Final Assignment: Database Management Tools in Python

Total Marks: 50 points
Submission Deadline: last day of exam period
Format: Individual

Overview

You must complete both **Part I (Core Programming and Algorithms)** and **Part II (Data Ingestion and Analytics)**.

---

1. Part I: Data Structures & Algorithms (25 points)

A. Node-Based Implementations

1. **Linked Nodes:**
   - Implement a **Doubly Linked Node** class.
   - Create a **Stack** class using these nodes.
   - Use two stacks to implement a **Queue** with `enqueue`, `dequeue`, and `peek` methods.
2. **Hash Table with Binary Search Trees (BST):**
   - Implement a Hash Table where each bucket uses a BST to handle collisions.
   - Demonstrate insert, search, and delete operations.
3. **Graph Implementation:**
   - Implement a Graph using a Node class (adjacency list representation).
   - Include traversal algorithms:
     - Depth-First Search (DFS)
     - Breadth-First Search (BFS)

---

2. Part II: Databases, Ingestion & Analytics (25 points)

Choose a dataset from any domain of your interest. You may use freely available public datasets (for example, from Kaggle, UCI Machine Learning Repository, or government open data portals). The dataset should be rich enough to support meaningful data storage, analysis, and visualization tasks.

A. Data & Database

1. **Database Selection:** Select an appropriate database system (e.g., relational, document, time-series, or graph database) to store your dataset. Clearly justify your choice based on data characteristics (structure, volume, relationships, update frequency, etc.).

2. **Data Ingestion:** Provide the code used to import or insert your chosen dataset into the selected database. Include any preprocessing or transformation steps required for proper data storage.

3. **Analytical Queries:** Write and execute a set of analytical queries that generate insights from your dataset. Examples include:

    - Identifying key trends or patterns (e.g., top-performing entities, high or low values).
    - Aggregating data over time or categories.
    - Detecting outliers or anomalies.

B. Analytics & Visualization

1. Use Python, SQL, or any suitable analytics tool to summarize and explore your data.

2. Create at least two visualizations (e.g., line charts, histograms, scatter plots, or heatmaps) that effectively communicate your findings.

3. Highlight any interesting trends, correlations, or anomalies observed in the data.

C. Machine Learning Component

1. Build and evaluate a suitable **Regression** or **Classification** model using your dataset. Clearly describe the problem you are addressing and justify your choice of model.

2. Perform **Feature Engineering**:

- Transform or encode features as needed (normalization, one-hot encoding, aggregations, etc.).
- Explain how feature design and storage can affect both model performance and database efficiency.

---

3. Submission Requirements

- The final submission must be in a single **Jupyter Notebook** file (`.ipynb`).
- The notebook should include both:
  - **Code cells** for implementation, data loading, analysis, and visualization.
  - **Markdown cells** for explanations, reasoning, and interpretation of results.
- Each section of the assignment (Data Structures, Database, Analytics, and Machine Learning) should be clearly separated with appropriate Markdown headings.
- All visual outputs (e.g., plots, graphs, tables) should be displayed directly within the notebook.
- The notebook must be self-contained and executable from start to finish without requiring external manual steps.
- Include comments or brief discussions in Markdown explaining:
  - The dataset and domain choice.
  - Implementation approach and design decisions.
  - Key analytical insights and model results.

---

4. Grading Breakdown (50 points)

Component	Points	Description
Linked Nodes, Stack, Queue	5	Functional and documented code
Hash Table + BST	5	Correct implementation and usage
Graph + Traversal Algorithms	5	Efficient and correct traversal
Database & Queries	15	Schema, data, and analytical queries
Visualization & ML Model	20	Quality of analysis, plots, and discussion
Total	**50 pts**	

Notes

- Submit all code and documentation in a single compressed folder or via GitHub.
- Late submissions are not accepted.
- You may discuss concepts with peers, but code and analysis must be original.