--- title: Final Machine Learning Project author: Azan & PB date: '2025-04-12' image: "image.jpg" description: "A blog post about our Final Machine learning project... " format: html ---

Necessary Imports:

In [16]:
```
# Install kagglehub if not already installed
%pip install kagglehub
%pip install latex

import kagglehub
import numpy as np
import pandas as pd
import re
import scipy
import seaborn as sns
import matplotlib.pyplot as plt

# Import the dataset
philipjames11_dark_net_marketplace_drug_data_agora_20142015_path = kagglehub.dataset_download('philipjames11/da
print('Data Import Success!')
```

```
Requirement already satisfied: kagglehub in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (0.3
.11)
Requirement already satisfied: packaging in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (fro
m kagglehub) (24.2)
Requirement already satisfied: pyyaml in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (from k
agglehub) (6.0.2)
Requirement already satisfied: requests in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (from
kagglehub) (2.32.3)
Requirement already satisfied: tqdm in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (from kag
glehub) (4.67.1)
Requirement already satisfied: charset_normalizer<4,>=2 in /Applications/anaconda3/envs/ml-045/lib/python3.9/sit
e-packages (from requests->kagglehub) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (
from requests->kagglehub) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-pack
ages (from requests->kagglehub) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-pack
ages (from requests->kagglehub) (2025.1.31)
Note: you may need to restart the kernel to use updated packages.
Collecting latex
  Downloading latex-0.7.0.tar.gz (6.5 kB)
  Preparing metadata (setup.py) ... done
Collecting tempdir (from latex)
  Downloading tempdir-0.7.1.tar.gz (5.9 kB)
  Preparing metadata (setup.py) ... done
Collecting data (from latex)
  Downloading data-0.4.tar.gz (7.0 kB)
  Preparing metadata (setup.py) ... done
Collecting future (from latex)
  Downloading future-1.0.0-py3-none-any.whl.metadata (4.0 kB)
Collecting shutilwhich (from latex)
  Downloading shutilwhich-1.1.0.tar.gz (2.3 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: six in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (from data
->latex) (1.17.0)
Requirement already satisfied: decorator in /Applications/anaconda3/envs/ml-045/lib/python3.9/site-packages (fro
m data->latex) (5.1.1)
Collecting funcsigs (from data->latex)
  Downloading funcsigs-1.0.2-py2.py3-none-any.whl.metadata (14 kB)
Downloading future-1.0.0-py3-none-any.whl (491 kB)
Downloading funcsigs-1.0.2-py2.py3-none-any.whl (17 kB)
Building wheels for collected packages: latex, data, shutilwhich, tempdir
  Building wheel for latex (setup.py) ... done
  Created wheel for latex: filename=latex-0.7.0-py3-none-any.whl size=7631 sha256=6ffaf07e41be2cb6bbb07143cfc839
28910fbe13b9a0bcec8a5daaf44a78c878
  Stored in directory: /Users/blank/Library/Caches/pip/wheels/94/84/e5/5ce582523fd479d00356867953085a67c47fbbc86
506aa92f8
  Building wheel for data (setup.py) ... done
  Created wheel for data: filename=data-0.4-py3-none-any.whl size=7272 sha256=8f0c07e57ce583754173702cab64585151
b247e6dd11f2eb839ef322be5de675
  Stored in directory: /Users/blank/Library/Caches/pip/wheels/8a/0b/a3/37ca07d5a2838bba2e475e8090455e40b94631bd5
7a99a35f4
  Building wheel for shutilwhich (setup.py) ... done
  Created wheel for shutilwhich: filename=shutilwhich-1.1.0-py3-none-any.whl size=2803 sha256=040d3efd78d6e5fa3a
59769172454cef143c6a2ae0c925eeff793f574e16fb81
  Stored in directory: /Users/blank/Library/Caches/pip/wheels/84/c7/f5/fed66dce1ed897b44e0da776b6a592dfad0a70f7d
d61f73a9d
  Building wheel for tempdir (setup.py) ... done
  Created wheel for tempdir: filename=tempdir-0.7.1-py3-none-any.whl size=2245 sha256=44dc116a16dc88d7c36edc4597
2584aaeb6ca875b82b070836f64127880fbd15
  Stored in directory: /Users/blank/Library/Caches/pip/wheels/31/7b/e3/af441c2f71a48c30809aada978c1433b163a0747e
73b5805ca
Successfully built latex data shutilwhich tempdir
Installing collected packages: tempdir, shutilwhich, funcsigs, future, data, latex
Successfully installed data-0.4 funcsigs-1.0.2 future-1.0.0 latex-0.7.0 shutilwhich-1.1.0 tempdir-0.7.1
Note: you may need to restart the kernel to use updated packages.
Data Import Success!
```

Pre-Processing:

Pretty much cleaning up the data to make sure it's useable, and to see its format properly.

