Twitter Sentiment Analysis: Apple vs Google

(GROUP 3 PRESENTATION)

Group Members

- 1. Maureen Auka
- 2. Ochieng' Ouma
- 3. Brian Kipchumba
- 4. Andrew Chege

INTRODUCTION

- In today's highly competitive technology industry, customer perception and sentiment play a crucial role in shaping brand reputation and influencing purchasing decisions.

- Apple and Google are two of the most recognized technology companies worldwide, and their products often generate strong opinions on social media platforms such as Twitter.

Project Goal:

- The goal of this project is to build a Natural Language Processing (NLP) model that can automatically classify the sentiment of Tweets related to Apple and Google products.

- By analyzing over 9,000 Tweets labeled as positive, negative, or neutral, the model will provide insights into how consumers feel about these brands and their products.

Objective:

- To build a model that can rate the sentiments of a Tweet based on its content.

DATA

- The dataset employed in the study was downloaded from https://data.world/crowdflower/brands-and-product-emotions/file/judge-1377884607_tweet_product_company.csv

- The crowd was asked if the tweet expressed positive, negative, or no emotion towards a brand and/or product. If some emotion was expressed they were also asked to say which brand or product was the target of that emotion.

- The dataset was taken through a series of data preparation, cleaning, and processing as detailed on the project's methodology.

METHODOLOGY

The adopted structure for the project was CRISP-DM that entails:

A. Business Understanding;

B. Data Understanding (Data Preparation, Data Cleaning, and Exploratory Data Analysis (EDA);

C. Modelling;

A. Business understanding:

This proof-of-concept sentiment analysis has several potential business applications:

- ❖ **Brand Monitoring:** Track changes in public perception of Apple and Google products over time.
- ❖ **Product Feedback:** Identify common sources of positive and negative sentiment to guide product improvements.
- ❖ **Competitive Insights:** Compare sentiment trends between Apple and Google to inform competitive strategy.
- ❖ **Customer Engagement:** Enable real-time responses to negative customer experiences and amplify positive ones.

Ultimately, this project demonstrates how sentiment analysis can help organizations leverage unstructured text data from social media to make data-driven marketing, customer service, and product development decisions.

