CS410 : US Midterm Election – Twitter Sentiment & Topic Analysis

Unnati Hasija (uhasija2@illinois.edu)

Prateek Dhiman (pdhiman2@illinois.edu)

Contents

1) **Project Background**
2) **Implementation Details**
   a. Requirements
   b. Setup & Usage Instructions with Results and Visualizations
   c. Errors and Solution
3) **Conclusion and Future Work**

Project Background

Background

As discussed in our initial project proposal we took **Twitter** and performed the following for **US Midterm Election** as a key area.

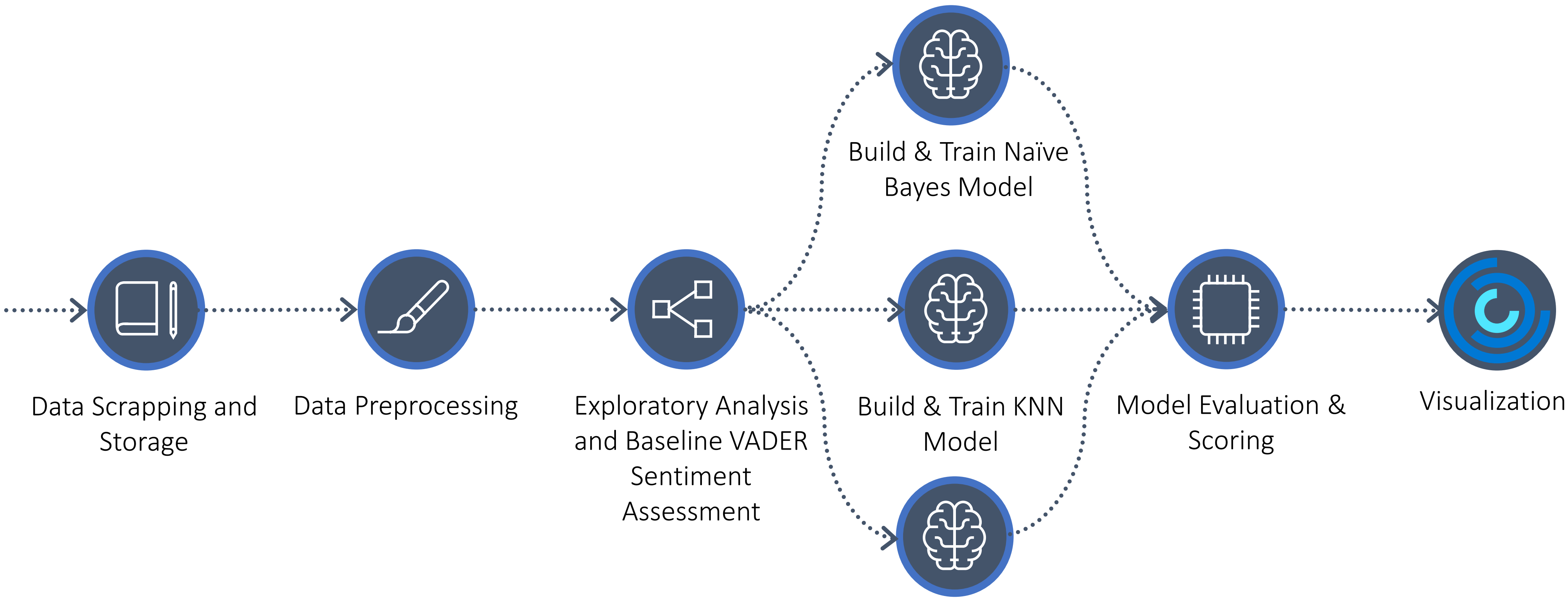
- **Sentiment Analysis** and prediction
  - With **Naïve Bayes** Classification
  - With **K-Nearest Neighbor** Classification
- **Topic Analysis** with **Latent Dirichlet Allocation**

Goal

Key Goals for the project that we set and achieved were

- **Sentiment Analysis** : Identify overall positive , negative and neutral sentiments towards elections and respective parties in US
- **Prediction** : Predict tweets for sentiments using Naïve Bayes and KNN. Also perform a comparison which Algorithm performs better and impact of data cleaning techniques on precision and accuracy of predictions.
- **Topic Analysis** : Identify top topics in tweets and predict the dominant topic and its percentage for each tweet.
- **Visualize** **:** Lastly compare our findings and present them as part of our final project.

Implementation Details



Step 1

Data Scrapping and Storage

Step 2

Data Preprocessing

Step 3

Exploratory Analysis and Baseline VADER Sentiment Assessment

Step 4

Build & Train Naïve Bayes Model

Build & Train KNN Model

Build LDA Model

Step 5

Model Evaluation & Scoring

Step 6

Visualization

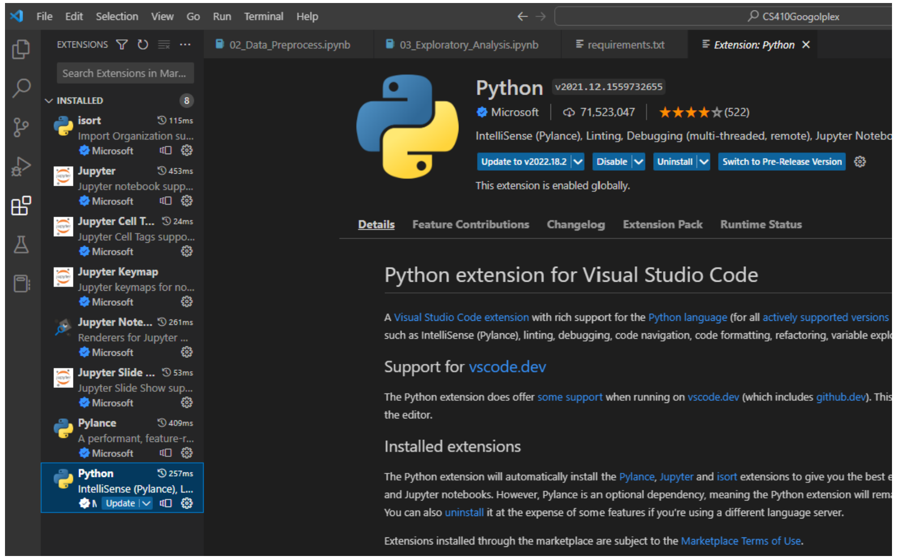
Implementation Details – Requirements

- Requirements
  - Local Jupyter Server / VSCODE for running Jupyter notebooks
  - Python 3.9+
  - git
  - Github Account
  - Snscrape
  - Numpy
  - Pandas
  - NLTK
  - Matplotlib
  - Wordcloud
  - Emot
  - Gensim
  - pyLDAVis

- Setup Instructions
  - Latest Python version and VScode can be installed via Anaconda distribution https://www.anaconda.com/
  - Details of PIP modules are available in in requirement.txt in the project repository. They will be installed as part of setup
  - If you are unable use the requirement.txt file , please use the PIP commands as mentioned in README.MD
  - Follow Screenshots from next slides to setup the project locally.

Setup and Usage Instructions

1. Install Anaconda with Python 3.9 from https://www.anaconda.com
2. Ensure you also install VSCODE via Anaconda Navigator
3. In VSCODE , please ensure you install the python extension


File Edit Selection View Go Run Terminal Help
CS410Googolplex
EXTENSIONS
Search Extensions in Mar...
INSTALLED 8
isort 115ms Import Organization su... Microsoft
Jupyter 453ms Jupyter notebook supp... Microsoft
Jupyter Cell T... 24ms Jupyter Cell Tags suppo... Microsoft
Jupyter Keymap Jupyter keymaps for no... Microsoft
Jupyter Note... 261ms Renderers for Jupyter ... Microsoft
Jupyter Slide ... 53ms Jupyter Slide Show sup... Microsoft
Pylance 409ms A performant, feature-r... Microsoft
Python 257ms IntelliSense (Pylance), L... Update
02_Data_Preprocess.ipynb
03_Exploratory_Analysis.ipynb
requirements.txt
Extension: Python
Python v2021.12.1559732655
Microsoft
71,523,047
(522)
Update to v2022.18.2
Disable
Uninstall
Switch to Pre-Release Version
Details
Feature Contributions
Changelog
Extension Pack
Runtime Status

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebo

This extension is enabled globally.

