Task 1

Question 1

1. **What is the rank of a matrix? Why is it important in machine learning?**
    - **Rank of the matrix is the number of independent rows or columns in the matrix and by independent means that It's not a linear combination of any other columns or row**
    - **The rank represents the maximum number of independent vectors that can span the columns of the matrix**
    - **In machine learning knowing the rank of the data help in understanding the dimensionality of the data so we can perform:**
        - **Dimensionality reduction: data with high dimensionality "high number of features" understanding the rank helps in knowing how many independent features represent**
        - **Principle component analysis uses the rank to reduce the data to its significant features, preserving as much variance as possible while reducing the number of the dimensions**

2. Explain eigenvalues and eigenvectors. How are they used in PCA?

- **The eigenvector is the vector which after applying the transformation stay at the same basis**
- **After transformation the vector stay the same but multiplied by a scalar "this scaler is called eigen value"**

PCA: the most common dimensionality reduction algorithm

And it's done by projecting each data point onto only the first few principle components to obtain lower dimensional data while preserving as much of the data variation as possible

Principle component tries to put the maximum possible information in the first component and the maximum remaining information in the second so we can make use of the first few components only as most of the information are compressed in

Principle components are the eigenvector of the covariance matrix

The steps of PCA:

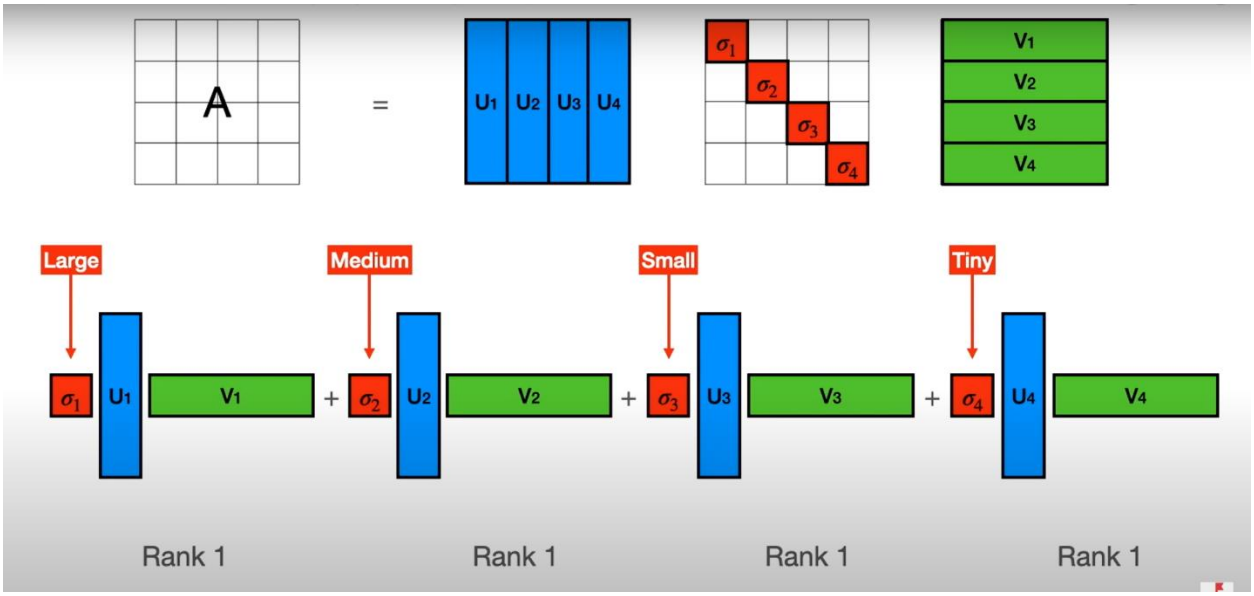
1. **Computing the covariance matrix "measures between to features and it only indicate the directional relationship between two variables"**
2. **Compute the eigen vectors and the eigen values of this matrix**
3. **Sort the eigen values in descending order and by dividing each value by the summation of the values you can get which vectors have the most variation of the data and can be used as principal component, "eigen vector corresponding to the highest eigen value will be the first principle component and so on"**
4. **Reduce the dimension of the data : transform the data using z score and then multiple it by the principal components**

Q3. What is singular value decomposition and why it is important?