In [11]:

```python
# Loading dataset:
df = pd.read_csv("/Users/blank/Desktop/Spring2025/CS451_ML/ML Project/Agora.csv", encoding="latin1")


# Now I'm going to pre process things that stood out to me whiel looking over the data manually:
# kill spaces and make everything lower-case overall
df.columns = (df.columns
                .str.strip()
                .str.lower()
                .str.replace(r'\s+', '_', regex=True))
```

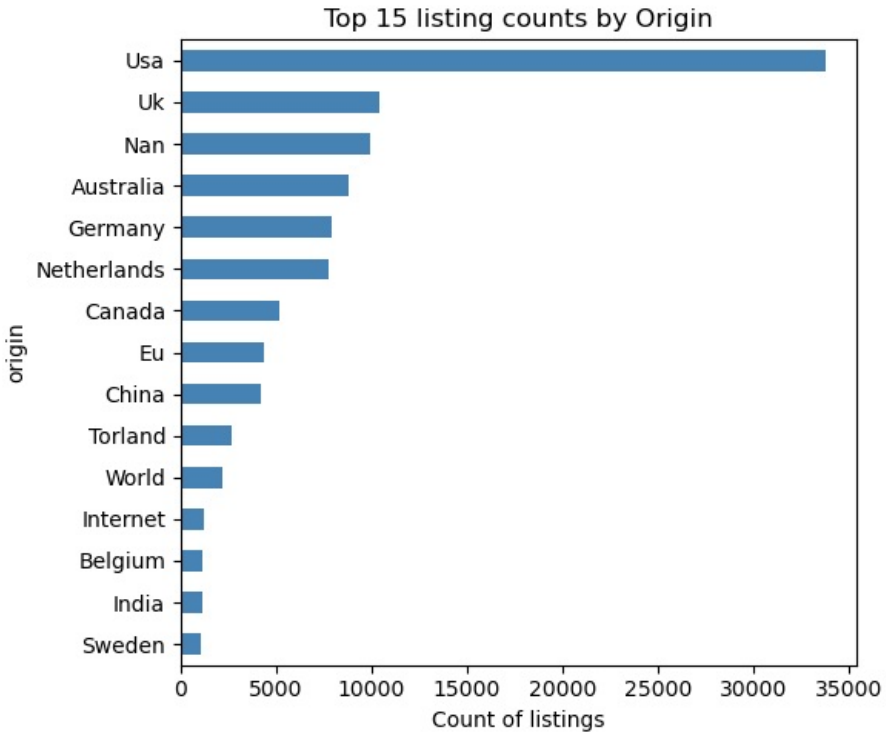
```
# Noticed some common use of words so used regex to generalize them. Def add more later.
clean_words = {
    r'\b(worldwide|global|everywhere)\b': 'Worldwide',
    r'\b(united\s*states|^us$|u\.s\.a?)\b': 'USA',
    r'\b(united\s*kingdom|^uk$|britain)\b': 'UK'
}
for col in ('origin', 'destination'):
    df[col] = (df[col].astype(str)
                      .str.lower()                                  # Lower cases it to have less unique set
                      .str.replace(r'[^\w\s]', ' ', regex=True)   # drop punctuation
                      .str.replace(r'\bonly\b', ' ', regex=True)   # Filters hype making words like “only”
                      .str.replace(r'\s+', ' ', regex=True)        # makes sure there's uniform single spacing
                      .str.strip())
    for pat, repl in clean_words.items():
        df[col] = df[col].str.replace(pat, repl, flags=re.I, regex=True) # Flag makes it case insensitive for r
    df[col] = df[col].str.title() # Making it title like to make the data visually more appealing and consisten

# Generic way of converting columns to numbers
num_cols = ['score',        # ratings we split out earlier
            'deals'         # deal counts
            # Add columns as you keep going down the project
           ]

# Filter out columns that do not exist in the dataset
num_cols = [col for col in num_cols if col in df.columns]

# convert each to float
for c in num_cols:
    df[c] = pd.to_numeric(df[c], errors='coerce')   # anything non-numeric treated as NaN

# Delete rows with missing numbers if any numeric columns exist
if num_cols:
    df = df.dropna(subset=num_cols).reset_index(drop=True)

if num_cols:
    print(df[num_cols].head())
else:
    print("No numeric columns found to process.")

print(f"Rows left: {len(df):,}")
```

