Project on Anomaly Detection

Problem Statement

Artifact Submission

Recommendation

Data Description

Challenge Evaluation Criteria

Vibelines & Bounties

Problem Statement - Identifying Outlier Reviewers on TripAdvisor

Goal

The primary goal of this project is to apply **unsupervised anomaly detection and density-based clustering techniques** to the TripAdvisor user ratings data. The aim is to **isolate user profiles** whose rating patterns significantly deviate from the majority, which may indicate either highly passionate experts or, potentially, fraudulent or non-genuine reviewers.

Dataset : TripAdvisor User Ratings Data , The dataset summarizes user behavior by providing the **average rating** given by each user across key categories:

avg_museum_rating
avg_park_rating
avg_restaurant_rating
avg_nightlife_rating

Objectives

Anomaly Detection with Isolation Forest: Apply the **Isolation Forest** algorithm to the multi-dimensional rating data.

- Identify and flag the top 1% of users who are classified as **outliers** (anomalies).
- Analyze the rating patterns of these flagged users to hypothesize why they are considered anomalies (e.g., users who rate *everything* a '1' or users who rate *everything* a '5').

Density-Based Clustering with DBSCAN: Apply **DBSCAN** to the dataset.

- Determine appropriate hyperparameters (Epsilon and Min_Samples) to identify dense clusters of "normal" users.
- The users **not assigned to any cluster** (labeled as noise by DBSCAN) will be treated as the second set of potential anomalies.

Comparative Analysis: Compare the lists of anomalies identified by Isolation Forest and the "noise" identified by DBSCAN.

Deliverable

A report detailing the parameters chosen for both models, a comparison of the anomalous users identified by each technique, and actionable insights for TripAdvisor regarding the characteristics of outlier reviewers.

Expected Outcome of the project

The outcome of this project will be the identification of TripAdvisor users whose review rating patterns significantly deviate from the norm within the dataset. This insight can be valuable for:

- **Detecting potential fake reviews or spammers:** Users with highly unusual or extreme rating patterns might warrant further investigation for fraudulent activity.
- **Identifying highly opinionated users:** Flagging users who consistently rate much higher or lower than the average, which might affect overall venue ratings.
- **Understanding niche preferences:** Discovering users with very specific or unusual interests that don't align with mainstream preferences, potentially for targeted marketing or content curation.
- **Improving data quality:** Pinpointing potential data entry errors or inconsistencies in the review collection process.

Data Description

About the Dataset:
The dataset provided summarizes user ratings across different categories of attractions, likely aggregated from individual reviews.

Column Name	Description
user_id	Unique identifier for each user.
avg_museum_rating	Average rating given by the user for museums.
avg_park_rating	Average rating given by the user for parks.
avg_restaurant_rating	Average rating given by the user for restaurants.
avg_nightlife_rating	Average rating given by the user for nightlife venues.

Artifact Submission

Your submission must include the following five artifacts, all packaged within a single GitHub repository.

1. Jupyter Notebook (.ipynb) (Add separate ipynb files for Isolation Forest and DBSCAN)

This is the core of your submission. Your Jupyter Notebook should be a complete, well-documented narrative of your data analysis journey. It must include:

- **Detailed Explanations:** Use Markdown cells to explain your thought process, the questions you are trying to answer, and the insights you've uncovered.
- **Clean Code:** The code should be well-structured, easy to read, and free of unnecessary clutter.
- **Comprehensive Comments:** Use comments to explain complex logic and the purpose of different code blocks.
- **Key Visualizations:** All visualizations should be clear, properly labeled, and directly support your findings.

2. Presentation (.pptx or .pdf)

Create a compelling presentation that summarizes your team's analysis and key findings. This presentation should serve as your final pitch. It must include:

- **Executive Summary:** A concise overview of your findings.
- **Key Insights:** The most important takeaways from your analysis.
- **Data-Driven Recommendations:** Actionable steps that can be taken based on your insights.
- **Supporting Visualizations:** A selection of your best visualizations to illustrate your points.

