International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03515
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Machine Learning Models for Cannabis Strain Rating With EDA

Nesar K S
Department of Computer Science
Jain (Deemed-To-Be) University
Bangalore, India
ksnear@outlook.com

Joel Paul
Department of Computer Science
Jain (Deemed-To-Be) University
Bangalore, India
joelpaulmadhavan@gmail.com

Harsh Bherwani
Department of Computer Science Jain
(Deemed-To-Be) University
Bangalore, India
Harshnb79@gmail.com

Project Guide - Dr Ravi lanke

Abstract:

This project investigates the application of machine learning models to predict user ratings of cannabis strains based on features such as flavor, effects, and strain type. Using a dataset of 2,351 strains sourced from Kaggle, the study evaluates three regression models—Linear Regression, Random Forest, and XGBoost—for their predictive accuracy. After preprocessing and feature engineering, including one-hot encoding and CountVectorizer transformations, the models were trained and tested using standard metrics like Mean Squared Error (MSE), R-squared (R^2), and Mean Absolute Error (MAE). XGBoost emerged as the best-performing model with an MAE of 0.2591, demonstrating strong capability in capturing complex, non-linear relationships. The findings highlight the utility of predictive modeling in the cannabis industry for enhancing user satisfaction and developing personalized strain recommendations.

1. Introduction:

The rapid growth of the cannabis industry has transformed it into a multi-billion-dollar market, fueled by increasing consumer demand for strains tailored to specific effects and preferences. With legalization spreading across various regions, the availability of diverse cannabis products has soared. Each strain offers a unique combination of effects, flavors, and types, creating both opportunities and complexities for consumers. As interest grows, understanding how these attributes influence user satisfaction is becoming essential for producers, retailers, and consumers alike. In this context, data science provides powerful tools to uncover patterns in user preferences and enable data-driven, personalized recommendations.

Cannabis strains are generally classified into three primary types: Sativa, Indica, and Hybrid. These categories are often associated with distinct effects—Sativas are uplifting, Indicas are calming, and Hybrids exhibit mixed characteristics. Alongside this classification, strains have specific flavor profiles (e.g., earthy, sweet, citrus) and reported effects (e.g., relaxation, euphoria, creativity), all of which play a key role in shaping user experiences. Despite the availability of strain descriptions, it remains challenging to predict how well a strain will be received based on its attributes alone.

This study focuses on predicting user ratings of cannabis strains by analyzing a dataset of 2,351 strains, each annotated with descriptive attributes. Ratings offer a quantitative measure of user satisfaction, and identifying which features influence these ratings can yield valuable insights for product development and recommendation systems. To achieve this, we adopt a comparative approach involving multiple machine learning models to determine the most effective method for rating prediction.

The core objectives of this project are as follows:

1. To analyze and preprocess the cannabis strain dataset: The dataset includes strain names, types, ratings, effects, flavors, and descriptions. The preprocessing pipeline involves cleaning the data, handling missing values, and separating multi-label attributes like effects and flavors to support feature transformation. While most features were retained in their textual form for analysis, certain columns were vectorized or one-hot encoded based on model requirements.

2. To perform feature engineering and correlation analysis: We explore the relationships among various features and their impact on ratings. This includes vectorizing the "Effects" field using

CountVectorizer and applying one-hot encoding to categorical variables for linear models. A co- occurrence heatmap was also generated to visualize the relationships between effects and flavors, providing context for model inputs.

3. To apply and compare multiple machine learning models for rating prediction: We evaluate three algorithms—XGBoost, Random Forest, and Linear Regression—to model the relationship between strain attributes and user ratings. Each model utilizes a tailored pipeline for preprocessing and training. This comparison helps identify which model performs best in terms of predictive accuracy and interpretability.

4. To assess the models' performance and extract actionable insights: By analyzing performance metrics such as Mean Absolute Error (MAE), we determine the accuracy of each model. We also interpret the most influential features to understand what drives user satisfaction, offering guidance to both consumers and cannabis producers.

