach that involves constraining the matrices to be nonnegative. It automatically extracts sparse but relevant features from a set of nonnegative data vectors. It is beneficial when working with image processing and text mining. Its flexibility which allows it to be applied to any kind of data would be its biggest advantage while the disadvantage would be its sensitivity to factorization. [20]

Convolutional Neural Networks (CNN) are a special type of artificial neural network created for processing and analyzing visual data. It is good at analyzing visual data through layers like convolutional, pooling, and fully connected. They're also effective in NLP for tasks like text classification, document similarity, and sentiment analysis. [37]

Recurrent Neural Networks (RNN) are designed to handle sequential data including time series and natural language, capturing time base dependencies. They are crucial in NLP for tasks like language modeling, enabling complex language processing by understanding word relationships through feedback loops.[16]

Transformer Modes (BERT, GPT) are essential for NLP tasks that involve text summarization, question answering, sentiment analysis, and language translation. Their strength lies in handling long-range dependencies and generating coherent, contextually relevant language representations through bidirectional context (BERT) and autoregressive language modeling (GPT). [38]

6. Train the Model:
   - Feed the training data into the chosen model.
7. Evaluate the Model:
   - Use the testing set to evaluate the model's performance
   - The Silhouette Score is not suitable measure for OPTICS or DBSCAN

Unsupervised learning methods such as DBSCAN, Mean Shift, Gaussian, OPTICS, and Self-Organizing Maps (SOM) don't support metrics such as MAE and R^2 hence these methods are used to evaluate supervised models performance. Silhouette score assesses cluster quality in unsupervised methods like SOM, emphasizing clustering quality based on data structure.

Classification

Accuracy is a metric that measures how accurate a prediction is, however it is not the best fit for datasets that have a lot of differences in the number of instances across different classes. [12]

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Instances}$$

Precision [**googleClassificationPrecision**] is used to predict the positive values among all instances, in the different classes . It is extremely handy, specially In cases where the false positives are wanted.[13]

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall [**googleClassificationPrecision**]is used to calculate models capacity to capture all the positive instances by determining the percentage of true positive predictions among all actual positive instances. It is useful when it is more critical to miss positive cases than to have false positives.

$$Recall = \frac{TruePositive}{TruePositive + FalsePositives}$$

F1 score provides a reasonable assessment of a classifier's performance and it is helpful in cases where the dataset exhibits an uneven class distribution. [13]

$$F1 = 2X\frac{Precicion \times Recall}{Precision + Recall}$$

Regression:

Mean Squared Error (MSE) is the average, or mean, of the squared difference between a dataset's actual and forecasted values., it is ideal for evaluating a regression model performance because of its assessing the accuracy of continuous numerical predictions. [18]

$$\sum_{i=1}^{D}(x_i - y_i)$$

Mean Absolute Error (MAE) is employed to calculate the average absolute differences between the actual and the forecasted values in a dataset by taking in considertion the average of absolute errors for a set of observations and forecasts. [30]

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

R-squared (R2) is a statistical measure that shows how well a regression model fits the data. Its value is located in a range of 0 and 1, where 1 denotes a perfect fit meaning that the forecasted values match the data perfectly [31].

Clustering

8. Hyperparameter Tuning:
    - Fine-tune the model by adjusting hyperparameters.
9. Feature Engineering:
    - Experiment with creating new features or transforming existing ones.
10. Cross-Validation:
    - Implement cross-validation techniques for a more robust model evaluation.
11. Model Interpretation (Optional):
    - Understand how the model is making predictions.
12. Deploy the Model (Optional):
    - If the model meets the desired performance, deploy it for making predictions on new data.
13. Monitor and Maintain (Optional):
    - Regularly monitor the model's performance and update it as needed.

The primary difference between the steps for machine learning in questioning the consistency of the topics they generate for practical applications. Future work should aim for models yielding stable and reliable topics. Neral and those specific to 'Classification' problem or 'Clustering' problem lies in the choice of algorithm and the evaluation metrics used.

For classification problems, you'd typically focus on algorithms designed for classification tasks (e.g., logistic regression, decision trees, etc.) and use metrics like accuracy, precision, recall, etc., for evaluation. On the other hand, for regression problems, you might use algorithms like linear regression and evaluate using metrics like mean squared error or R-squared. The specific steps can be adapted based on the nature of the problem you're solving.

SECTION II: Machine Learning.

ABSTRACT

The Machine Learning literatures and methodology are examined, followed by an exploration of models and their evaluations. The emphasis was on supervised and unsupervised algorithms, covering classification, clustering, and deep learning-based regression. Additionally, typical ML steps were performed on the provided datasets, including but not limited to, preprocessing, building models, evaluating the model. The results were: 1) Random Forest was the best for classifying levels of health risk with an accuracy of 33.5%. 2) K-Mean performs best in clustering commute patterns into 9 clusters. 3) A Deep Learning Model was created using keras dense layers to predict the property price. Which gave a 0,997 r^2 score.

INTRODUCTION

This report is structured into two different sections. The first section focuses on machine learning and presents a comprehensive exploration of both supervised and unsupervised algorithms. The supervised algorithm focuses on Classification that involves tasks such as data processing, model creation and analysis to predetermine labels to the "Health Monitoring" dataset. Additionally, it explores clustering, using the "Urban Mobility Patterns" dataset to identify distinct commuting patterns through data preprocessing, model building, and detailed analysis. The final task in this section involves Deep Learning-based Regression that utilizes the "House_Price_Prediction" dataset to predict property prices.

