**MSc in Data Analytics**

Author: Muhammad Rizwan

e-mail: 2024665@student.cct.ie

Student ID: 2024665

**Abstract**

*In the tourism sector, recommendation systems play a critical role in helping users make tailored decisions. This study introduces a recommender system that uses a dataset of ratings and reviews to offer tailored hotel recommendations. The work tackles two prevalent issues in tourism recommendation systems: the cold-start problem and data sparsity. To increase recommendation diversity and accuracy, the system makes use of a number of machine learning methods, such as Random Forest (RF), K-Nearest Neighbors (KNN), Naive Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR). When the efficacy of these algorithms is assessed, Logistic Regression performs best and has the highest accuracy. The outcomes demonstrate how well the system can produce varied and accurate recommendations, which enhances user satisfaction and the hotel choosing process.*

**Introduction**

Personalized recommendations have emerged as a key component in the tourist sector for improving the trip experience. Travelers frequently rely on recommendation systems to help them make well-informed decisions that suit their individual needs and interests because there are so many hotels, sights, and activities to choose from. To assist consumers in navigating this plethora of possibilities and locating appropriate lodging that improves their travel enjoyment, sophisticated recommender systems must be developed.

A sophisticated recommender system created especially for hotel recommendations is presented in this paper. The system tackles two of the most difficult problems in recommendation technology—the cold-start problem and data sparsity—by utilizing a dataset of user ratings and reviews. These problems are prevalent in the travel industry, as consumers might not have had any prior experience with particular hotels or locations, which makes it challenging to provide recommendations that are pertinent. A variety of machine learning methods, such as Random Forest (RF), K-Nearest Neighbors (KNN), Naive Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR), are used in this work to address these problems.

After extensive testing, the method with the best recommendation accuracy was found to be Logistic Regression. The results of the study demonstrate the efficacy of this strategy since it encourages diversity in the solutions that are recommended in addition to providing correct advice. By ensuring that visitors have a well-balanced range of both new and familiar options, this diversity improves user happiness and streamlines the hotel decision process.

**Materials and Methods**

This chapter is dedicated to presenting a comprehensive methodology aimed at inspiring the travel experience by leveraging the huge potential of data analysis and recommendation systems. At its core, our proposed methodology comprises four key modules: Data Extraction, Preprocessing, Development, and the Recommender Engine. In order to conduct this study, we gather a dataset crawled from TripAdvisor, a renowned platform for rating and reviewing tourist attractions around the world. Our objective is to harness this data to enhance the travel planning process and ultimately provide travelers with more personalized and enjoyable experiences.

* **Proposed Methodology Architecture**

The proposed methodology consists of the following modules

* Data Preprocessing.
* Recommendation (Review, Ratings)
* Anaconda (Jupyter notebook)
* **Data Collection**

In the proposed methodology, dataset is gathered from the TripAdvisor website using a Python script. This dataset contains 8,906 ratings of 88 attractions by the users including User\_IDs. The ratings range from 1 to 5 on a scale of 1 to 5. Using this vast amount of data, algorithms can create tailored recommendations based on user preferences and tastes for places to visit. Since Pakistan has a worldwide reputation as a tourist destination and has a wide variety of tastes and preferences, the dataset was obtained specifically from TripAdvisor. This choice ensures that the system's suggestions are relevant and applicable to Pakistan's tourism industry.

* **Recommender Engine**

In the proposed methodology, dataset that i used for this report/work was obtained from a research project conducted at the University of Gujrat in Pakistan. The researcher created this dataset by performing web scraping on various tourism and hotel websites to collect hotel reviews and ratings. I also used this dataset for my analysis. She gathered the dataset from the TripAdvisor website using a Python script. This dataset contains 8,906 ratings of 88 attractions by the users including User\_IDs. The ratings range from 1 to 5 on a scale of 1 to 5. Using this vast amount of data, algorithms can create tailored recommendations based on user preferences and tastes for places to visit. Since Pakistan has a worldwide reputation as a tourist destination and has a wide variety of tastes and preferences, the dataset was obtained specifically from TripAdvisor. This choice ensures that the system's suggestions are relevant and applicable to Pakistan's tourism industry. This study aims to create a better way for travel recommendation systems to make users happier. For achieving this different ML algorithm are used and recommend the hotels on the basis of the customer review and the ratings. By using these different methods and mixing them in special ways, the study wants to make travel planning more personal and enjoyable for each user, so they can have a more special and satisfying experience when planning their trips.

