Machine Learning, Data analysis and visualization Task.

**Part 1.0**

1. Recommender systems

These are algorithms or models that provide recommendations about products or services that are of interest to a consumer. They use existing data to find the most relevant and personalized content for users.

Areas of applications for recommendation systems include google search recommendations, online shopping recommendations, movie recommendations, and restaurants, among others.

1. Challenges of recommendation systems.

*Data sparsity* – when the users do not provide ratings for a product or service.

Changing user preferences – users’ tastes may shift.

*New or changing data* – as characteristic of products and services, they are highly dynamic over time to keep up with current trends.

*Frequent updating* – is required to capture new trends that are relevant to users.

*Complexity trade-off* - of incorporating more item details and more user information, while maintaining a fast performing and accurate recommender system.

1. Approaches to Solve the recommender systems task

The approaches used for implementing recommender systems are content-based approaches and collaborative filtering approaches. These are described in the table below.

|  |  |
| --- | --- |
| Content-based methods | Collaborative filtering methods |
| Also known as an item-based recommendation because it uses item features. | Uses implicit similarities between items and users simultaneously to provide recommendations. |
| Uses explicit features about an item and a user. | Uses implicit features about items and uses. |
| Uses ratings, likes, or dislikes to make recommendations. | Uses user behavior such as views, clicks, searches, and purchases. |
| Waits on user feedback or reactions to an item, before providing recommendations. | Generates feature or vector embeddings to make recommendations to the user. |
| Complex and high computational cost to incorporate all the relevant items features to make recommendations. | It does not provide the most personalized recommendations since it is based on a consensus system, which may cause only saturation of the most popular items to uninterested users. |

**Part 2.0**

I chose the open-source House Prices dataset from the Kaggle website. Imported it to Jupiter notebook using pandas read\_csv library and stored the file in a data frame called *dataset*, then started to explore my data.

1. **Dataset description**

Dataset.shape – shows the structure of the data. It has 5000 records(data points) and 7 columns(features). The features are:

Avg. Area Income – this is the average income in an area

Avg Area House Age – this is the average age of the house

Avg. Area Number of Rooms – the average number of rooms in a house

Avg. Area Number of Bedrooms – the average number of bedrooms in a house

Area population – the population of people living in the area

Price – the house price.

Address – the address of a house

A screenshot of a computer

Description automatically generated

1. **Data Preprocessing includes:**

Renaming the columns - The name of the columns needs to be renamed to make sense.

Check for Na values and solved them.

Also, we note, that all the data points in our dataset are of type int.

A screenshot of a computer

Description automatically generated

From the dataset.isna().sum(), there are two NA values in the No\_of \_rooms column.

From the human judgment, it does not make sense that a house has zero rooms, so I fill the NA with the mean of the field.

A screenshot of a computer

Description automatically generated

Now, the dataset doesn’t contain NA values.

1. **Feature selection** – from visual examination, we can drop the Income, Address, and Area Population columns since they will not provide useful information in price determination in the same region.

I created a new data frame called *cdataset* with the following fields below:

The independent features are HouseAge, No\_of\_rooms and No\_of\_Bedrooms, while the dependent feature will be Price.

A screenshot of a computer

Description automatically generated

1. **Next, use matplotlib and seaborn to visualize the relation between the features.**

Using the matplotlib, the below scatter plot provides a view of the data points distribution. We can draw a linear regression line on the plot.

A screenshot of a computer

Description automatically generated

Using the seaborn library, the correlation of the features can be visualized as below: where,

1.0 represents a higher influence between the coupled variables, and 0.0 represents a small influence. Positive values show direct linear correlation, negative values show inverse correlation. Examination between the features and price:

No\_of\_rooms (0.34) have a small direct linear influence on the house price.

No\_of\_Bedrooms (0.17) have the smallest direct linear influence on the house price.

HouseAge (0.45) has a larger direct linear influence on the house price.

A screenshot of a computer

Description automatically generated

1. **Use Linear regression to predict the house price.**

From the above scatter plot and heatmap, there is an observed linear relationship between our independent fields and dependent field, Price.

Next is to build the model and start making predictions.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Results after testing the model prediction for prices of houses with the following features:

A 20-year-old house, with 4 rooms and 2 bedrooms predicts price to be *3,149,707*.

A 10-year-old house, with 4 rooms and 2 bedrooms predicts price to be *1,527,187*.

A 20-year-old house, with 2 rooms and 2 bedrooms predicts the price to be 2*,915,658*.

A 20-year-old house, with 2 rooms and 4 bedrooms predicts the price to be 3*,157,764*.

**Part 3**

Conclusion

I have learned a lot about Artificial Intelligence and its many applications in solving real-world problems. I was able to understand the differences and applications of Artificial Intelligence, Machine learning, Deep learning, and Data Science.

In summary, Deep learning involves using neural networks to train computers to perform smart tasks and is powerful when dealing with image and video processing. Machine learning involves using statistical models, such as regression, SVM, and decision trees, among others, to train computers. We may also use deep learning together with statistical models.

Artificial intelligence is a larger umbrella that involves using machine learning and deep learning to implement solutions in robotics systems. Data science involves getting insights from available data and can be performed on Excel, Tableau, or Power BI.

Challenges

Leaning to use python so that I can perform data analysis and implement models using machine learning.

Understanding the statistical equations for different algorithms, which are complex.

There are many options to select from to solve a given problem in a dataset, deciding the optimal one is difficult.

Future work

I am interested in researching more on recommendation systems and their wide range of applications. I will research more on these systems and implement recommendations systems. It could also lead to a career in Artificial Intelligence.