Classification and Clustering are the two main areas of focus in the Machine Learning, section I. The tasks which use pre-provided real-world datasets require for the implementation of preprocessing methods, model creation to tackle the problems and to extract analytical insights. In the classification task the primary goal is to develop a model that can predict the target variable "Risk_Level" and in the Clustering task the main goal is to identify distinct mobility patterns and gain insights into urban transportation challenges.

METHODOLOGY

Question 1: Classification

Data Preprocessing: With the "Health Monitoring" dataset in which there are two columns, namely 'Heart_Rate', and 'Blood_Sugar' with missing values in the same 100 rows, out of 3,000 rows.

We use a predictive imputation method, specifically the KNNImputer package from sklearn, to fill all the missing values in the two columns based on the other 3 columns; 'Blood_Pressure', 'Cholesterol', and 'Risk_Level.'

The KNNImputer uses the rows without missing values to train the imputation model, and then it imputes the missing values

Normalization and splitting dataset: After all missing values have been filled by the KNNImputer, the data is then normalized using StandardScaler, a package from sklearn, followed by splitting it into training, and testing sets.

Models: The following are the models we built so we can compare how well each of them performs on this dataset.

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine(SVM)
- K-Nearest Neighbours (KNN)
- Naive-Bayes

A loop is then created to collect the results and its accuracy score in preparation for visualization and analysis.

Question 2: Clustering

Data Preprocessing: The dataset "Urban Mobility Patterns" contains missing values in the same 120 rows out of 5,000 rows, in two of the columns; 'Heart_Rate', and 'Blood_Sugar.' These missing values are handled by the use of a LinearRegression model to impute the missing values through this code.

Normalization: StandardScaler is subsequently used to normalize the data prior to splitting it into training and testing sets.

Models: The main objective of this question is to create clusters by assembling similar data points. There are several machine learning models commonly used. These are the clustering models we built.

Not all the models support the use of the same evaluation metrics. Consequently, the silhouette score can be applied to many of the selected models. Additionally, MSE, R^2, and MAE are employed as scoring metrics for models where they are suitable.

Question 3: Deep Learning

Data Preprocessing: There are two issues with the provided "House_Price_Prediction" dataset.

- There is one row (#2902) that has 0 square_feet for a two-beds, 4-baths property.

This row is dropped on the ground that it is just one row, and an increased probability of overfitting would be an unintended result of predictive imputation.

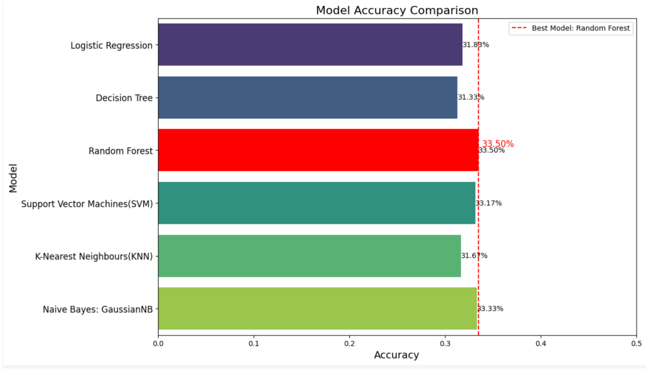
- There are 200 rows of missing values (out of 15,000 rows, a 1.33% of the overall data. They are in three columns; 'Age_of_Property', 'Proximity_to_City_Center', and 'Property_Price'. In handling these missing values, the IterativeImputer with LinearRegression is used.
- Additionally, there were 49 rows of data with property prices ranging below 0. We decided to not impute these 49 rows, to avoid increasing the risk of overfitting . Additionally, it does not provide a big data loss hence it only affects 0.33% of data.
- Normalization: StandardScaler is used to normalize the data.

Building Model: A function is created to help identify the 'best model' by comparing the MSE loss, using 'adam' optimiser, and employing dropout layers between input, hidden, and output layers to avoid overfitting.

ANALYSIS OF RESULTS

Question 1: Classification

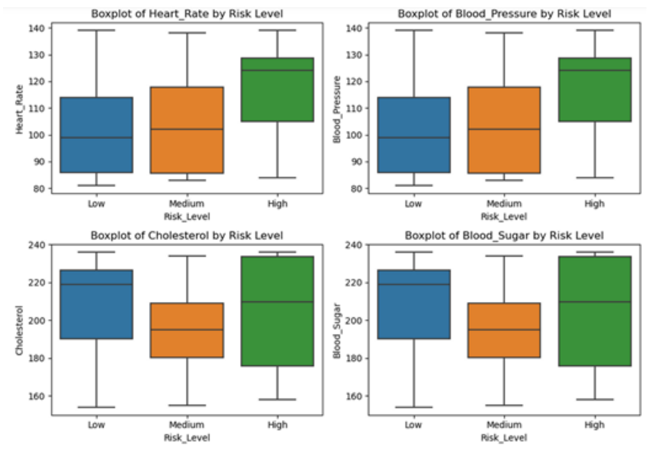
The bar graph below compares the accuracy scores of six different ML models.



Random forest showed the highest accuracy score of 33.50%. This can be attributed to its proficiency in detecting complex patterns in the data. Unlike other models, Random Forest uses a group of classifiers to generate varied decision trees for higher performance on unknown data, thus mitigating the issues associated with training and testing data. [11].