* **Tourism Recommendation**

A carefully chosen dataset gathered from TripAdvisor is at the heart of this tourism recommendation system. This dataset contains a wealth of useful information, such as information about 88 distinctive attractions and a large collection of 9,640 user reviews. This rich and diverse dataset forms a basis over the recommendation engine, that generates personalized and useful recommendations for travelers. The recommendation system makes use of TripAdvisor's extensive content to learn about what users like and dislike. It looks at reviews left by other travelers to learn about tourist attractions, memorable encounters, and favorite parts of various locations. The system acknowledges patterns and trends in data using smart techniques and uses this data to create suggestions that are tailored to each user's interests. Every suggestion is thus based on practical knowledge and the pooled knowledge of fellow travelers.

Travel recommendation approach is designed using a knowledge-based filtering method that leverages hotel reviews and ratings from our dataset, developed through Python code and machine learning algorithms. This approach gathers user preferences during registration, allowing us to suggest destinations that closely match their interests, even for new users without historical data. By focusing on a knowledge-based model, we ensure personalized recommendations based solely on the user’s stated preferences, providing a customized and user-centric travel planning experience. This method enables us to deliver relevant recommendations that enhance each user's travel planning journey.

* **Problem Statement**

Recommendation systems have found successful implementation in various countries around the world, including popular destinations like Paris, Morocco, Tehran, Isfahan, Shiraz, Indonesia, Jakarta, Tainan, Spain, Tarragona, Milan, America, Europe, and Oman. Pakistan still lacks a tailored solution for the tourism sector. Two significant challenges commonly faced in tourism recommendations are the choosing a better place while traveling. The problem arises when there is a scarcity of data available for new users or items. This information gap is a significant impediment to providing precise and personalized recommendations. Similarly, the issue of sparsity stems from a scarcity of relevant data, resulting in a lack of comprehensive information needed for making meaningful recommendations.

* **Research Objectives**

The following are the precise goals of this research project:

* Establish a travel recommendation system in Pakistan on the basis of ratings and reviews on different places.
* Recommend places according to user's taste and rating.
* **Data Preprocessing**

The proposed methodology preprocesses the data collected from TripAdvisor to guarantee the dataset's reliability and accuracy. To accomplish this, a Python script in a Jupyter notebook surroundings is used to clean the data. During the cleaning process, irrelevant and raw values are removed or filtered out, enhancing the overall quality and usability of the dataset. By utilizing a Python script in the data cleaning phase, the obtained cleaned data becomes suitable for further analysis and development, ultimately leading to clearer and more accurate results in the recommendation system.

Based on our previous discussions, we have used some of the visualizations and techniques mentioned above, but not all. Here's a recap of what we have already covered and what could be added to your analysis:

* **Distribution of Ratings**:
* We created **histograms** to visualize the distribution of ratings.
* sns.histplot(df['Ratings'], kde=True).

