Vistora AI Assignment for AI ML Intern

Name : Ayush Parwal

Branch : CSE AIML

Roll Number : bt22csa023@iiitn.ac.in

Thanks for considering my application.
Hi, I'm Ayush Parwal, currently pursuing a B.Tech in AIML at Indian Institute of Information Technology, Nagpur.
I am passionate about machine learning and AI, with hands-on experience in large language models, generative AI, and NLP.
I enjoy building intelligent systems that solves real-world challenges.

Question 1) Introduction to Feature Engineering?

Answer : Feature Engineering is the process of converting raw data into meaningful and semantic input features that can be used to improve the performance of machine learning models.

This includes creating new features, modifying existing ones, or selecting the most relevant features from data we have.

It involves domain expertise, statistical knowledge, and a solid understanding of the data, and it is often considered one of the most critical steps in the machine learning pipeline.

Improves Model Accuracy: Quality features allow models to better detect patterns and relationships in the data.

Reduces Overfitting: Proper feature selection and transformation can help generalize models better we can use regularization as well.

Boosts Efficiency: Clean and well-prepared features can accelerate training and reduce model complexity.

Normalization is the process of scaling numerical values to a common range, usually between 0 and 1. This helps machine learning models train faster and perform better, especially those sensitive to the scale of data.

Example:

Original values: 10, 20, 30, after transforming this we can get like this 1,2,3 that is simple to compute.

Encoding converts categorical data into numerical format so that machine learning algorithms can use it.

Common Methods:

One-Hot Encoding: Converts each category into a separate binary column.

Example: Color = [Red, Blue, Green] → [1, 0, 0], [0, 1, 0], [0, 0, 1]

But this will create the high dimensional data.

Label Encoding: Assigns a unique number to each category.

Example: Red = 0, Blue = 1, Green = 2 like this.

Time-based Aggregations generate features by summarizing data over specific time windows.

Examples:

1)Total sales in last 7 days.

2)Average temperature over past month.

I can write the code as well

"SELECT customerid, COUNT(*) AS orders_last_7_days FROM

orders WHERE orderdate >= CURRENT_DATE - INTERVAL '7 days'

GROUP BY customerid;"

Question 2) Using Snowflake for Data Storage & Processing?

Answer : Snowflake is a cloud-based data platform that supports storing and querying structured e.g., tables, CSV and semi-structured e.g., JSON, Parquet data.

Structured Data: Stored in traditional table format using rows and columns e.g., customer info, sales records.

Semi-Structured Data: Handled natively using the VARIANT data type, allowing you to store JSON and XML inside regular tables and query them using SQL.

Example SQL Query to Extract & Preprocess Data:

"SELECT customeid, COUNT(orderid) AS total_orders_30d, AVG(order_amount) AS avg_order_value_30d, MAX(order_date) AS last_order_date FROM orders WHERE order_date >= CURRENT_DATE - INTERVAL '30 days' GROUP BY customerid;"

Snowflake integrates smoothly with modern ML workflows using tools such as:

Snowpark Python, Scala: Lets you write feature engineering logic in Python or Scala and run it inside Snowflake.

External Functions: Call external services (e.g., REST APIs or ML models hosted outside) from within SQL queries.

Python ML Libraries + Connectors:

Use snowflake-connector-python to fetch training data from Snowflake into a Pandas DataFrame.

Connect Snowflake to ML platforms like AWS SageMaker, Databricks, or Azure ML.

Streamlit in Snowflake: Build and deploy interactive ML dashboards directly within the Snowflake ecosystem.

Question 3: Feature Store Concepts?

Answer: A Feature Store is a centralized system for managing, storing, and serving features used in machine learning models.

It is designed to Store preprocessed features so they can be reused across different models and teams. Ensure consistency between training and inference (same logic, same results). Track feature versions and their lineage for reproducibility. Serve features in real time or in batch depending on model needs.

Lets see the comparison in AWS Sage Maker Feature Store, Snowflake Feature Store:

AWS SageMaker Feature Store is a fully managed service focused on ML workflows within AWS. It supports both real-time (online) and batch (offline) feature storage and integrates seamlessly with other AWS ML tools, making it ideal for low-latency model serving and automated feature management.

Snowflake Feature Store leverages Snowflake's cloud data platform and SQL capabilities for feature storage and engineering. It is best suited for teams already using Snowflake, focusing mainly on batch processing and seamless integration with data analytics and ETL pipelines.

Question 4: Implementing Feature Engineering with Snowflake & Feature Store?

Answer : lets see this one by one so moving with first that is…

1) <u>Extract: Fetch Raw Data Using Snowflake SQL</u>

First, use Snowflake SQL queries to extract relevant raw data from tables. For example, to get customer purchase data.

With this sql query:

"*SELECT customerid, orderid, order_amount, order_date FROM orders WHERE order_date >= CURRENT_DATE - INTERVAL '90 days';"*

2) <u>Transform: Perform Feature Engineering</u>

Use SQL to create features like aggregations or encodings. For example, calculate total spending and number of orders per customer.

With this sql query:

“SELECT customer_id, COUNT(order_id) AS total_orders, SUM(order_amount) AS total_spent, AVG(order_amount) AS avg_order_value FROM orders WHERE order_date >= CURRENT_DATE - INTERVAL '90 days' GROUP BY customer_id;”

3) Load into Feature Store

Features can be stored back into a Feature Store for reuse and consistency. In Snowflake, this often means storing the engineered features in a dedicated, versioned table, e.g., customer_features.

Alternatively, in cloud platforms like AWS SageMaker Feature Store or Databricks Feature Store, you load features using their APIs or SDKs to register and store features with metadata and versioning.

4) Access for ML Models

ML models access features by querying the feature store:

Batch mode: Retrieve feature tables from Snowflake or cloud feature stores into data frames for training.

Online mode: Use low-latency APIs (like AWS SageMaker Feature

Store endpoints) to fetch features in real time during inference.

Integrated pipelines: Use Snowpark or data connectors to integrate feature retrieval seamlessly within ML pipelines or notebooks.

Question 5) Practical Task?

Answer : lets see the demonstration…