

File Edit Selection View Go Run ...

python.py crud shop manager.py # Sentiment Analysis (Beginner Version).py X

C:\> Users > rashi > Desktop > # Sentiment Analysis (Beginner Version).py > ...

```
3 # Step 1: Import the tools we need
4 from sklearn.feature_extraction.text import CountVectorizer
5 from sklearn.naive_bayes import MultinomialNB
6
7 # Step 2: Create a small dataset (posts + their sentiment labels)
8 posts = [
9     "I love this app, it's awesome!",
10    "This is the worst update ever.",
11    "Not bad, but could be better.",
12    "Amazing experience, I'm so happy!",
13    "I hate using this, very annoying."
14 ]
15
16 labels = ["positive", "negative", "neutral", "positive", "negative"]
17
18 # Step 3: Convert text into numbers (Bag of Words)
19 vectorizer = CountVectorizer()
20 X = vectorizer.fit_transform(posts)
21
22 # Step 4: Train a simple model (Naive Bayes)
23 model = MultinomialNB()
24 model.fit(X, labels)
25
26 # Step 5: Test the model with new posts
27 new_posts = [
28     "I really enjoy using this app!",
29     "Terrible service, I'm angry.",
30     "It's okay, nothing special."
31 ]
32
33 X_new = vectorizer.transform(new_posts)
34 predictions = model.predict(X_new)
35
```

Ln 28, Col 38 Spaces: 4 UTF-8 CRLF Python Signed out 3.13.5 (base) Go Live

19°C Mostly clear

01:27 25-11-2025

File Edit Selection View Go Run ...

python.py crud shop manager.py # Sentiment Analysis (Beginner Version).py 7 X

C:\> Users > rashi > Desktop > # Sentiment Analysis (Beginner Version).py > ...

```
69 import numpy as np
70
71 def predict_sentiment(model, vectorizer, text):
72     """
73     Takes cleaned text → converts to numbers → model predicts → returns label.
74     """
75
76     # Convert single text to list for consistent processing
77     single = False
78     if isinstance(text, str):
79         text = [text]
80         single = True
81
82     # Convert text to numeric form for the ML model
83     vectorized = vectorizer.transform(text)
84
85     # Predict the sentiment (positive/negative/neutral)
86     prediction = model.predict(vectorized)
87
88     # If model supports probability scores
89     if hasattr(model, "predict_proba"):
90         probability = model.predict_proba(vectorized)
91         confidence = probability.max(axis=1)
92     else:
93         confidence = [1.0] * len(prediction) # basic fallback
94
95     results = []
96     for label, score in zip(prediction, confidence):
97         results.append({
98             "label": label,
99             "confidence": float(score)
100         })
101
```

Ln 28, Col 38 Spaces: 4 UTF-8 CRLF Python Signed out 3.13.5 (base) Go Live

19°C Mostly clear

01:27 25-11-2025



File Edit Selection View Go Run ...

python.py crud shop manager.py # Sentiment Analysis (Beginner Version).py 7 X

C: > Users > rashi > Desktop > # Sentiment Analysis (Beginner Version).py > ...

```
154 def train(csv_path):
168
169     # Convert text to numeric features
170     vectorizer = TfidfVectorizer(max_features=15000)
171     X_train_vec = vectorizer.fit_transform(X_train)
172     X_test_vec = vectorizer.transform(X_test)
173
174     # Train the model
175     model = LogisticRegression(max_iter=1000)
176     model.fit(X_train_vec, y_train)
177
178     # Test performance
179     predictions = model.predict(X_test_vec)
180     accuracy = accuracy_score(y_test, predictions)
181     f1score = f1_score(y_test, predictions, average="weighted")
182
183     logger.info(f"Accuracy: {accuracy:.3f}")
184     logger.info(f"F1 Score: {f1score:.3f}")
185     logger.info("\n" + classification_report(y_test, predictions))
186
187     # Save model + vectorizer
188     save_model(model, vectorizer)
189
190     print("Training complete. Model saved.")
191     """
192
193 main.py
194 This is the user-facing program.
195 User types text + system cleans + predicts + shows result.
196 """
197
198 from model_loader import load_model_and_vectorizer
199 from preprocess import clean_text
200 from validator import
```

Ln 28, Col 38 Spaces: 4 UTF-8 CRLF Python Signed out 3.13.5 (base) Go Live

19°C Mostly clear 01:28 25-11-2025

File Edit Selection View Go Run ...

python.py crud shop manager.py # Sentiment Analysis (Beginner Version).py 7 X

C: > Users > rashi > Desktop > # Sentiment Analysis (Beginner Version).py > ...

```
200     is_non_empty_text,
201     is_text_reasonable_length,
202     contains_allowed_characters,
203 )
204 from predictor import predict_sentiment
205
206 def start():
207     print("=== Sentiment Analysis System ===")
208     print("Type a message to analyze. Type 'exit' to quit.\n")
209
210     # Load trained model
211     try:
212         model, vectorizer = load_model_and_vectorizer()
213     except FileNotFoundError:
214         print("Model not found. Train the model first using train.py")
215         return
216
217     while True:
218         user_input = input("Enter text: ")
219
220         if user_input.lower() == "exit":
221             print("Goodbye!")
222             break
223
224         # Validate input
225         if not is_non_empty_text(user_input):
226             print("! Please type something.")
227             continue
228         if not contains_allowed_characters(user_input):
229             print("! Text doesn't contain valid characters.")
230             continue
231         if not is_text_reasonable_length(user_input):
232             print("! Text too long or too short.")
```

Ln 28, Col 38 Spaces: 4 UTF-8 CRLF Python Signed out 3.13.5 (base) Go Live

19°C Mostly clear 01:28 25-11-2025

File Edit Selection View Go Run ...

python.py crud shop manager.py # Sentiment Analysis (Beginner Version).py 7

C: > Users > rashi > Desktop > # Sentiment Analysis (Beginner Version).py > ...

```
206 def start():
234
235     cleaned = clean_text(user_input)
236     result = predict_sentiment(model, vectorizer, cleaned)
237
238     print(f"\n🌸 Sentiment: {result['label']}")
239     print(f"📊 Confidence: {result['confidence']:.2f}\n")
240
241 if __name__ == "__main__":
242     start()
```

Ln 28, Col 38 Spaces: 4 UTF-8 CRLF Python Signed out 3.13.5 (base) Go Live

19°C Mostly clear 01:28 25-11-2025

