```
# Clean text
def clean_text(text):
    text = str(text).lower()
    text = re.sub(f"[{re.escape(string.punctuation)}]", "", text)
    text = re.sub(r"\d+", "", text)
    return text

df['clean_query'] = df['query'].apply(clean_text)

# Label Encoding
le = LabelEncoder()
df['label'] = le.fit_transform(df['category'])

# TF-IDF
tfidf = TfidfVectorizer(max_features=1000)
X_tfidf = tfidf.fit_transform(df['clean_query']).toarray()

# Add query length as a feature
X_combined = np.hstack((X_tfidf, df[['query_length']].values))

# Dimensionality reduction
pca = PCA(n_components=50)
X_pca = pca.fit_transform(X_combined)

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X_pca, df['label'], test_size=0.2,
random_state=42)
```