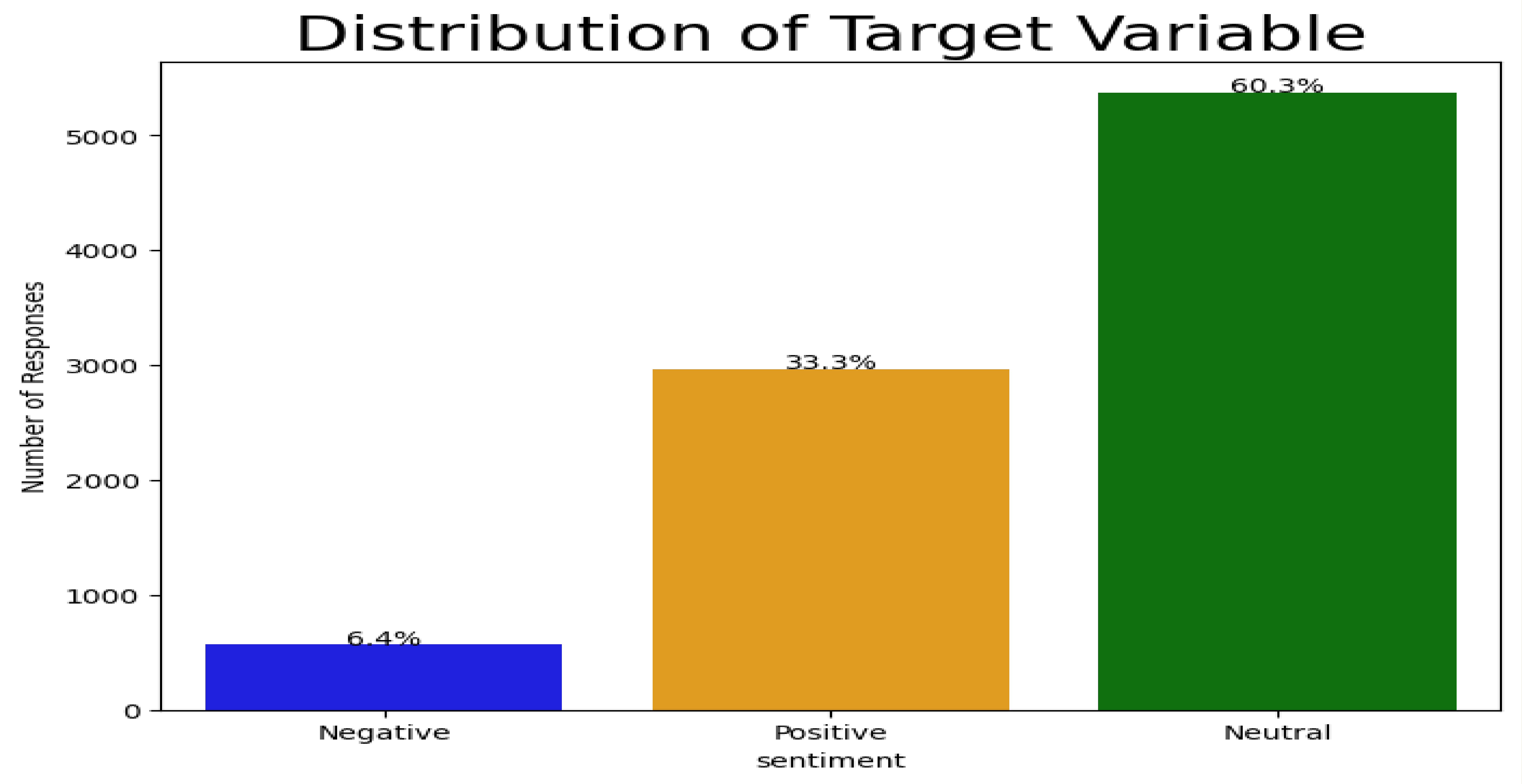
B. Data Understanding

Data Preparation & Cleaning;

- Started with importing standard packages as shown below:

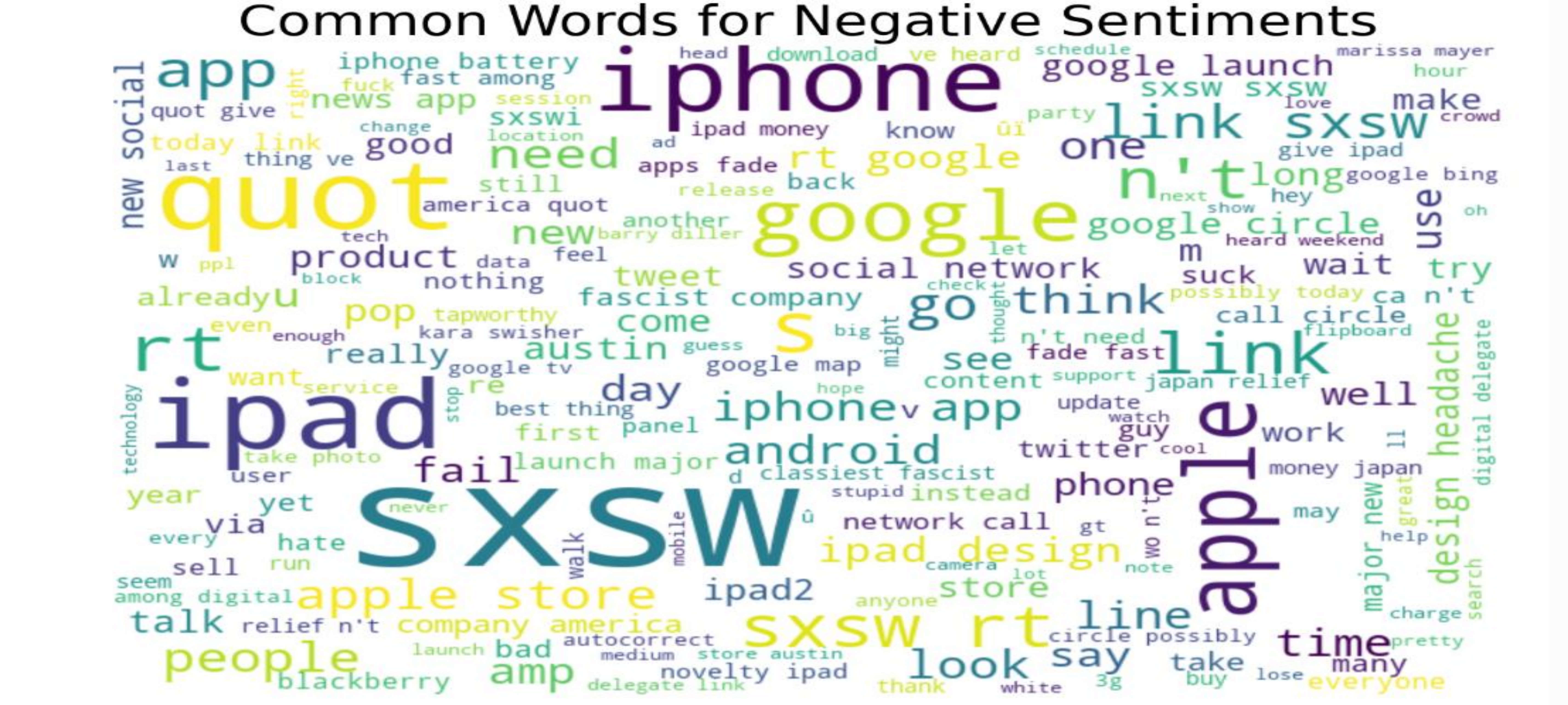
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from textwrap import fill

from sklearn.model_selection import train_test_split, cross_val_score, RepeatedStratifiedKFold,
GridSearchCV
import warnings
import math
import os
warnings.filterwarnings("ignore")