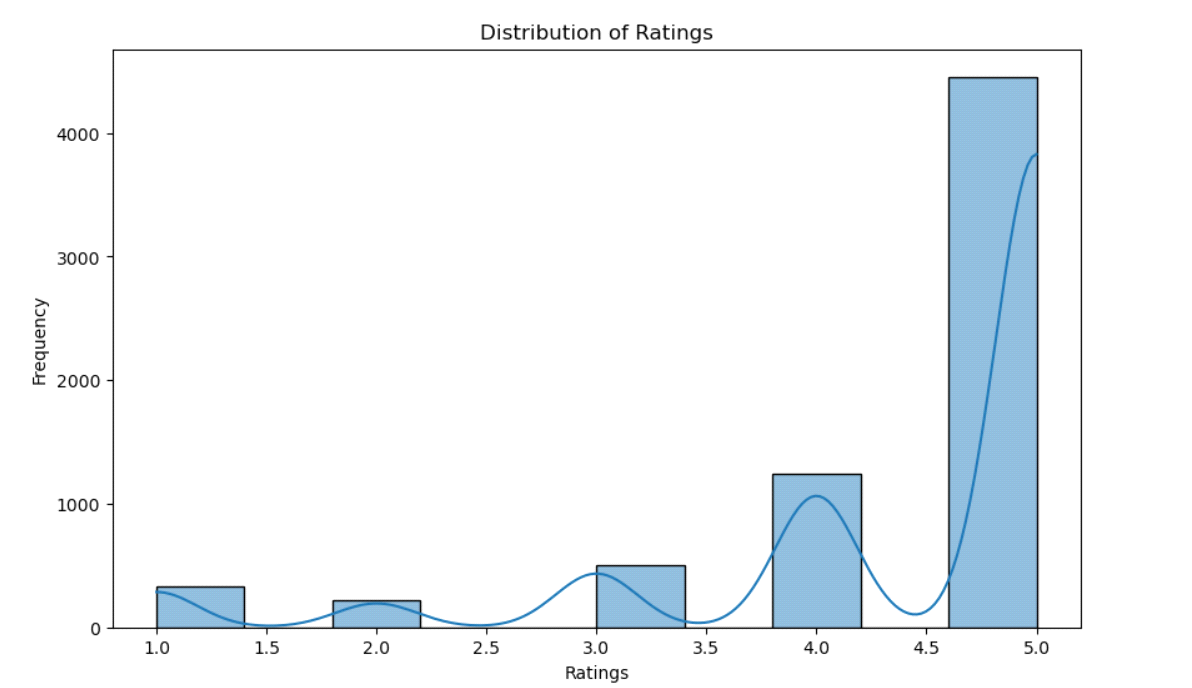


Figure 1: Distribution of Rating

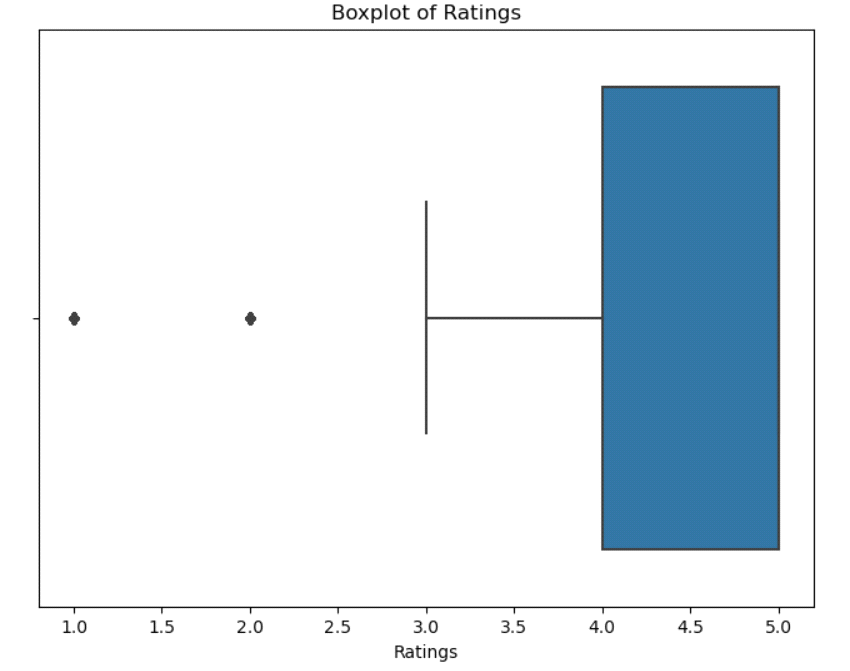
* **Boxplot for Ratings**:
* We used **boxplots** to identify the spread and outliers in the ratings.
* sns.boxplot(x=df['Ratings']).

