```
!pip install pyLDAvis
!pip install gensim
!pip install --upgrade numpy
!pip install pandas==1.5.3
```

1/2 **1. Data acquisition, description, and preparation 2.Research Question**

The research question I aim to explore with topic modeling is: "**How did regional parties in the UK, along with UKIP, adapt and respond to the realities of Brexit post-referendum?**" I would like to delve into the understanding of the thematic shifts and focuses in party manifestos and communications during a crititcal point in UK politics. In using the Latent Dirichlet Allocation (LDA) by examining the frequency and co-occurrence of words across various texts, the LDA can reveal the dominant themes and concerns in these political manifestos. My dataset, comprising of seven party platforms from 2017-2019 of UKIP, SNP, Plaid Cymru (Party of Wales), and DUP, provides a good an varied dataset for this analysis. I have chosen to go with regional parties and UKIP to together to ge to see how all parties tried to take a postive spin of Brexit regional and UKIP after the referendum. The chosen time frame captures the immediate aftermath of the Brexit referendum, a period likely to be ripe with scottish independece desire, fear in Norhtern Ireland over a hard border, worries in all regions about econmic impacts of brexit and UKIP's push for more sovereignty being a tying theme for all the parties.

Import Libraries

```
import os
import pandas as pd
import re
import numpy
import gensim
import pyLDAvis
import pyLDAvis.gensim_models
import matplotlib.pyplot as plt
from gensim.models import LdaModel
from gensim.corpora import Dictionary
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
nltk.download('punkt')
nltk.download('stopwords')
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
[nltk_data] Downloading package punkt to /root/nltk_data...
```

```
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

Preprocessing and Document Matrix

```
def preprocess_text(text):
    text = text.lower()
    text = re.sub(r'\W', ' ', text)
    text = re.sub(r'\s+[a-zA-Z]\s+', ' ', text)
    text = re.sub(r'\^[a-zA-Z]\s+', ' ', text)
    text = re.sub(r'\s+', ' ', text, flags=re.I)
    return text

```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
```

```
folder_path = '/content/Data'

all_texts = pd.DataFrame(columns=['text'])

for filename in os.listdir(folder_path):
    if filename.endswith('.csv'):
        file_path = os.path.join(folder_path, filename)
        manifesto = pd.read_csv(file_path, usecols=[0], header=0)
        manifesto.columns = ['text']
        manifesto['processed_text'] = manifesto['text'].apply(preprocess_tex
        all_texts = pd.concat([all_texts, manifesto], ignore_index=True)

```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
```

```
all_texts['text'] = all_texts['text'].apply(preprocess_text)

tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(all_texts['text'])
tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns=tfidf_vectorizer.get
tfidf_df
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
```

	00	000	000per	049	067	10	100	1000	100bn	108	...	ynys	young
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0

4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6237	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
6238	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
6239	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
6240	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
6241	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0

6242 rows × 7395 columns

Tokenization

```
stop_words = set(stopwords.words('english'))

def tokenize_and_preprocess(text):
    tokens = word_tokenize(text)
    tokens = [token for token in tokens if token not in stop_words]
    tokens = [token for token in tokens if token not in string.punctuation]
    tokens = [token for token in tokens if token.isalnum()]
    return tokens

all_texts['tokens'] = all_texts['processed_text'].apply(tokenize_and_preproc
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
```

LDA

3

LDA analyzes words and phrases across documents (for example political manifestos) and identifies clusters of terms that frequently appear together, which it then interprets as topics.

In executing this model for the analysis, I made critical choices regarding its configuration. One such choice was the number of topics to extract, I've have set at five. This defines how many distinct themes you expect to find in the collection of documents. Choosing fewer topics might lead to a simplistic view where distinct themes of the manifestos are merged, while too many topics can fragment the analysis, leading to overly specific topics. Another key parameter was the number of 'passes' the LDA model makes over the data, set at 25 in my case. More passes typically lead to more refined and accurate topic assignments but require more computational resources and time.

These choices significantly influence how effectively the model can evaluate the research question: "How did the regional parties, along with UKIP, attempt to reconcile with the reality of Brexit post-referendum?" By opting for five topics, the goal was to strike a balance between capturing a broad spectrum of themes and maintaining analytical clarity.

Corpus for LDA