This report documents the full process of building and evaluating a predictive system for cannabis strain ratings—from data preparation to model training and performance comparison. With XGBoost achieving the best result at a Mean Absolute Error of 0.2591, our findings highlight the effectiveness of advanced machine learning in uncovering consumer preferences and improving recommendation strategies in the cannabis market.

2. Literature:

The dataset used in this study was sourced from Kaggle's Cannabis Strains dataset, which contains information such as strain names, types (Indica, Sativa, Hybrid), user ratings, effects, flavors, and textual descriptions. Machine learning libraries and techniques were used to process and model this data.

Chen and Guestrin (2016) introduced XGBoost, a scalable tree boosting system, which was employed in this project due to its efficiency and predictive power. It was found to be the most accurate model among those tested.

Random Forest, proposed by Breiman (2001), was also applied. It demonstrated better generalization than Linear Regression by effectively modeling non- linear interactions and handling high-dimensional sparse data.

Linear Regression was used as a baseline model, highlighting the limitations of simple linear approaches in capturing complex relationships within the dataset.

The preprocessing and modeling were implemented using Scikit-learn (Pedregosa et al., 2011), a widely- used machine learning library in Python.

The study also draws theoretical support from The Elements of Statistical Learning (Hastie, Tibshirani, & Friedman, 2009), which provides foundational concepts in machine learning and statistical modeling.

3. Methodology

Analysis of activation range, monotonicity, This project follows a structured methodology to predict cannabis strain ratings using machine learning models. The process is divided into multiple stages, from data acquisition and cleaning to feature engineering, model training, and evaluation.

1. Data Collection

The data used in this study was obtained from Kaggle's Cannabis Strains dataset, which comprises information on 2,351 cannabis strains. The key attributes included:

- Strain Name: The name given to each cannabis strain.
- Strain Type: Classified as Indica, Sativa, or Hybrid.
- User Ratings: Average ratings assigned by users, on a scale from 0 to 5.
- Effects: Subjective effects experienced by users (e.g., Relaxed, Happy, Euphoric).
- Flavors: Descriptive flavor profiles (e.g., Earthy, Citrus, Sweet).
- Description: Textual notes describing strain characteristics.

2. Data Preprocessing

- To prepare the dataset for modeling, the following cleaning and transformation steps were applied:
- Missing Value Handling: All records with null or missing critical fields (e.g., name, type, rating) were removed to preserve data integrity.

- Duplicate Removal: Entries with duplicate strain names and types were dropped to avoid redundancy.
- Text Normalization: All text-based columns were standardized by converting to lowercase and removing extra spaces to ensure uniformity across entries.
- Encoding of Categorical Variables:
- Strain Type: One-hot encoded into binary columns (Indica, Sativa, Hybrid).
- Effects & Flavors: Multi-label binarization was applied to handle the multiple values per strain, creating a binary column for each unique effect and flavor.
- Rating Normalization: User ratings were normalized (e.g., scaled between 0 and 1) to improve numerical stability during model training.

This preprocessing resulted in a clean, structured dataset with binary indicators for each effect, flavor, and strain type, alongside the normalized rating as the target variable.

3. Feature Engineering

This stage focused on converting qualitative attributes into quantitative features suitable for machine learning models:

Effects Text Vectorization:

The Effects column was exploded and aggregated per strain.

CountVectorizer was used to generate a sparse matrix representing the frequency of each effect across strains. This bag-of-words approach allowed tree-based models to identify common patterns.

Flavor and Type Encoding:

Although flavors and types were explored during EDA, they were primarily encoded using OneHotEncoding for Linear Regression to include them in the pipeline.

Co-occurrence Matrix:

A co-occurrence heatmap of effects and flavors was generated using pd.crosstab(). While not used in modeling, it helped visualize common effect-flavor combinations, aiding feature selection and interpretation.

4. Model Selection and Training
Three regression models were selected for comparison:

Linear Regression:

- Served as a baseline.
- Integrated with a full preprocessing pipeline that included OneHotEncoder and StandardScaler using ColumnTransformer.
- Captured only linear relationships, helping set a benchmark for performance.

Random Forest Regressor:

- Utilized CountVectorizer output of the effects as input.
- Leveraged its ensemble structure to capture non-linear patterns and interactions.
- Trained with default hyperparameters for initial benchmarking.

