**Machine learning assignment**

**2a). briefly describe the working principles of the following algorithms:**

**i. Linear Regression**

Linear Regression is a supervised learning algorithm used for predicting a continuous output variable (dependent variable) based on the relationship between one or more independent variables. It assumes a linear relationship between the input features and the output.

**Working Principle:**

1. **Model Representation**:
   * The algorithm models the relationship as: y=β0+β1x1+β2x2+⋯+βnxn+ϵy = \beta\_0 + \beta\_1x\_1 + \beta\_2x\_2 + \dots + \beta\_nx\_n + \epsilony=β0​+β1​x1​+β2​x2​+⋯+βn​xn​+ϵ where yyy is the predicted output, xix\_ixi​ are input features, βi\beta\_iβi​ are coefficients, and ϵ\epsilonϵ is the error term.
2. **Objective**:
   * Find the coefficients (β\betaβ) that minimize the error between the predicted values and the actual values in the training data.
   * The most common error metric is **Mean Squared Error (MSE)**: MSE=1n∑i=1n(yi−y^i)2\text{MSE} = \frac{1}{n} \sum\_{i=1}^{n}(y\_i - \hat{y}\_i)^2MSE=n1​i=1∑n​(yi​−y^​i​)2
3. **Training Process**:
   * Use optimization techniques (e.g., Gradient Descent or the Normal Equation) to compute the best-fit line or hyperplane by minimizing MSE.
4. **Prediction**:
   * Once trained, the model can predict yyy for unseen xxx values using the learned coefficients.

**Applications:**

* Predicting house prices based on features like area, number of rooms, etc.
* Forecasting stock prices or sales.

**ii. K-Nearest Neighbors**

K-Nearest Neighbors (KNN) is a supervised learning algorithm used for both classification and regression. It is a non**-**parametric method, meaning it makes no assumptions about the underlying data distribution.

**Working Principle:**

1. **Data Representation**:
   * Each data point in the training set is plotted in feature space.
   * The algorithm stores all training data and uses it during prediction.
2. **Classification**:
   * For a new data point, the algorithm identifies the kkk closest data points (neighbors) using a distance metric such as:
     + **Euclidean Distance**: d(p,q)=∑i=1n(pi−qi)2d(p, q) = \sqrt{\sum\_{i=1}^{n} (p\_i - q\_i)^2}d(p,q)=i=1∑n​(pi​−qi​)2​
     + Manhattan or Minkowski distances can also be used.
   * It assigns the class label that is most frequent among these kkk neighbors (majority voting).
3. **Regression**:
   * For regression tasks, KNN predicts the output as the average of the outputs of the kkk nearest neighbors.
4. **Key Parameters**:
   * **Value of kkk**: Determines the number of neighbors considered.
   * **Distance Metric**: Determines how neighbors are measured.
5. **Training Process**:
   * No explicit training phase; all computations are deferred to prediction (lazy learning).
6. **Prediction**:
   * The algorithm searches for neighbors and applies the majority vote (classification) or averaging (regression).

**Applications:**

* Handwritten digit recognition (classification).
* Predicting house prices based on nearby property values (regression).

**b) Machine learning is used in recommendation systems. Explain how collaborative filtering works in building recommendations.**

Collaborative filtering is a machine learning technique employed by recommendation systems to identify the preference of a user based on the historic interactions or preferences of a user group. It assumes that users with similar preferences in the past will share similar preferences in the future also.

**How Collaborative Filtering Works**

There are two major kinds of collaborative filtering: **User-based** and **Item-based.**

**1. User-Based Collaborative Filtering.**

This technique finds similar users and recommends items liked or interacted with by similar users.

The following are theSteps:

1. **Create a User-Item Interaction Matrix:**

A matrix is created in which rows represent users, columns represent items, and entries represent the interaction of a user with an item. Examples include ratings, purchases, clicks, etc.

Example:

Mathematically;

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | User | Item | Item A | Item B | Item C | | User 1 | | 5 | 4 | ? | | User 2 | | 3 | ? | 5 | | User 3 | | 3 | 2 | 4 | |  | | | | | |

Here, "?" denotes unknown preferences.

1. **Similarity Computation:**

It Calculate the similarity between users using metrics such as:

* **Cosine Similarity:**

Sim (u,v) = u.v / ||u|| ||v|| ​

* **Pearson Correlation Coefficient**:

It determines the strength of a linear relationship between the preferences of users.

1. **Recommendation:**

It finds similar users for a target user and recommends items that those users have interacted with but the target user has not.

**2. Item-Based Collaborative Filtering**

It takes into consideration item similarity, rather than user similarity. Items that are similar to what a user liked are recommended.

**The following are the Steps:**

1. **Calculate the Item-Item Similarity Matrix:**

Similarity between two items is obtained by comparing what users have selected/liked/rated about them.

1. **Similarity Computation**:

Compute the similarity between two items using such metrics as cosine similarity.

1. **Recommendation:**

It gives a target user recommendation of items similar to his/her already interacted-with items.

**Advantages**

* **Scalable:** Scales well with large datasets.
* **Domain Agnostic**: It does not require item-specific information.

**Challenges**

* **Cold Start Problem**: It is hard to make recommendations for new users or items with no interaction record available.
* **Data Sparsity**: The user-item interaction matrix usually has a very big sparsity which will be a nightmare while calculating the similarity.

**Applications**

* **Movie Recommendation:** Netflix, IMDB.
* **E-commerce**: Amazon, eBay
* **Music Recommendation**: Spotify, Pandora.

**Sources**:

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning*.
2. Towards Data Science: [Linear Regression Explained](https://towardsdatascience.com/).
3. Alpaydin, E. (2021). *Introduction to Machine Learning*.
4. Scikit-learn Documentation: KNN Algorithm.
5. Analytics Vidhya: [Understanding KNN](https://www.analyticsvidhya.com/).
6. Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook*.
7. Koren, Y., Bell, R., & Volinsky, C. (2009). *Matrix Factorization Techniques for Recommender Systems*.
8. Towards Data Science: [Collaborative Filtering Explained](https://towardsdatascience.com/).
9. Scikit-learn Documentation: Recommender Systems.