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable expl

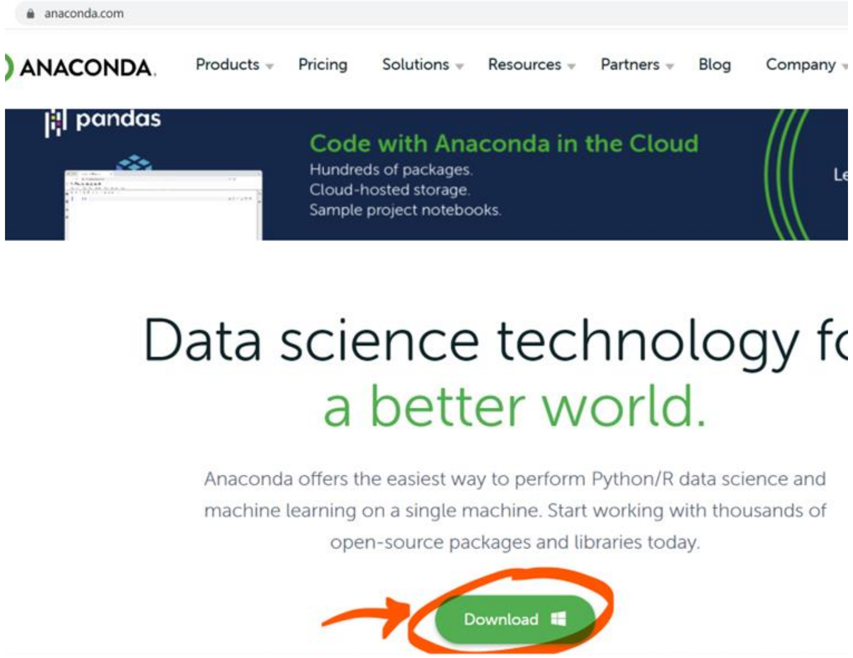
Support for vscode.dev

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This the editor.

Installed extensions

The Python extension will automatically install the Pylance, Jupyter and isort extensions to give you the best e and Jupyter notebooks. However, Pylance is an optional dependency, meaning the Python extension will rema You can also uninstall it at the expense of some features if you're using a different language server.

Extensions installed through the marketplace are subject to the Marketplace Terms of Use.

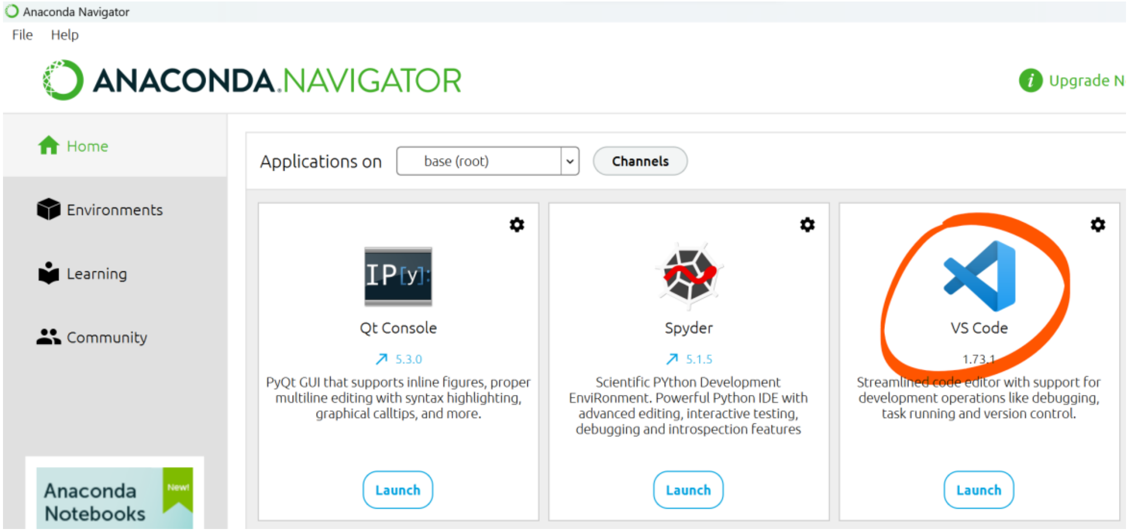


Code with Anaconda in the Cloud

Hundreds of packages.
Cloud-hosted storage.
Sample project notebooks.

Data science technology fo a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.



PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Streamlined code editor with support for development operations like debugging, task running and version control.

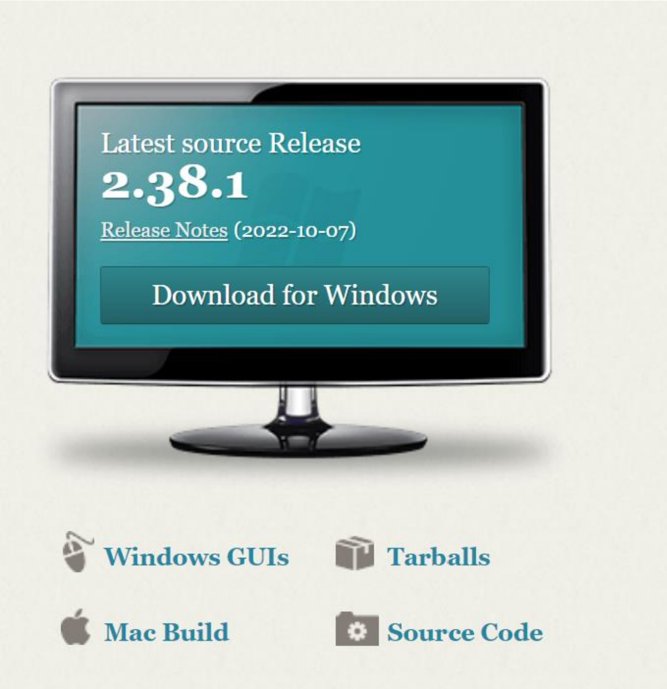
Setup and Usage Instructions

3. Install Git in your local machine if it does not exist from Git (git-scm.com)

4. Perform the initial git setup if not done in past

```
$ git config --global user.name “<your github username> "
```

```
$ git config --global user.email <your github email>
```



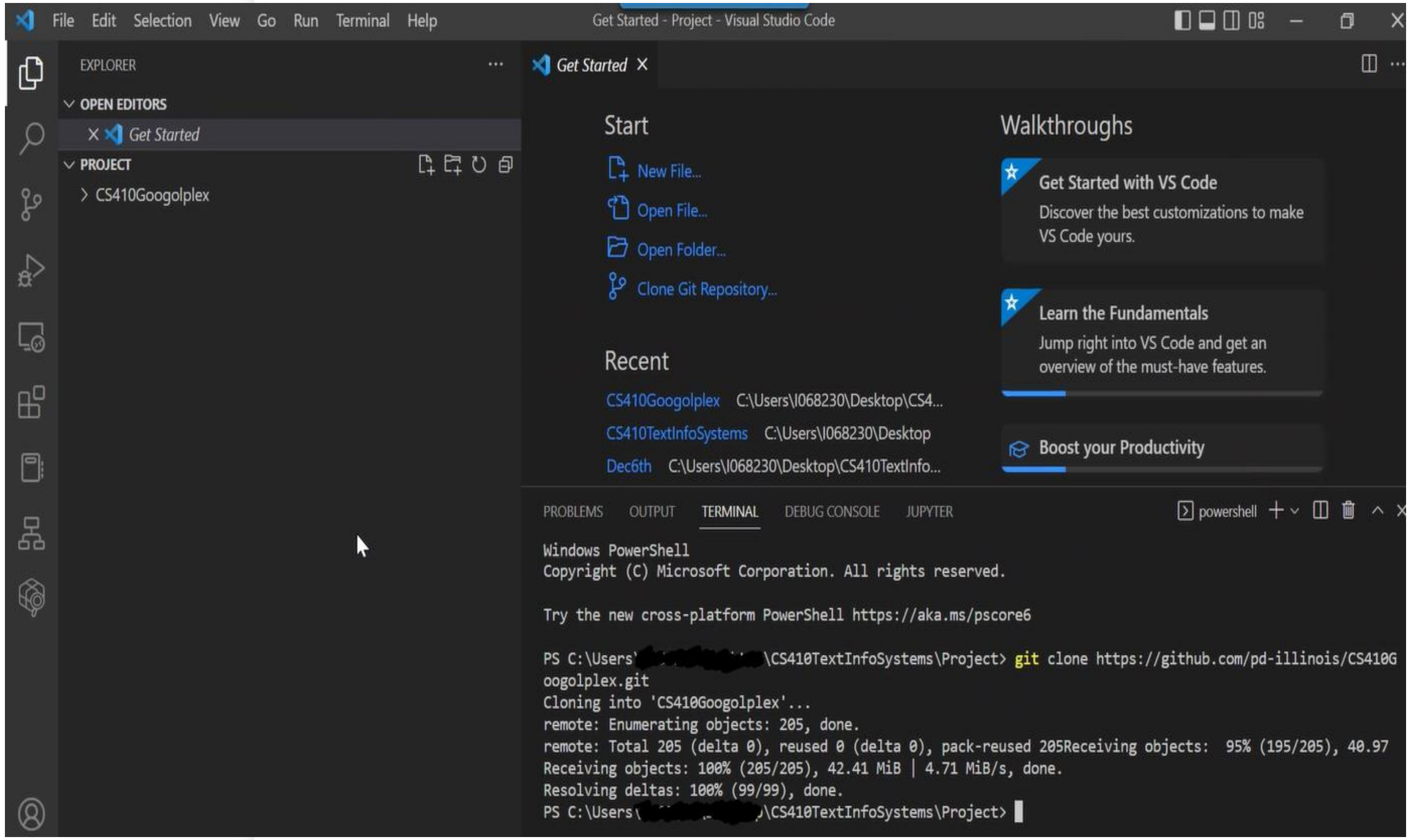
Latest source Release
2.38.1
Release Notes (2022-10-07)

