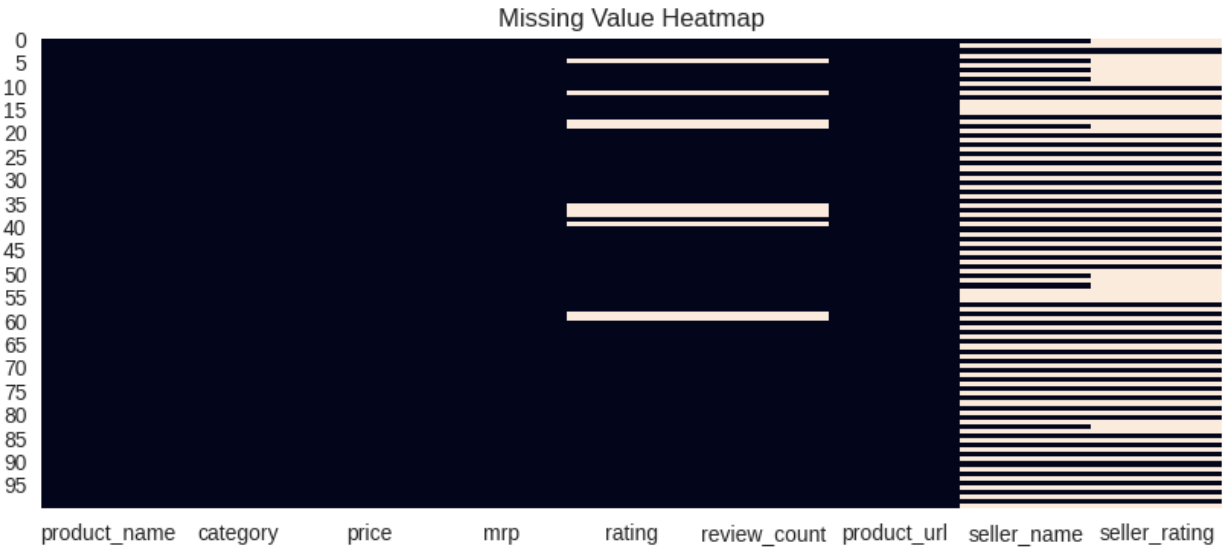
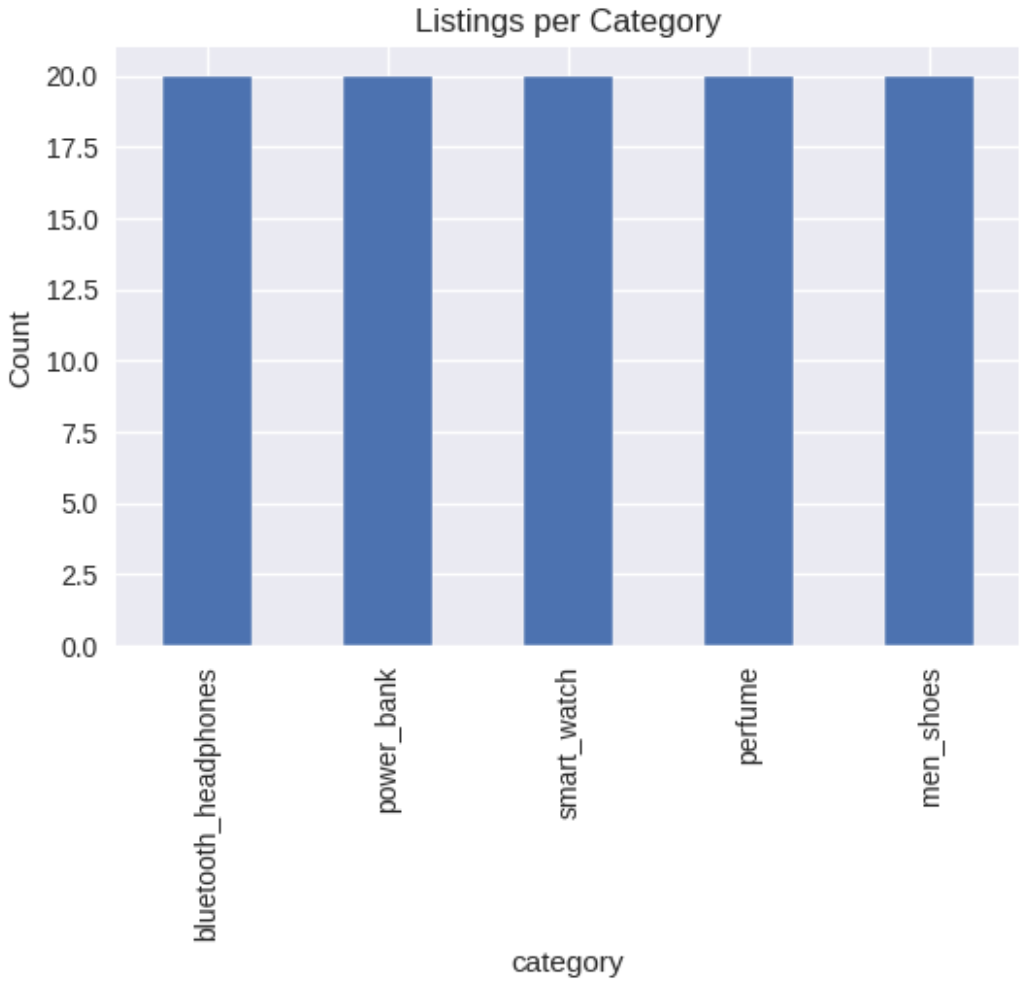
```
# ================================
# ROUND 2 — PREPROCESSING & VISUALIZATION
# ================================

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.cluster import KMeans
from google.colab import files

plt.style.use("seaborn-v0_8")

# ================================
# 1 LOAD RAW DATA
# ================================
df = pd.read_csv("/content/snapdeal_raw_data.csv")
print("Raw dataset shape:", df.shape)
df.head()
```

```
Raw dataset shape: (100, 9)
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 100,\n  \"fields\": [\n    {\n      \"column\": \"product_name\",\n      \"properties\": {\n\"dtype\": \"string\",\n        \"num_unique_values\": 89,\n\"samples\": [\n          \"OLIVE OPS Sports Edition Smartwatch | Heart Rate | Steps | Wireless Charging Smartwatch (Black-Grey Strap, 1.73 Inch Big Sunlight Proof Display)\",\n          \"Ramsons - MIDNIGHT Eau De Parfum Perfume For Men Long Lasting Premium Perfume 40ml (Pack of 1)\",\n          \"Zebronics 10000 -mAh 22.5W Li-Polymer Power Bank\"\n        ],\n        \"semantic_type\": \"\",\n\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"category\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 5,\n        \"samples\": [\n          \"power_bank\",\n          \"men_shoes\",\n\"smart_watch\"\n        ],\n        \"semantic_type\": \"\",\n\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"price\",\n      \"properties\": {\n        \"dtype\": \"number\",\n\"std\": 291,\n        \"min\": 129,\n        \"max\": 1248,\n\"num_unique_values\": 77,\n        \"samples\": [\n          318,\n673,\n          488\n        ],\n        \"semantic_type\": \"\",\n\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"mrp\",\n      \"properties\": {\n        \"dtype\": \"number\",\n\"std\": 1324,\n        \"min\": 185,\n        \"max\": 6999,\n\"num_unique_values\": 34,\n        \"samples\": [\n          349,\n799,\n          2299\n        ],\n        \"semantic_type\": \"\",\n\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"rating\",\n      \"properties\": {\n        \"dtype\": \"number\",\n\"std\": 0.39630629146429436,\n        \"min\": 3.5,\n        \"max\":
```

```
5.0,\n        \"num_unique_values\": 16,\n        \"samples\": [\n
4.2,\n          3.9,\n          4.4\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"review_count\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1212.122512071145,\n        \"min\": 1.0,\n        \"max\": 7954.0,\n
\"num_unique_values\": 68,\n        \"samples\": [\n          272.0,\n
134.0,\n          7954.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"product_url\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 100,\n
\"samples\": [\n          \"https://www.snapdeal.com/product/hotstyle-