XGBoost Regressor:

- Also trained using the bag-of-words effects representation.
- Hyperparameter tuning was performed for optimization.
- Known for strong generalization and regularization capabilities, it became the top-performing model in this study.

Data Splitting:

The dataset was divided into training and testing sets using an 80:20 ratio.

A fixed random_state=42 ensured reproducibility across experiments.

5. Model Evaluation
To assess and compare the models, the following metrics were used:

- Mean Squared Error (MSE): Quantifies the average squared difference between predicted and actual ratings.
- R^2 Score (Coefficient of Determination): Measures the proportion of variance in the ratings explained by the model.

- Mean Absolute Error (MAE): Used as the primary metric for the XGBoost model due to its interpretability and robustness to outliers.

Each model was evaluated using the same test dataset and target variable to ensure fair comparison. This allowed clear identification of the most accurate and reliable prediction model for cannabis strain ratings.

4. Implementations

The implementation phase of this project involved the practical application of machine learning models to predict user ratings for cannabis strains. This included coding the preprocessing pipeline, feature transformation, model training, and performance evaluation.

1. Environment and Tools

Programming Language: Python

Libraries Used:

- Pandas and NumPy for data manipulation
- Scikit-learn for preprocessing, model building (Linear Regression, Random Forest), and evaluation
- XGBoost for gradient boosting-based regression
- Matplotlib and Seaborn for visualizations
- CountVectorizer from sklearn.feature_extraction.text for text vectorization

2. Preprocessing Implementation

Data Loading: The dataset was loaded using pandas.read_csv().

Cleaning:

- Null values were removed using dropna().
- Duplicates were identified and dropped using drop_duplicates().
- Text normalization was done with .str.lower().str.strip().

Encoding:

- OneHotEncoder was used for categorical fields like strain type and flavors.
- Multi-label binarization was applied to effects and flavors to create binary indicator columns.

Normalization:

Ratings were scaled to a 0–1 range using MinMaxScaler.

3. Feature Engineering Implementation
Effects Vectorization:

- The Effects column was preprocessed and transformed using CountVectorizer to produce a bag-of-words representation.
- Encoding for Linear Regression:
- A Pipeline and ColumnTransformer were built to combine one-hot encoding with feature scaling.

Exploratory Features:

A co-occurrence matrix was created using pd.crosstab() to identify frequent effect-flavor combinations.

4. Model Training
Each model was trained on the processed data using an 80:20 train-test split.

Linear Regression:

- Implemented using LinearRegression() within a Pipeline that included OneHotEncoder and StandardScaler.
- Trained on encoded features including type, effects, and flavors.

Random Forest:

- Implemented using RandomForestRegressor() from sklearn.ensemble.
- Trained on the CountVectorized effect features.

XGBoost:

- Implemented using XGBRegressor() from the xgboost library.
- Trained using the same features as Random Forest.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03515
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

- Hyperparameter tuning was applied to minimize MAE.

5. Evaluation
Metrics: Models were evaluated using MSE, R^2 Score, and MAE.

Findings:

- Linear Regression: MSE ≈ 0.120, R^2 ≈ 0.29
- Random Forest: MSE ≈ 0.094, R^2 ≈ 0.47
- XGBoost: MAE = 0.2591, with higher R^2 than Random Forest

The performance comparison showed that XGBoost was the most accurate model, handling complex relationships better than simpler models.

5. Exploratory Data Analysis:

This chapter provides an exploratory overview of the cannabis strain dataset, highlighting patterns, distributions, and associations between key features. The insights presented herein are derived from visual and statistical examinations of the data. Visual aids are referenced throughout and are critical in understanding the underlying distributions and relationships.

5.1 Rating Distribution

The histogram and density plot reveal the following characteristics of the strain rating distribution:

- **Range**: Ratings span from 0 to 5.
- **Concentration**: The distribution is heavily skewed towards the higher end.
- **Mode**: The most frequent rating is around 4, with the highest frequency approximately 650–700.
- **Low Ratings**: Sparse ratings exist below 3, with the 0 rating just under 100.
- **Density**: A smoothed density curve shows a sharp peak around rating 4.