Setup and Usage Instructions

5. Create a folder in your machine where you would clone the project files.

6. Open git bash and run the following

git clone https://github.com/pd-illinois/CS410Googolplex.git


File Edit Selection View Go Run Terminal Help
Get Started - Project - Visual Studio Code
EXPLORER
OPEN EDITORS
Get Started
PROJECT
CS410Googolplex
Get Started
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Recent
CS410Googolplex C:\Users\I068230\Desktop\CS4...
CS410TextInfoSystems C:\Users\I068230\Desktop
Dec6th C:\Users\I068230\Desktop\CS410TextInfo...
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.
Boost your Productivity
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE JUPYTER
powershell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\[redacted]\CS410TextInfoSystems\Project> git clone https://github.com/pd-illinois/CS410G
oogolplex.git
Cloning into 'CS410Googolplex'...
remote: Enumerating objects: 205, done.
remote: Total 205 (delta 0), reused 0 (delta 0), pack-reused 205Receiving objects:  95% (195/205), 40.97
Receiving objects: 100% (205/205), 42.41 MiB | 4.71 MiB/s, done.
Resolving deltas: 100% (99/99), done.
PS C:\Users\[redacted]\CS410TextInfoSystems\Project>
```

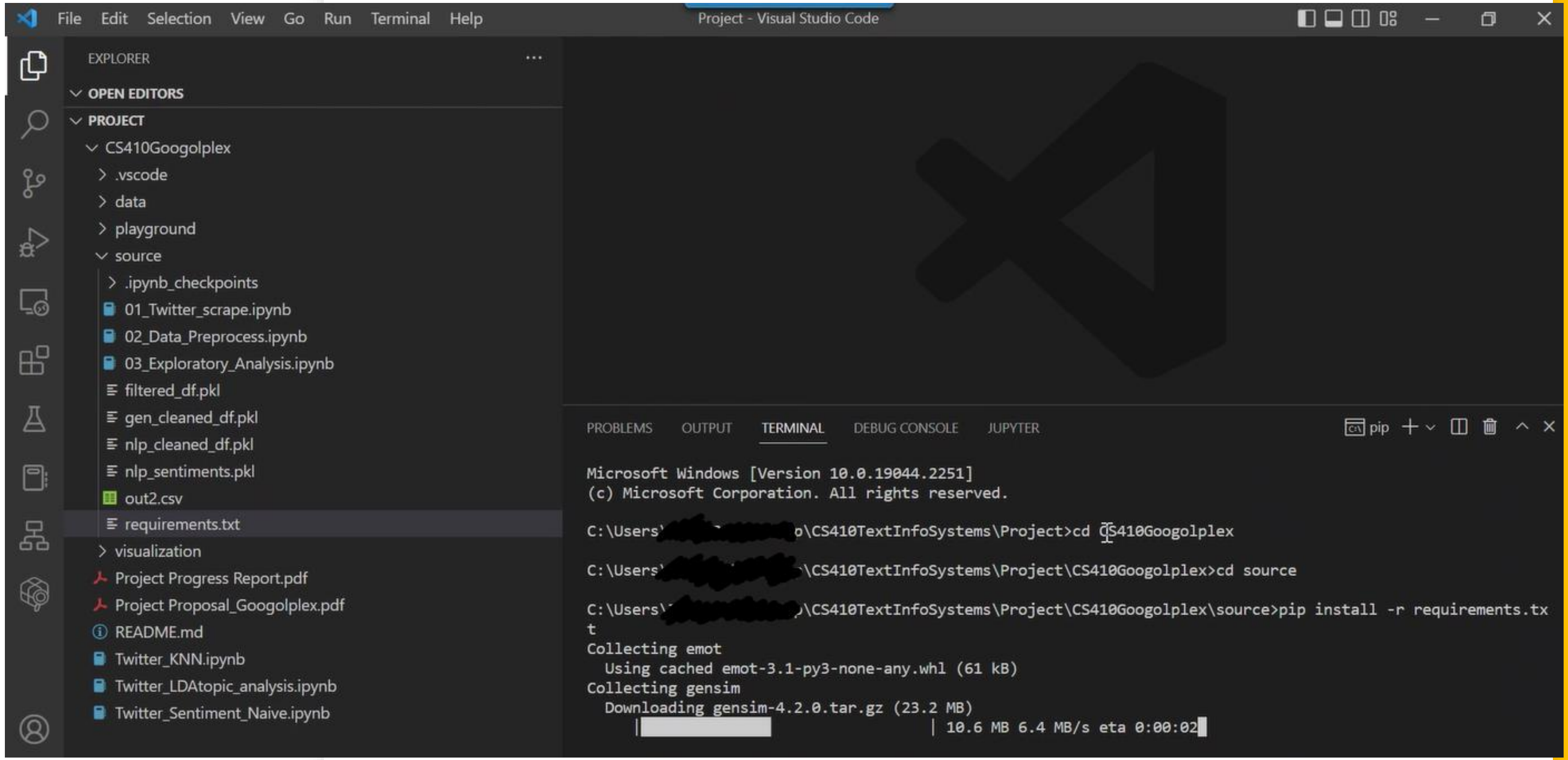
Setup and Usage Instructions

7. Open the folder in VSCODE and go to source folder.

8. Open Terminal in VSCODE and run

pip install -r requirements.txt

Wait for the installation to complete. If installation fails due to genism please see section – Expected Errors and Solution



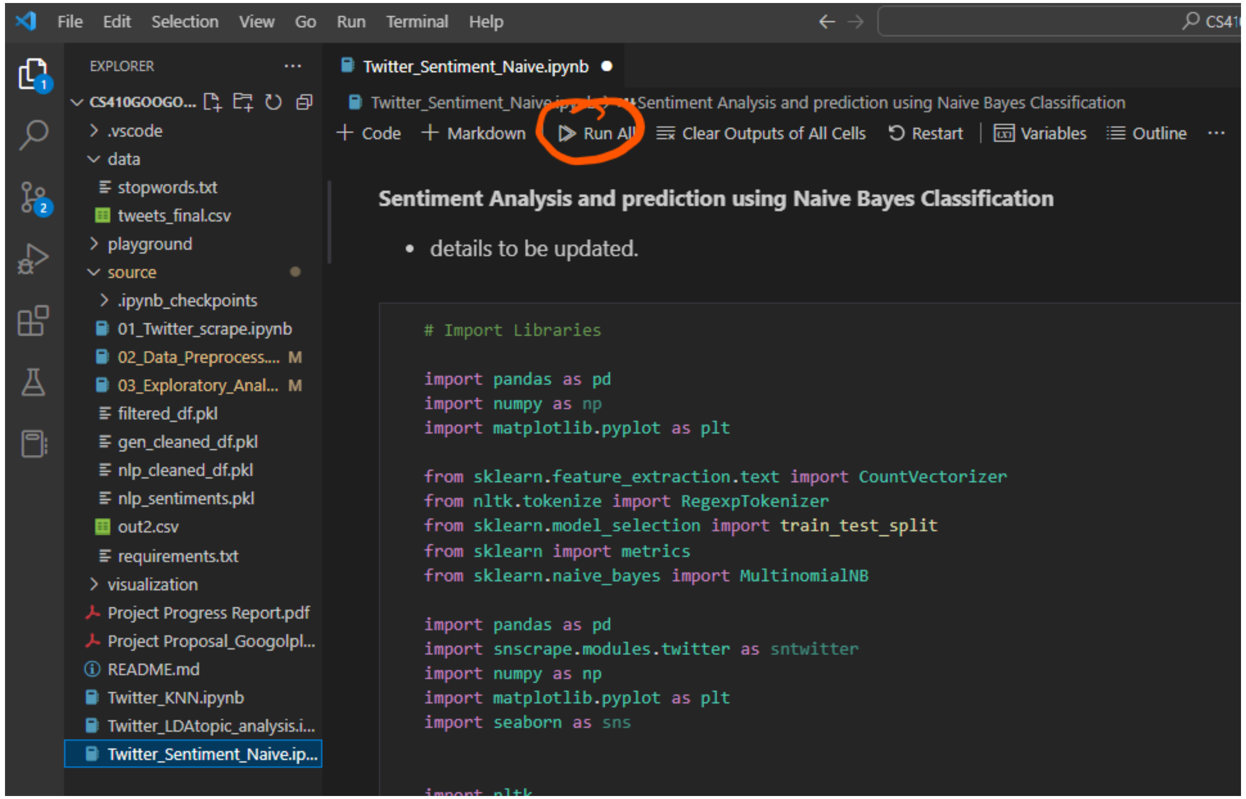
```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\[redacted]\CS410TextInfoSystems\Project>cd CS410Googolplex

