PS-12 Knowledge Representation

Intel Unnati Industrial Training -Domain Masters[Appl.ID:131]

Problem Statement:

Title: **Knowledge Representation and Insight Generation from Structured Datasets**

Objective:

The goal of this project is to create an AI-based solution that can process, analyze, and generate insights from structured datasets. The solution should be capable of:

1. **Processing and Analyzing Structured Data**:
    - Efficiently handling various types of structured datasets.
    - Analyzing the data to extract valuable information.
2. **Identifying Patterns**:
    - Detecting patterns, trends, and correlations within the data.
    - Highlighting significant findings that are not immediately obvious.
3. **Generating Meaningful Insights**:
    - Producing insights that are actionable and can inform decision-making processes.
    - Presenting insights in a comprehensible and usable format.

Explanation of Problem Statement and Objectives

Problem Statement: Organizations and individuals often have access to large volumes of structured data (e.g., spreadsheets, databases) but may struggle to extract meaningful insights from this data. The challenge lies in effectively representing the knowledge contained within these datasets and identifying patterns that can lead to actionable insights. Manual analysis is time-consuming and prone to human error, which creates a need for automated solutions.

Objectives:

1. **Processing and Analyzing Structured Data**:
    - **Handling Various Types of Data**: The AI solution should be versatile enough to work with different formats of structured data, such as CSV files, SQL databases, and Excel spreadsheets.
    - **Efficient Analysis**: It should be able to perform data cleaning, normalization, and transformation to prepare the data for analysis. This includes dealing with missing values, outliers, and inconsistent data entries.

2. **Identifying Patterns**:
    - **Detecting Trends**: The solution should be capable of identifying trends over time, such as sales growth or seasonal patterns.
    - **Finding Correlations**: It should be able to determine relationships between different data variables, such as the correlation between marketing spend and sales performance.
    - **Highlighting Anomalies**: The solution should identify outliers and anomalies that could indicate significant events or errors in the data.
3. **Generating Meaningful Insights**:
    - **Actionable Insights**: The insights generated should be actionable, providing users with clear recommendations or highlighting areas that need attention.
    - **Comprehensible Presentation**: Insights should be presented in a way that is easy to understand, using visualizations, summaries, and explanations that make the data accessible to non-experts.
    - **Supporting Decision-Making**: The insights should aid in making informed decisions by providing evidence-based analysis.

By achieving these objectives, the AI-based solution will empower users to unlock the full potential of their structured data, leading to better decision-making and improved outcomes.

Data Description

adult.csv dataset

Dataset Description

Overview:

- **Rows**: 48,842
- **Columns**: 15

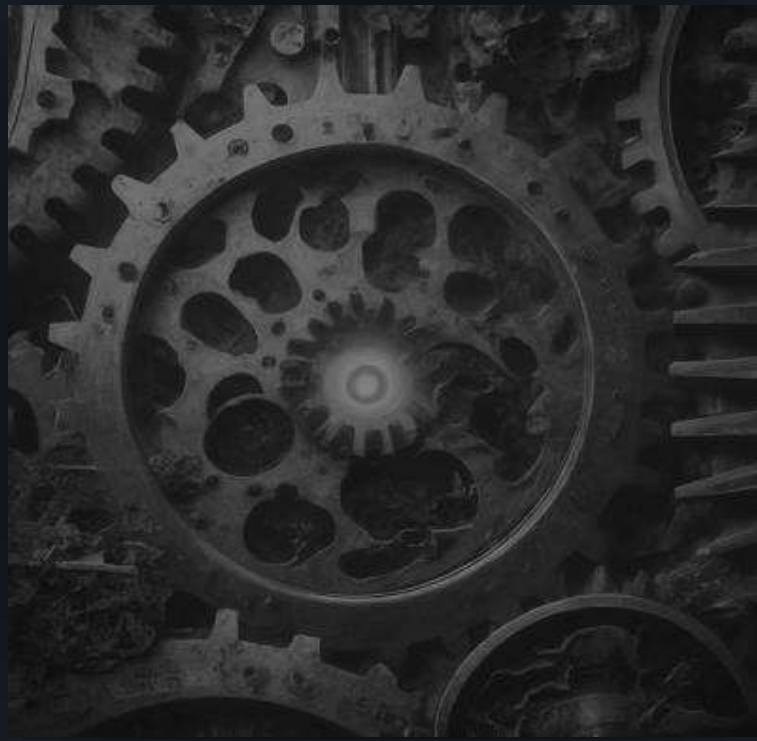
Columns:

1. **age**: Integer, Age of the individual.
2. **workclass**: Categorical, Type of employer.
3. **fnlwgt**: Integer, Final weight.
4. **education**: Categorical, Highest level of education achieved.
5. **educational-num**: Integer, Number of years of education.
6. **marital-status**: Categorical, Marital status.
7. **occupation**: Categorical, Job type.
8. **relationship**: Categorical, Relationship status.
9. **race**: Categorical, Race.
10. **gender**: Categorical, Gender.
11. **capital-gain**: Integer, Capital gain.
12. **capital-loss**: Integer, Capital loss.
13. **hours-per-week**: Integer, Hours worked per week.
14. **native-country**: Categorical, Country of origin.
15. **income**: Categorical, Income category (<=50K, >50K).