Figure 2:Boxplot Figure

* **Correlation Heatmap**:
* A **correlation heatmap** was created to understand relationships between numerical features.
* sns.heatmap(df.corr(), annot=True).

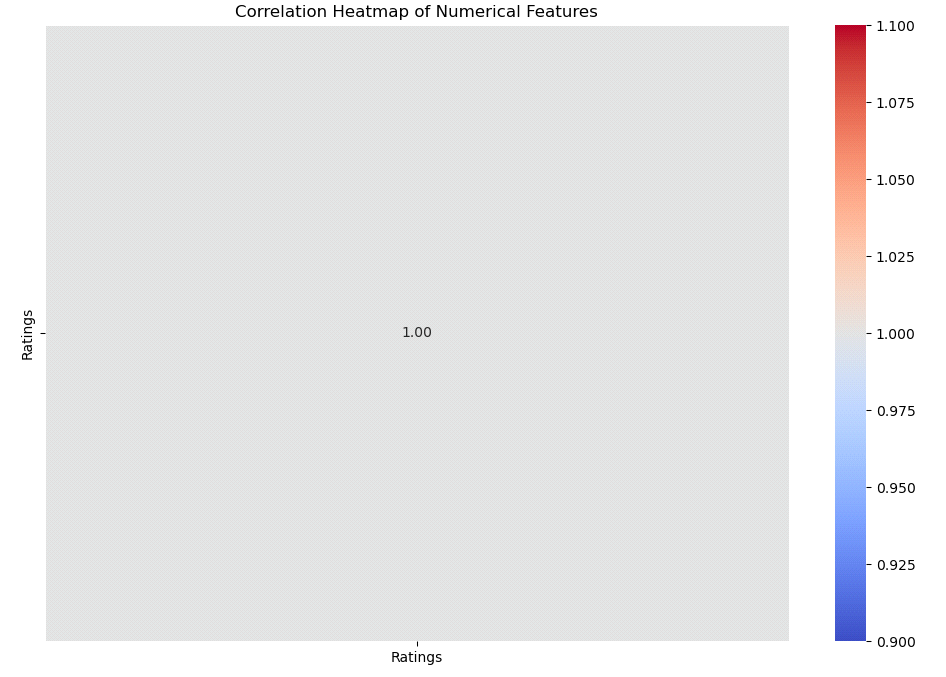


Figure 3: Correlation Heatmap

* **Barplot for Rating Categories**:
* We visualized the number of hotels in different rating categories (like Excellent, Good, Average, etc.).
* sns.barplot(x=rating\_categories, y=rating\_counts).