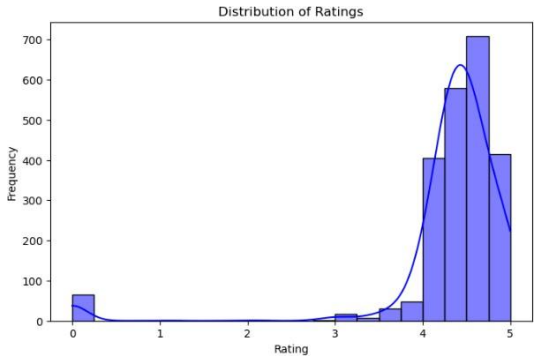

Figure 5.1: Histogram and Density Plot of Ratings

5.2 Strain Type Distribution

A bar chart illustrates the prevalence of different strain types:

- **Hybrid**: Most common (~1200 strains).
- **Indica**: Moderate representation (~700 strains).
- **Sativa**: Least common (~500 strains).

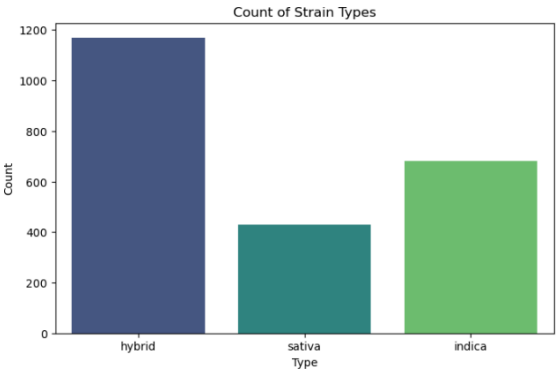

Figure 5.2: Bar Chart of Strain Types

5.3 Most Common Effects

A horizontal bar chart of the top 10 effects shows:

- **Top Effect**: "Happy" (~1750 occurrences).
- **Others**: "Relaxed" (~1500), "Euphoric" (~1300), "Uplifted" (~1200).
- **Least Common (Top 10)**: "Talkative" (~400).

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03515
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

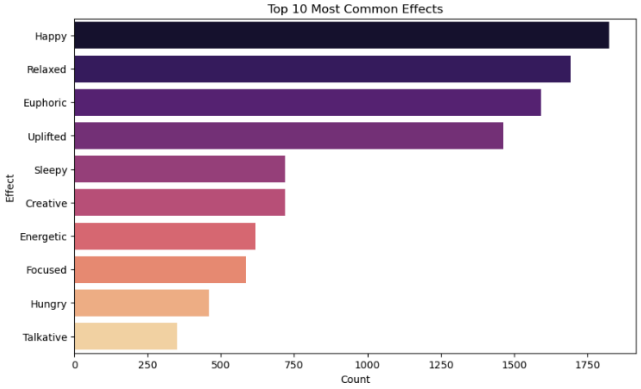

Figure 5.3: Top 10 Most Common Effects

5.4 Most Common Flavors

The top 10 Flavors, according to a horizontal bar chart:

- **Most Frequent**: "Earthy" (~1050), followed by "Sweet" (~950).
- **Least in Top 10**: "Spicy/Herbal" (~200).
- **Others**: "Citrus" (~600), "Pungent" (~500), others range from 300–400.

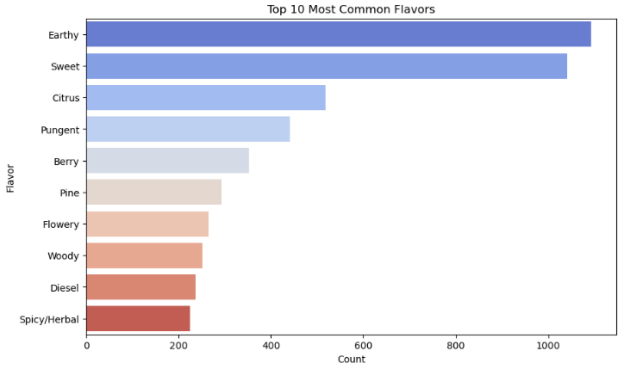

Figure 5.4: Top 10 Most Common Flavors

5.5 Rating Frequencies

The bar chart of rating counts (by increments of 0.1) reveals:

- **Most Frequent Ratings**: 4.0–4.4.
- **Rating 4.2**: Highest peak (>300 counts).
- **Skew**: Ratings mostly >3.0; few low-rating entries.