Key Insights:

- The dataset includes demographic information and income.
- It contains both numerical and categorical variables.
- No missing values are present in the dataset.
- Primary variables of interest for analysis could include `age`, `education`, `occupation`, `hours-per-week`, and `income`.

This dataset is well-suited for analyzing relationships between demographic factors and income levels.



Unique Idea Brief (Solution)

Knowledge Representation and Insight Generation from Structured Datasets

Solution Overview: This project aims to develop an AI-based solution to process, analyze, and extract meaningful insights from the structured Adult Income Dataset. The dataset is sourced from the UCI Machine Learning Repository and contains information about individuals' demographic and employment attributes. The primary goals are to preprocess the data, effectively represent the knowledge, and identify patterns to support decision-making processes. The solution leverages various data preprocessing techniques and machine learning models to achieve these objectives.

The unique aspect of this solution lies in its comprehensive approach to data preprocessing, scalability using distributed computing, and the implementation of multiple machine learning models to ensure robust insight generation. By focusing on the entire pipeline from data cleaning to model evaluation and insight generation, the project provides a holistic framework for handling structured datasets.

Features Offered

Data Cleaning:

- **Handling Missing Values:** Using KNN Imputer, missing values in the dataset are imputed based on the nearest neighbors' values, ensuring the integrity and completeness of the data.
- **Removing Duplicate Records:** Duplicate records are identified and removed to maintain data quality and avoid redundancy.
- **Normalizing Numerical Features:** StandardScaler is used to normalize numerical features, ensuring that they have a mean of 0 and a standard deviation of 1, which helps in improving the performance of machine learning models.
- **Encoding Categorical Variables:** OneHotEncoder is employed to convert categorical variables into a format that can be provided to machine learning algorithms to do a better job in prediction.

Data Transformation:

- **Normalization:** Ensuring numerical features are scaled to a standard range.
- **Encoding:** Converting categorical variables into numerical formats using techniques like OneHot Encoding.

Feature Engineering:

- **Creating New Features:** Deriving new features from existing ones to better capture the underlying patterns in the data.
- **Selecting Relevant Features:** Using techniques to identify and retain features that have the most significant impact on the target variable, which improves model accuracy and reduces complexity.

Scalability:

- **PySpark Integration:** Leveraging PySpark for large-scale data processing and analysis. PySpark enables distributed computing, making it possible to handle and process large datasets efficiently.

Correlation Analysis:

- **Generating Correlation Heatmaps:** Visualizing the correlation between features and the target variable before and after dropping low-correlation columns. This step helps in understanding which features are most impactful.
- **Dropping Low-Correlation Columns:** Simplifying the model by removing features that have a correlation below a predefined threshold (e.g., 0.1) with the target variable.

Modeling:

- **Random Forest:** Implementing a Random Forest classifier to predict income categories by combining the predictions of multiple decision trees.
- **Support Vector Machine (SVM):** Using SVM to find the hyperplane that best separates the income categories.
- **XGBoost Classifier:** Applying the XGBoost algorithm for efficient and accurate classification.
- **Adaboost Classifier:** Using Adaboost to boost the performance of a weak classifier.
- **Logistic Regression:** Training a logistic regression model to predict the probability of an individual's income exceeding $50K/year.
- **Decision Trees:** Using decision tree algorithms to learn patterns and make predictions based on the trained data.

Insight Generation:

- **Analyzing Model Outputs:** Extracting actionable insights from the predictions made by various models. This involves understanding the importance of different features and their impact on the target variable.
- **Web-Based Insights:**Upon CSV file upload, the web interface prompts the user to select two parameters. Based on the selected parameters, the system generates insights, including graphical representations and basic computational analyses.

The website, hosted on PythonAnywhere, can be accessed at insightgenerator.pythonanywhere.com. The backend of this web application is developed using Flask, a lightweight WSGI web application framework in Python.

Key Features:

1. **CSV File Upload**:
   - Users can upload CSV files through a user-friendly interface.
   - The application handles file parsing and data extraction to ensure smooth processing.
2. **Parameter Selection**:
   - After uploading a CSV file, users can select specific parameters for analysis.
   - The interface dynamically updates based on the data in the uploaded file, allowing for customizable insights.
3. **Insight Generation**:
   - The application processes the selected parameters and generates insights.
   - Insights are displayed in the form of graphs, making data visualization intuitive and easy to understand.