There clearly is a certain level of predictive difficulties as even the best model can only accurately predict the level of health risk 33.5% of the time. This could be due to the fact that determining the level of health risk may require more information that is not currently present in the provided dataset, such as age, BMI, alcohol consumption, tobacco or drugs use, health history, sleep pattern, stress level, physical activity. and etc.

Patterns and insight from the dataset:



Median Heart rate and average blood pressure increase across the three risk levels. However, Cholesterol and Blood sugar are not as indicative of the level of health risk, this is due to the fact that the averages for the low risk group is slightly higher than those of the high risk group, while the values for the medium risk level are at the lowest among the three groups.

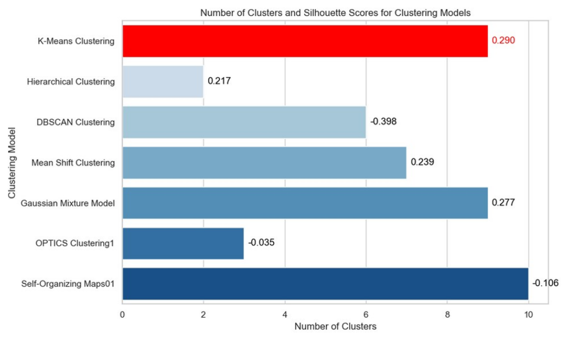
It is notable that based on the high risk values on Cholesterol and Blood sugar levels covering the range in the other risk groups, that Cholesterol levels, and the level of blood sugar are probably NOT determinant factors to health-risk.

Recomendations for future improvements

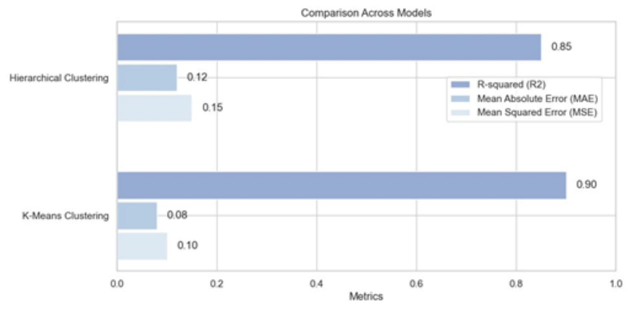
Accuracy can possibly be increased by incorporating more health information (age, weight, BMI, alcohol/drug use, health history, dietary habit, stress level, and sleep patterns, etc..) Having access to this information about a person enables the model to perform a more complex regression analysis to extract relationships of these different pieces of information (which might not be linear and the relationship could be extracted using Polynomial-Regression). Additionally, having more features allows a proper feature extraction to be performed. This will create a more complete picture of what factors actually contribute to the different levels of health risk.

Question 2: Clustering

The bar graphs below gives an overview of the silhouette scores for the selected 7 models.



K-Means clustering produced the highest value at 0.290 for 9 clusters. K-Means outperforms other models in classifying Urban Mobility Patterns. This is indicative that the model is doing a good job at extracting the pattern of commute and grouping them together.



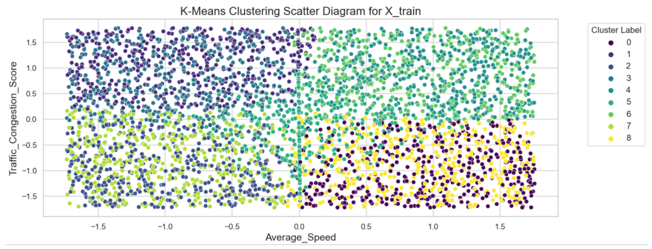
The other models are not designed for predictive modeling hence they do not support R^2, MAE and MSE performance measures. However, there are two other models that support these measurements. The performance metrics confirms that K-means clustering outperforms Hierarchical clustering on this dataset.

Performance analysis:

Metric comparison shows that K-Means having higher R-squared value and a lower MAE is a more accurate model compared to Hierarchical Clustering. Furthermore, K-Means has a substantially lower MSE, suggesting fewer big clustering mistakes and a tighter fit to the data.

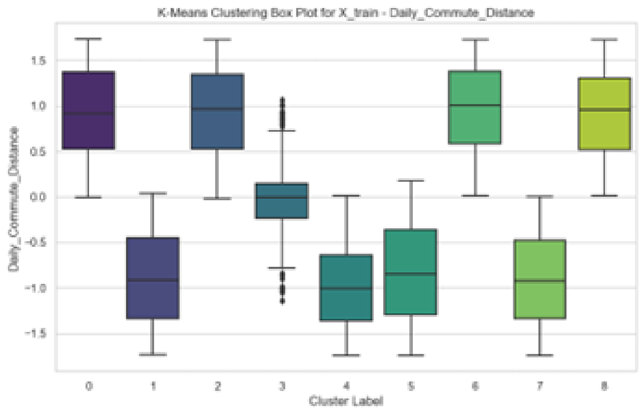
Patterns and insight from the dataset:

There are two clusters in each of the four quadrants and one in the center of the four.



The clusters 0 and 4 on the bottom right are the best of the matrix, with higher avg speed and lower traffic congestion time.Cluster 8 is the avg on both of the features. C7 and C6 have the worst position in regards to the features in this plot.

The normalization process scales features to a common range. This allows for a fair comparison between features with different units and magnitudes. We recognize a pattern in 2 major groups of clusters regarding average speed. One with higher average and one with lower than average.