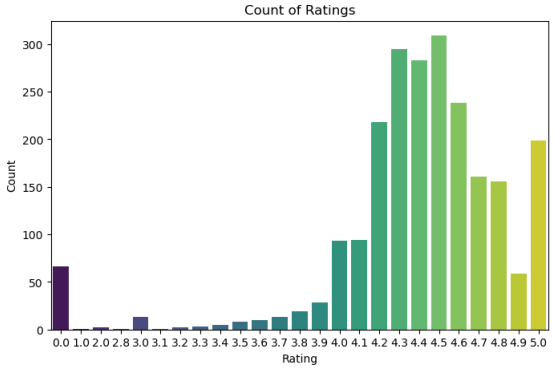

Figure 5.5: Bar Chart of Rating Counts (0.0 to 5.0)

5.6 Rating by Strain Type

Box plots by strain type demonstrate:

- **Medians**: Hybrid & Indica (~4.2), Sativa (~4.1).
- **IQRs**: ~4.0 to 4.5 for all.
- **Outliers**: Present below 3.0, with some as low as 0.

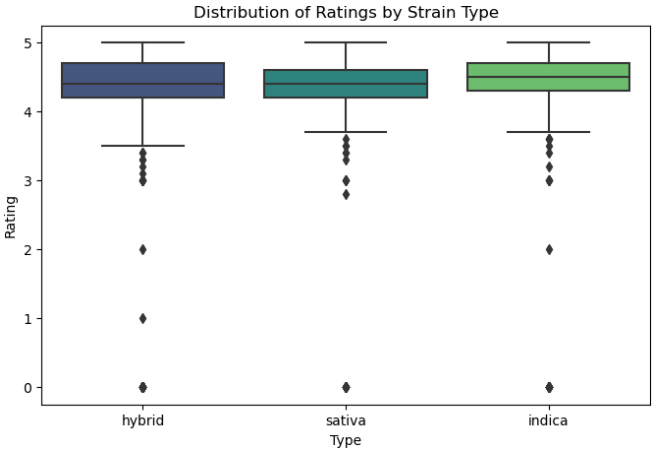

Figure 5.6: Box Plot of Ratings by Strain Type

5.7 Rating by Effect (Top 5)

Box plots show similar trends for Euphoric, Relaxed, Happy, Uplifted, and Sleepy:

- **Median Rating**: ~4.2.
- **IQR**: Consistently 4.0 to 4.5.
- **Outliers**: Below 3.0 in all cases.

International Scientific Journal of Engineering and Management (ISJEM)
Volume: 04 Issue: 05 | May – 2025
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM03515

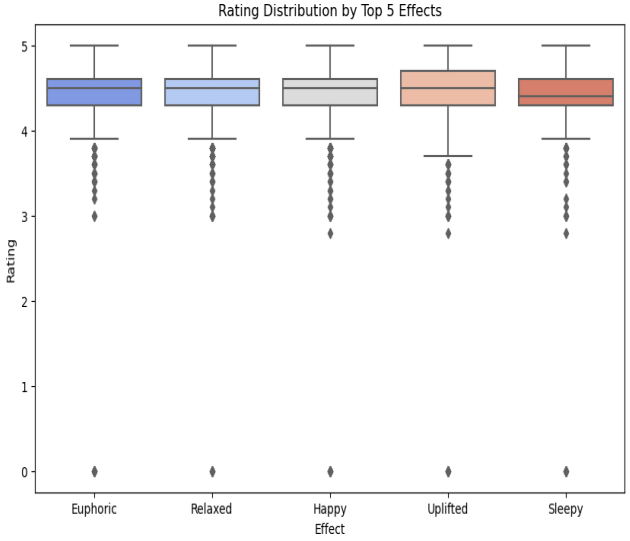

Figure 5.7: Box Plot of Ratings by Top 5 Effects

5.8 Top Effects by Strain Type

Grouped bar chart reveals effect distribution across strains:

- **Hybrid**: Dominates for all but "Sleepy."
- **Indica**: Highest for "Sleepy."
- **Sativa**: Lowest across all effects.