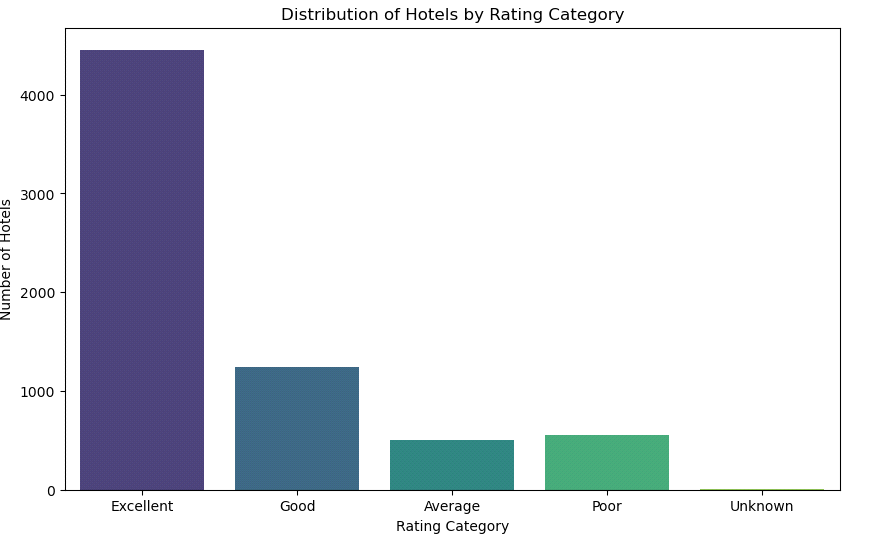


Figure 4:Barplot for Rating Categories

* **Countplot for Sentiment**:

We might have also used a **countplot** to show sentiment distribution (e.g., positive, negative, neutral) if you created sentiment labels.

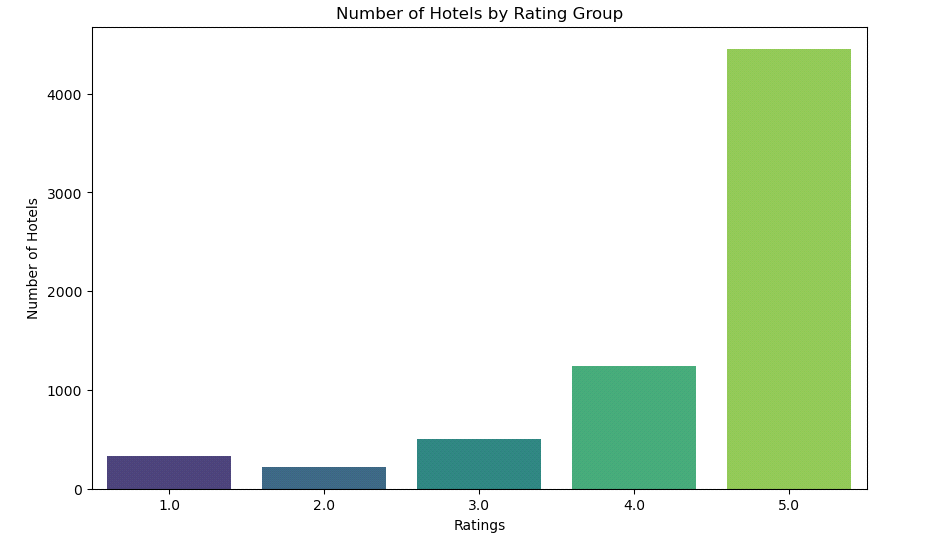


Figure 5: Countplot for Sentiment

* **Supervised Machine Learning for the Dataset**

Supervised learning algorithms learn from labeled data to make predictions or classifications. The model is trained using a known output (the label) and a set of input features (predictors), which allows the model to learn the relationship between the input and output variables. Common examples of supervised learning algorithms include Linear Regression, Logistic Regression, Decision Trees, Support Vector Machines (SVM), and Random Forests.

In the case of our tourists’ hotel recommendation dataset where reviews and ratings are provided by users, supervised learning is appropriate for several reasons:

* **Labeled Data**: The dataset contains labeled data. Each hotel review is accompanied by a numerical rating like label, such as 1-5 stars, which allows us to treat this as a supervised learning problem. We can predict the rating based on features like hotel name, review text, and amenities.
* **Prediction Task**: The goal is to predict the rating or categorize the review sentiment, which is a classical supervised learning task. The model learns to map the input features to the corresponding output label.e.g., predicting if a review is positive, neutral, or negative).
* **Clear Objective**: Supervised learning algorithms are ideal when there is a clear, well-defined target the rating. In a hotel recommendation system, you can predict ratings based on customer reviews and other hotel features. This allows for easy performance evaluation through metrics like accuracy, precision, recall, and F1-score.
* **Regression vs Classification**: Depending on your target variable, you can either perform regression predicting continuous ratings like 1-5 stars or classification categorizing reviews as "Excellent," "Good," "Average," etc. This is another advantage of supervised learning since it can accommodate both types of tasks.
* **Hotel Rating Prediction**: Given a set of reviews and hotel characteristics, a supervised learningmodel like Random Forest or SupportVector Machine can be trained to predict the rating that a user might give to a hotel. By training on labeled reviews and corresponding ratings the model learns the underlying patterns that influence ratings.