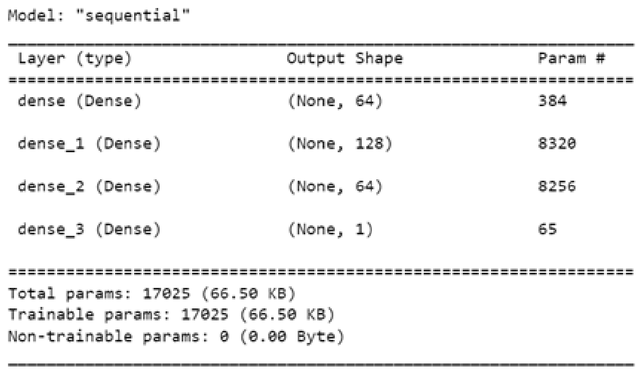
The image displays that clusters 1, 4, 5, and 7 are the closest to the city center while clusters 0, 2, 6 and 8 are the groups furthest away from the city center. Cluster 3 falls in-between the two groups as we can see in the plot.

Recommendations based on the clustering result:

Individuals can utilize the model to get an overview of the commute speed and distance in the process of considering moving to one of these clusters. Additional point to consider is using the coverage of public transportation to the clusters (which is not available in the provided dataset) to help make a decision.

Question 3: Deep Learning

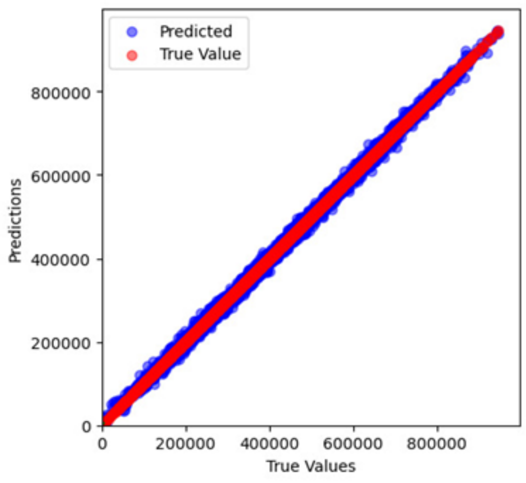
Below is the deep learning model summary



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                384

 dense_1 (Dense)             (None, 128)               8320

 dense_2 (Dense)             (None, 64)                8256

 dense_3 (Dense)             (None, 1)                 65

=================================================================
Total params: 17025 (66.50 KB)
Trainable params: 17025 (66.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

The graph below depicts the predicted values with the true value overlaid indicating that the overall prediction fits the true values .

```
Test Mean Squared Error: 103913544.0
Test Mean Absolute Error: 8030.862230866559
R^2 Score on Test Set: 0.9973327490739607
```



R^2 score of 0.997 measures the proportion of the variance in the house price based on the features. The model processes exceptionally high predictive power. This means that the model can capture the underlying patterns in the data very well, resulting in its near perfect predictions.

The mean squared error between the expected and actual numbers is assessed by the Mean Squared Error (MSE). A relatively low MSE of 20,786,412 is relatively high in absolute terms, but that could be explained by a large range of property prices (min of 409 and max of 984,821.9).

The combination of MSE and R^2 score indicates that the model performs very well on the test set and learns the patterns in the data that generalize effectively to unseen data.

Recommendations to improve the model:

Ensure the integrity of the data (abnormal data ranges), monitor and maintain the model as well as gathering more data to feed it back to the model, perform feature engineering to finetune parameters and features.

CONCLUSION

In creating a classification for the health monitoring dataset; after exploring multiple models and evaluating their performance based on the evaluation matrix, Random Forest achieved the highest accuracy score at 33.50%. One of the key findings is that Heart rate, and blood pressure are the two factors directly correlated to the level of health risk. However, we discover that the Cholesterol and levels of blood sugar are not determining factors for health risk levels.

Regarding the clustering question; K-Means clustering outperformed other models with a higher silhouette score producing 9 clusters. The model recognizes patterns in commute-related features across clusters, such as average speed, commute distances, and waiting times in the Q2. Additionally, it identifies differences in commute distances, waiting time and travel distance for the 9 clusters of property.

With respect to the Deep Learning question; The created model can predict the house price based on the features in the provided data at the 99.7% accuracy.

SECTION III: NATURAL LANGUAGE PROCESSING

ABSTRACT

A dataset of 4,803 rows by 20 columns was provided. Data exploration is done to gain an understanding (top 10 movie language, budget and profitability over time), then preprocessing is performed (among others, 3 rows were dropped, and a new column called ‘description’ was created.)

Text processing is performed on the created movie description, the feed to TF and TF-IDF vectorizer. After which, topic Modeling (LDA, Truncated SVD, and NMF) were created. However, comparisons are primarily done between LDA and NMF. The result is that NMF out performs as it creates the best fit of topics.

The last section deals with identifying similar and dissimilar movies to the “Harry Potter and the Half Blood Prince”. As expected, the list of movies that are most similar to Harry Potter are the other Harry Potter movies. On the other extreme, the list of movies that are dissimilar to Harry Potter includes John Carter, Spider man 3, and Avengers.

INTRODUCTION

The second part of the report focuses on NLP and involves tasks such as text processing, feature extraction and representation using both TF and TF-IDF models. The analysis task focuses on finding movie recommendations to the users based on their preferences and dislikes.

