

Code Portfolio

Name: Manasa P

UGID: 21WU0101055

Team 3

Repository URL: <https://github.com/manasa-x/Code-Portfolio>

Trained by: Dr. Thasni T

Overview

This portfolio showcases projects and tasks across various domains, including Python programming, SQL, data manipulation with Pandas, exploratory data analysis (EDA), Machine learning project on license plate detection and Learning Management System and finally, deploying the app using Flask. Each project highlights problem-solving approaches, implementation details, and outputs.

4x4 Training Program

Day 1:

- Appointed as Cerebra Tech employees
- Had an overview about python programming
- Python tasks focusing on scripting, problem-solving, and logic building.
- Examples include working with control structures, functions, and object-oriented programming.

Key Files:

- Python-Programming Folder: Includes all the 13 assignments covering various topics

Python:

Topics covered include:

- Introduction, variables, datatypes, indentation, strings, lists, tuples, sets, dictionary, standard input output, operators and its types
- Conditional statements (if, if-elif-else, if-else)
- Loops (For, While, Do-while)
- Object Programming (class, objects, encapsulation, inheritance, polymorphism)
- Exception handling (try, except, finally)
- File modes

Day 2:

- 50 SQL Challenges were given to code and submit using Github
- SQL queries to interact with relational databases.
- Includes database creation, table manipulation, and advanced queries.

Key Files:

- The SQL pdf documentation contains 25 questions along with their code and solutions.

Day 3:

- Overview about Pandas and Numpy python libraries
- Discussed about exploratory data analysis and applied it to an existing dataset
- Insights into datasets using Python visualization libraries.
- Steps include data cleaning, exploratory analysis, and visualization.
- Also discussed about machine learning algorithms including supervised (Linear regression and classification), unsupervised and reinforcement learning

Key Files:

- Pandas-Task Folder: Includes sales dataset and data manipulation task performed on it using pandas
- Exploratory-Data-Analysis Folder: Includes covid-19 dataset country wise dataset and EDA performed on it along with insights inferred.
- ML Project Folder: License plate detection project done with my teammate.

Pandas:

- Data manipulation and analysis in Python.
- Data Structures: Series and DataFrame objects for handling 1D and 2D data.
- Data Manipulation:
 - Importing/exporting data from various formats (CSV, Excel, JSON).
 - Filtering, sorting, and aggregating data.
 - Handling missing values.

EDA:

- Focus: Gaining insights and understanding data distributions.
- Overview statistics: describe(), info().
- Distribution checks using histograms, boxplots, and scatter plots.
- Heatmaps for correlation analysis, pair plots and trend analysis.

ML:

Focus: Building predictive models using datasets.

- **Supervised Learning:**
 - Regression: Predicting continuous values using models like Linear Regression.
 - Classification: Identifying categories using algorithms like Decision Trees, Random Forests, or Logistic Regression.
- **Unsupervised Learning:**
 - Clustering: Grouping data points using methods like K-Means.
 - Dimensionality Reduction: Techniques like PCA to reduce data complexity.
- **Model Evaluation:**
 - Metrics: Accuracy, Precision, Recall, F1-score, RMSE.
 - Validation: Train-test splits and cross-validation techniques.
- **Model Deployment:**
 - Integration into applications.

Day 4:

- Overview on Artificial Neural Network and the basics of perceptron and neural network
- Classifying iris dataset
- Task: the goal of the project is to classify handwritten digits (0-9) from the MNIST dataset using a CNN. The MNIST dataset is a collection of 28x28 pixel grayscale images of handwritten digits.

Key Files:

- The DL Folder consists of the jupyter notebook performing cnn on mnist dataset.

Perceptron:

A simple neural network used for binary classification. It consists of:

- **Input Layer:** Takes input features.
- **Weights & Bias:** Determines the importance of features.
- **Activation Function:** Produces the output (0 or 1).

It works by calculating the weighted sum of inputs and applying an activation function. The perceptron is limited to solving only linearly separable problems (e.g., basic classification tasks).

CNN:

A deep learning model designed for image and visual data classification. It consists of:

1. **Convolutional Layers:** Extracts features from images.

2. **Pooling Layers:** Reduces dimensions and helps with computational efficiency.
3. **Fully Connected Layers:** Makes final classification decisions.

CNNs are highly effective for tasks like image recognition, object detection, and segmentation, automatically learning hierarchical features from raw data.