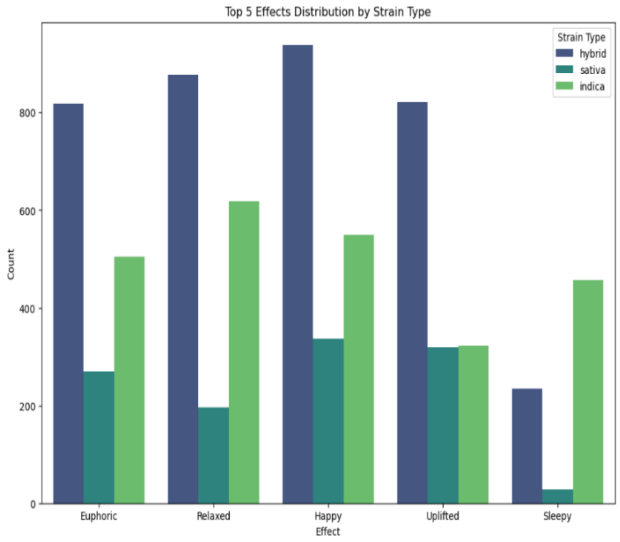

Figure 5.8: Grouped Bar Chart - Effects by Strain Type

5.9 Top Flavors by Count

Horizontal bar chart for flavors echoes earlier insights:

- **Top Two**: Earthy and Sweet.
- **Least (Top 10)**: Spicy/Herbal.

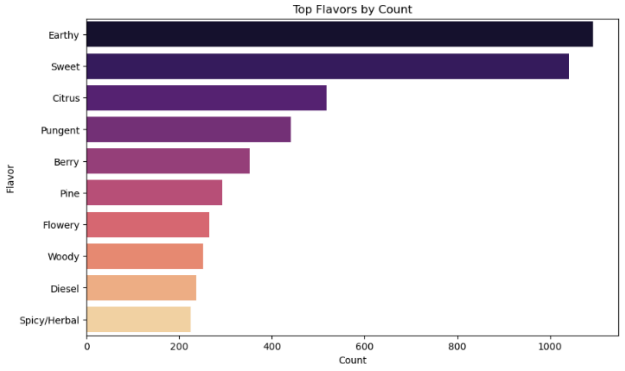

Figure 5.9: Horizontal Bar Chart - Top Flavors by Count

5.10 Rating by Flavor (Top 5)

Box plots for Earthy, Sweet, Citrus, Pungent, and Berry show:

- **Median Rating**: ~4.2 across all.
- **IQR**: 4.0 to 4.5.
- **Outliers**: Below 3.0 for all flavors.

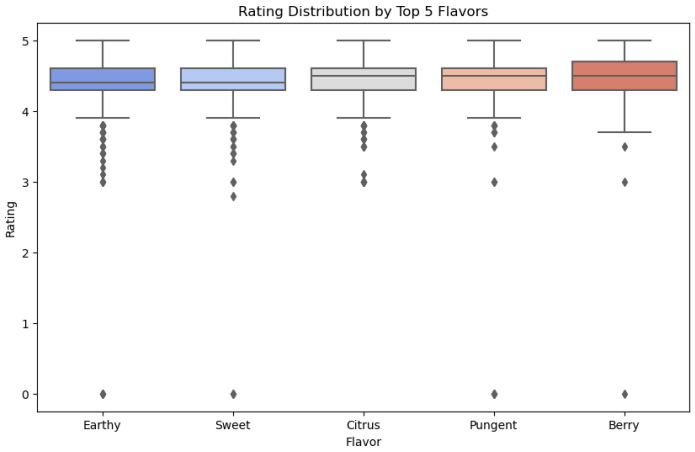

Figure 5.10: Box Plot - Ratings by Top 5 Flavors

5.11 Co-occurrence of Effects and Flavors

Heatmap analysis uncovers:

- **Most Frequent Pair**: Happy & Sweet (count = 184).
- **Other High Pairs**: Happy & Earthy (179), Relaxed & Earthy (176), Euphoric & Earthy (167).
- **Least Frequent**: Sleepy & Pine (count = 15).