```
dictionary = Dictionary(all_texts['tokens'])
corpus = [dictionary.doc2bow(tokens) for tokens in all_texts['tokens']]
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
```

Train LDA Model and Print Topics

```
num_topics = 5

lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, pass

topics = lda_model.print_topics(num_topics=num_topics, num_words=10)
for topic_number, topic_words in topics:
    print(f"Topic {topic_number + 1}: {topic_words}")
```

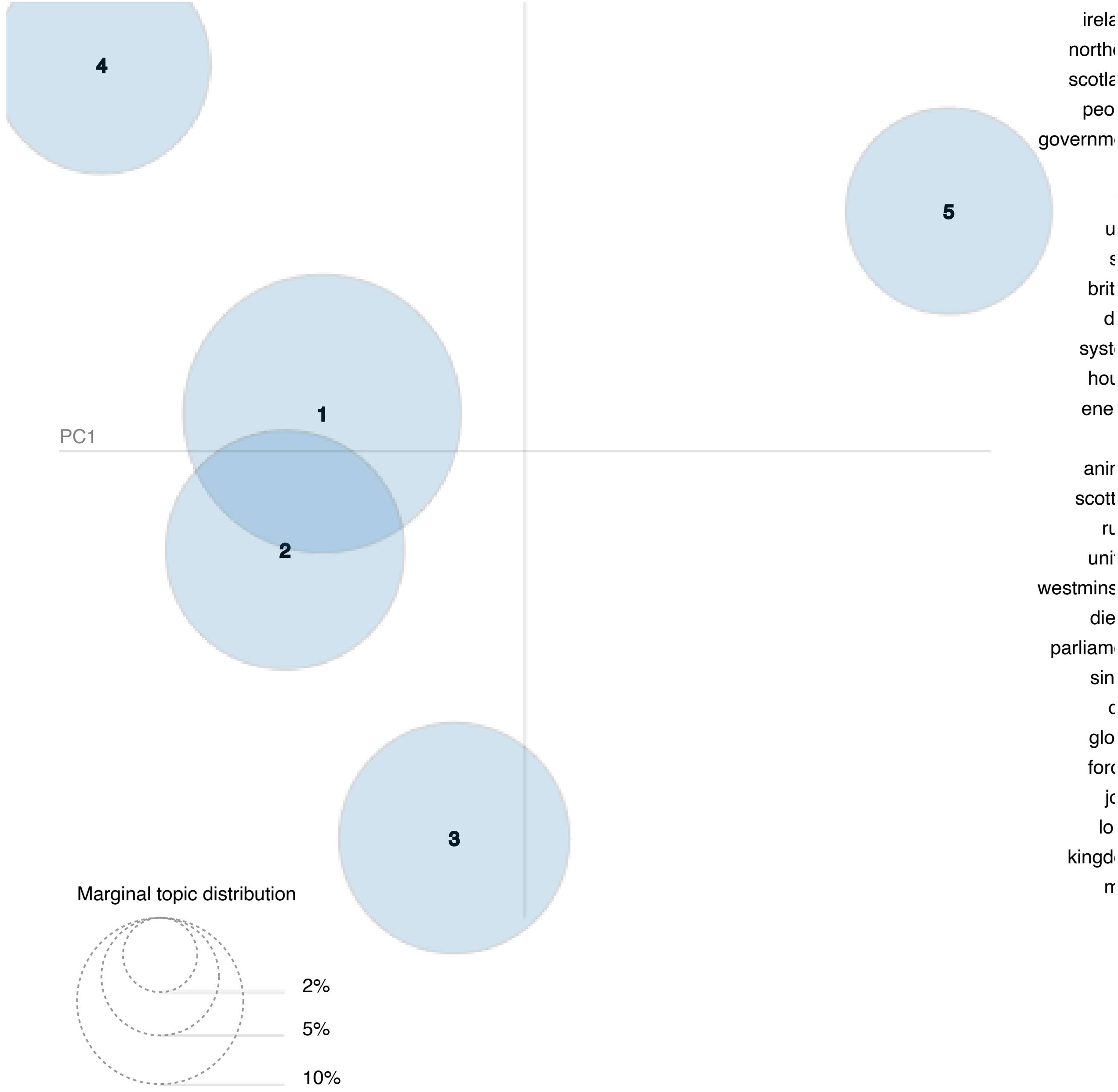
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
Topic 1: 0.018*"people" + 0.012*"system" + 0.008*"house" + 0.006*"health" +
Topic 2: 0.018*"eu" + 0.017*"ireland" + 0.017*"northern" + 0.012*"uk" + 0.0
Topic 3: 0.021*"ukip" + 0.011*"energy" + 0.010*"britain" + 0.008*"new" + 0.
Topic 4: 0.018*"scotland" + 0.015*"government" + 0.011*"uk" + 0.008*"snp" +
Topic 5: 0.015*"government" + 0.014*"uk" + 0.011*"tax" + 0.011*"snp" + 0.00
```