C:\Users\[redacted]\CS410TextInfoSystems\Project\CS410Googolplex>cd source

C:\Users\[redacted]\CS410TextInfoSystems\Project\CS410Googolplex\source>pip install -r requirements.tx
t
Collecting emot
  Using cached emot-3.1-py3-none-any.whl (61 kB)
Collecting gensim
  Downloading gensim-4.2.0.tar.gz (23.2 MB)
     |██████████                    | 10.6 MB 6.4 MB/s eta 0:00:02
```

Setup and Usage Instructions

9. Ensure you have the python 3.9 selected as your main python interpreter .

10. Now Open **Twitter_Sentiment_Naive.ipynb** and Run the Jupyter Notebook

11. Validate the successful run and results.



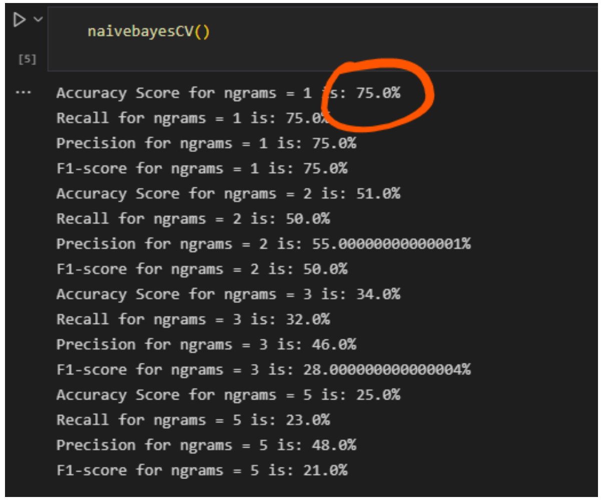
Sentiment Analysis and prediction using Naive Bayes Classification

- details to be updated.

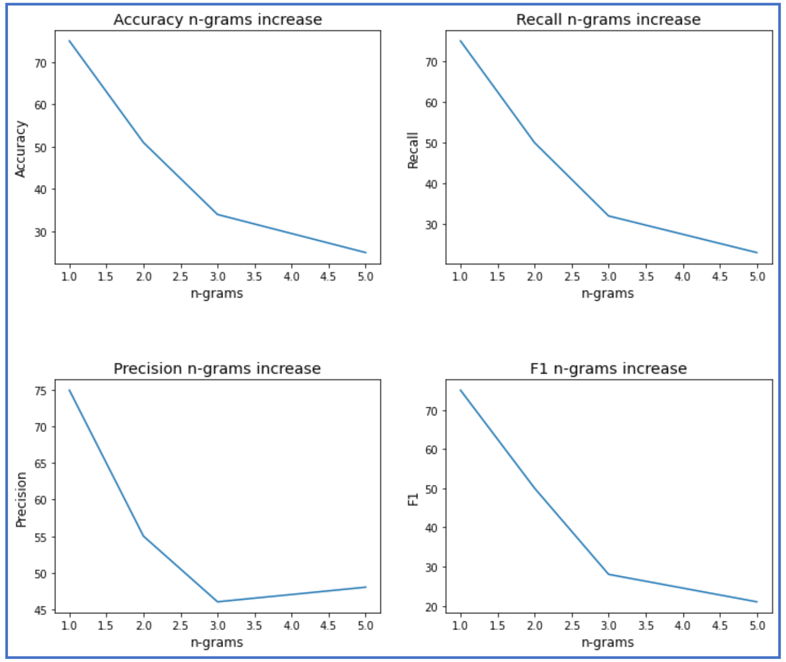
```
# Import Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.naive_bayes import MultinomialNB

import pandas as pd
import snscrape.modules.twitter as sntwitter
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


import nltk
```

Setup and Usage Instructions

12. Key Naïve Bayes Results

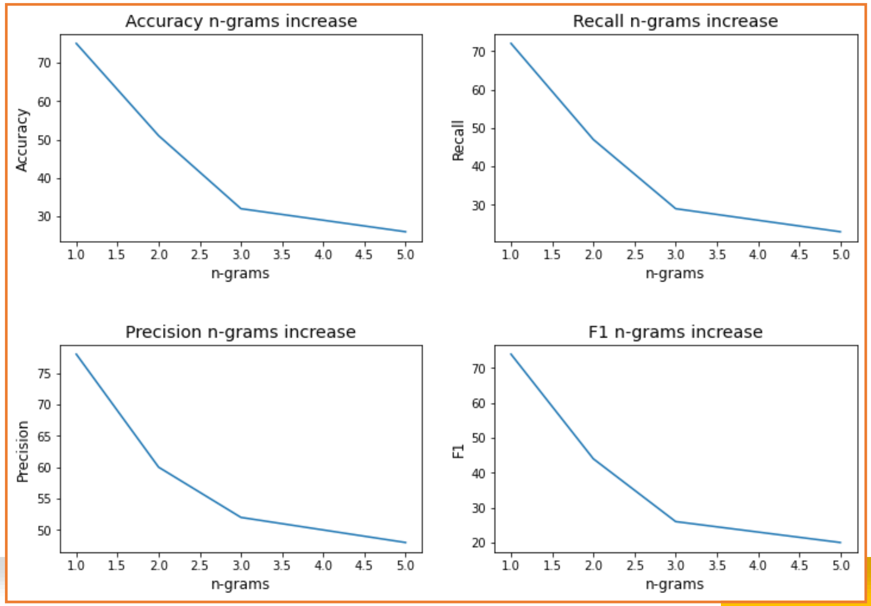


```
[5]  naivebayesCV()

