

```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.feature_extraction.text import CountVectorizer
import networkx as nx

# Load the dataset
data = pd.read_csv('dataset.csv')

# 1. Correlation Heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(data[['AminerCitationCount', 'CitationCount_CrossRef', 'PubsCited_CrossRef',
'Downloads_Xplore']].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# 2. Trend Analysis Over Time
data.groupby('Year')['CitationCount_CrossRef'].sum().plot(kind='line', title='Citation
Trend Over Years', marker='o')
plt.show()

data.groupby('Year')['Downloads_Xplore'].sum().plot(kind='line', title='Downloads Trend
Over Years', marker='s', color='g')
plt.show()

# 3. Topic Modeling on Abstracts
vectorizer = CountVectorizer(max_df=0.95, min_df=2, stop_words='english')
abstracts_matrix = vectorizer.fit_transform(data['Abstract'].dropna())
lda = LatentDirichletAllocation(n_components=5, random_state=0)
lda.fit(abstracts_matrix)

# Display top words in each topic
def display_topics(model, feature_names, no_top_words):
    for topic_idx, topic in enumerate(model.components_):
        print(f"Topic {topic_idx}: ", " ".join([feature_names[i] for i in
topic.argsort()[::-no_top_words - 1::-1]]))

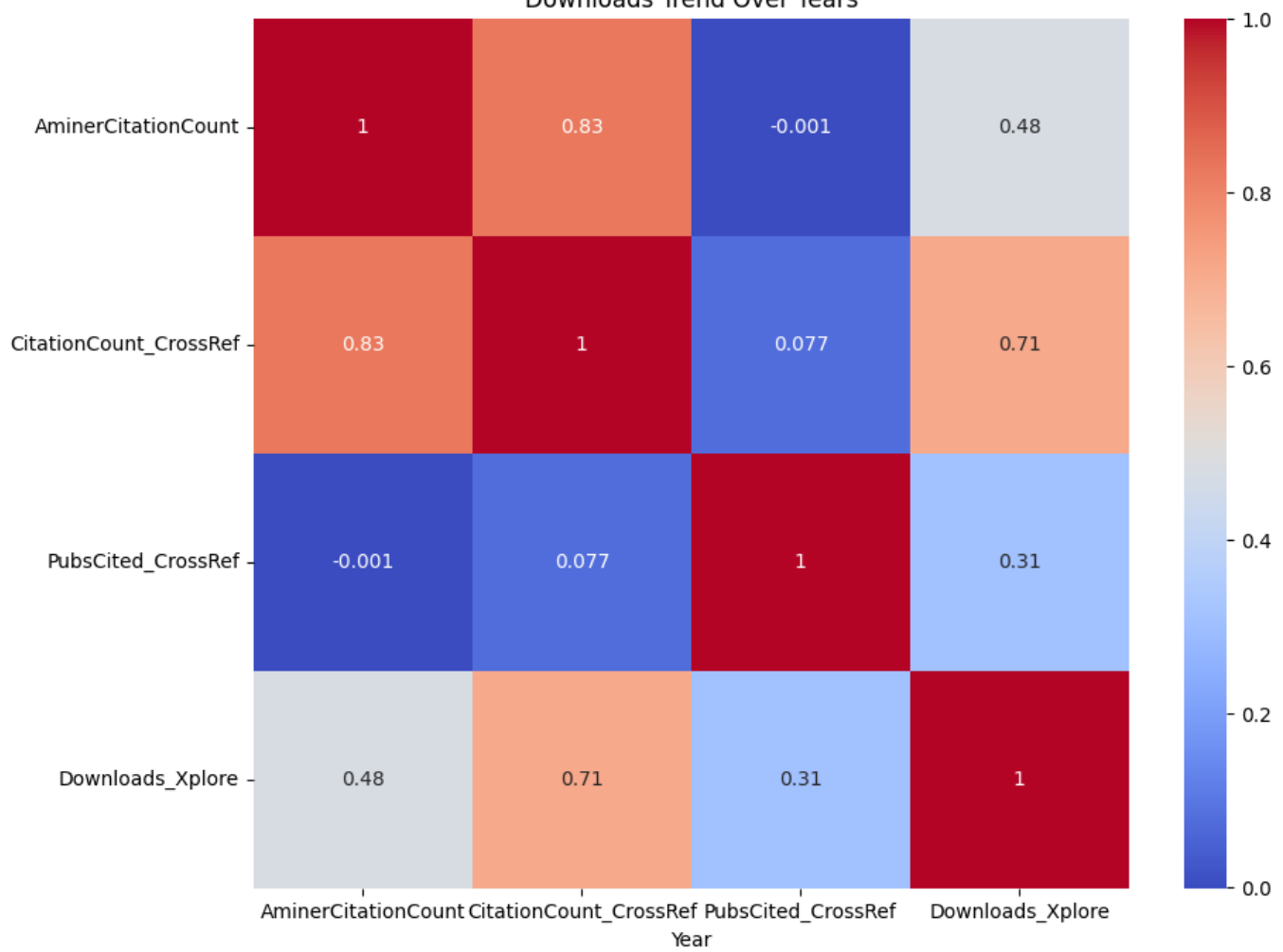
display_topics(lda, vectorizer.get_feature_names_out(), 10)

# 4. Author Network Analysis
edges = []
for authors in data['AuthorNames-Deduped'].dropna():
    author_list = authors.split(';')
    edges.extend([(author_list[i], author_list[j]) for i in range(len(author_list)) for
j in range(i+1, len(author_list))])

G = nx.Graph()
G.add_edges_from(edges)
plt.figure(figsize=(12, 12))
nx.draw(G, with_labels=True, node_size=50, font_size=8)
plt.title('Author Collaboration Network')
plt.show()

```


Downloads Trend Over Years



Author Collaboration Network