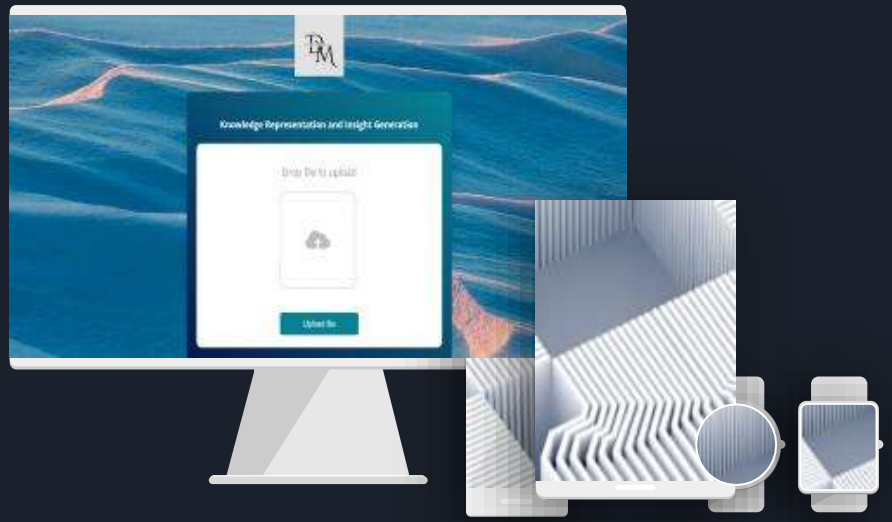
Technical Stack for the website:

- **Backend**: Flask framework is used to manage server-side logic, handle user requests, and interact with the file system.
- **Frontend**: HTML, CSS, and JavaScript are used for creating a responsive and interactive user interface.
- **Hosting**: The website is hosted on PythonAnywhere, a cloud-based platform that supports Python web applications.

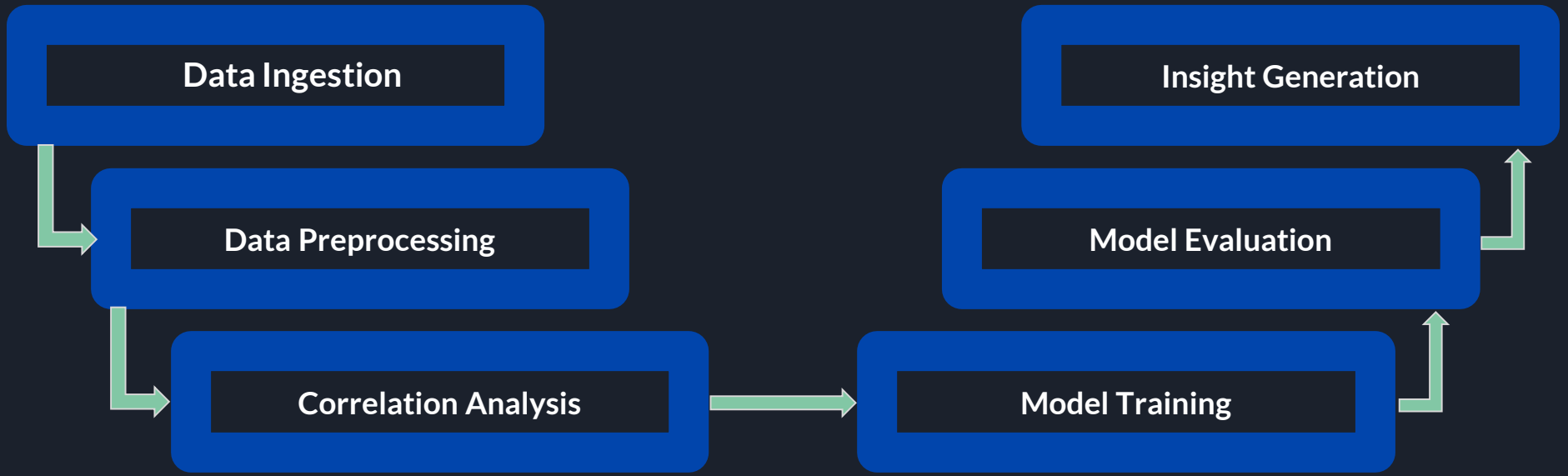
Performance Considerations:

- **Load Time**: The website might take 1-2 minutes to load initially, a common characteristic of applications hosted on shared cloud platforms like PythonAnywhere.
- **Data Handling**: Efficient data handling mechanisms are in place to process CSV files and generate insights without significant delays.

This web application serves as a powerful tool for users to upload their data, customize their analysis, and visualize results seamlessly.