3. README File (.md)

The README file is the first thing we'll look at. It should serve as a quick guide to your project and provide essential details. It must include:

- **Project Title :**
- **Brief Problem Statement:** A summary of the project and your approach.
- **Summary of Findings:** A bullet-point summary of your most significant insights.

4. Attached Dataset

Please include the original dataset (.csv or other format) within your repository. This ensures the judges can reproduce your analysis without any issues.

5. GitHub Repository

Your final submission will be your GitHub repository. The repository name **must follow this exact format:**
Anomaly_Detection_ProjectName_TMP

Challenge Evaluation Criteria

Criteria Name	Criteria weight
Data Understanding and Exploratory Data Analysis	20%
Data preprocessing and feature engineering	25%
Model building and evaluation	30%
Business Recommendation	15%
Coding guidelines and standards	10%

Recommendation for Isolation Forest Implementation

1. **Data Preprocessing:**

    Selecting only the numerical columns representing average ratings (avg_museum_rating, avg_park_rating, avg_restaurant_rating, avg_nightlife_rating).

    While Isolation Forest is less sensitive to feature scaling than distance-based methods, it's generally good practice to consider it, especially if the ranges of the ratings vary significantly or if there are extreme outliers that could disproportionately influence tree splits.

2. **Isolation Forest Implementation:**

    Applying the IsolationForest algorithm to the average rating data.

    Tuning the contamination parameter (e.g., setting it to a small percentage like 0.01 or 0.05) to define the expected proportion of anomalies in the dataset. This helps the algorithm set an appropriate threshold for anomaly scores.

3. **Anomaly Identification:**

    The model will output an anomaly score for each user.

    Based on the contamination parameter, users will be classified as either "normal" or "outlier" (anomaly).

4. **Analysis and Interpretation of Anomalies:**

    Investigating the characteristics of the users flagged as anomalies. For example, an anomalous user might have:

    Extremely high or low ratings across all categories consistently.

    A rating profile that is highly inconsistent (e.g., very high for museums but very low for restaurants, when most users show some correlation).

    A rating pattern that doesn't fit any common profile observed in the majority of users.

    Understanding *why* these users are considered anomalous based on their specific rating patterns.

5. **Visualization (Optional but Recommended):**

If the data is reduced to 2D or 3D (e.g., using PCA or t-SNE) before or after applying Isolation Forest, the normal points and identified anomalies can be visualized, making the outliers visually apparent as distinct points.

Recommendation for DBSCAN Implementation

1. Data Preprocessing:
    - Selecting only the numerical columns representing average ratings (avg_museum_rating, avg_park_rating, avg_restaurant_rating, avg_nightlife_rating).
    - **Crucially, performing feature scaling** on these rating columns (e.g., using StandardScaler). This is essential for DBSCAN, as it is a distance-based algorithm, and unscaled features can lead to biased distance calculations if rating scales vary or if one category has inherently wider rating distributions.
2. DBSCAN Implementation:
    - Applying the DBSCAN algorithm to the scaled average rating data.
    - **Careful tuning of the eps and min_samples hyperparameters** will be critical. The choice of these parameters will directly influence what is considered a "dense region" of normal user rating behavior and, thus, what points are identified as "noise" (anomalies).
3. Anomaly Identification:
    - DBSCAN will assign a cluster label to each data point. Users labeled as -1 are identified as noise points or outliers. These are the anomalies.
4. Analysis and Interpretation of Anomalies:
    - Investigating the characteristics of the users flagged as anomalies. For example, an anomalous user might have:
        - Extremely high ratings across all categories compared to others.
        - Extremely low ratings across all categories (a "negative reviewer").
        - Highly inconsistent ratings (e.g., very high for museums but very low for restaurants, when most users show some correlation).
        - A rating profile that doesn't fit any common pattern.
    - Understanding *why* these users are considered anomalous based on their specific rating patterns.
5. Visualization (Optional but Recommended):
    - If the data is reduced to 2D or 3D (e.g., using PCA or t-SNE) before or after applying DBSCAN, the clusters of normal users and the identified noise points can be visualized, making the anomalies visually apparent as isolated points.

VIBELINES

Kickoff Vibes - T
Wrapoff Vibes - T + 7
Spotlight Vibes - T + 14

Lets Go