2024

List of Python Project Assignment

PROJECTS FOR DATA SCIENCE WITH PYTHON

PYTHON-DOCX

Contents

1. Smart Data Analyzer

Objective:
Develop a data analysis tool in Python that reads CSV and Excel files, automatically analyzes the dataset, categorizes columns as categorical or numerical, and identifies the best possible analysis methods.

Requirements:

- - File Input: Read CSV and Excel files.
- - Automatic Analysis: Identify categorical and numerical columns.
- - Best Analysis Methods: Suggest appropriate analysis techniques for each column type.
- - Graphical Output: Plot selected graphs and compile them into a PDF.
- - Statistical Inference: Generate a human-readable summary of statistical insights.

Tools & Libraries:

Pandas, Matplotlib, Seaborn, ReportLab

Expected Outcome:

A user-friendly console-based or GUI application that guides users through uploading files, displays data insights, and optionally saves graphical data in a PDF.

2. Probability-Based Machine Learning System for Gender Identification

Objective:
Build a machine learning system that learns from a database of names and predicts the gender based on new inputs, allowing incremental learning with new names.

Requirements:

- - Data Input: Database of names with gender labels for initial training.
- - Learning System: Model that can be updated with new names and labels.
- - Prediction System: Users input names to receive gender predictions.
- - Incremental Storage: Save newly learned names to the model for future predictions.

Tools & Libraries:

Scikit-Learn, Pandas, Joblib

Expected Outcome:

A command-line or GUI-based application where users enter names, see gender predictions, and add new names to improve model accuracy.

3. Deep Learning Image Classifier

Objective:
Create a deep learning model that processes a folder of images and identifies the content in each image.

Requirements:

- - Image Input: Ability to input a folder containing images.
- - Model Training: Train a deep neural network on a predefined dataset.
- - Prediction: Recognize content in new images and label each accordingly.

Tools & Libraries:

TensorFlow, Keras, OpenCV, PIL

Expected Outcome:

A tool that can take a folder of images and output labeled predictions for each, potentially with a simple UI or command-line interface.

4. Tic-Tac-Toe with Reinforcement Learning

Objective:
Develop a Tic-Tac-Toe game that employs reinforcement learning to learn and improve with each game against the user.

Requirements:

- - Game Interface: User interface for playing Tic-Tac-Toe.
- - Q-Learning: Implement Q-learning for the AI to learn from user moves.
- - Persistent Learning: Save the AI's learning state for future games.

Tools & Libraries:

Python, Pickle

Expected Outcome:

A Tic-Tac-Toe game where the AI gradually improves and can be saved for continuous learning.

5. Mathematical Formula Extractor and Evaluator

Objective:
Develop an application that extracts mathematical formulas from images and evaluates them.

Requirements:

- - Image Input: Accept an image containing handwritten or printed formulas.
- - Text Extraction: Use OCR to detect and extract formulas.
- - Evaluation: Parse and evaluate mathematical expressions.

Tools & Libraries:

Tesseract OCR, SymPy

Expected Outcome:

An application that extracts mathematical expressions from images and provides solutions, usable via command-line or basic GUI.

6. Interactive HTML Dashboard for Data Visualization

Objective:
Build an interactive HTML dashboard allowing users to upload CSV, Excel, or PDF files and visualize the data in user-selected graphs with customizable colors.

Requirements:

- - File Upload: Interface to upload CSV, Excel, or PDF files.
- - Data Processing: Parse uploaded data and prepare for visualization.
- - Graph Options: Display four customizable graphs with selectable colors.
- - Dynamic Visualization: Users choose graphs to display with selected colors.

Tools & Libraries:

Flask, Django, D3.js, Plotly, Pandas

Expected Outcome:

A web-based dashboard for visualizing uploaded data with an intuitive interface for selecting graph types and colors.

7. Web-based Market Information Analyzer

Objective:
Develop a program that accepts user input to search the web, extract relevant market data, and perform basic analysis on the mined information.

Requirements:

- - User Input: Accept queries related to market information (e.g., stock prices, commodity prices, market trends).
- - Web Scraping: Search the web and scrape relevant data.
- - Data Analysis: Perform trend analysis, calculate averages, and provide a summary.
- - Output: Display analysis results in a readable format, optionally save them to a file.

Tools & Libraries:

BeautifulSoup, Requests, Pandas, Matplotlib

Expected Outcome:

An application that takes user queries, mines relevant market data from the web, and outputs analyzed trends and summaries to help users make data-driven decisions.