...  Accuracy Score for ngrams = 1 is: 75.0%
     Recall for ngrams = 1 is: 75.0%
     Precision for ngrams = 1 is: 75.0%
     F1-score for ngrams = 1 is: 75.0%
     Accuracy Score for ngrams = 2 is: 51.0%
     Recall for ngrams = 2 is: 50.0%
     Precision for ngrams = 2 is: 55.00000000000001%
     F1-score for ngrams = 2 is: 50.0%
     Accuracy Score for ngrams = 3 is: 34.0%
     Recall for ngrams = 3 is: 32.0%
     Precision for ngrams = 3 is: 46.0%
     F1-score for ngrams = 3 is: 28.000000000000004%
     Accuracy Score for ngrams = 5 is: 25.0%
     Recall for ngrams = 5 is: 23.0%
     Precision for ngrams = 5 is: 48.0%
     F1-score for ngrams = 5 is: 21.0%
```



Naïve Bayes

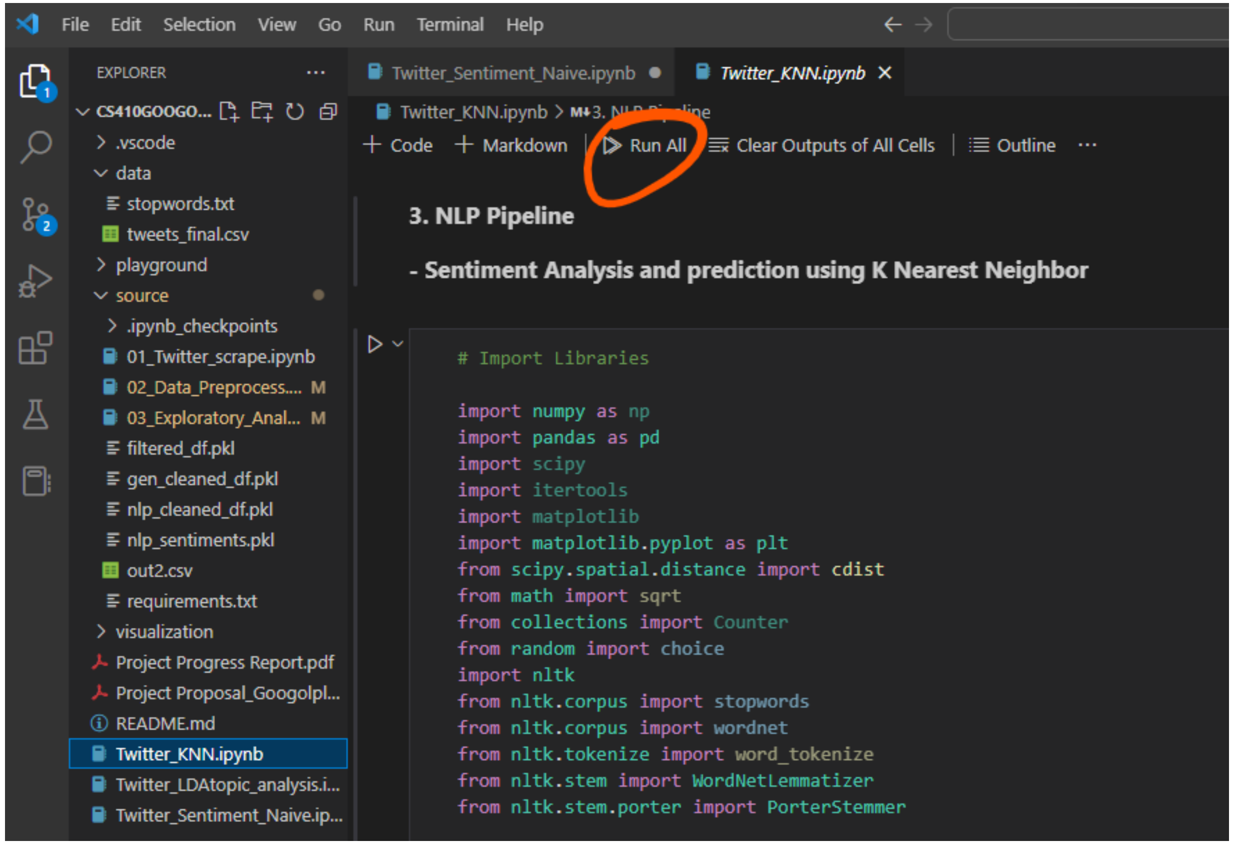
Naïve Bayes with TD-IDF



Setup and Usage Instructions

13. Open **Twitter_KNN.ipynb** and Run the Jupyter Notebook

11. Validate the successful run and results.



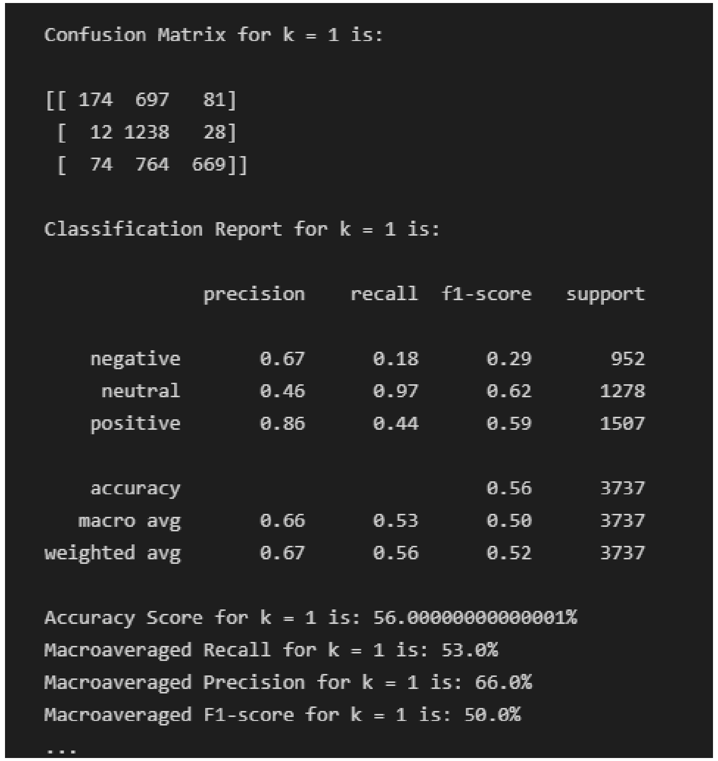
3. NLP Pipeline

- Sentiment Analysis and prediction using K Nearest Neighbor

```
# Import Libraries

import numpy as np
import pandas as pd
import scipy
import itertools
import matplotlib
import matplotlib.pyplot as plt
from scipy.spatial.distance import cdist
from math import sqrt
from collections import Counter
from random import choice
import nltk
from nltk.corpus import stopwords
from nltk.corpus import wordnet
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import PorterStemmer
```

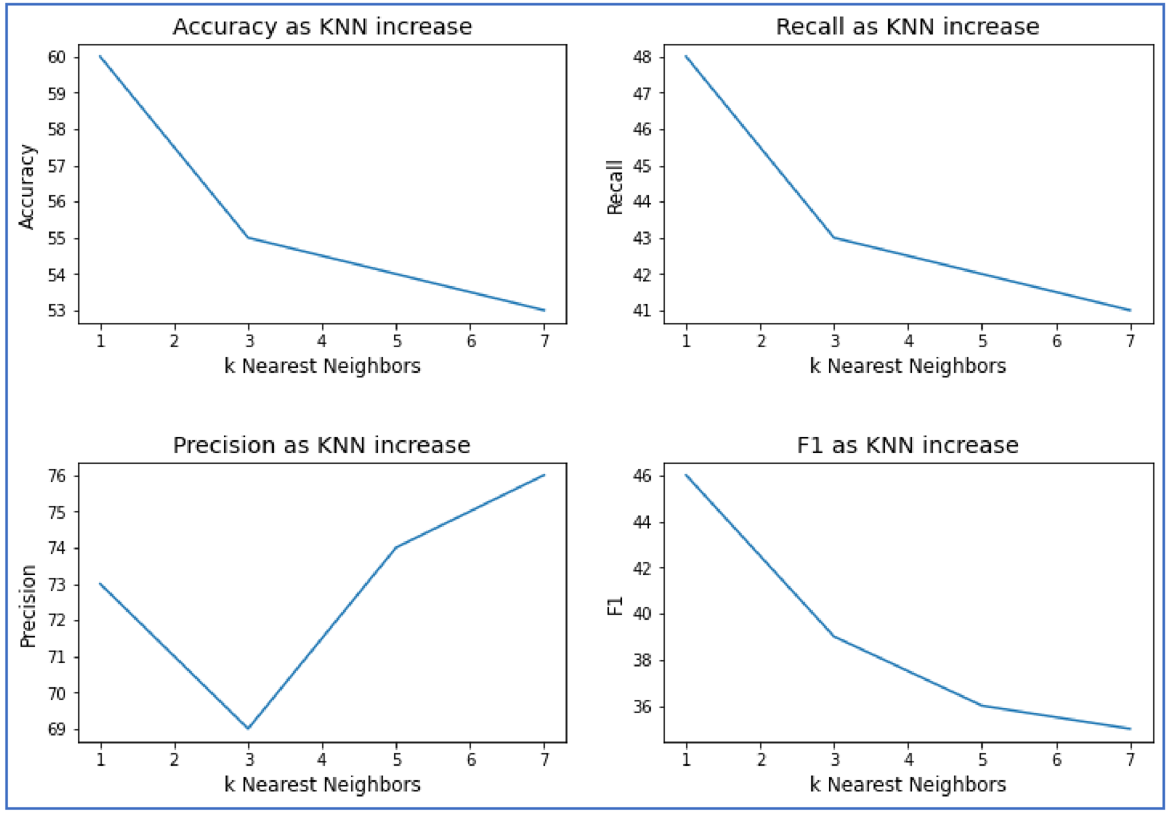
Setup and Usage Instructions

12. Key KNN Results