It is an approximation of the matrix by rank one matrices

$$A = U\Sigma V^T$$

- **U is an m*m orthogonal matrix whose columns equal the unit eigen vectors of AA transpose**
- **V n*n orthogonal matrix whose columns equal the unit eigen Vectors of A^TA**
- **Σ m*n diagonal matrix and each value is the square root of the positive eigen values of AA^T**



σ_1, σ_2 are the singular numbers

Assume that are large enough compared to the other values thus we can igonre the other values so we can get a very closer matrix to the origial matrix with less rankk and dimensionality

Application of singular value decomposition

- **Dimensionality reduction**

- **matrix approximation**: Like I mentioned above if the top singular values are larger enough compared to the other values you can approximate this matrix by the first singular values
- **image compression**

Q4. what is the difference between causation and correlation

Correlation is when two variables appear to change in such a pattern for example one might decrease as the other increase

In statistics correlation expresses the degree to which two variables change with one another, but it doesn't indicate that one variable is causing the other's change

Causation means one variable directly influences **another, for instance, one variable increase *because* the other decreases**

Testing and analysis confirm whether two variables are merely correlated or have a cause-and-effect relationship

How to measure the correlation

Statistical correlation quantifies the strength and direction of the relationship between two variables. It's measured by the 'correlation coefficients'1 and -1.

A positive value suggests that the variables are increasing or decreasing together: there's a positive correlation. A negative value indicates they are moving in the opposite direction (a negative correlation), and 0 means there is no linear relationship.

How to measure causation

Once you find a correlation, you can test for causation by running experiments that control the other variables and measure the difference. You can use two experiments to identify causation within your product:

- **Hypothesis testing**
- **A/B/n experiments**

Example to indicate the difference between correlation and causation

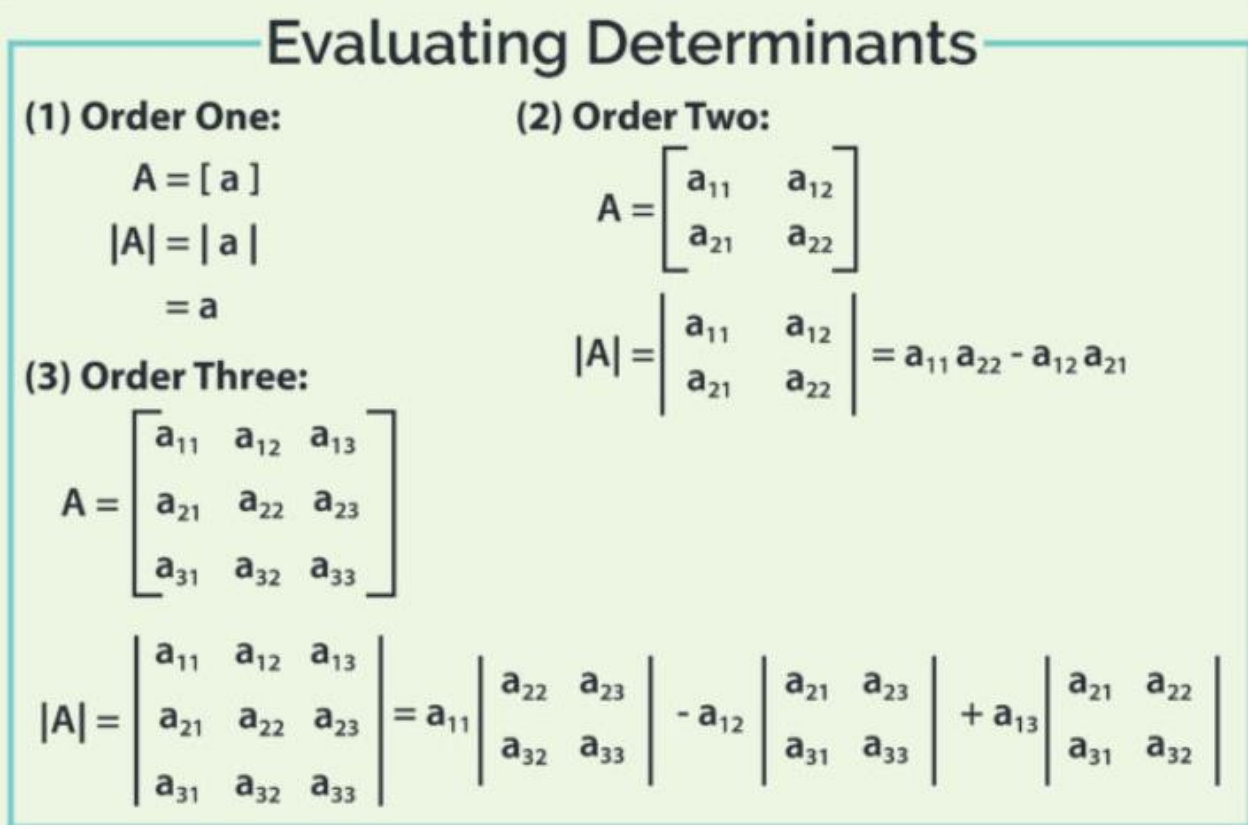
You can notice from statically analysis that number of books that one reads is increasing by increase the size of the shoes so we can define a correlation here but the causation is that there is another factor who causes that which is the age that causes the both factors 'size if the shoes and number of books ' to increase