from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
```

Exploratory Data Analysis (EDA);



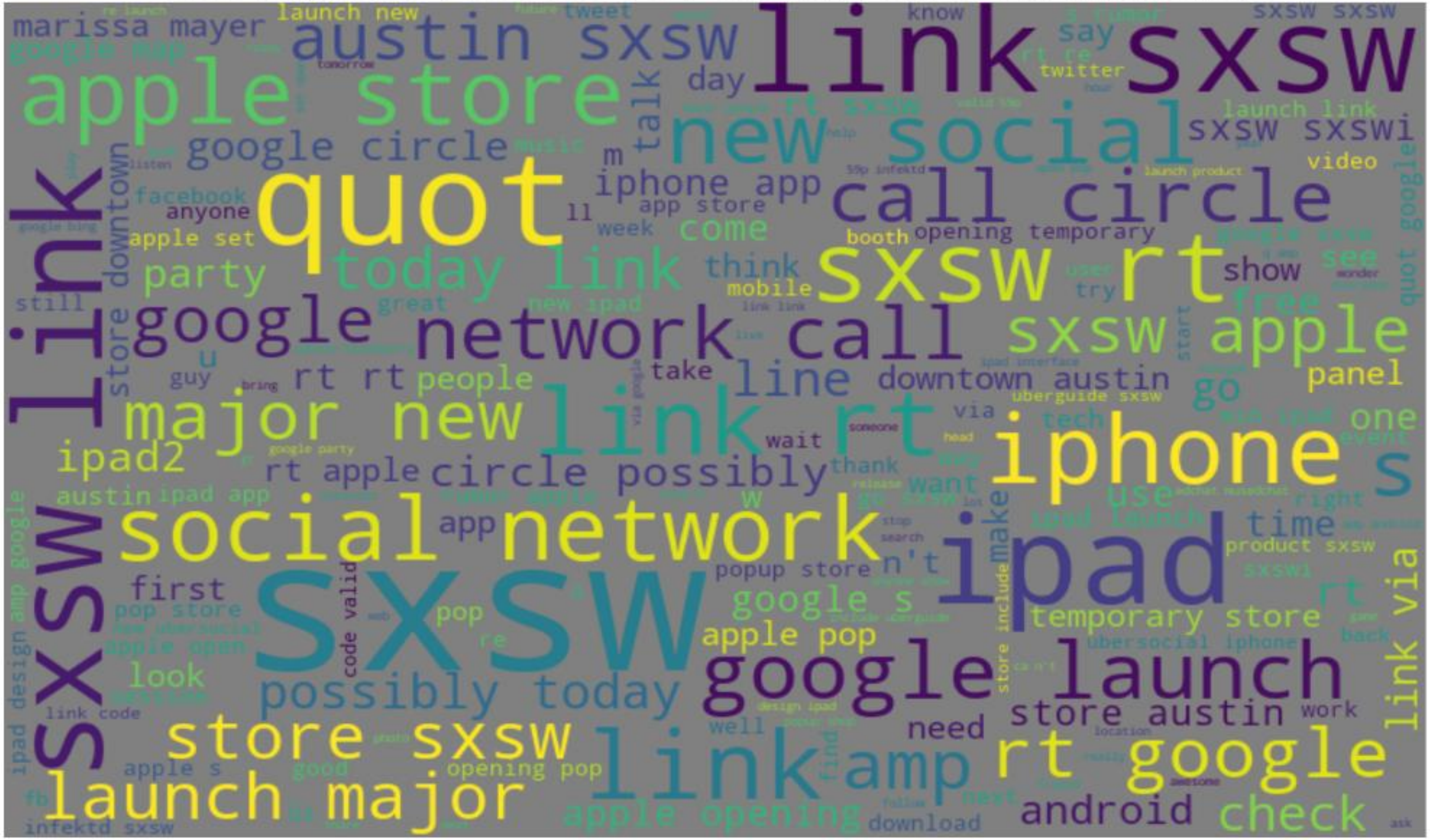
Most Common word in the Dataset:

- Frequently used words appear enlarged as compared to less frequently used words.



Common Words for Positive Sentiments

Common Words for Neutral Sentiments



C. Modelling

Data Processing:

- **split data into X and y;**

```
X = sentiment_data["cleaned_text"] #feature variable
y = sentiment_data["sentiment"] # target variable
```

- **data split into train and test;**

```
X_train, X_test, y_train,  y_test = train_test_split(X,y, test_size=0.25,
stratify=y, random_state=42,shuffle=True)
```

Models used:

i. Dummy Classifier Model

ii. Logistic Regression

iii. Decision Tree Model

iv. Random Forest

v. Naive Bayes and Support Vector Machine (SVM) models

Conclusion and Recommendations

- Logistic Regression and SVC improved recall for the negative class, making the model less biased toward neutral.

- Positive sentiment performance is moderate, while neutral remains strong.

- Next steps could include:
  - Further threshold optimization per class.
  - Feature engineering (e.g., combining char- and word-level n-grams).
  - Addressing class imbalance to improve negative and positive precision.

THANKS!

Does anyone have any question?