```
Confusion Matrix for k = 1 is:

[[ 174  697   81]
 [  12 1238   28]
 [  74  764  669]]

Classification Report for k = 1 is:

              precision    recall  f1-score   support

    negative       0.67      0.18      0.29       952
     neutral       0.46      0.97      0.62      1278
    positive       0.86      0.44      0.59      1507

    accuracy                           0.56      3737
   macro avg       0.66      0.53      0.50      3737
weighted avg       0.67      0.56      0.52      3737

Accuracy Score for k = 1 is: 56.00000000000001%
Macroaveraged Recall for k = 1 is: 53.0%
Macroaveraged Precision for k = 1 is: 66.0%
Macroaveraged F1-score for k = 1 is: 50.0%
...
```

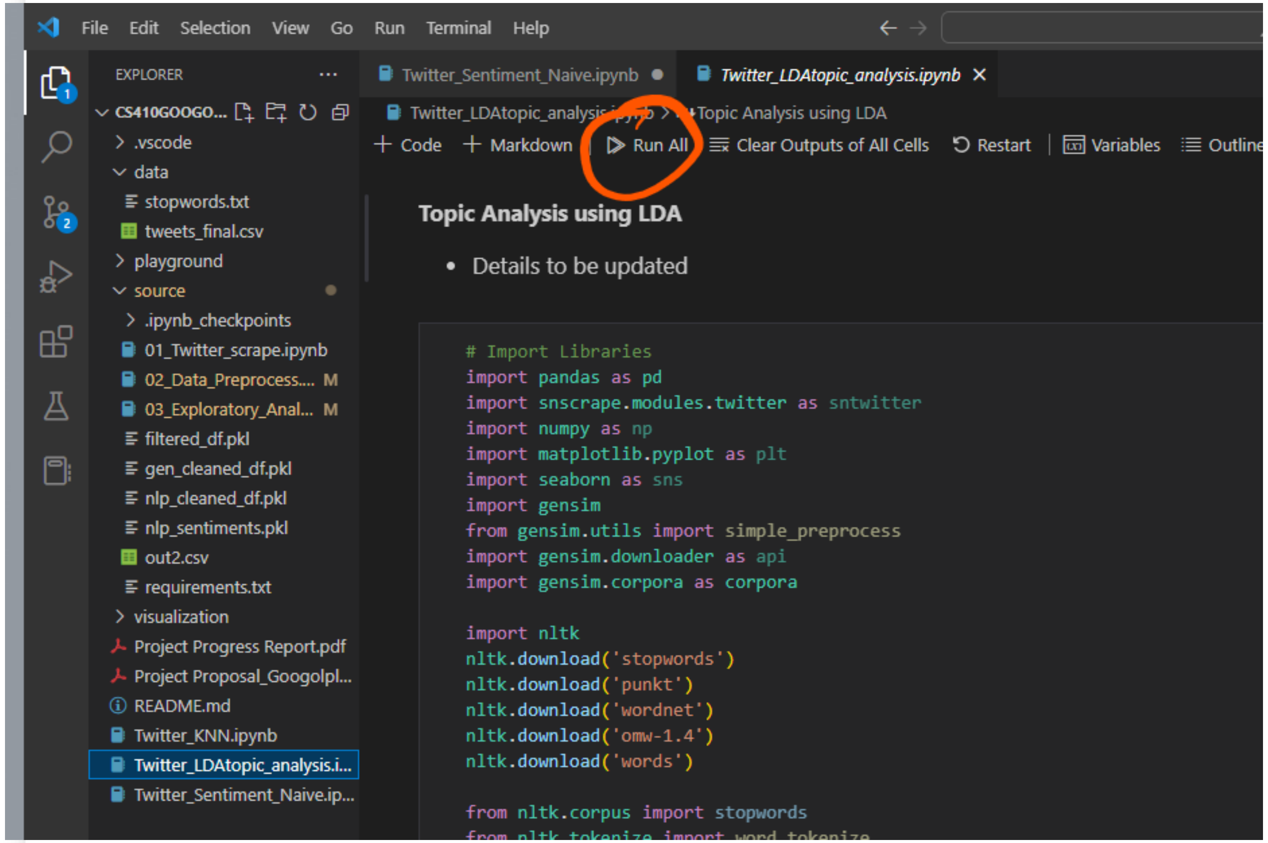
KNN



Setup and Usage Instructions

13. Open **Twitter_LDAtopic_Analysis.ipynb** and Run the Jupyter Notebook

11. Validate the successful run and results.



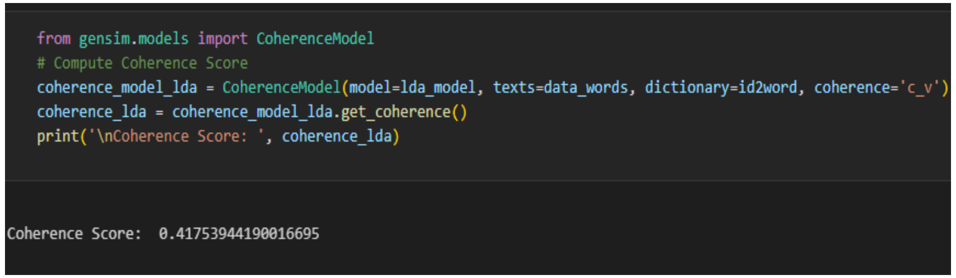
Topic Analysis using LDA

- Details to be updated

```
# Import Libraries
import pandas as pd
import snscrape.modules.twitter as sntwitter
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import gensim
from gensim.utils import simple_preprocess
import gensim.downloader as api
import gensim.corpora as corpora

import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('words')

