**1. Three Things to evaluate a Classification model.**

**Accuracy** - It is the ratio of true prediction versus the total number of predictions.

**Loss** - based on training set, it tells how wrong is our model. Loss is some go with a model that had balanced bias and variance. In the case of the classification task, we use binary cross-entropy, categorical cross-entropy, sparse categorical cross-entropy.

**Confusion matrix** - With the help of the confusion matrix we calculate different kinds of evaluation metrics, like precisely, precision, F1-score, etc.

**2. Transform a term to numerical representation.**

**Bag of words**

A BOW is represented by text that describes the occurrence of words within documents. Binary BOW contains one of the word is present any number of times is represented as 1 otherwise 0.

**TF-IDF(Term frequency-Inverse data frequency)**

TFIDF is a measure of how a word appears in a document divided by the total number of words in documents. It gives high weight to a word that appears many times in a document but also frequently in many documents. It gives low weight if the word appears in the document and frequently in the corpus. It gives low weight if the word appears in the document and frequently in the corpus.

by the number of the document that contains the word.

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

**Workflow :-**

```
```

```
https://www.kaggle.com/datasets/jillanisofttech/medium-articles-dataset
```

As The Print website had many news category and each category had hundreds of pages and each pages had 10 to 15 articles because of that only indicated 20 pages per category. Total 3000 articles were enough to extract 1000 articles.

```
!wget -q -O /tmp/chromedriver.zip https://chromedriver.storage.googleapis.com/114.0.5735.90/chromedriver_linux64.zip
!unzip -q -o /tmp/chromedriver.zip -d /tmp/
!cp /tmp/chromedriver /usr/bin/
!chmod +x /usr/bin/chromedriver
!pip -q install selenium webdriver_manager
import os, time, re, pandas as pd
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.exceptions import TimeoutException, NoSuchElementException, StaleElementReferenceException
from webdriver_manager.chrome import ChromeDriverManager
from datetime import datetime, timedelta
from collections import Counter, defaultdict
from urllib.parse import urlparse, unquote
import nltk, spacy, textstat
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from textblob import TextBlob
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
print("✅ Imports ready")
```

```
✅ Imports ready
```

```
# ── Setup ──
BASE = "https://theprint.in/category/politics/"
CHROME = "/usr/bin/chromedriver"
MAX_PAGES = 20
MIN_ARTICLES = 1000
OUT = "/tmp/theprint_politics.csv"

def new_driver():
    opts = Options()
    opts.binary_location = "/usr/bin/chromium-browser"
    opts.add_argument("--headless=new")
    opts.add_argument("--no-sandbox")
    opts.add_argument("--disable-dev-shm-usage")
    opts.add_argument("--disable-gpu")
    opts.add_argument("--window-size=1920,1080")
    opts.add_argument("--disable-blink-features=AutomationControlled")
    opts.add_argument("--user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36")
    return webdriver.Chrome(service=Service(CHROME), options=opts)

d = new_driver()
print("✅ Driver ready")
```

```
✅ Driver ready
```

```
# ── Collect article URLs from category pages ──
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.exceptions import TimeoutException, NoSuchElementException, StaleElementReferenceException
from webdriver_manager.chrome import ChromeDriverManager
from datetime import datetime, timedelta
from collections import Counter, defaultdict
from urllib.parse import urlparse, unquote
import nltk, spacy, textstat
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from textblob import TextBlob
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
print("✅ Imports ready")
# ── Setup ──
BASE = "https://theprint.in/category/politics/"
CHROME = "/usr/bin/chromedriver"
MAX_PAGES = 20
MIN_ARTICLES = 1000
OUT = "/tmp/theprint_politics.csv"

def new_driver():
    opts = Options()
    opts.binary_location = "/usr/bin/chromium-browser"
    opts.add_argument("--headless=new")
    opts.add_argument("--no-sandbox")
    opts.add_argument("--disable-dev-shm-usage")
    opts.add_argument("--disable-gpu")
    opts.add_argument("--window-size=1920,1080")
    opts.add_argument("--disable-blink-features=AutomationControlled")
    opts.add_argument("--user-agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36")
    return webdriver.Chrome(service=Service(CHROME), options=opts)

d = new_driver()
print("✅ Driver ready")
# ── Collect article URLs from category pages ──
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.exceptions import TimeoutException, NoSuchElementException, StaleElementReferenceException
from webdriver_manager.chrome import ChromeDriverManager
from datetime import datetime, timedelta
from collections import Counter, defaultdict
from urllib.parse import urlparse, unquote
import nltk, spacy, textstat
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from textblob import TextBlob
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
print("✅ Imports ready")
```

```
✅ Imports ready
```

```
```