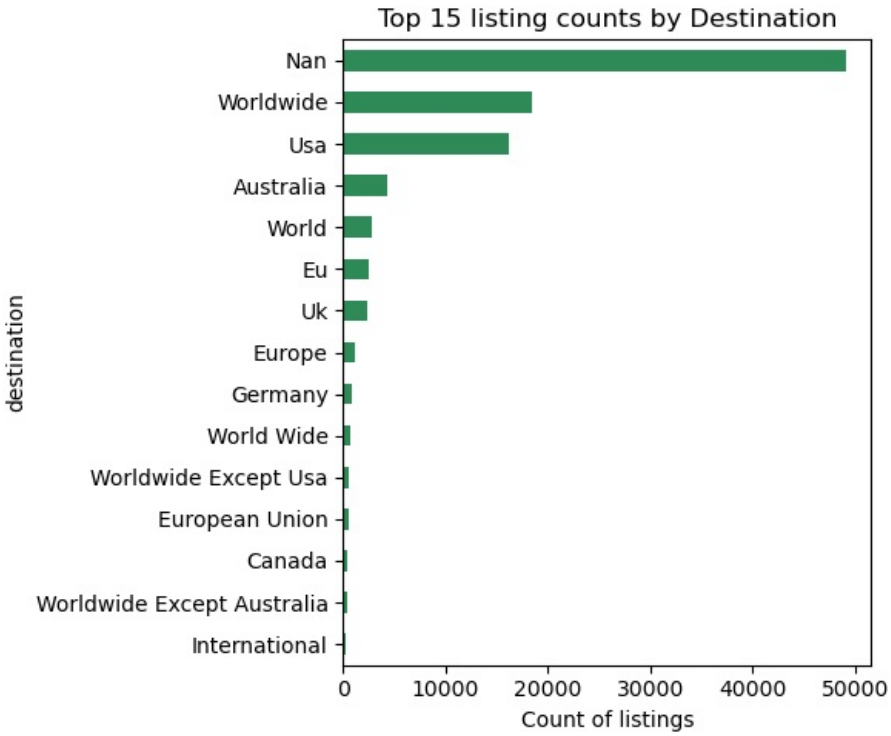
```
No numeric columns found to process.
Rows left: 109,689
```

Data Visualization and Analysis:

Now I will go through some columns that stood out to me and see if we can figure out some stuff that we might be able to research further.

In [14]:
```
# Look into origin and destination of the products:

# Origin
origin_counts = (df.groupby('origin').size().sort_values(ascending=False).head(15))

plt.figure(figsize=(6, 5))
origin_counts.plot(kind='barh', color='steelblue')
plt.gca().invert_yaxis()
plt.xlabel('Count of listings')
plt.title('Top 15 listing counts by Origin')
plt.tight_layout()
plt.show()

# Product destination
destination_counts = (df.groupby('destination').size().sort_values(ascending=False).head(15))

plt.figure(figsize=(6, 5))
destination_counts.plot(kind='barh', color='seagreen')
plt.gca().invert_yaxis()
plt.xlabel('Count of listings')
plt.title('Top 15 listing counts by Destination')
plt.tight_layout()
plt.show()

# Print the table
print("\n Top 5 Origins")
print(origin_counts.head(5).to_frame(name='count'))

print("\n Top 5 Destinations")
print(destination_counts.head(5).to_frame(name='count'))
```






```
 Top 5 Origins
             count
origin
Usa          33746
Uk           10373
Nan           9882
Australia     8767
Germany       7877

 Top 5 Destinations
              count
destination
Nan           49161
Worldwide     18487
Usa           16190
Australia      4331
World          2738
```

Plot analysis:

Based on the two plots that I just saw, as well as the printed table, it's evident that most product listings come from the USA, with the UK, Australia, and Germany following behind. A noticeable number of listings don't include origin info. On the destination side, a huge chunk is also missing, which is not surprising given it's a darkweb dataset. From what is listed, many products are shipped worldwide, or specifically to the USA and Australia. This shows the USA is a major player on both ends selling and buying while the missing destination

data might just be sellers choosing not to share where they ship, or skipping the detail altogether or straight up lying about it on both cases.