```
# prompt: from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from scikeras.wrappers import KerasClassifier
import joblib
import os
# write all the installing statment
```

```
# prompt: I want to install import time
# import os
# import re
# import string
# import cv2
# import numpy as np
# import pandas as pd
# import tensorflow as tf
# from sklearn.model_selection import train_test_split
# from sklearn.feature_extraction.text import TfidfVectorizer
# from sklearn.linear_model import LogisticRegression
# from sklearn.ensemble import RandomForestClassifier, VotingClassifier
# from sklearn.naive_bayes import MultinomialNB
# from sklearn.metrics import classification_report, accuracy_score
# from tensorflow.keras.applications import VGG16
# from tensorflow.keras.layers import Dense, Flatten, BatchNormalization, Dropc
# from tensorflow.keras.models import Model
# from tensorflow.keras.optimizers import Adam
# all of this

!pip install opencv-python
!pip install scikit-learn
!pip install tensorflow
!pip install joblib
!pip install tensorflow
!pip install scikeras
!pip install joblib

import time
import os
import re
import string
import cv2
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, accuracy_score
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, Flatten, BatchNormalization, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from scikeras.wrappers import KerasClassifier
import joblib
import os
```

```
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.11
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/pyt
Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/di
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.1
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/pyt
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /us
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/pyth
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dis
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/pyth
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/di
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/pyth
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/
Requirement already satisfied: numpy<2.1.0,>=1.26.0 in /usr/local/lib/pyt
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /u
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/pytho
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/pytho
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/pytho
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/py
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/di
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.1
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/pyt
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /us
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/pyth
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dis
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/pyth
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/di
```

```
from google.colab import files
files.upload()
```

Choose Files no files selected Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle.json
{'kaggle.json':
```

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

```
!kaggle datasets download -d divg07/casia-20-image-tampering-detection-dataset
```

```
Dataset URL: https://www.kaggle.com/datasets/divg07/casia-20-image-tamperin
License(s): unknown
```

```
import pandas as pd
import zipfile

# Extract the contents of the zip file to a folder
with zipfile.ZipFile('/content/casia-20-image-tampering-detection-dataset.zip',
    zip_ref.extractall('/content/CASIA2') # Extract to a directory named CASIA2

# Now you can access the images within the extracted folder:
# tampered_dir = "/content/CASIA2/Tp"  # Path to tampered images
# authentic_dir = "/content/CASIA2/Au"  # Path to authentic images
# ... (rest of your code to load and process images)


!mkdir -p /content/image_data/authentic
!mkdir -p /content/image_data/tampered

!mv /content/CASIA2/CASIA2/Au/* /content/image_data/authentic/
!mv /content/CASIA2/CASIA2/Tp/* /content/image_data/tampered/



print("its running")
import time
start = time.time()
import pandas as pd
import csv
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os
import cv2
import re
import string
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, accuracy_score, confusion_mat
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import joblib
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from scikeras.wrappers import KerasClassifier
import joblib
import os
from tensorflow.keras.models import Sequential # Import the Sequential model



# --- IMAGE-BASED FAKE NEWS DETECTION MODEL ---
# Set paths for the dataset
# Data generator (on-the-fly loading)




image_size = (224, 224)
batch_size = 32
epochs = 15

# Define data directories
base_dir = "/content/image_data"

# Data augmentation and rescaling
datagen = ImageDataGenerator(
    rescale=1.0/255,
    validation_split=0.2
)

# Train and validation generators
train_generator = datagen.flow_from_directory(
    base_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='binary',
    subset='training'
)

val_generator = datagen.flow_from_directory(
    base_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='binary',
    subset='validation'
)

# Load pre-trained VGG16 model
vgg16 = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Freeze pre-trained layers
for layer in vgg16.layers:
```

```
for layer in vgg16.layers:
    layer.trainable = False

# Add custom layers on top of VGG16
x = Flatten()(vgg16.output)
x = Dense(1024, activation='relu')(x)
x = Dense(512, activation='relu')(x)
output = Dense(1, activation='sigmoid')(x)

# Build and compile the model
image_model = Model(inputs=vgg16.input, outputs=output)
image_model.compile(optimizer=Adam(learning_rate=0.0001), loss='binary_crossentr

# Train the model using generators
image_model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=epochs
)

# Save model in HDF5 format
# Save in native Keras format (recommended for Keras 3)
image_model.save('/content/image_model.keras')  # DON'T use .h5
print("Model saved as image_model.keras")

# Optional: Save model path using joblib
import joblib
joblib.dump({'model_path': '/content/image_model.keras'}, '/content/image_model.
print("Image model path saved to image_model.pkl")

# Optional: download files (if needed)
from google.colab import files
files.download('/content/image_model.keras')
files.download('/content/image_model.pkl')
```

```
its running
Found 10092 images belonging to 2 classes.
Found 2522 images belonging to 2 classes.
Epoch 1/15
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py
  self._warn_if_super_not_called()
316/316 ━━━━━━━━━━━━━━━━━━━━ 73s 224ms/step - accuracy: 0.6064 - loss: 0.67
Epoch 2/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 69s 217ms/step - accuracy: 0.7094 - loss: 0.55
Epoch 3/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 68s 216ms/step - accuracy: 0.7487 - loss: 0.50
Epoch 4/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 69s 217ms/step - accuracy: 0.7820 - loss: 0.44
Epoch 5/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 83s 219ms/step - accuracy: 0.7962 - loss: 0.42
Epoch 6/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 68s 216ms/step - accuracy: 0.8130 - loss: 0.39
Epoch 7/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 75s 238ms/step - accuracy: 0.8222 - loss: 0.36
Epoch 8/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 76s 241ms/step - accuracy: 0.8352 - loss: 0.34
Epoch 9/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 69s 218ms/step - accuracy: 0.8477 - loss: 0.33
Epoch 10/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 75s 238ms/step - accuracy: 0.8401 - loss: 0.33
Epoch 11/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 69s 218ms/step - accuracy: 0.8508 - loss: 0.31
Epoch 12/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 68s 216ms/step - accuracy: 0.8643 - loss: 0.28
Epoch 13/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 69s 219ms/step - accuracy: 0.8667 - loss: 0.28
Epoch 14/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 68s 216ms/step - accuracy: 0.8690 - loss: 0.27
Epoch 15/15
316/316 ━━━━━━━━━━━━━━━━━━━━ 68s 217ms/step - accuracy: 0.8689 - loss: 0.27
Model saved as image_model.keras
Image model path saved to image_model.pkl
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Start coding or generate with AI.

```
from tensorflow.keras.callbacks import EarlyStopping
path = '/content/'

df_fake = pd.read_csv(path + "Fake.csv", header=None, delimiter="\t", quoting=c
df_true = pd.read_csv(path + "True.csv", encoding='utf-8', engine='python', on_

# Label and merge
df_fake["class"] = 0
df_true["class"] = 1
df_text = pd.concat([df_fake, df_true], axis=0).sample(frac=1).reset_index(drop
df_text = df_text.drop(["title", "subject", "date"], axis=1)

# Preprocessing
def wordopt(text):
    text = str(text).lower()
    text = re.sub(r'\[.*?\]', '', text)
    text = re.sub(r'\\W', ' ', text)
    text = re.sub(r'https?://\S+|www\.\S+', '', text)
    text = re.sub(r'<.*?>+', '', text)
    text = re.sub(r'[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub(r'\n', '', text)
    text = re.sub(r'\w*\d\w*', '', text)
    return text

df_text["text"] = df_text["text"].apply(wordopt)

# Train-test split
x_text = df_text["text"]
y_text = df_text["class"]
x_train_text, x_test_text, y_train_text, y_test_text = train_test_split(x_text,

# TF-IDF vectorization
vectorizer = TfidfVectorizer(max_features=5000)
xv_train_text = vectorizer.fit_transform(x_train_text).toarray()
xv_test_text = vectorizer.transform(x_test_text).toarray()

# Build Keras model
model = Sequential([
    Dense(512, activation='relu', input_shape=(xv_train_text.shape[1],)),
    Dropout(0.3),
    Dense(256, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'

# Train
model.fit(xv_train_text, y_train_text, epochs=5, batch_size=32, validation_spli

# Evaluate
loss, acc = model.evaluate(xv_test_text, y_test_text)
```

```
print(f"Text Classification Accuracy (Keras): {acc:.4f}")

# Save model and vectorizer
model.save("/content/text_model.h5")
joblib.dump(vectorizer, "/content/vectorizer.pkl")
print("Keras model saved as text_model.h5 and vectorizer as vectorizer.pkl")
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87:
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/5
948/948 ━━━━━━━━━━━━━━━━━━━━ 9s 6ms/step - accuracy: 0.9980 - loss: 0.0428
Epoch 2/5
948/948 ━━━━━━━━━━━━━━━━━━━━ 3s 3ms/step - accuracy: 1.0000 - loss: 7.3672e
Epoch 3/5
948/948 ━━━━━━━━━━━━━━━━━━━━ 3s 3ms/step - accuracy: 1.0000 - loss: 2.6187e
Epoch 4/5
948/948 ━━━━━━━━━━━━━━━━━━━━ 3s 3ms/step - accuracy: 1.0000 - loss: 2.3647e
Epoch 5/5
948/948 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 1.0000 - loss: 4.8187e
351/351 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 1.0000 - loss: 1.5118e
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` o
Text Classification Accuracy (Keras): 1.0000
Keras model saved as text_model.h5 and vectorizer as vectorizer.pkl
```

```
# Create a new test generator
test_datagen = ImageDataGenerator(rescale=1./255)

# Assuming your test images are in '/content/test_images/'
test_generator = test_datagen.flow_from_directory(
    '/content/image_data',            # You MUST have subfolders for each class
    target_size=image_size,
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False
)
y_test = test_generator.classes
y_test_hybrid = y_test[:min_len]
```

```
Found 12614 images belonging to 2 classes.
```

```
!pip install streamlit tensorflow opencv-python pillow
import numpy as np
import pandas as pd
from tensorflow.keras.models import load_model
import joblib
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score
```

```
# Load saved models and vectorizer
text_model = load_model("text_model.h5")
vectorizer = joblib.load("vectorizer.pkl")
image_model = load_model("image_model.keras")

# Text preprocessing
xv_test_text = vectorizer.transform(x_test_text).toarray()
text_preds = text_model.predict(xv_test_text).flatten()  # Probabilities

# Image predictions (probabilities)
image_preds = image_model.predict(test_generator, verbose=1).flatten()

# Check shapes match
min_len = min(len(text_preds), len(image_preds))
text_preds = text_preds[:min_len]
image_preds = image_preds[:min_len]

# Combine predictions as hybrid features
hybrid_features = np.vstack((text_preds, image_preds)).T  # Shape: (N, 2)

# Create corresponding labels
# (you must align y_test with these samples — below assumes y_test is already a
y_test_hybrid = y_test[:min_len]

# Train a classifier on hybrid features
clf = LogisticRegression()
clf.fit(hybrid_features, y_test_hybrid)

# Predict and evaluate
hybrid_preds = clf.predict(hybrid_features)
acc = accuracy_score(y_test_hybrid, hybrid_preds)

print(f"\n✅ Hybrid Model Accuracy: {acc:.4f}")
print("\nClassification Report:\n", classification_report(y_test_hybrid, hybric
```

```
Collecting streamlit
  Downloading streamlit-1.44.1-py3-none-any.whl.metadata (8.9 kB)
Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/di
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.1
Requirement already satisfied: blinker<2,>=1.0.0 in /usr/local/lib/python
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/pytho
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.11
Requirement already satisfied: numpy<3,>=1.23 in /usr/local/lib/python3.1
Requirement already satisfied: packaging<25,>=20 in /usr/local/lib/python
Requirement already satisfied: pandas<3,>=1.4.0 in /usr/local/lib/python3
Requirement already satisfied: protobuf<6,>=3.20 in /usr/local/lib/python
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.11/
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python
Requirement already satisfied: tenacity<10,>=8.1.0 in /usr/local/lib/pyth
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.
Requirement already satisfied: typing-extensions<5,>=4.4.0 in /usr/local/
Collecting watchdog<7,>=2.1.5 (from streamlit)
  Downloading watchdog-6.0.0-py3-none-manylinux2014_x86_64.whl.metadata (
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.3/44.3 kB 2.3 MB/s eta 0
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in /usr/local
Collecting pydeck<1,>=0.8.0b4 (from streamlit)
  Downloading pydeck-0.9.1-py2.py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.1
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/pyt
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /us
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/pyth
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/di
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/pyth
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /u
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.
Requirement already satisfied: narwhals>=1.14.2 in /usr/local/lib/python3
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/pytho
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/p
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.1
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/pytho
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/pytho
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.
```

Start coding or generate with AI.

```
!streamlit run /usr/local/lib/python3.11/dist-packages/colab_kernel_launcher.py
```

```
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.28.0.12:8501
  External URL: http://34.83.135.107:8501

  Stopping...
  Stopping...
```