pyLDAvis

```
pyldavis_data = pyLDAvis.gensim_models.prepare(lda_model, corpus, dictionary
pyLDAvis.display(pyldavis_data)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/pandas/core/dtypes/cast.py:1641: De
See https://numpy.org/devdocs/release/1.25.0-notes.html and the docs for mo
  return np.find_common_type(types, [])
```

Selected Topic: 0 Previous Topic Next Topic Clear Topic



4

There are 5 topics that cover the most frequent terms of the party platforms focusing on the issues that each region and UKIP must focus on to have a "successfull" brexit. The topics I have found are Societal Systems and Brexit Implications, Geopolitical Dynamics and Regional Identities, Economic and Environmental Strategies, and Governance and Fiscal Policies.

Save pyLDAvis as HTML

```
pyLDAvis.save_html(pyldavis_data, 'visualization.html')
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: Deprecat
  and should_run_async(code)
```

5. **Societal Systems and Brexit Implications (Topic 1)**

In Topic 1 the data reveals an interpretation by regional parties such as UKIP and the DUP of Brexit's ramifications on societal infrastructures. The prevalent terms - "people," "system," "house," "health" - underscore a deep-seated concern for the holistic well-being of citizens within the post-Brexit context. The focus on "health," "ukip," "united," and "services" reflects that there was a critical discourse on the reconfiguration of national healthcare and service frameworks, adapting to new socio-political realities. Furthermore, the incorporation of terms like "diesel" and "single" an awareness of environmental and economic challenges.

Geopolitical Dynamics and Regional Identities (Topics 2 and 4)

Topics 2 and 4 encapsulate the intricate geopolitical and regional identity shifts in the Brexit epoch. Topic 2, with its emphasis on "eu," "ireland," "northern," and "uk," brings to light pivotal issues such as the Northern Ireland border dilemma and the UK's redefined relationship with the EU. This analysis elucidates concerns over international security, global positioning, and financial resilience in a post-Brexit world. Simultaneously, Topic 4 offers insights into Scotland and Wales' strategic positioning, with terms like "scotland," "government," "snp," and "wales." The prominence of "westminster" and "parliament" in this discourse suggests a critical engagement with power dynamics and autonomy within the UK, highlighting regional parties' advocacy for increased self-governance post-Brexit.

Economic and Environmental Strategies (Topic 3)

Topic 3's is characterized by "ukip," "energy," "britain," and "new," indicates a strategic redirection towards economic and environmental issues. The inclusion of "animal," "local," "jobs," and "cost" points to a strategic balance between economic growth, environmental conservation, and animal welfare. This implies that regional parties are proactively adapting to Brexit-induced economic transformations while also prioritizing sustainable, community-oriented policies.

Governance and Fiscal Policies (Topic 5):

Topic 5 centeres around governance and fiscal management, with key terms like "government," "uk," "tax," and "snp." The analysis highlights a focus on reorganizing government spending and taxation strategies in the post-Brexit scenario. The mention of "women" in this context indicates a recognition of gender-sensitive policy-making.

Conclusion Overall, the topic modeling analysis portrays how UK regional parties and UKIP are strategically navigating the complex challenges in post-Brexit Britain. Their approaches, encompassing societal, healthcare, geopolitical, economic, and governance aspects, demonstrate a concerted effort to address regional nuances alongside broader national concerns within an evolving political and economic framework while trying to focus on moving foward past brexit onto other pollcy topics.