In summary, all the tasks go through three crucial steps such as data preprocessing, model building and analysis. These provide a comprehensive understanding of different methods and models in both ML and NLP.

METHODOLOGY

The following processes were performed onto the supplied “Exam_NLP.csv” file.

- Understanding the dataset.

Examined“Exam_NLP.csv” dataset using methods like.unique(), .describe(), and .value_counts(), and visualized key features.

- Text preprocessing.

Removed rows with NULL in both ’tagline’ and ’overview’, reducing data from 4,803 to 2,800 rows.

Merged ’tagline’ and ’overview’ into a new ’description’ column, the remaining 4,800 rows of movie data.

Applied ’process_text’ function on ’description’ to create ’processed_description’ for vectorization, this contains a ‘cleaned’ description ready to be turned into TF and TF-IDF vectors.

- Creating TF and TF-IDF vectors.

Generated TF and TF-IDF vectors using CountVectorizer and TfidfVectorizer from Sklearn.

- Topic Modeling.

Compared three models: LDA, TruncatedSVD, and NMF, using TF and TF-IDF vectors for internal and cross-model comparison.

- Model Evaluation.

In addition to utilizing visualization to compare the results both within the same model and across the 3 topic modeling models, three measures were used: Perplexity score, Topic Stability score, and Topic Diversity score.

- Similarity Analysis.

The exam asks for a list of movies to recommend to a viewer who likes the ’Harry Potter and the Half-Blood Prince’ movie. To get the list of recommended movies, the TF-IDF matrix was fed through the ‘cosine_similarity’ package where the ’description’ of the movie is used as the point of comparison.

At this stage, a new function named ‘get_similar_movie’ function was created whereby the similarity score of all the other movie descriptions are measured against the ‘index’ movie ‘Harry potter and the Half Blood Prince.’ After the comparison, a list of movies is returned.

One can refine the list further by adding a ‘filter’ to take into consideration only movies containing the words “Harry Potter” in the movie title.

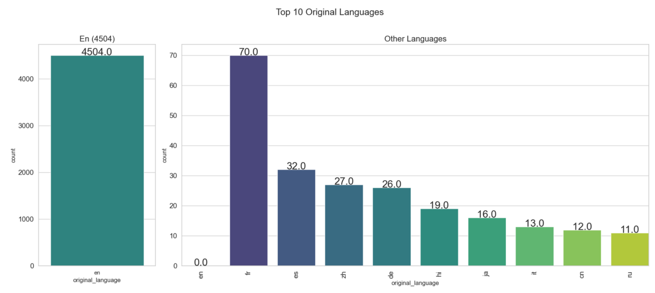
- Dissimilarity Analysis.

In the case that the preference of the audience is known, and s/he dislikes the “Harry Potter and the Half Blood Prince”. A minor modification of the ‘get_similar_movie’ function, that is adding the reverse sorting of the similar scores. Then simply return the list of movies with the least similarity score.

ANALYSIS OF THE RESULTS

The TF-IDF vectorization process is applied to the textual descriptions of the movies and not to their titles. This process converts the textual information into numerical vectors, capturing the importance of words in the descriptions. The TF-IDF outperformed the TF therefore we decided to take a closer at the evaluation of models in combination with TF-IDF.

Top 10 Languages chart

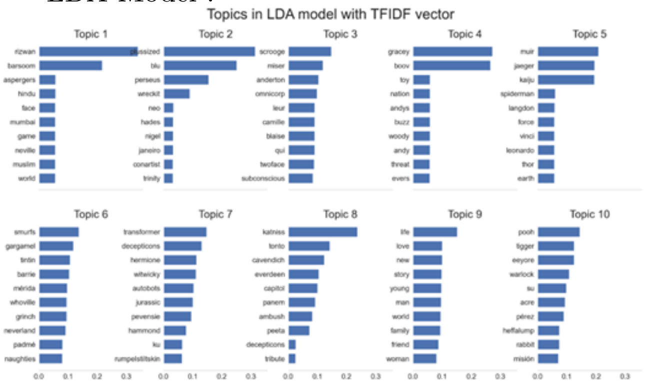


The bar chart shows the top 10 languages, the highest being English (4,505 movies). The next 9 highest languages make up 298 movies, mainly in French, Spanish and simplified Chinese.

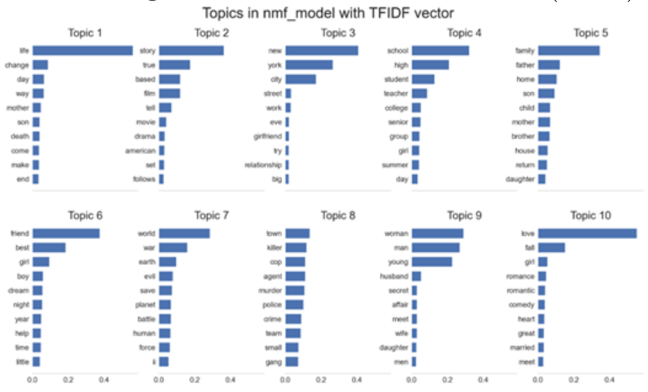
Topic modeling

Three models; LDA, TruncatedSVD, and NMF, were created using both TF and TF-IDF vectors. However, perplexity, diversity and stability scores indicate that the models performed better with TF-IDF. Therefore we only discuss the results from LDA, and NMF.

