CAPSTONE PROJECT OVERVIEW

Project Objective

The goal of this capstone project is to provide students with hands-on experience in end-to-end data science workflows, from identifying and cleaning real-world datasets to building, fine-tuning, and deploying machine learning models. Students are expected to:

- Identify and work with a dataset relevant to their field of study.
- Perform data preprocessing, including cleaning and handling missing values, where applicable.
- Build, evaluate, and fine-tune machine learning models.
- Deploy the final model using a local deployment and a user-friendly web interface such as Streamlit.

Dataset Requirements

To ensure students engage with real-world data and are able to experience the challenges of data cleaning, feature engineering, and modeling, the dataset they select should meet the following criteria:

1. **Source**: The dataset should be obtained from open data sources such as Kaggle or other publicly available platforms.
2. **Size**: The dataset must contain at least 2,000 entries but not exceed 10,000 entries. Larger datasets are acceptable if manageable within the project timeline.
3. **Originality**: Students should select a dataset that is not widely used or well-known in common tutorials, as this will enhance learning through problem-solving rather than mimicking existing solutions.
4. **Data Quality**: The dataset should ideally require data cleaning, including handling missing values or inconsistencies. This will enable students to practice preprocessing techniques.
5. **Model Performance**: The performance of models trained on the dataset should not exceed 97% in order to allow for observable improvements during the model fine-tuning process.
6. **Imbalanced Data**: It is preferable if the dataset is imbalanced, as this provides an opportunity to apply resampling techniques and understand the challenges of working with skewed datasets.

Capstone Project Structure

1. Introduction

1.1 Project Information

This section should provide an overview of the project, including the problem statement and the goals the students aim to achieve through their analysis. Students are required to clearly articulate the business problem or research question their model aims to solve.

1.2 Dataset Description

Students must introduce and describe the dataset they are working with, including its source, the number of entries and features, and the type of data it contains (e.g., tabular, time series, etc.).

1.3 Column Descriptions

Provide a comprehensive description of each column in the dataset:

- **Target Variable**: The outcome the model will predict.
- **Feature Variables**: Detailed descriptions of the predictor variables and their data types.

2. Preprocessing

2.1 Data Cleaning

Students must document the cleaning process, detailing any issues they encountered, such as:

- Duplicate entries
- Inconsistent data formats (e.g., date formats or case sensitivity)
- Data errors (e.g., incorrect entries such as negative values where inappropriate)

2.2 Missing Value Analysis

Students should identify and address missing values in the dataset. The analysis should include:

- The percentage of missing values for each feature.
- The approach used to handle missing values.

2.3 Outlier Analysis

Students should perform outlier detection and analysis to identify unusual data points and determine whether to remove or transform them.

2.4 Feature Engineering

If applicable, students should apply feature engineering techniques, including:

- **New Feature Creation**: Generate new features that could enhance model performance.
- **Transformation**: Apply necessary transformations (e.g., logarithmic scaling) to improve linearity or model fit.

3. Exploratory Data Analysis (EDA)

3.1 Data Visualization

Students should use visual tools to explore and understand the data. Suggested visualizations include:

- Histograms and bar plots to explore feature distributions.
- Boxplots to examine outliers and data spread.
- Scatterplots to explore relationships between features.
- Heatmaps to visualize correlations between features.

3.2 Correlation Analysis

Students should conduct a correlation analysis to identify multicollinearity or strong relationships between variables.

4. Scaling, Categorical Variables, and Splitting

4.1 Scaling

Students are expected to apply feature scaling to normalize or standardize features, especially when using algorithms that are sensitive to feature magnitude (e.g., SVM, KNN). Techniques such as StandardScaler or MinMaxScaler should be applied.

4.2 Encoding Categorical Variables

All categorical variables must be appropriately encoded to ensure compatibility with machine learning algorithms. Students should apply one-hot encoding or label encoding depending on the variable type.

4.3 Splitting the Data

The data should be split into training and test sets to evaluate model performance. Students should ensure that the data is split in a way that prevents data leakage. For imbalanced datasets, students may also consider stratified sampling.

5. Models

5.1 Creating Models and Initial Results

Students will create multiple machine learning models, evaluating the performance of each using relevant metrics. Students must document the initial model results, including metrics such as RMSE, accuracy, precision, recall, F1-score, and AUC-ROC for classification tasks.

5.2 Fine-Tuning and Hyperparameter Optimization

Once baseline models are created, students will fine-tune model performance by optimizing hyperparameters.

5.3 Model Comparison

Students should compare all models based on their performance metrics. Graphical representations such as bar charts or line graphs should be used to highlight performance differences.

5.4 Feature Importance

For models that support it (e.g., tree-based models), students should analyze and report the most important features.

5.5 Final Model

Based on model comparison, students should select the final, best-performing model. This model will be used in the subsequent steps, including deployment.

5.6 Create a Model with Fewer Features (if needed)

If it is possible and necessary, students can create models with less features according to feature importance results.

5.7 Pickle the Model

Once the final model is trained and selected, students should serialize and save the model using the pickle library for deployment.

6. Deployment

6.1 Local Deployment

Students are required to deploy their model locally.

6.2 Deployment with Streamlit

To make the model more accessible, students should create a web application using Streamlit. The application should allow users to input values and receive predictions.

Conclusion

At the end of the project, students must present a summary of their findings, highlighting key learnings from the data cleaning, modeling, and deployment phases. This reflection should include challenges faced and how they were overcome, as well as suggestions for potential improvements or extensions of the project.