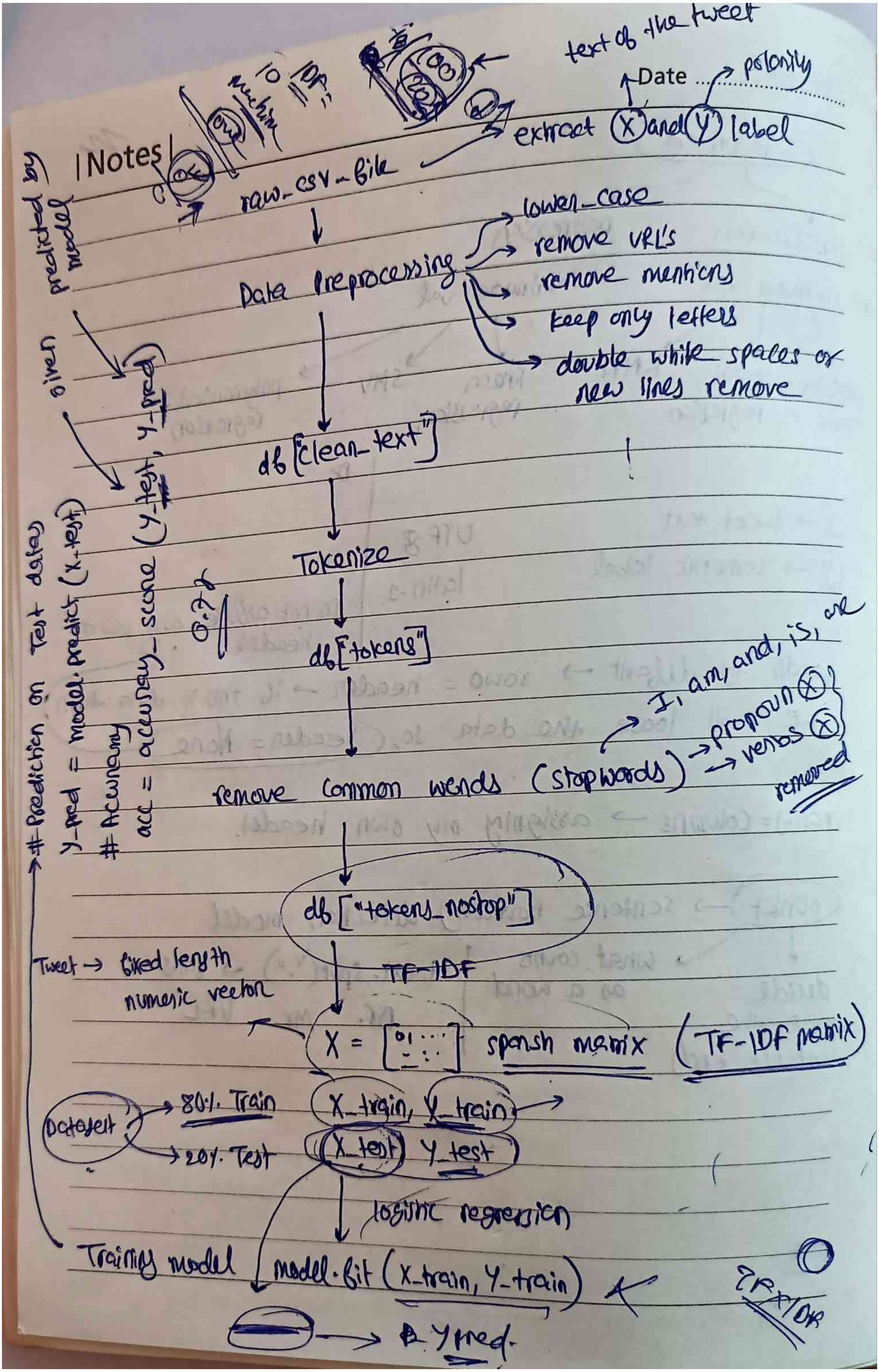
|Notes



raw_csv_file → extract (X) and (Y) label

X → Date ... text of the tweet

Y → polarity

Data Preprocessing
- lower_case
- remove URL's
- remove mentions
- keep only letters
- double white spaces or new lines remove

df["clean_text"]

Tokenize

df["tokens"]

remove common words (stopwords) → I, am, and, is, are
- pronoun (X)
- verbs (X)

removed

df["tokens_nostop"]

TF-IDF

Tweet → fixed length numeric vector

X = [0 1 ...] sparsh matrix (TF-IDF matrix)

Dataset → 80% Train, 20% Test

X_train, Y_train

X_test, Y_test

logistic regression

Training model model.fit (X_train, Y_train)

→ Y pred.

Prediction on Test data

Y_pred = model.predict (X_test)

Accuracy

acc = accuracy_score (Y_test, Y_pred)

given

predicted by model

0.78

1

TF-IDF (Term frequeny - Inverse Document frequeney)

↳ converts ~~text~~ sentence → vector of number while preserving meaning.

(s-1) vocabulary creation

↳ scans all tweets and builds dictionary of words.

love → 0 hate → 1 machine → 2 learning → 3 ⋮	TF-IDF shape (1,048,572, 15,000) ↑ Tweets ↑ unique words.

→ importance

$$\boxed{TF(\text{word}, \text{Tweet}) = \frac{\text{count of word in a tweet} \uparrow}{\text{total words in a tweet}}}$$

$$\boxed{IDF\left(\text{Inverse Document frequeny}\right) = \log\left(\frac{\text{total\_}\overset{\text{tweets}}{~}}{\text{tweets\_containing\_word} \uparrow}\right)}$$

→ How-rare

IDF↑ → rare words

TF↑ → important

TF×IDF

(i) love machine learning ≡ [0.32, 0.52, 0.52, 0.11]

max_features = 5000 (most frequent words)

=