The requirement of using hyper-parameter tuning is suitable for our dataset, particularly when we're working with machine learning models like classification or regression models. Hyper-parameter tuning helps optimize your machine learning model's performance by systematically exploring different combinations of hyperparameters.

For any machine learning model, the performance can significantly vary based on the values of hyperparameters. Hyperparameters are external configurations to the model and finding the optimal set of hyperparameters can improve the model's ability to generalize well to unseen data.

* **Random Forest**: Tuning hyper-parameters like n estimators (number of trees), max-depth, min-samples-split, and max-features can impact performance.
* **Support Vector Machine (SVM)**: Tuning hyper-parameters like C (regularization) and kernel linear, polynomial, RBF can optimize performance.

There are several approaches for hyperparameter tuning, and while GridSearchCV and RandomizedSearchCV are the most commonly used techniques, other methods also exist.

* **GridSearchCV:** Exhaustively searches through a manually specified hyperparameter grid, evaluating all possible combinations of hyperparameters.
* It's computationally expensive because it tries every combination of hyperparameters.
* **RandomizedSearchCV:** More efficient when you have a large search space. It randomly samples from a specified distribution of hyperparameters rather than trying all possible combinations.
* This approach can be more efficient for finding optimal hyperparameters with less computational cost.

**Results and Discussion**

Here, a detail discussion about the results that got after performing all the needed steps. Dataset cleaning, check for missing values and removing the missing values and also different preprocessing techniques and also done the visualization of the dataset before and after the preprocessing steps. After preprocessing different ML Classifiers are performed on the dataset. For the ML classifier firstly, we spilt the dataset into the train and test than apply ML classifiers.

The ML classifiers that we apply included SVM, RF, KNN, LR and NB. in the results the LR and SVM are likely have equally accuracy 74%. But LR out performs the SVM because the accuracy of SVM is 0.07400 but the LR’s accuracy is 0.07417. so, in this case we can say that the LR outperforms the all other Ml classifiers.