LDA Model :



Non-Negative Matrix Factorization(NMF):

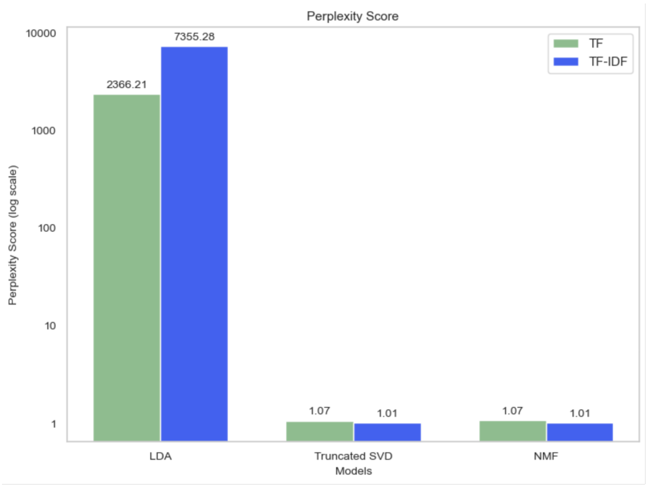


Based on visual inspection of the graphs above; it is evident that the two models picked out 10 very different vocabulary words to create 10 topics. NMF is capturing the level of importance of the words better than LDA, based on the height difference of the bars. Stronger/more influential words have longer bars.

To analyze the results, we have used perplexity, topic stability, and topic diversity as evaluation metrics.

Perplexity Score

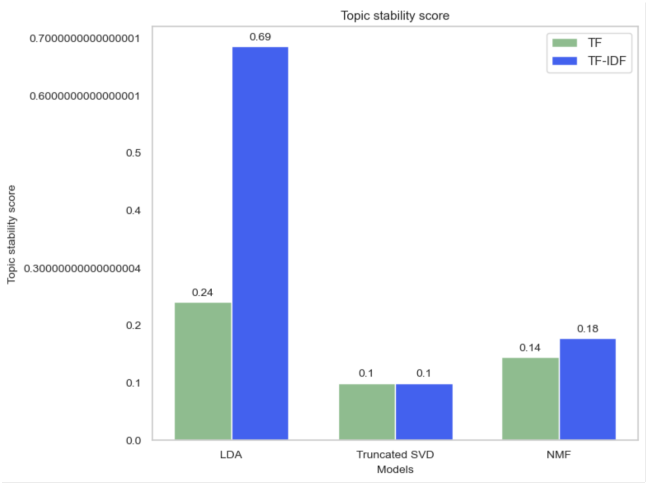
Perplexity score measures the performance of the topic model. It measures how well the model predicts a held-out test set. Lower perplexity values indicate better performance; lower scores equals better prediction accuracy.



Based on the perplexity score, Truncated SVD and NMF outperform the LDA model. However, it's important to note that perplexity may not always correlate perfectly with the human interpretability of topics.

Topic Stability Score

Topic stability assesses how consistent topics – its robustness are across different runs or subsets of the data. A higher score indicates consistency. This can be measured using techniques like Jaccard similarity or topic overlap.

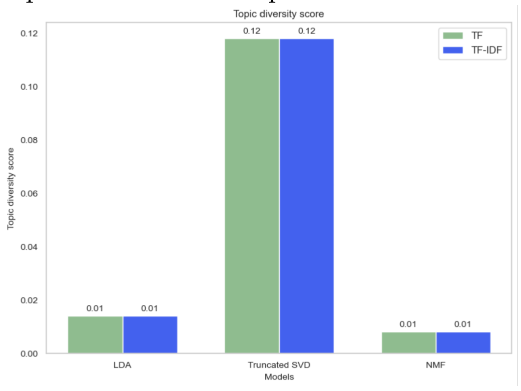


LDA shows a highest stability score in TF-IDF at ∼0.89 compared to when using TF score at ∼0.19, the topics produced are more consistently the same across different runs, given the model is reliable and keeps giving the same results.

Truncated SVD and NMF on the other hand, show low stability scores, both with TF and TF-IDF, questioning the consistency of the topics they generate for practical applications. Future work should aim for model stability of the model and reliable topics.

Topic Diversity

Assess whether topics cover a diverse range of themes or if they are too similar. A good topic model should capture diverse aspects of the dataset. A high score means there is little overlap between the top words of different topics.



LDA scored low on diversity for both TF and TF-IDF, implying overlapping topics. Truncated SVD scored higher, suggesting a wider variety of topics. NMF with TF-IDF falls between LDA and SVD, indicating moderate topic uniqueness.

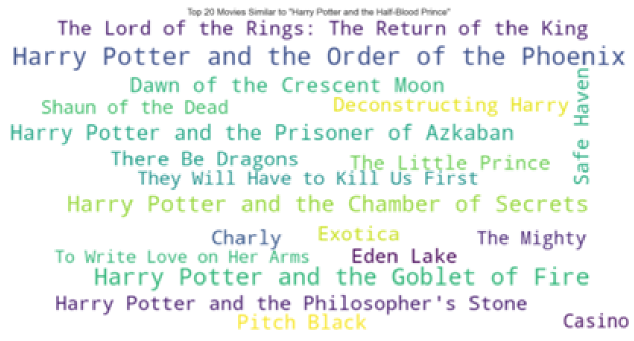
Overall, each model and vectorization technique has its diverse strengths and weaknesses.LDA is more consistent with TF-IDF showing the same topic each time making it less predictable regarding new topics. NMF offers a balance between topic diversity and stability, especially with TF-IDF, making it a potentially good middle ground in between LDA and SVD. NMF is capturing the level of importance of the words better than LDA so if we only had to choose one, NMF would be our choice of model.

