```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer, ENGLISH_STOP_WORDS
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, ConfusionMatrixDisplay

# Example: Load your data (replace this with your actual dataset)
# For example, df = pd.read_csv("your_data.csv")
# Make sure your DataFrame has columns: "text" and "label"
# Here's a dummy placeholder for demonstration:
# df = pd.DataFrame({
#     "text": ["I love this!", "Terrible experience", "Great service", "Worst ever", "Will buy again"],
#     "label": [1, 0, 1, 0, 1]
# })

# Vectorize text using TF-IDF
vectorizer = TfidfVectorizer(stop_words='english')
features = pd.DataFrame(
    vectorizer.fit_transform(df["text"]).toarray(),
    columns=vectorizer.get_feature_names_out()
)

# Generate length-based metrics
df["char_count"] = df["text"].str.count(r"\S")
df["word_count"] = df["text"].str.count(r"\w+")
df["avg_word_length"] = df["char_count"] / df["word_count"].replace(0, np.nan)
```

```
# Combine TF-IDF features with length-based features
x = pd.concat([features, df.loc[:, "char_count":]], axis=1)
y = df["label"]

# Train/test split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)

# Train a Random Forest classifier
rf = RandomForestClassifier(random_state=42).fit(x_train, y_train)

# Predict the labels
y_pred = rf.predict(x_test)

# Print classification metrics
print(classification_report(y_test, y_pred))

# Plot confusion matrix
ConfusionMatrixDisplay.from_estimator(rf, x_test, y_test, normalize="all")
plt.title("Confusion Matrix")
plt.show()

# Show feature importances
feature_importance_df = pd.DataFrame({
    "feature": x.columns,
    "importance": rf.feature_importances_
}).sort_values(by="importance", ascending=False)

print(feature_importance_df.head(10))
```

```
# Temporary dummy dataset for testing
df = pd.DataFrame({
    "text": [
        "I love this product!",
        "Terrible service and rude staff.",
        "Absolutely fantastic experience.",
        "Worst purchase ever.",
        "I would buy this again."
    ],
    "label": [1, 0, 1, 0, 1]  # 1 = Positive, 0 = Negative
})
```