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03515
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

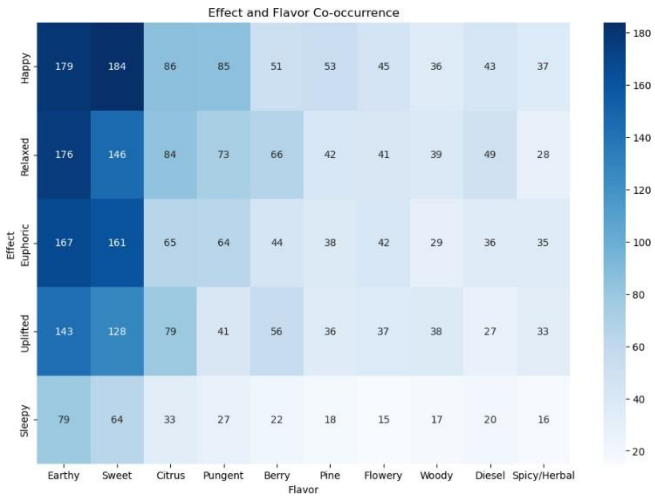

Figure 5.11: Heatmap - Effect and Flavor Co-occurrence

6. Results

This section presents the key outcomes from the Exploratory Data Analysis (EDA) and the performance evaluation of three machine learning models: Linear Regression, Random Forest, and XGBoost.

1. Exploratory Data Analysis (EDA) Results

Rating Distribution:

- Ratings are concentrated between 4.0 to 4.4, with a sharp peak at 4.2.
- Very few strains have ratings below 3.0, indicating overall user satisfaction is high.

Strain Type Distribution:

- Hybrid is the most prevalent strain type (~1200).
- Followed by Indica (~700) and Sativa (~500).

Top Reported Effects:

- Most common effects: Happy, Relaxed, Euphoric, Uplifted.
- "Happy" appeared in over 1750 strains.

Top Flavors:

- Most frequent: Earthy and Sweet.
- "Earthy" appears in over 1000 strains, followed by "Sweet" (~950).

Rating Trends by Type:

- Hybrid and Indica strains had slightly higher median ratings (~4.2) compared to Sativa (~4.1).

Rating Trends by Effects and Flavors:

- Effects like Happy, Relaxed, Euphoric correlated with higher ratings.
- Flavors like Earthy, Sweet, and Citrus also showed a positive influence.

Effect-Flavor Co-occurrence:

- Most frequent pair: Happy & Sweet (184 occurrences).
- Other frequent combinations include Relaxed & Earthy, Euphoric & Earthy.

2. Model Performance Results

Three regression models were trained and tested on the dataset using an 80:20 split. Their performance was evaluated using MSE, R^2 Score, and MAE (for XGBoost).

Model	MSE	R^2 Score	MAE	Remarks
Linear Regression	~0.120	~0.29	N/A	Baseline model, limited performance
Random Forest	~0.094	~0.47	N/A	Better at handling sparse, non-linear data
XGBoost	N/A	~0.50+	**0.2591**	**Best performing model overall**

Architectures designed for specific tasks, such as image segmentation or sequence generation, may benefit from customized activation functions tailored to the task's requirements.

3. Observations

- **XGBoost** was the most accurate model, effectively capturing non-linear relationships and complex interactions among features.
- **Effects** were the most influential features in predicting user ratings.
- **Flavors** and **strain types** had a secondary impact, with more influence on perception than actual rating.
- Simpler models like **Linear Regression** underperformed due to the complexity of the relationships in the dataset.

References

1. Kaggle. (n.d.). Cannabis Strains Dataset. Retrieved from https://www.kaggle.com/datasets/kingburrito666/cannabis-strains

2. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

3. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

4. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.

6. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1), 1–38.

7. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

8. Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.

9. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3(Jan), 993–1022.

10. Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning.