AI & ML Workshop: Text Classifier

Student Lab Manual

Workshop Overview

In this hands-on workshop, you will build a simple machine learning model that classifies short text reviews, specifically *Yelp* reviews, as **positive** or **negative**.
Estimated Time: **1.5 - 2 hours**

You'll use `Python` in `Google Colab to`:

- Prepare and explore a text dataset.
- Clean and organize the data for training.
- Split examples into training and testing sets.
- Convert text into numerical features.
- Train a sentiment classification model.
- Evaluate how well the model performs.
- Experiment with your own review inputs.
- Save your trained model for reuse.

Getting Started

Please Read!

Important Reminder: To qualify for your **reward**, you must complete the entire workshop and carefully follow all instructions. Every reflection in the `Google Colab` must include a thoughtful response of at least **50 words**. Finally, make sure to complete the **Final Reflection** in the `Google Form` before submitting your work. Incomplete reflections or skipped steps may result in ineligibility for the reward.

Now, let us get started with this workshop! There are files that you need to download to your computer in order to continue. First, you need to get the `.ipynb` file for the `Google Colab`. One more thing that you will need is the dataset from *Yelp*.

Instruction

Navigate to our official website, csed-research-lab.org. On the navigation bar, click `Projects` and you can find the link to the `Google Colab`. Click the link and make a copy of this `Google Colab`.

Instruction

Below the link to the `Google Colab`, click the button to download the **CSV** file. You will need it for later in the workshop.

Step 0: Colab Setup

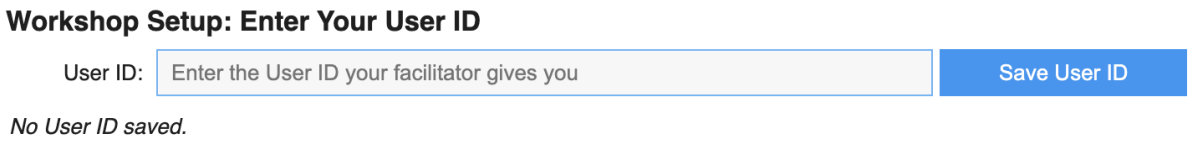
Before diving into building your model, let's get your workspace ready in Google Colab.

Please Read!

Important Reminder: Always click the **triangle play button** to run the cell *before* typing anything in the textbox. If you type first and then click the play button, your text will disappear, and you'll need to re-enter it. Running the cell first makes sure your reflection box works properly and saves your response when you click **Submit Reflection**.

Instruction

Run the cell labeled **Step 0.1: Setup (run once)** and **Step 0.2: Reflection helpers (run once)**. Enter your User ID and click the blue button **Save User ID**.

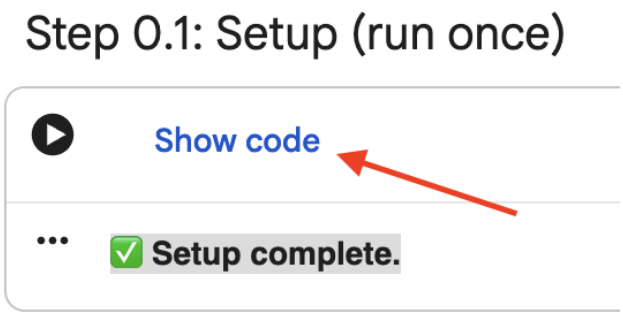


Instruction

Run the cell **Step 0.3** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on. *Notes:*

1. *Highlighted lines have arrows like this* `# <------------` .
2. *Click Show code for the cell to expand.*

```
try:
    import sklearn, pandas, numpy, matplotlib, joblib, seaborn    # <------------------ This Line
except Exception:
```



Reflect

Take a look at **Step 0.1**. Based on the import statements you've seen in the highlighted portion of **Step 0.1**, what do you think the code is doing? Which parts look familiar or new?

Step 1: Loading and Cleaning Yelp data

Now that your setup is ready, it's time to meet your data — the foundation of every machine learning project!