gray-mens-sports-running/623981372905#bcrumbSearch:men%20shoes\",\n
\"https://www.snapdeal.com/product/olive-ops-sports-edition-
smartwatch/655462266845#bcrumbSearch:smart%20watch\",\n
\"https://www.snapdeal.com/product/st-john-one-men-army/659679519339#b
crumbSearch:perfume\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"seller_name\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 24,\n
\"samples\": [\n          \"POOJA ENTERPRISES\",\n          \"HELIOS
LIFESTYLE PVT LTD\",\n          \"Varni Enterprise\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"seller_rating\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.3104070941563096,\n        \"min\": 3.5,\n        \"max\": 5.0,\n
\"num_unique_values\": 9,\n        \"samples\": [\n          4.3,\n
3.9,\n          3.5\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}

# Missing values
df.isnull().sum().sort_values(ascending=False)

plt.figure(figsize=(10,4))
sns.heatmap(df.isnull(), cbar=False)
plt.title("Missing Value Heatmap")
plt.show()

# Category distribution
plt.figure(figsize=(6,4))
df["category"].value_counts().plot(kind="bar")
plt.title("Listings per Category")
plt.ylabel("Count")
plt.show()

df.dtypes
```



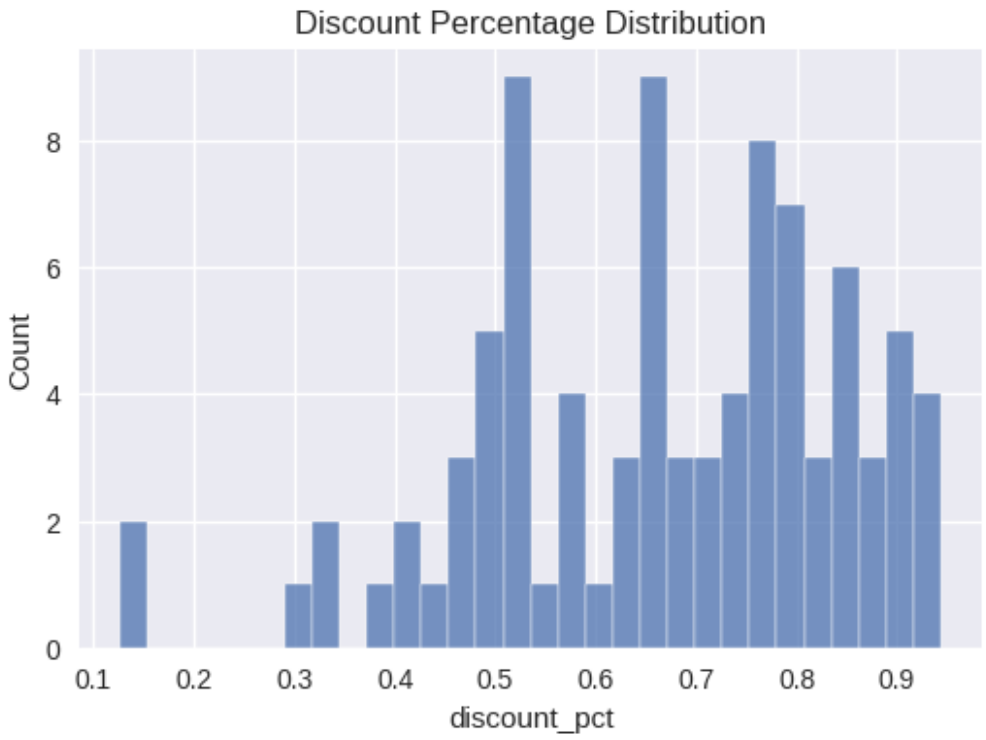

```
product_name      object
category          object
```

```
price              int64
mrp                int64
rating           float64
