Pipeline

Raw Dataset
↓
Data Inspection & EDA
↓
Text Cleaning & Preprocessing
↓
Feature Engineering
↓
Rule-Based Fake Scoring
↓
Train–Test Split
↓
ML Model Training
↓
Prediction
↓
Evaluation Metrics
↓
Visualization & Analysis

1. load and inspect dataset

check columns, label,missing values, review length description
qns:
no of fake and real reviews
are rating extreme,etc --- *part of EDA*

2. Data cleaning and Preprocessing

lowercase,remove punctuation,numbers ,extra whitespaces, stopwords ,lemmatization, contraction-expansion

stopwords: _is, the, and, of,etc - it doesnot add meaning in classification
lemmatiation: running->run ,better->good,etc
- reduces vocab size and improves generalization
contraction-expansion : don't ->do not ,can,t -> can not
- improves sentiment analysis

Libraries used
`re` → Regular expressions (cleaning text)
`nltk` → Stopwords removal
`spaCy` → Lemmatization (advanced preprocessing)

```
can also remove
        html tags, URLs, emojis
```

Step	Method	Library
Lowercasing	`.lower()`	Python
Remove punctuation	Regex	`re`
Remove numbers	Regex	`re`
Remove extra spaces	Regex	`re`
Stopwords removal	Stopword list	`nltk`
Lemmatization	Base form conversion	`spaCy`

3. Feature Engineering

involves extracting meaningful numerical features from textual reviews that help distinguish fake reviews from genuine ones.
a. Review length - Fake reviews are often **unnaturally short** (generic praise) or **unusually long** (promotional content) . counts total no of characters after text cleaning

b. Word Count - Fake reviews tend to be **shallow** and lack detailed explanation.

c. Adjective Ratio - Fake reviews rely heavily on **emotional adjectives** such as *amazing, excellent, perfect* rather than factual descriptions.
Adjective Ratio=Number of Adjectives/Total Words

d. Sentiment Score - Overall polarity score of a review text.
Fake reviews usually show **extreme positivity** or **extreme negativity**, unlike genuine balanced opinions.
Apply sentiment analysis using tools like **VADER** and extract the compound score (range −1 to +1).
Example:
- Fake: +0.95 (Over-positive)
- Genuine: +0.35 (Moderate sentiment)

e .Generic Phrase Flag - Binary feature indicating the presence of commonly used **template phrases**.
Example:
- "Highly recommended"
- "Best product ever"
- "Must buy"
- "Worth every penny"

f . TF-IDF vectors- TF-IDF converts text into numeric vectors by emphasizing important and rare words, which helps detect repetitive and promotional phrases commonly found in fake reviews.

Fake reviews:
- Use **repetitive promotional phrases**
- Contain **similar word patterns**
- Avoid detailed descriptions

```
TF-IDF:
     Captures repeated fake-style phrases
     Reduces influence of common stopwords
    Works well with classical ML models
```

Feature	Library Used	Reason
Review length	Python / pandas	Character counting
Word count	Python	Token splitting
Adjective ratio	spaCy	Accurate POS tagging
Sentiment score	NLTK (VADER)	Robust sentiment polarity
Generic phrase flag	re / Python	Pattern matching
TF-IDF vectors	scikit-learn	Text vectorization

1. Sentiment–Rating Mismatch
   when the **sentiment of review text does not align with the given rating**.

Rating	Sentiment	Interpretation
5★	Negative	Suspicious
1★	Positive	Suspicious
5★	Highly Positive	Normal
3★	Neutral	Normal

2. Rule based fake review scoring-Before applying machine learning, a **rule-based scoring system** is built to detect fake reviews in an **interpretable way**.

Condition	Score
Short review	+1
Extreme rating (1★ or 5★)	+1
Generic phrases present	+1
Sentiment–rating mismatch	+1
High adjective ratio	+1

Decision Rule:
If Total Score ≥ 3 → Fake Review
Else → Genuine Review

4.Machine Learning Models

objective of this phase is to **train, evaluate, and compare multiple machine learning models** that can automatically classify reviews as **Fake or Genuine** using the engineered features and text representations obtained in previous steps.

Input to ML models
Each review is represented by a **combined feature set**:
1. **Textual Features**
- TF-IDF vectors (word importance)
2. **Statistical & Linguistic Features**
- Review length
- Word count
- Adjective ratio
- Sentiment score
- Generic phrase flag
- Sentiment–rating mismatch
- Rule-based fake score

These features allow models to learn **both linguistic patterns and behavioral signals** of fake reviews.

Classification type
Binary Classification
- Fake Review → **1**
- Genuine Review → **0**

Train and Test Split Strategy
80-20 ratio

Model Categories

Baseline Models

Baseline models provide a **reference performance** and help verify whether the extracted features are meaningful.

1. Logistic Regression

Purpose:
Acts as a strong linear baseline classifier
Estimates the probability of a review being fake

Why chosen:

- Works extremely well with TF-IDF vectors
- Interpretable and fast
- Ideal for text classification tasks

Learning behavior:

- Assigns weights to each feature
- Learns which words and behavioral features increase the likelihood of a review being fake

2. Naive Bayes

Purpose:

- Probabilistic baseline model for text data

Why chosen:

- Performs well for bag-of-words and TF-IDF features
- Extremely fast and memory efficient

Learning behavior:

- Assumes conditional independence between words
- Learns fake-review word distributions (e.g., “best ever”, “highly recommended”)

Baseline Models Goal

- Validate feature usefulness

- Provide a performance benchmark
- Ensure model simplicity before moving to complex methods

Advanced Models

1. Random Forest

The **Random Forest Classifier** is used as an **ensemble-based machine learning model** to classify reviews into:
Fake (1)
Genuine (0)

It combines the predictions of multiple decision trees to improve **classification accuracy, robustness, and generalization**.