Q4. How would you compute the inverse of a matrix? What is its relevance in Machine Learning?

$$A^{-1} = \frac{adj\ A}{|A|}$$

Which |A| is the determent

Adj or adjoint matrix which is the transpose of the cofactor matrix



- **Singular matrix**

A singular matrix is a square matrix with a determinant value equal to zero. We cannot find the inverse of a singular matrix. For a singular matrix, |A| = 0.

Cofactor matrix

A cofactor is a number derived by removing the row and column of a certain element in a matrix, which is a numerical grid in the shape of a rectangle or a square. A positive (+) or negative (-) sign usually precedes the cofactor. The formula for determining the cofactor for a particular element is as follows,

$$A_{ij} = (-1)^{i+j} \det M_{ij}$$

$$\begin{bmatrix} 1 & -1 & 2 \\ 4 & 0 & 6 \\ 0 & 1 & -1 \end{bmatrix}$$

Solution: We will first evaluate the cofactor of every element,

$$\text{cof}(a_{11}) = +\begin{vmatrix} 0 & 6 \\ 1 & -1 \end{vmatrix} = -6 \qquad \text{cof}(a_{12}) = -\begin{vmatrix} 4 & 6 \\ 0 & -1 \end{vmatrix} = 4$$

$$\text{cof}(a_{21}) = -\begin{vmatrix} -1 & 2 \\ 1 & -1 \end{vmatrix} = 1 \qquad \text{cof}(a_{22}) = +\begin{vmatrix} 1 & 2 \\ 0 & -1 \end{vmatrix} = -1$$

$$\text{cof}(a_{31}) = +\begin{vmatrix} -1 & 2 \\ 0 & 6 \end{vmatrix} = -6 \qquad \text{cof}(a_{32}) = -\begin{vmatrix} 1 & 2 \\ 4 & 6 \end{vmatrix} = 2$$

$$\text{cof}(a_{13}) = +\begin{vmatrix} 4 & 0 \\ 0 & 1 \end{vmatrix} = 4$$

$$\text{cof}(a_{23}) = -\begin{vmatrix} 1 & -1 \\ 0 & 1 \end{vmatrix} = -1$$

$$\text{cof}(a_{33}) = +\begin{vmatrix} 1 & -1 \\ 4 & 0 \end{vmatrix} = 4$$

Applications in Machine Learning

The matrix inverse is important in:

1. **Linear Regression: Solving for weights in the normal equation:**

The Normal Equation

To find the value of $\boldsymbol{\theta}$ that minimizes the MSE, there exists a *closed-form solution*—in other words, a mathematical equation that gives the result directly. This is called the *Normal equation* (Equation 4-4).

