Advanced Questions

1) What is the rank of a matrix? Why is it important in machine learning?

- The rank of a matrix is the number of linearly independent rows or columns in the matrix.

- It is important in ML because it tells us how much unique information we have. Like, if you have a matrix with 100 features but a rank of 5, that means you could probably describe your data just as well with 5 features ,the rest is just redundant information.

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

- Think of an eigenvector as a special direction where when you transform the data. The eigenvalue tells you how much stretching or squeezing happens. In PCA, we use these to find the most important directions in our data - the directions where our data varies the most. That's how we can reduce dimensions while keeping the most important information.

3) What is a singular value decomposition (SVD), and why is it important?

- The Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. It is like breaking down a complicated matrix into simpler parts.

-It is important because SVD does something similar with data - it identifies the most important patterns or "features" that really define your dataset. This

is very useful when you're dealing with things like recommendation systems. It is used in Data reduction , Data driven generalization of Fourier transform.

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

Computing a matrix inverse is like finding the "opposite" of a matrix.

We have multiple ways to compute it like:

1. Using Python/NumPy: The easiest way is just numpy.linalg.inv(matrix)
2. Gaussian elimination: Solving step by step
3. Using built-in algorithms in ML libraries

It is useful in ML in:

1. Linear Regression: When we're trying to find the best-fit line through data points, we use matrix inverse to solve for the coefficients. It helps us find the best model parameters.
2. Feature Selection: Matrix inverse helps us understand which features are most important in our model and how they relate to each other.
3. Data Transformation: Sometimes we need to "undo" transformations we've applied to our data - matrix inverse helps us do this.

5) What is the difference between correlation and causation?

Let me explain it with example to be more clear:

Imagine you notice that ice cream sales and shark attacks both increase during summer. That's correlation - they happen together, but one doesn't

cause the other. The real cause? Hot weather! It makes people buy more ice cream AND go swimming more (leading to more shark encounters).

Correlation just means two things tend to happen together or follow similar patterns. Like how people carrying umbrellas correlates with rainy days. But causation? That's when one thing directly causes another - like how rain actually causes people to open their umbrellas.

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

Think of gradient descent like trying to find the bottom of a valley. You're standing somewhere on a hill, and you want to get to the lowest point. That's basically gradient descent!

In machine learning, we're trying to find the "valley" - the point where our model makes the fewest mistakes. Here's how it actually works:

1. Then we check how wrong we are (calculate the error)
2. We figure out which direction would reduce our error the most
3. Take a small step in that direction
4. Repeat until we're not getting much better

The "small step" part is super important - we call it the learning rate. If your steps are too big, you might overshoot the bottom of the valley. Too small, and you'll take forever to get there!

Why is it so important? Because:

- It helps models learn from their mistakes
- Works well with huge amounts of data
- Can handle complex problems with many variables
- It's like having an automatic fine-tuning system

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

1. Supervised Learning - Think of this as learning with a teacher. The algorithm gets labeled examples to learn from, like a student getting solved problems to understand patterns. A classic example is email spam detection - we show the system lots of emails marked as "spam" or "not spam" so it learns what to look for.
2. Unsupervised Learning - This is more like exploring and finding patterns on your own. The algorithm looks at data without any labels and tries to find structure. It's like giving someone a bunch of photos and asking them to group similar ones together without telling them what to look for. Think of how Netflix groups similar movies together based on viewing patterns.
3. Reinforcement Learning - This one's fascinating - it's like learning through trial and error with rewards. Imagine teaching a dog new tricks by giving treats when they get it right. That's how AI learned to play games like Chess and Go - by playing millions of games and learning from wins and losses.