Architecture Diagram



Architecture Diagram

1. **Data Ingestion:**
   - Load the dataset using PySpark to create a distributed DataFrame, enabling scalable data processing.
2. **Data Preprocessing:**
   - Perform data cleaning (handling missing values, removing duplicates).
   - Normalize numerical features and encode categorical variables.
   - Engineer new features and select relevant ones.
   - Split the dataset into training and test sets.
3. **Correlation Analysis:**
   - Generate initial and refined correlation heatmaps to visualize feature relationships.
4. **Model Training:**
   - Train multiple machine learning models (Random Forest, SVM, XGBoost, Adaboost, Logistic Regression, Decision Trees).
5. **Model Evaluation:**
   - Evaluate model performance using metrics like accuracy, precision, recall, and F1 score.
6. **Insight Generation:**
   - Analyze model outputs to extract actionable insights.

Team Contribution

Name: Sai Manikanta Patro

1. Data Cleaning:

- Handled missing values by replacing '?' with NaN.
- Imputed numerical features with mean and categorical features with mode/forward fill.

2. Knowledge Representation:

- Used Matplotlib, Pandas, and Seaborn for visualizations.
- Created scatter plots, bar graphs, box plots, histograms between various parameters.

3. Model Training & Evaluation:

- Implemented SVM, XGBoost, and AdaBoost with hyperparameter tuning.
- Evaluated models using accuracy, precision, recall, F1 score, and confusion matrix.

4. **Ensuring Scalability:**

- Utilized PySpark for scalable data processing and analysis.
- Converted data to Pandas for graph generation and model training.

5. Web Designing:

- Developed a Flask app for CSV upload and insight generation.
- Deployed the app on PythonAnywhere: insightgenerator.pythonanywhere.com

Name: Lohith Konchada

1.Logistic Regression Development

- Developed the logistic regression model for predictive analysis within the project.
- Conducted data preprocessing and feature selection to enhance model accuracy.

2. Deployment Assistance

- Contributed significantly to the deployment process of the web application on PythonAnywhere.

3. Version Control and Documentation

- Assisted in pushing the project code to GitHub, ensuring proper version control and collaboration.
- Created detailed reports and PowerPoint presentations to summarize findings and methodologies.

Name: Shreya Allupati

1. Knowledge Representation:

- Used Matplotlib, Pandas, and Seaborn for visualizations.
- Created scatter plots, bar graphs, box plots, histograms between various parameters.
- Summarized the generated plots manually for manual insights.
- Used Correlation analysis to discover if there is a relationship between two parameters, and how strong that relationship might be.

2. Model Training and Evaluation:

- Implemented Random Forest Classifier and done Hyper Parameter Tuning using Grid Search CV to create the best fit Random Forest model
- Evaluated models using accuracy, precision, recall, F1 score, and confusion matrix.

3. Web Designing:

- Frontend part of the web design using HTML and CSS. The web application allows users to upload a CSV file

Name:Kunjal Grover

1.Model Training and Evaluation

- The Decision tree algorithm was used to recognize a pattern and then predict a value for a desired attribute by taking into account other factors or attributes.
- Metrics such as accuracy, precision, recall, and F1 score were also calculated to evaluate the model.

2.Insight Generation

- GPT2 model for text generation as well as sshleifer/distilbart-cnn-12-6 for summarization was used to generate textual insights and to explain the statistics of the data.
- Both numerical and catagorical data was identified and insights were generated to help explain its significance as well as understand the data distribution, central tendencies, and variability,

3.Data Preprocessing:

- Data normalization was performed using standard scaler, itss purpose is to normalize features by removing the mean and scaling to unit variance. It ensures datasets of different scales are treated equally in the ML algorithm.

Technologies Used

1. **Pandas:** For data manipulation and analysis, providing powerful data structures to handle structured data effectively.
2. **NumPy:** For numerical operations, offering a comprehensive library for mathematical functions and operations.
3. **Scikit-learn:** For data preprocessing (KNNImputer, StandardScaler, OneHotEncoder) and implementing machine learning models.
4. **Scipy:** For statistical functions and operations.
5. **PySpark:** For large-scale data processing, enabling distributed computing and efficient handling of large datasets.
6. **Matplotlib/Seaborn:** For data visualization, helping in generating plots and heatmaps to visualize data relationships and model outputs.
7. **Flask**:The backend part of the website is mainly built using flask.

Conclusion

This project outlines a comprehensive approach to processing and analyzing the Adult Income Dataset. By leveraging advanced data preprocessing techniques and multiple machine learning models, the solution aims to generate meaningful insights that can support decision-making processes. The use of PySpark ensures scalability, making it feasible to handle large datasets efficiently. The solution's unique approach lies in its thorough data preprocessing, robust model training and evaluation, and insightful analysis of model outputs, providing a complete framework for knowledge representation and insight generation from structured datasets.