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

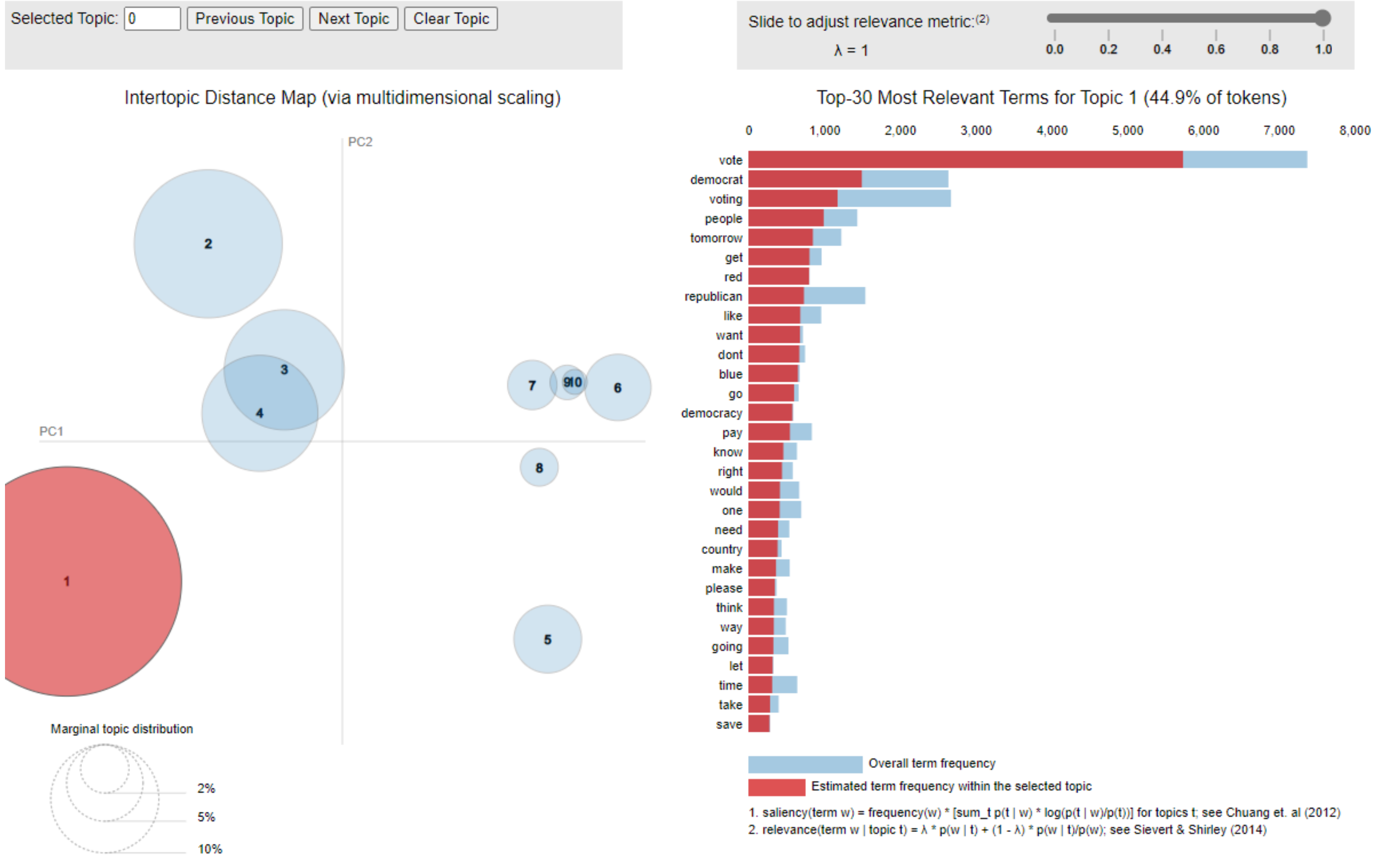
Setup and Usage Instructions

12. Key LDA Results



```
from gensim.models import CoherenceModel
# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_words, dictionary=id2word, coherence='c_v')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Coherence Score:  0.41753944190016695
```

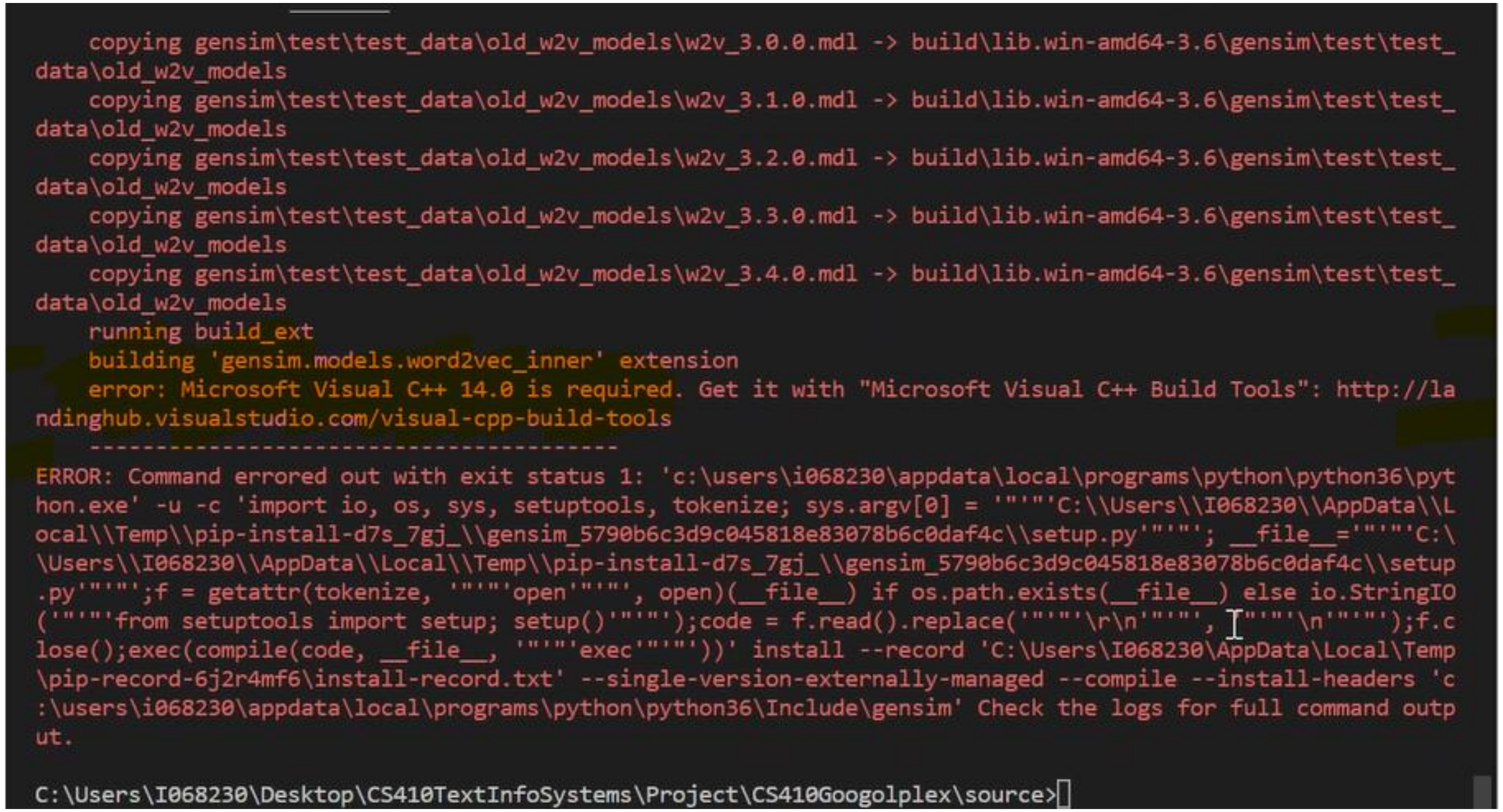


1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Expected Errors and Solutions