**Cross Validation with Different ML Algorithms:**

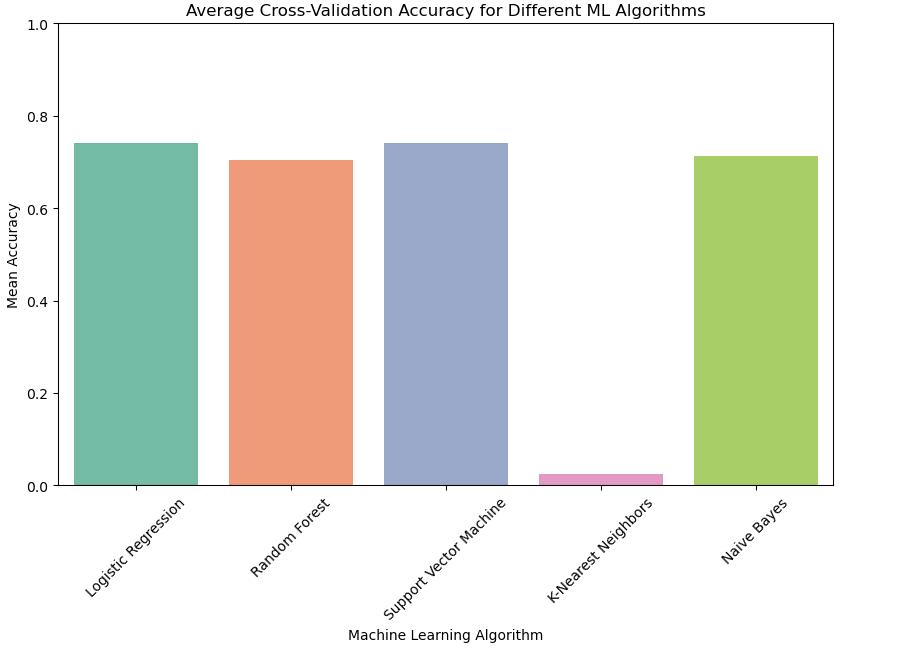


Figure 6: Cross Validation with different ML Algorithm

fig.0.1The chart (fig.0.1) shows that Logistic Regression, SVM, and Naive Bayes have high cross-validation accuracy, with SVM performing best, while K-Nearest Neighbors has a very low accuracy. This indicates KNN may not be suitable for this dataset without further tuning.

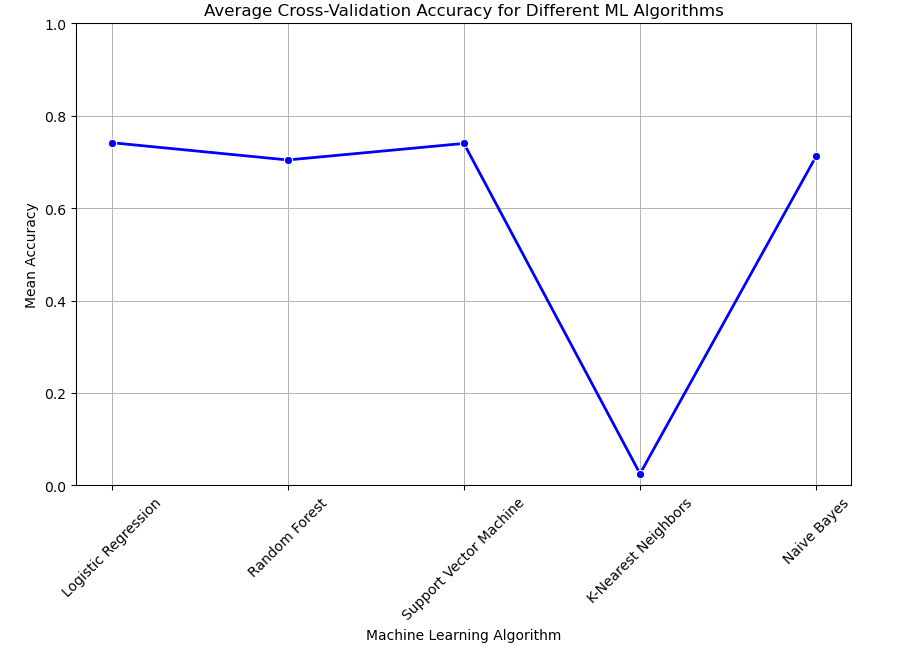


Figure 7: Average Cross-Validation Accuracy

The plot (fig.0.2) shows average cross-validation accuracy for different ML algorithms, with Logistic Regression, Random Forest, and SVM performing well, while K-Nearest Neighbors performs poorly. Naive Bayes has moderate accuracy, lower than the top performers.

**Results**

After apply all Machine learning models the result shows that the Logistic Regression and Support Vector Machine (SVM) achieved the highest mean accuracy (around 0.74), making them the top-performing models in this set. Naive Bayes followed closely with an accuracy of 0.71, while Random Forest had a slightly lower accuracy at 0.70. K-Nearest Neighbors, however, performed significantly worse, with a mean accuracy of only 0.03, indicating it may not be suitable for this dataset.