Equation 4-4. Normal equation

$$\widehat{\boldsymbol{\theta}} = (\mathbf{X}^{\intercal}\mathbf{X})^{-1}\,\mathbf{X}^{\intercal}\,\mathbf{y}$$

In this equation:

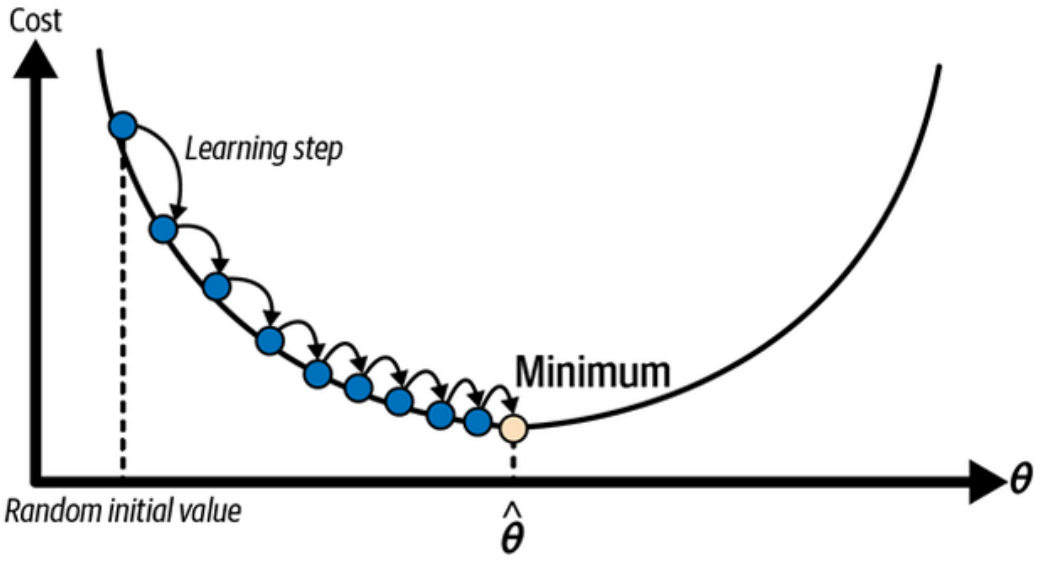
- $\widehat{\boldsymbol{\theta}}$ is the value of $\boldsymbol{\theta}$ that minimizes the cost function.
- $\mathbf{y}$ is the vector of target values containing $y^{(1)}$ to $y^{(m)}$.

2. **Optimization: Calculating gradients and Hessians in advanced models.**
3. **Transformations: Inverting transformations for feature scaling or spatial transformations.**

Q6.How does gradient descent work? Explain its importance in ML.

Gradient descent is a generic optimization algorithm capable of finding optimal solutions to a wide range of problems. The general idea of gradient descent is to tweak parameters iteratively in order to minimize a cost function.

In practice, you start by filling θ with random values (this is called random initialization). Then you improve it gradually, taking one baby step at a time, each step attempting to decrease the cost function (e.g., the MSE), until the algorithm converges to a minimum



Equation 4-7. Gradient descent step

$$\boldsymbol{\theta}^{(\text{next step})} = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \operatorname{MSE}(\boldsymbol{\theta})$$

Θ is the parameters of the model

Ƞ learning rate hyperparameter

▼MSE(Θ) The gradient vector, noted ∇ MSE(θ), contains all the partial derivatives of the cost function (one for each model parameter).

Equation 4-5. Partial derivatives of the cost function

$$\frac{\partial}{\partial \theta_j} \operatorname{MSE}(\boldsymbol{\theta}) = \frac{2}{m} \sum_{i=1}^{m} \left(\boldsymbol{\theta}^{\intercal} \mathbf{x}^{(i)} - y^{(i)} \right) x_j^{(i)}$$

Instead of computing these partial derivatives individually, you can use Equation 4-6 to compute them all in one go. The gradient vector, noted $\nabla_{\boldsymbol{\theta}} \operatorname{MSE}(\boldsymbol{\theta})$, contains all the partial derivatives of the cost function (one for each model parameter).

