```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd

# Define file path
file_path = '/content/drive/My Drive/NLPHW4/hw4.pk'

# Load the pickle file into a DataFrame
data = pd.read_pickle(file_path)

# Display the first few rows to confirm the data
print(data.head())

```

```
                                                body                    label
0   We use essential cookies to make Venngage wor...  legal_contract_examples
1  A legal contract is a written document that is...  legal_contract_examples
2   November 27 2023 14 min Author Olga Asheychik...  legal_contract_examples
3  Accelerate contracts with AI native workflows ...  legal_contract_examples
4  Create smarter agreements commit to them more ...  legal_contract_examples
```

```
import pickle
import nltk
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string

# Download NLTK resources
nltk.download('punkt_tab')
nltk.download('stopwords')

# Define file path for pickle file
file_path = '/content/drive/My Drive/NLPHW4/hw4.pk'

# Load the pickle file
with open(file_path, 'rb') as f:
    data = pickle.load(f)

# Extract body (text of document) and labels
documents = data['body']
labels = data['label']

# Preprocess text: tokenization, lowercasing, removing stopwords and punctuation
stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    # Tokenize the text
    tokens = word_tokenize(text.lower())  # Convert to lowercase and tokenize
    # Remove stopwords and punctuation
    tokens = [word for word in tokens if word.isalpha() and word not in stop_words]
    return " ".join(tokens)

# Preprocess all documents
preprocessed_documents = [preprocess_text(doc) for doc in documents]

# Split the data into training and test sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(preprocessed_documents, labels, test_size=0.3, random_state=42)

# Convert text data into TF-IDF features
tfidf_vectorizer = TfidfVectorizer()
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Train a Random Forest classifier
classifier = RandomForestClassifier(n_estimators=100, random_state=42)
classifier.fit(X_train_tfidf, y_train)

```

```
# Make predictions on the test set
y_pred = classifier.predict(X_test_tfidf)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Classify all documents in the dataset (train and test combined)
all_tfidf = tfidf_vectorizer.transform(preprocessed_documents)
all_predictions = classifier.predict(all_tfidf)

# Print the predictions for all documents (first 10 for brevity)
for doc, label in zip(documents[:10], all_predictions[:10]):  # Displaying the first 10 for brevity
    print(f"Document: {doc[:100]}... => Predicted label: {label}")

# Optionally: Save the predictions with document texts
classified_documents = pd.DataFrame({
    'Document': documents,
    'Predicted Label': all_predictions
})

# Save to a CSV file
classified_documents.to_csv('/content/drive/My Drive/NLPHW4/classified_documents.csv', index=False)

```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Accuracy: 0.7794117647058824

Classification Report:
                                  precision    recall  f1-score   support

engineering_specification_examples       0.62      0.95      0.75        22
           legal_contract_examples       0.89      0.80      0.84        20
       marketing_material_examples       1.00      0.62      0.76        26

                          accuracy                           0.78        68
                         macro avg       0.84      0.79      0.78        68
                      weighted avg       0.84      0.78      0.78        68

Document:  We use essential cookies to make Venngage work By clicking Accept All Cookies you agree to the stor... => Predicted label: le
Document: A legal contract is a written document that is drawn up by a party and is agreed upon by all parties... => Predicted label: le
Document:  November 27 2023 14 min Author Olga Asheychik Senior Web Analytics Manager at PandaDoc Choosing the... => Predicted label: le
Document: Accelerate contracts with AI native workflows Advanced electronic signature on any device Create con... => Predicted label: le
Document: Create smarter agreements commit to them more efficiently and manage them to realize their full valu... => Predicted label: le
Document: A contract is an agreement between two parties that creates an obligation to perform or not perform ... => Predicted label: le
Document: Please enable JS and disable any ad blocker... => Predicted label: legal_contract_examples
Document:  WEBINAR NOV 14th Use Contract Data to Drive New Insights REGISTER NOW In the realm of legal agreeme... => Predicted label: le
Document:  luctus etiam leo nulla etiam convallis tincidunt integer Pellentesque suscipit adipiscing nullam lu... => Predicted label: le
Document: Entering into contracts is part of running a small business and it s important to manage your contra... => Predicted label: le
```