8) Explain this code and provide a way to optimize

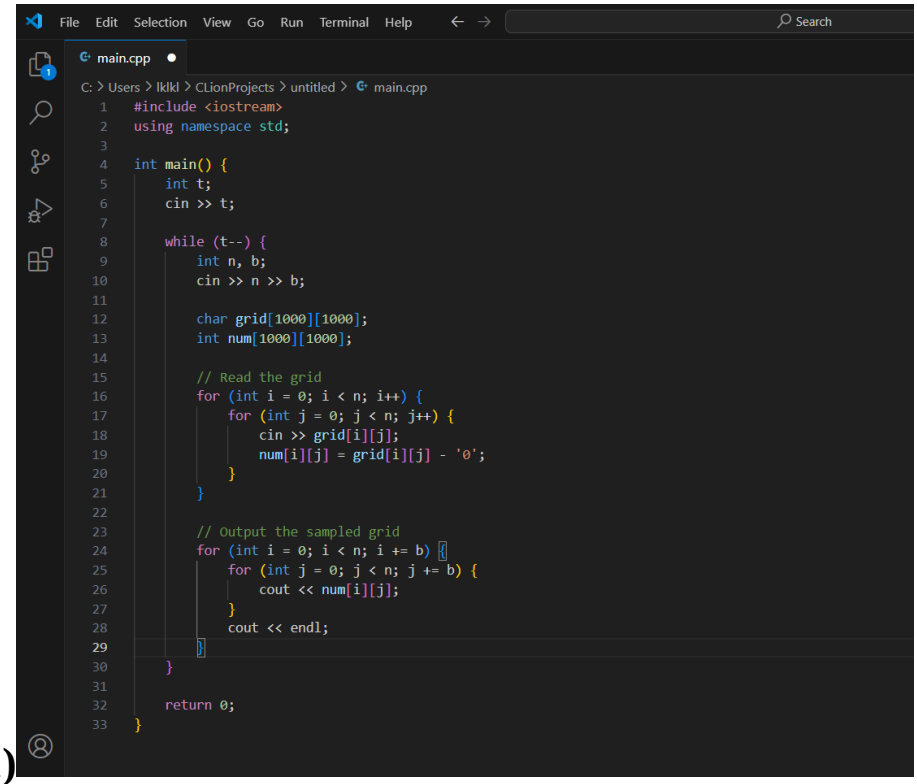
This SQL query calculates the total sales for each distinct product in the Sales table and uses a correlated subquery to compute the total sales amount for each product.

The optimized version:

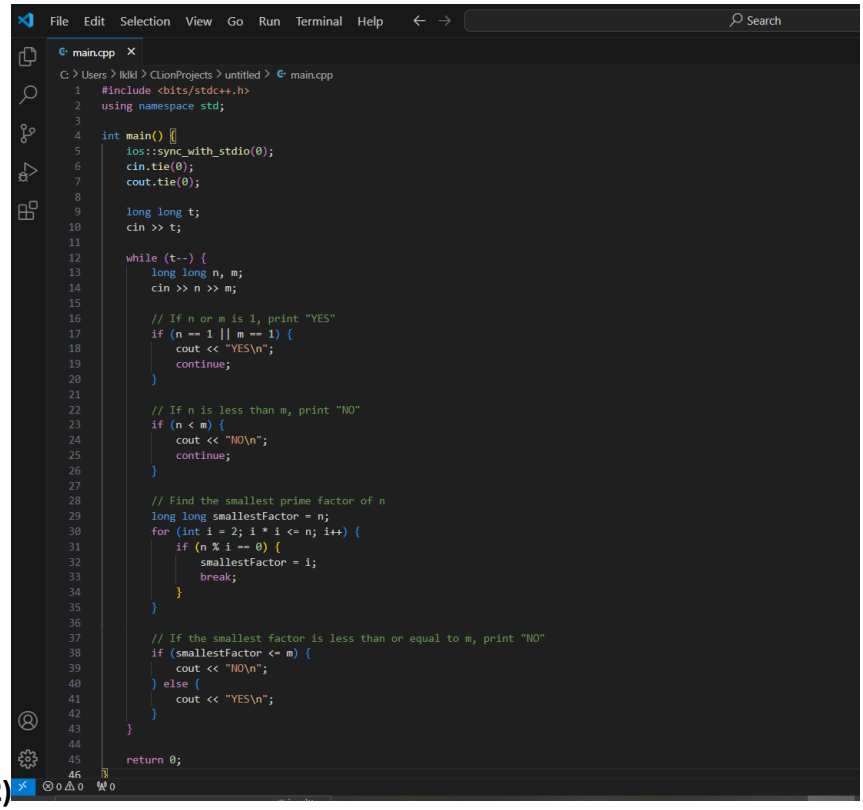
```sql
SELECT
    ProductID,SUM(Amount) AS TotalSales
FROM Sales
GROUP BY ProductID;
```

Problem Solving Questions:

1)



2)



3)

```sql
SELECT
    s.student_id,
    s.student_name,
    sub.subject_name,
    COUNT(e.subject_name) as attended_exams
FROM Students s
CROSS JOIN Subjects sub
LEFT JOIN Examinations e
    ON s.student_id = e.student_id
    AND sub.subject_name = e.subject_name
GROUP BY s.student_id, s.student_name, sub.subject_name
ORDER BY s.student_id, sub.subject_name;
```

4)

```sql
WITH tweet_counts AS (
    SELECT user_id,COUNT(*) AS tweet_bucket
    FROM tweets
    WHERE YEAR(tweet_date) = 2022
    GROUP BY user_id
)
SELECT tweet_bucket,COUNT(*) AS users_num
FROM tweet_counts
GROUP BY tweet_bucket
ORDER BY tweet_bucket;
```