Equation 4-6. Gradient vector of the cost function

$$\nabla_{\boldsymbol{\theta}} \operatorname{MSE}(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial}{\partial \theta_0} \operatorname{MSE}(\boldsymbol{\theta}) \\ \frac{\partial}{\partial \theta_1} \operatorname{MSE}(\boldsymbol{\theta}) \\ \vdots \\ \frac{\partial}{\partial \theta_n} \operatorname{MSE}(\boldsymbol{\theta}) \end{pmatrix} = \frac{2}{m} \mathbf{X}^{\intercal} (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$$

ALL these photos from hands on machine learning book

What are the types of ML? Discuss each type and give examples on them.

There are many criteria to classify machine learning systems like :

- How they are supervised during training (supervised, unsupervised, semi-supervised, self-supervised, and others)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model, much like scientists do (instance-based versus model-based learning)

I will focus on the first one as it is the most used criteria

Supervised learning

In supervised learning, the training set you feed to the algorithm includes the desired solutions, called labels

A typical supervised learning task is classification. The spam filter is a good example of this: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails.

Another typical task is to predict a target numeric value, such as the price of a car, given a set of features (mileage, age, brand, etc.). This sort of task is called regression

Unsupervised learning

involves training models on unlabeled data, where the system identifies patterns and structures without explicit guidance. Key tasks include:

Clustering: Algorithms group similar data points together. For instance, a clustering algorithm might categorize blog visitors into groups (e.g., teenagers who enjoy comic books and adults who enjoy sci-fi).

Visualization: Algorithms like t-SNE help visualize high-dimensional data in 2D or 3D, making it easier to spot patterns and clusters.

Dimensionality Reduction: Reduces the number of features while preserving information (e.g., merging age and mileage of a car into a single feature to represent wear and tear).

Anomaly Detection: Identifies unusual or abnormal instances in the data, such as fraudulent credit card transactions or manufacturing defects, based on patterns learned from normal data.

Association Rule Learning: Uncovers relationships between attributes in large datasets. For example, discovering that customers who buy barbecue sauce and potato chips are likely to also buy steak, which could inform product placement in a store.

Semi-supervised learning

you will often have plenty of unlabeled instances, and few labeled instances.

Some photo-hosting services, such as Google Photos, are good examples of this. Once you upload all your family photos to the service, it automatically recognizes that the same person A shows up in photos 1, 5, and 11, while another person B shows up in photos 2, 5, and 7. This is the unsupervised part of the algorithm (clustering). Now all the system needs is for you to tell it who these people are. Just add one label per person and it is able to name everyone in every photo, which is useful for searching photos.

Reinforcement learning

Reinforcement learning is a very different. The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return. It must then learn by itself what is the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation

For example, many robots implement reinforcement learning algorithms to learn how to walk. DeepMind's AlphaGo program is also a good example of reinforcement learning: it made the headlines in May 2017 when it beat Ke Jie, the number one ranked player in the world at the time, at the game of Go. It learned its winning policy by analyzing millions of games, and then playing many games against itself.

Q8. Explain this code and provide a way to optimize it:

```
SELECT DISTINCT ProductID,
       (
       SELECT SUM(Amount)
        FROM Sales S2
        WHERE S2.ProductID = S1.ProductID
        ) AS TotalSales
FROM Sales S1;
```

SELECT DISTINCT ProductID:

To only get the unique values with no duplicates

And notice there is outer query and inner subquery

Subquery (SUM(Amount) FROM Sales S2 WHERE S2.ProductID = S1.ProductID):

- **For each unique ProductID found in the main query (which refers to S1), a subquery is used to calculate the sum of the Amount for that specific product from the Sales table.**
- **This subquery is related with the outer query because it uses S1.ProductID in the WHERE clause, meaning it will calculate the sum for each ProductID in the main query.**

FROM Sales S1:

- **The main query is working with the Sales table (aliased as S1), which is the outer where it get the distinct ProductID.**

Optimization:

The query can be optimized by using a GROUP BY approach instead of using a subquery.

Query | Query History

```
1  SELECT ProductID,
2         SUM(Amount) AS TotalSales
3  FROM Sales
4  GROUP BY ProductID;
```

Using GROUP BY product and aggregate the Amount for each group and group here is the unique ProductID