**Table of Cross Validation:**

|  |  |  |
| --- | --- | --- |
|  | **Classifiers** | **Accuracy** |
| 1 | Logistic Regression | 0.741723 |
| 2 | Random Forest | 0.704317 |
| 3 | Support Vector Machine | 0.740096 |
| 4 | K-Nearest Neighbors | 0.025133 |
| 5 | Naive Bayes | 0.712450 |

Table 1:Table of Cross Validation

**REFERENCES**

Abbasi-Moud, Z., Vahdat-Nejad, H., & Sadri, J. (2021). Tourism recommendation system based on semantic clustering and sentiment analysis. *Expert Systems with Applications*, *167*, 114324. <https://doi.org/10.1016/j.eswa.2020.114324>

Alrasheed, H., Alzeer, A., Alhowimel, A., shameri, N., & Althyabi, A. (2020). A Multi-Level Tourism Destination Recommender System. *Procedia Computer Science*, *170*, 333–340. <https://doi.org/10.1016/j.procs.2020.03.047>

An, H., & Moon, N. (2022). Design of recommendation system for tourist spot using sentiment analysis based on CNN-LSTM. *Journal of Ambient Intelligence and Humanized Computing*, *13*(3), 1653– 1663. <https://doi.org/10.1007/s12652-019-01521-w>

Badouch, M., & Boutaounte, M. (2023). Personalized Travel Recommendation Systems: A Study of Machine Learning Approaches in Tourism. *Journal of Artificial Intelligence, Machine Learning and Neural Network*, *33*, 35–45. <https://doi.org/10.55529/jaimlnn.33.35.45>

Bahramian, Z., Ali Abbaspour, R., & Claramunt, C. (2017). A Cold Start Context-Aware Recommender System for Tour Planning Using Artificial Neural Network and Case Based Reasoning. *Mobile Information Systems*, *2017*, 1–18. <https://doi.org/10.1155/2017/9364903>

Choi, I. Y., Ryu, Y. U., & Kim, J. K. (2021). A recommender system based on personal constraints for smart tourism city. *Asia Pacific Journal of Tourism Research*, *26*(4), 440–453. <https://doi.org/10.1080/10941665.2019.1592765>

Chouiref, Z., & Hayi, M. Y. (2022). Toward Preference and Context-Aware Hybrid Tourist Recommender System Based on Machine Learning Techniques. *Revue d’Intelligence Artificielle*, *36*(2), 195–208. <https://doi.org/10.18280/ria.360203>

Fang, H., Chen, C., Long, Y., Xu, G., & Xiao, Y. (2022). DTCRSKG: A Deep Travel Conversational Recommender System Incorporating Knowledge Graph. *Mathematics*, *10*(9), 1402. <https://doi.org/10.3390/math10091402>

Fararni, K. A., Nafis, F., Aghoutane, B., Yahyaouy, A., Riffi, J., & Sabri, A. (2021). Hybrid recommender system for tourism based on big data and AI: A conceptual framework. *Big Data Mining and Analytics*, *4*(1), 47–55. <https://doi.org/10.26599/BDMA.2020.9020015>

Gamidullaeva, L., Finogeev, A., Kataev, M., & Bulysheva, L. (2023). A Design Concept for a Tourism Recommender System for Regional Development. *Algorithms*, *16*(1), 58. <https://doi.org/10.3390/a16010058>

Hassannia, R., Vatankhah Barenji, A., Li, Z., & Alipour, H. (2019). Web-Based Recommendation System for Smart Tourism: Multiagent Technology. *Sustainability*, *11*(2), 323. <https://doi.org/10.3390/su11020323>

Herce-Zelaya, J., Porcel, C., Tejeda-Lorente, Á., Bernabé-Moreno, J., & Herrera-Viedma, E. (2022). Introducing CSP Dataset: A Dataset Optimized for the Study of the Cold Start Problem in Recommender Systems. *Information*, *14*(1), 19. <https://doi.org/10.3390/info14010019>

Hong, M., & Jung, J. J. (2021). Multi-criteria tensor model for tourism recommender systems. *Expert*

*Systems with Applications*, *170*, 114537. <https://doi.org/10.1016/j.eswa.2020.114537>

Julashokri, M., Leonardi, S., & Seabra, P. (2022). Tourism Integrated Recommender System: Setubal

Peninsula Case Study: *Proceedings of the 24th International Conference on Enterprise Information Systems*, 623–627. <https://doi.org/10.5220/0011102100003179>