It's essential to use a combination of these evaluation methods, as no single metric can fully capture the quality of generated topics and have in consideration that the objectives of the research and the properties of the data might influence the selection of the evaluation metrics.

Searching for similar and dissimilar movies. To identify movies

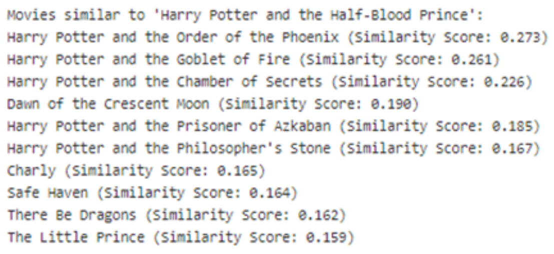
Identifying movies that are similar to "Harry Potter and the Half-Blood Prince" using cosine similarity scores.

These are the movies (presented in WordCloud plot) that are most similar to Harry Potter.


Top 20 Movies Similar to "Harry Potter and the Half-Blood Prince"
The Lord of the Rings: The Return of the King
Harry Potter and the Order of the Phoenix
Dawn of the Crescent Moon
Shaun of the Dead
Deconstructing Harry
Safe Haven
Harry Potter and the Prisoner of Azkaban
There Be Dragons
The Little Prince
They Will Have to Kill Us First
Harry Potter and the Chamber of Secrets
Charly
Exotica
The Mighty
To Write Love on Her Arms
Eden Lake
Harry Potter and the Goblet of Fire
Harry Potter and the Philosopher's Stone
Pitch Black
Casino

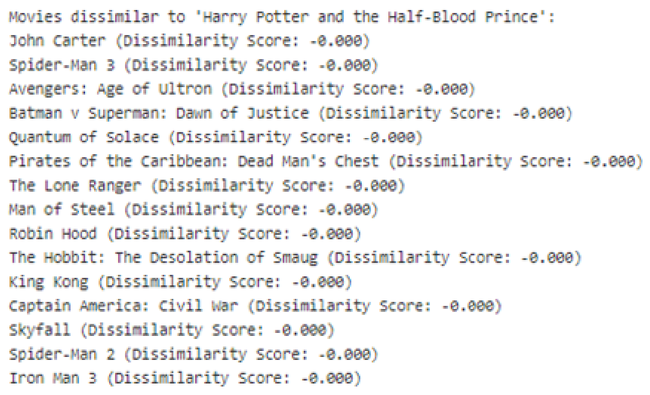
Note that the top 3 Harry Potter movies dominate the space.

This is the output from the cosine Similarity, all harry potter movies receive high rank on the list.


```
Movies similar to 'Harry Potter and the Half-Blood Prince':
Harry Potter and the Order of the Phoenix (Similarity Score: 0.273)
Harry Potter and the Goblet of Fire (Similarity Score: 0.261)
Harry Potter and the Chamber of Secrets (Similarity Score: 0.226)
Dawn of the Crescent Moon (Similarity Score: 0.190)
Harry Potter and the Prisoner of Azkaban (Similarity Score: 0.185)
Harry Potter and the Philosopher's Stone (Similarity Score: 0.167)
Charly (Similarity Score: 0.165)
Safe Haven (Similarity Score: 0.164)
There Be Dragons (Similarity Score: 0.162)
The Little Prince (Similarity Score: 0.159)
```

Dissimilarity Scores (Negative cosine similarity score)

Arranging the score in descending order gives us the list of movies that are NOT similar to "Harry Potter." The output is shown below:



```
Movies dissimilar to 'Harry Potter and the Half-Blood Prince':
John Carter (Dissimilarity Score: -0.000)
Spider-Man 3 (Dissimilarity Score: -0.000)
Avengers: Age of Ultron (Dissimilarity Score: -0.000)
Batman v Superman: Dawn of Justice (Dissimilarity Score: -0.000)
Quantum of Solace (Dissimilarity Score: -0.000)
Pirates of the Caribbean: Dead Man's Chest (Dissimilarity Score: -0.000)
The Lone Ranger (Dissimilarity Score: -0.000)
Man of Steel (Dissimilarity Score: -0.000)
Robin Hood (Dissimilarity Score: -0.000)
The Hobbit: The Desolation of Smaug (Dissimilarity Score: -0.000)
King Kong (Dissimilarity Score: -0.000)
Captain America: Civil War (Dissimilarity Score: -0.000)
Skyfall (Dissimilarity Score: -0.000)
Spider-Man 2 (Dissimilarity Score: -0.000)
Iron Man 3 (Dissimilarity Score: -0.000)
```

Above shows the output of a list of movies dissimilar to "Harry Potter and The Half-Blood Prince", sorting in ascending order to find the most dissimilar movie.

CONCLUSION

Using TF-IDF as an input in all three selected models(LDA, Truncated SVD and NMF) outperforms the same models using TF. When comparing the models using perplexity, diversity and stability scores; the NMF model is the best overall. But this depends on the use case of the model. The use of cosine similarity score to identify movies similar to Harry Potter and the half blood

Prince identified, among other movies, the other Harry Potter movies. Whereas, a list of the dissimilar movies includes: John Carter, Spiderman-3 and Avengers and others.