Random Forest is particularly effective for fake review detection due to the following reasons:

- It can handle **non-linear relationships** between features such as sentiment score, adjective ratio, and review length.
- It learns **complex decision rules** from a combination of:
  - TF-IDF vectors
  - Behavioral features (review length, word count)
  - Linguistic features (adjective ratio, generic phrases)
  - Rule-based fake score
- It significantly **reduces overfitting** by averaging the predictions of multiple decision trees instead of relying on a single tree.

This makes Random Forest robust against noise and biased patterns commonly found in fake review datasets.

Learning Behavior in Fake Review Detection

Each decision tree in the Random Forest learns **different fake-review signals** by using:

- A random subset of training reviews
- A random subset of features at each split

As a result:

- Some trees may focus on **textual patterns** (e.g., repetitive promotional phrases)
- Others may focus on **behavioral indicators** (e.g., very short or very long reviews)
- Some trees may learn **sentiment–rating mismatch patterns**

The **final prediction** for a review is obtained using **majority voting**, where the class predicted by most trees is selected as the final output.

Feature Signal	Tree Decision
Short review + generic phrase	Fake
Extreme rating + high sentiment	Fake
Balanced sentiment + detailed content	Genuine
Mixed indicators	Depends on majority

5. Model Evaluation

Evaluation Metrics
The following metrics are considered:

- Accuracy
- Precision
- Recall
- Confusion Matrix

Each metric provides a different perspective on model performance and helps in understanding the nature of classification errors.

In fake review detection, a wrong prediction can lead to:

- Genuine reviewers being falsely accused
- Loss of trust in the platform
- Legal and ethical issues

Therefore, **how** a model makes mistakes is more important than **how many** mistakes it makes.

Confusion Matrix

	Predicted Fake	Predicted Genuine
Actual Fake	True Positive (TP)	False Negative (FN)
Actual Genuine	False Positive (FP)	True Negative (TN)

Where:

- **TP (True Positive)** → Fake review correctly detected
- **FP (False Positive)** → Genuine review wrongly flagged as fake

- **FN (False Negative)** → Fake review missed
- **TN (True Negative)** → Genuine review correctly identified

Accuracy

Accuracy measures the overall correctness of the model and is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Although accuracy is a commonly used metric, it can be **misleading in fake review detection**. In highly imbalanced datasets, a model may achieve high accuracy by predicting most reviews as genuine while failing to detect fake reviews effectively.

Precision

Precision measures the reliability of fake review predictions and is defined as:

$$Precision = \frac{TP}{TP + FP}$$

Precision indicates **how many of the reviews predicted as fake are actually fake**. This metric is critically important in fake review detection because a **false positive** (labeling a genuine review as fake) can lead to loss of customer trust, unfair penalization of honest users, and reduced credibility of the platform.

Therefore, a high precision value ensures that when the system flags a review as fake, it is highly likely to be correct.

Recall

Recall measures the ability of the model to identify all fake reviews and is defined as:

$$Recall = \frac{TP}{TP + FN}$$

A high recall indicates that the model successfully detects a large proportion of fake reviews. However, improving recall alone may increase the number of false positives, which negatively affects system reliability.

Summary

- Accuracy alone is insufficient for evaluating fake review detection models.
- Precision is the most critical metric, as it minimizes false accusations of genuine reviews.

- Recall complements precision by measuring fake review detection coverage.
- The confusion matrix provides a clear understanding of different types of classification errors.

6. Visualization

Create:
- Fake vs real review count
- Review length distribution
- Rating distribution
- Confusion matrix heatmap

Libraries:

✓ Matplotlib

- Core plotting library
- Used for basic plots and full customization
- Foundation for all other visualization tools

✓ Seaborn

- Built on top of Matplotlib
- Provides clean, statistical, and publication-ready plots
- Ideal for distributions and heatmaps

✓ Pandas (Built-in Plotting)

- Quick exploratory plots
- Useful for initial EDA

✓ Scikit-learn (Metrics Visualization Support)

- Provides confusion matrix data
- Used in combination with Seaborn/Matplotlib

Fake vs Genuine Review Count

Tool Used:
✓ Pandas
✓ Seaborn

Visualization Type:

- Bar chart

Purpose:

- Show class distribution
- Detect dataset imbalance
-

Review Length Distribution

Tool Used:
✓ Seaborn
✓ Matplotlib

Visualization Type:

- Histogram
- Kernel Density Plot (optional)

Purpose:

- Compare writing behavior of fake vs genuine reviews
- Validate short/long review assumptions

Rating Distribution

Tool Used:
✓ Seaborn
✓ Matplotlib

Visualization Type:

- Count plot
- Bar chart

Purpose:

- Identify extreme rating patterns
- Support rule-based extreme rating detection

Sentiment vs Rating Analysis (Optional but Strong)

Tool Used:
✓ Seaborn

Visualization Type:

- Scatter plot
- Box plot

Purpose:

- Visualize sentiment–rating mismatch
- Validate fake review heuristics

Confusion Matrix Heatmap

Tool Used:
✓ Scikit-learn
✓ Seaborn
✓ Matplotlib

Visualization Type:

- Heatmap

Purpose:

- Visual evaluation of model performance
- Analyze false positives and false negatives

Feature Importance Visualization (Random Forest)**

Tool Used:
✓ Matplotlib
✓ Pandas

Visualization Type:

- Horizontal bar chart

Purpose:

- Identify most influential fake-review features
- Improve explainability