Usecase 8 - (Project 5)

By: eng. Esraa Madhi

Leveraging your expertise in machine learning, you should develop an appropriate model for the given dataset.

This project must at least satisfy the following minimum requirements:

Usecase 8

For any **Data project** we should go through these steps:

Step 1: Defining the Problem Statement

When browsing websites, it's common to encounter data that could be valuable for building models or conducting exploratory data analysis (EDA) to glean insights. Identify a specific issue for which you would need to scrape data.

If you haven't yet defined a specific problem for which you'd need to scrape data, consider this scenario:
Riyadh is known for its wide variety of dining options. The goal is to create a model that categorizes these restaurants from a unique perspective, considering attributes such as ratings, the number of ratings, geographic location, and the type of cuisine offered.

Step 2: Collecting Data

We lack a compiled dataset at the moment 😁. Please consider using web scraping method to extract the needed data from any platform to solve your problem.

In case you have consider the above scenario, You can extract restaurants data from any platform, for example, Google Maps.

Bounce: Upload your dataset to Kaggle platform to be your first uploaded dataset, Yay 😁

Step 3: Data Quality Checking and Remediation

Step 4: Exploratory Data Analysis

- For these two steps, make sure to do:
    a. Data Profiling: apply the 7 types of data profiling
    b. Data Cleaning: handle missing values, correcting errors, and dealing with outliers.
    c. Univariate Analysis &Bivariate/Multivariate Analysis: to understand their distribution and look at the relationships between variables. For your visualizations make sure to:
        - Drive meaningful insights that would help you in building models (at least 3 different charts).

Step 5: Building Machine Learning Models

Step 6: Model Evaluation

During training model and evaluation, make sure to do:

- Selection of Models: Choose at least two different unsupervised machine learning models to train on your dataset. Examples include DBSCAN, kmeans and so on.
- Feature Engineering: Apply feature engineering techniques to create new features or modify existing ones to improve model performance such as :
    - Encode categorical variables
    - Normalize or standardize numerical features.
- Model Training: Train your selected models on the training dataset.
- Hyperparameter Tuning: Fine-tune the hyperparameters of each model to optimize performance.
- Performance Metrics: Use appropriate performance metrics to evaluate the models.
- Model Comparison: Compare the models based on their performance metrics to determine which model performs the best on your dataset.

Step 7: Communicating Results

Document the methodology and conclusions from your model training experience in a one-page README markdown file comprising the following sections:

- Team Members: List all individuals who contributed to the project.
- Introduction: Briefly describe the problem being addressed and the goals of the project.
- Dataset Synopsis and Origin: Provide a summary of the dataset utilized and its source.
- Model Selection: Train at least 2 different unsupervised machine learning models for the task.
- Feature Engineering: Outline the steps taken to manipulate or create new features to improve model performance.
- Hyperparameter Optimization: Explain the process and methods used to fine-tune model hyperparameters.
- Performance Metric Visuals: Include charts or graphs that illustrate the performance of each model across various metrics.
- Best Model Determination: Explain the criteria for selecting the best-performing model.

- Feature and Prediction Insights: Offer an interpretation of how different features influence the model's predictions.

Implement the deployment of our most proficient model as follows:

- Construct a FastAPI endpoint to serve the model
- Develop a Streamlit application that features:
  a. A visualization displaying the results of the model’s clustering.
  b. A user interface that interacts with the established FastAPI endpoint to classify new data points into clusters.

Step 9 : Model Performance Maintenance in Production

Not applicable

Note: the red steps means they are Not applicable in the project