Error : Gensim fails due to missing C++ Build

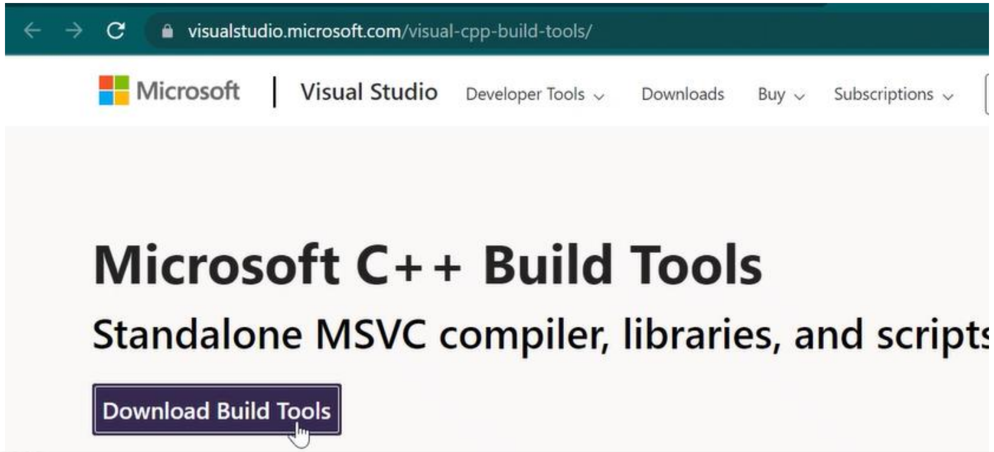
- Solution Install Microsoft Visual C++ Build Tools.



```
    copying gensim\test\test_data\old_w2v_models\w2v_3.0.0.mdl -> build\lib.win-amd64-3.6\gensim\test\test_
data\old_w2v_models
    copying gensim\test\test_data\old_w2v_models\w2v_3.1.0.mdl -> build\lib.win-amd64-3.6\gensim\test\test_
data\old_w2v_models
    copying gensim\test\test_data\old_w2v_models\w2v_3.2.0.mdl -> build\lib.win-amd64-3.6\gensim\test\test_
data\old_w2v_models
    copying gensim\test\test_data\old_w2v_models\w2v_3.3.0.mdl -> build\lib.win-amd64-3.6\gensim\test\test_
data\old_w2v_models
    copying gensim\test\test_data\old_w2v_models\w2v_3.4.0.mdl -> build\lib.win-amd64-3.6\gensim\test\test_
data\old_w2v_models
    running build_ext
    building 'gensim.models.word2vec_inner' extension
    error: Microsoft Visual C++ 14.0 is required. Get it with "Microsoft Visual C++ Build Tools": http://la
ndinghub.visualstudio.com/visual-cpp-build-tools
    ----------------------------------------
ERROR: Command errored out with exit status 1: 'c:\users\i068230\appdata\local\programs\python\python36\pyt
hon.exe' -u -c 'import io, os, sys, setuptools, tokenize; sys.argv[0] = '"'"'C:\\Users\\I068230\\AppData\\L
ocal\\Temp\\pip-install-d7s_7gj_\\gensim_5790b6c3d9c045818e83078b6c0daf4c\\setup.py'"'"'; __file__='"'"'C:\
\Users\\I068230\\AppData\\Local\\Temp\\pip-install-d7s_7gj_\\gensim_5790b6c3d9c045818e83078b6c0daf4c\\setup
.py'"'"';f = getattr(tokenize, '"'"'open'"'"', open)(__file__) if os.path.exists(__file__) else io.StringIO
('"'"'from setuptools import setup; setup()'"'"');code = f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.c
lose();exec(compile(code, __file__, '"'"'exec'"'"'))' install --record 'C:\Users\I068230\AppData\Local\Temp
\pip-record-6j2r4mf6\install-record.txt' --single-version-externally-managed --compile --install-headers 'c
:\users\i068230\appdata\local\programs\python\python36\Include\gensim' Check the logs for full command outp
ut.

C:\Users\I068230\Desktop\CS410TextInfoSystems\Project\CS410Googolplex\source>
```

Expected Errors and Solutions

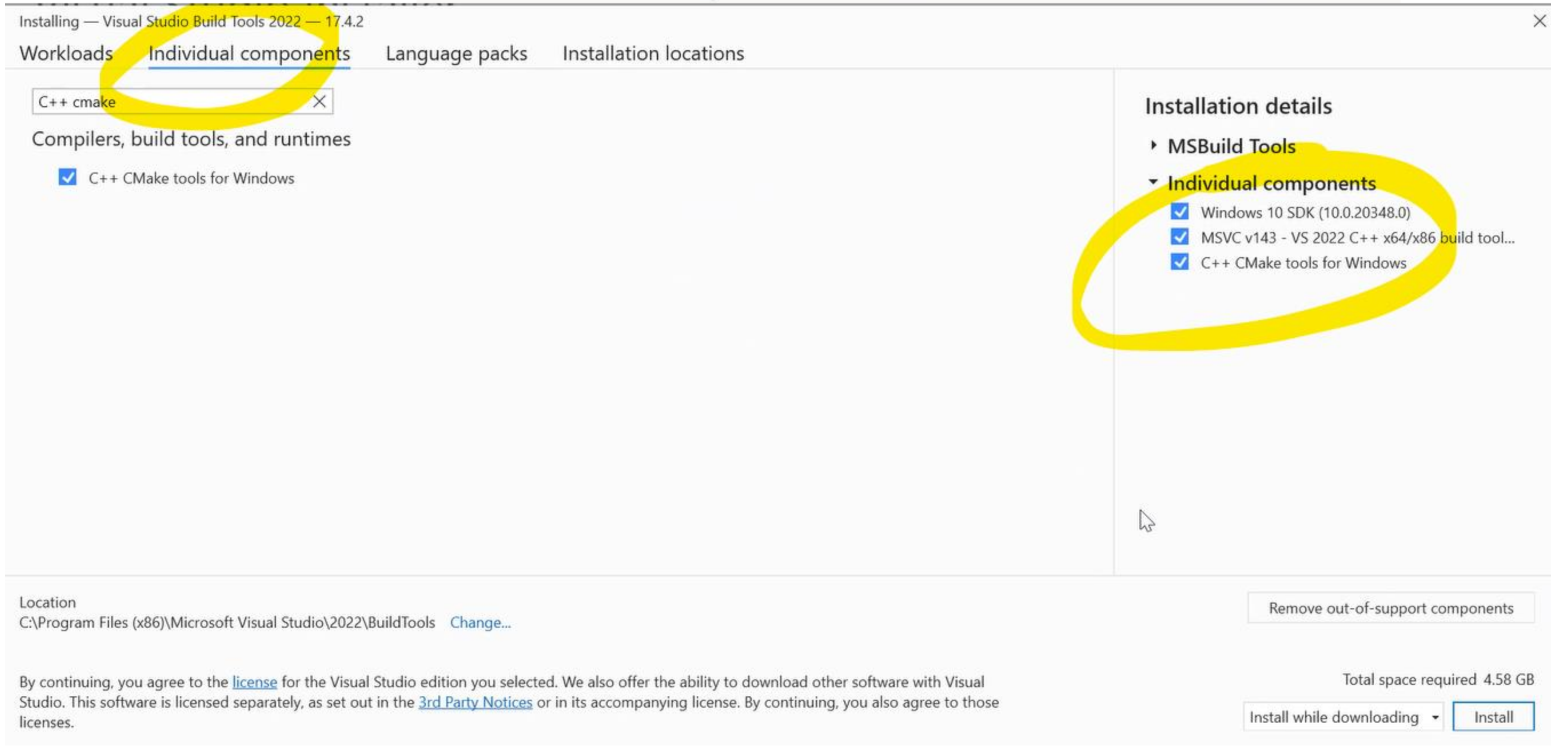
https://visualstudio.microsoft.com/visual-cpp-build-tools/



visualstudio.microsoft.com/visual-cpp-build-tools/
Microsoft | Visual Studio
Developer Tools
Downloads
Buy
Subscriptions

Microsoft C++ Build Tools

Standalone MSVC compiler, libraries, and scripts



Installing — Visual Studio Build Tools 2022 — 17.4.2

C++ cmake

Compilers, build tools, and runtimes

☑ C++ CMake tools for Windows

Installation details

- MSBuild Tools
- Individual components
  - ☑ Windows 10 SDK (10.0.20348.0)
  - ☑ MSVC v143 - VS 2022 C++ x64/x86 build tool...
  - ☑ C++ CMake tools for Windows

Location
C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 4.58 GB

Conclusion and future work

❑ - Accuracy improves if the data has been cleansed properly. We removed the words that are not a part of nltk.words. This helped in improving the accuracy for Naïve Bayes from 65% to 75%.

❑ - For KNN, difficult to build a feature vector of huge number of tweets.

❑ - For sentiment analysis on tweets data, we found Naïve Bayes to show better accuracy over K-NN.

❑ - It was more difficult to train K-NN as compared with Naïve Bayes. Hence, we conclude to use Naïve Bayes for sentiment analysis on Tweets data.

❑ - For future work, the same model could be evaluated for different subjects other than US midterm elections by simply scraping the data on another topic from Twitter using the code provided and plugging that data for Topic Modeling and Text Categorization and could be evaluated for accuracy.

❑ - Future work could also include modeling other text categorization techniques like SVM, Deep Learning (LSTM), etc. with word embedding techniques such as Word2vec could also be applied and compared for accuracy.

Thank you