References

[1] In: *Academic.oup.com* (). URL: https://academic.oup.com/aje/article/172/9/1070/148540.

[2] *1.10. Decision Trees*. No date. URL: https://scikit-learn.org/stable/modules/tree.html.

[3] *1.9. Naive Bayes*. No date. URL: https://scikit-learn.org/stable/modules/naive_bayes.html.

[4] *1.9. Naive Bayes*. No date a. URL: https://scikit-learn.org/stable/modules/naive_bayes.html.

[5] *2.1. Gaussian Mixture Models*. No date. URL: https://scikit-learn.org/stable/modules/mixture.html.

[6] *6.2. Feature Extraction*. No date. URL: https://scikit-learn.org/stable/modules/feature_extraction.html.

[7] Ajisamudra. *NLP - Count, TF-IDF, Hashing Vectorizer*. Accessed on: Accessed: 2023-11-28. Publication date not specified. URL: https://www.kaggle.com/code/ajisamudra/nlp-count-tf-idf-hashing-vectorizer.

[8] Fatih Karabiber Ph.D. in Computer Engineering et al. *Hierarchical Clustering*. No date. URL: https://www.learndatasci.com/glossary/hierarchical-clustering/#:~:text=Hierarchical%20clustering%20is%20a%20popula.

[9] W. baeldung. *Stratified Sampling in Machine Learning*. 2023. URL: https://www.baeldung.com/cs/ml-stratified-sampling#:~:text=Stratified%20Sampling%20is%20a%20sampling,Random%20Sampling%20within%20each%20subgroup.

[10] Lorenzo Beretta and Alessandro Santaniello. *Nearest neighbor imputation algorithms: A critical evaluation - BMC medical informatics and decision making*. 2016. URL: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-016-0318-z.

[11] H. Byeon and B. B. “Is the Random Forest Algorithm Suitable for Predicting Parkinson’s Disease with Mild Cognitive Impairment out of Parkinson’s Disease with Normal Cognition?” In: *International Journal of Environmental Research and Public Health* (2020). Accessed: 02 December 2023. URL: https://pubmed.ncbi.nlm.nih.gov/32290134/.

[12] *Classification: Accuracy.* No date. URL: https://developers.google.com/machine-learning/crash-course/classification/accuracy.

[13] *Classification: Precision and Recall.* No date. URL: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall%7D.

[14] Various contributors. *In Natural Language Processing (NLP), how do you make an efficient dimension reduction?* Accessed on: Date accessed not specified. Publication date not specified. URL: https://stats.stackexchange.com/questions/124908/in-natural-language-processing-nlp-how-do-you-make-an-efficient-dimension-red.

[15] C. Khanna. *Text pre-processing: Stop words removal using different libraries.* 2021. URL: https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a.

[16] W. Koehrsen. *Recurrent Neural Networks by Example in Python.* 2018. URL: https://towardsdatascience.com/recurrent-neural-networks-by-example-in-python-ffd204f99470.

[17] *Latent Dirichlet Allocation.* 2021. URL: https://www.geeksforgeeks.org/latent-dirichlet-allocation/.

[18] *Mean Squared Error.* No date. URL: https://www.britannica.com/science/mean-squared-error.

[19] Andreas C. Müller and Sarah Guido. *Introduction to Machine Learning with Python.* ISBN-13: 978-1449369415. O’Reilly Media, Inc., 2016.

[20] *Non-negative Matrix Factorization.* 2023. URL: https://www.geeksforgeeks.org/non-negative-matrix-factorization/.

[21] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[22] V.K.A. et al. Researcher. *All You Need to Know About Support Vector Machines.* 2022. URL: https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector-machine/#:~:text=A%20support%20vector%20machine%20(SVM)%20is%20a%20m.

[23] *Self Organizing Maps - Kohonen Maps.* 2023. URL: https://www.geeksforgeeks.org/self-organising-maps-kohonen-maps/.

[24] *sklearn.cluster.DBSCAN.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html.

[25] *sklearn.cluster.KMeans.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html.

[26] *sklearn.cluster.MeanShift.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html#:~:text=Mean%20shift%20clustering%20aims%20to.

[27] *sklearn.cluster.OPTICS.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html.

[28] *sklearn.ensemble.RandomForestClassifier.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[29] *sklearn.impute.KNNImputer.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html.

[30] *sklearn.metrics.mean_absolute_error.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html.

[31] *sklearn.metrics.r2$_s$core.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html%7D.

[32] *sklearn.preprocessing.MinMaxScaler.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

[33] *sklearn.preprocessing.RobustScaler.* No date. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html.

[34] Author not specified. *Two Minutes NLP: Learn TF-IDF with Easy Examples.* Accessed on: Accessed: 2023-11-28. Publication date not specified. URL: https://medium.com/nlplanet/two-minutes-nlp-learn-tf-idf-with-easy-examples-7c15957b4cb3.

[35] Author not specified. *Understanding TF-IDF (Term Frequency-Inverse Document Frequency).* Accessed on: Accessed: 2023-11-28. Publication date not specified. URL: https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/.

[36] *Stemming vs. Lemmatization.* 2023. URL: https://databasecamp.de/en/data/stemming-lemmatization.

[37] *What Are Convolutional Neural Networks?* No date. URL: https://www.ibm.com/topics/convolutional-neural-networks.

[38] *What Is a Transformer Model?* No date. URL: https://www.ibm.com/topics/transformer-model.