Instruction

Perfect! Moving on to **Step 1.1** and **Step 1.2**. Run these cells and fill in your reflections. Remember to click the green button **Submit Reflection** before moving on.

Reflect

After you download the **CSV** file, open it with **Excel**. What do you see inside? Now, open it with **Notepad**. Now what do you see? What do you think is happening?
If you had to classify a review as positive or negative yourself, what words or phrases would you look for?

Instruction

Examine then run the code inside **Step 1.4** and **Step 1.5**, which starts out like the code below. Use the **CSV** file you downloaded at the beginning of the workshop to upload. Then, run **Step 1.6** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

```
#@title Step 1.4: Upload or Use Sample Data
import pandas as pd, io
from IPython.display import display
try:
    from google.colab import files
```

```
    USING_COLAB = True
...
#@title Step 1.5: Cleaning the dataset
import pandas as pd
valid = {'positive','negative','pos','neg','1','0','true','false'}
df = df.rename(columns={c:c.lower() for c in df.columns})
...
```

Reflect

What do you think the cleaning code is doing? Hint: Look at the highlighted lines in **Step 1.4 and 1.5**. Why do you think we need to normalize labels and remove duplicates?

Step 2: Split train/test

In this step, you'll explore a key idea in machine learning — how models learn and how we test what they've learned.

Instruction

Examine then run the code in **Step 2.1** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

```
#@title Step 2.1: Train/test split
import numpy as np
from sklearn.model_selection import train_test_split
...
```

Reflect

What do you think this code is doing? Why do you think we split the data into two groups?

Step 3: TF-IDF (Term Frequency–Inverse Document Frequency)

In this step, you'll discover how computers can understand raw text and make machine learning with language possible.

Instruction

Examine then run the code in **Step 3.1** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

```
#@title Step 3.1: TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer
...
```

Reflect

What do you think TF-IDF is doing? What do you think 'Term Frequency' and 'Inverse Document Frequency' might mean?

Step 4: Training

Now it's time for the exciting part — teaching your model how to think!

Instruction

Examine then run the code in **Step 4.1** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

Reflect

How is using a trained model to make predictions different from writing explicit rules to classify text?

Instruction

Run the code in **Step 4.2** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

```
#@title Step 4.2: Fit classifiers
from sklearn.svm import LinearSVC
...
```

Reflect

What do you think happens when we call .fit()? What do you think the model is learning?

Step 5: Evaluate your model(s)

You've trained your model — now it's time to see how well it actually performs!

Instruction

Examine then run the code in **Step 5.1** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

```
#@title Step 5.1: Evaluation
from sklearn.metrics import accuracy_score, classification_report,
    ↪ confusion_matrix
import matplotlib.pyplot as plt, numpy as np, itertools
import seaborn as sns
...
```

Reflect

What do you think is happening in Step 5.1? Why is it necessary to evaluate our model?

Step 6. Try it

You did it — your model is built, trained, and ready to take the stage! Now it's your turn to see how smart your AI really is.

Instruction

Congrats on reaching this step. This means that you have finished building your own model. Now, let's have some fun testing your own model. Run **Step 6.1: Predict your own review**, write a sample Yelp review, then click the button **Predict**. After that, run **Step 6.2** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

Reflect

What was the review you wrote. Was it classified correctly when you hit Predict?

Step 7: Save the model

You've built, trained, and tested your sentiment model — now it's time to make sure your hard work doesn't disappear!

Instruction

Examine then run the code in **Step 7.1** and fill in your reflection. Remember to click the green button **Submit Reflection** before moving on.

```
#@title Step 7.1: Save model
import joblib
...
```

Reflect

Why do you think it is important to save our model?

Final Reflection

You are almost done with this workshop! The only thing left for you to do is to fill out the final reflection into the google form posted in our official website.

Instruction

Navigate to our official website, csed-research-lab.org. On the navigation bar, click `Projects` and you can find the links there besides the links to `Google Colab`.