review_count     float64
product_url       object
seller_name       object
seller_rating    float64
dtype: object
```

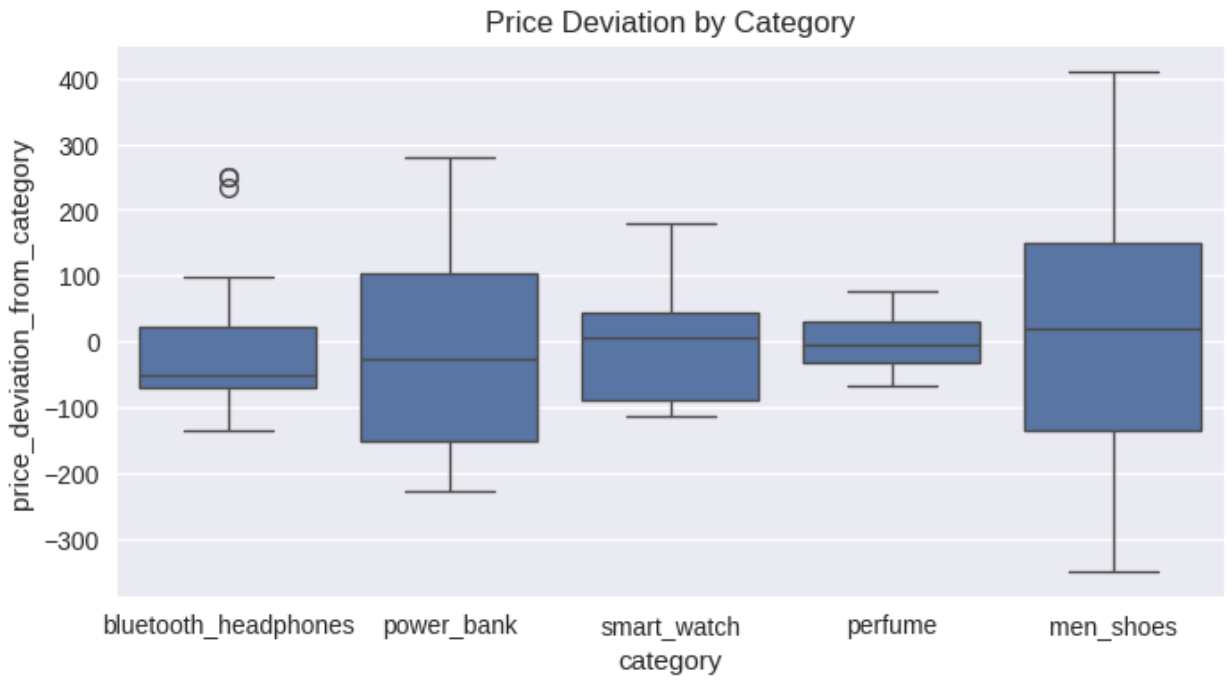
```
before = df.shape[0]

df["price"] = pd.to_numeric(df["price"], errors="coerce")
df["mrp"] = pd.to_numeric(df["mrp"], errors="coerce")
df["rating"] = pd.to_numeric(df["rating"], errors="coerce")
df["review_count"] = pd.to_numeric(df["review_count"],
errors="coerce").fillna(0)
df["seller_rating"] = pd.to_numeric(df["seller_rating"],
errors="coerce")

df = df.dropna(subset=["price", "mrp", "rating"])

after = df.shape[0]
print("Rows before cleaning:", before)
print("Rows after cleaning:", after)

plt.figure(figsize=(6,4))
sns.boxplot(x=df["price"])
plt.title("Price Distribution After Cleaning")
plt.show()
```

```
Rows before cleaning: 100
Rows after cleaning: 90
```


```
plt.figure(figsize=(6,4))
sns.histplot(df["discount_pct"], bins=30)
plt.title("Discount Percentage Distribution")
plt.show()

plt.figure(figsize=(6,4))
sns.scatterplot(data=df, x="price", y="rating", hue="category")
plt.title("Price vs Rating")
plt.show()

plt.figure(figsize=(8,4))
sns.boxplot(data=df, x="category", y="price_deviation_from_category")
plt.title("Price Deviation by Category")
plt.show()
```

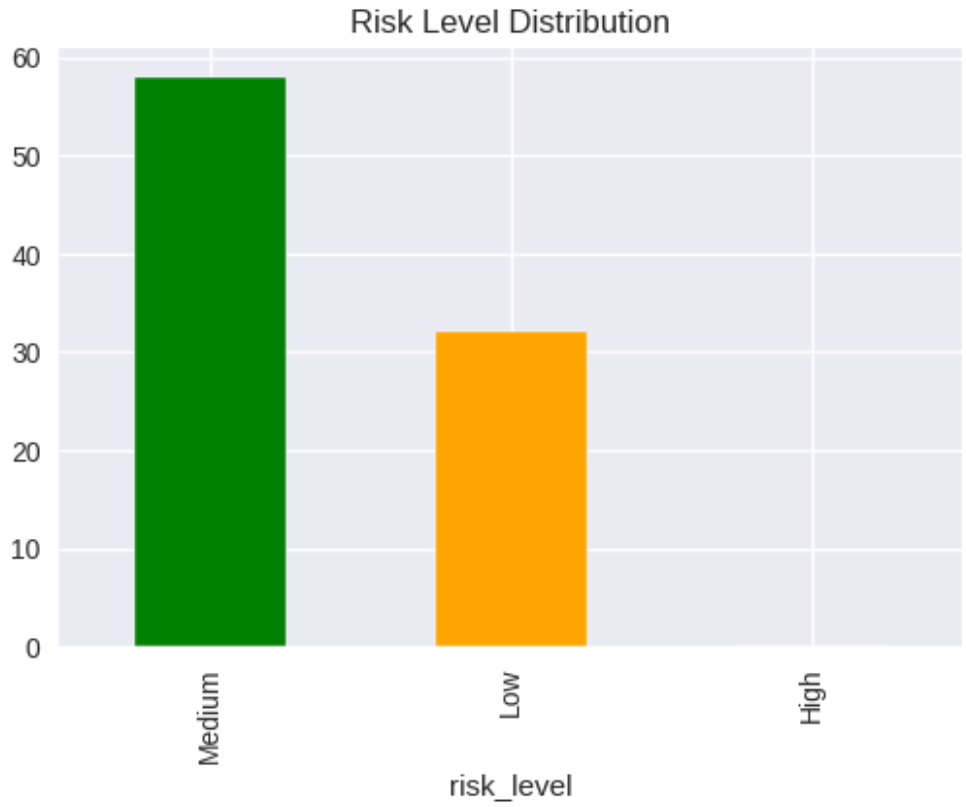

Discount Percentage Distribution
Count
discount_pct
0
2
4
6
8
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

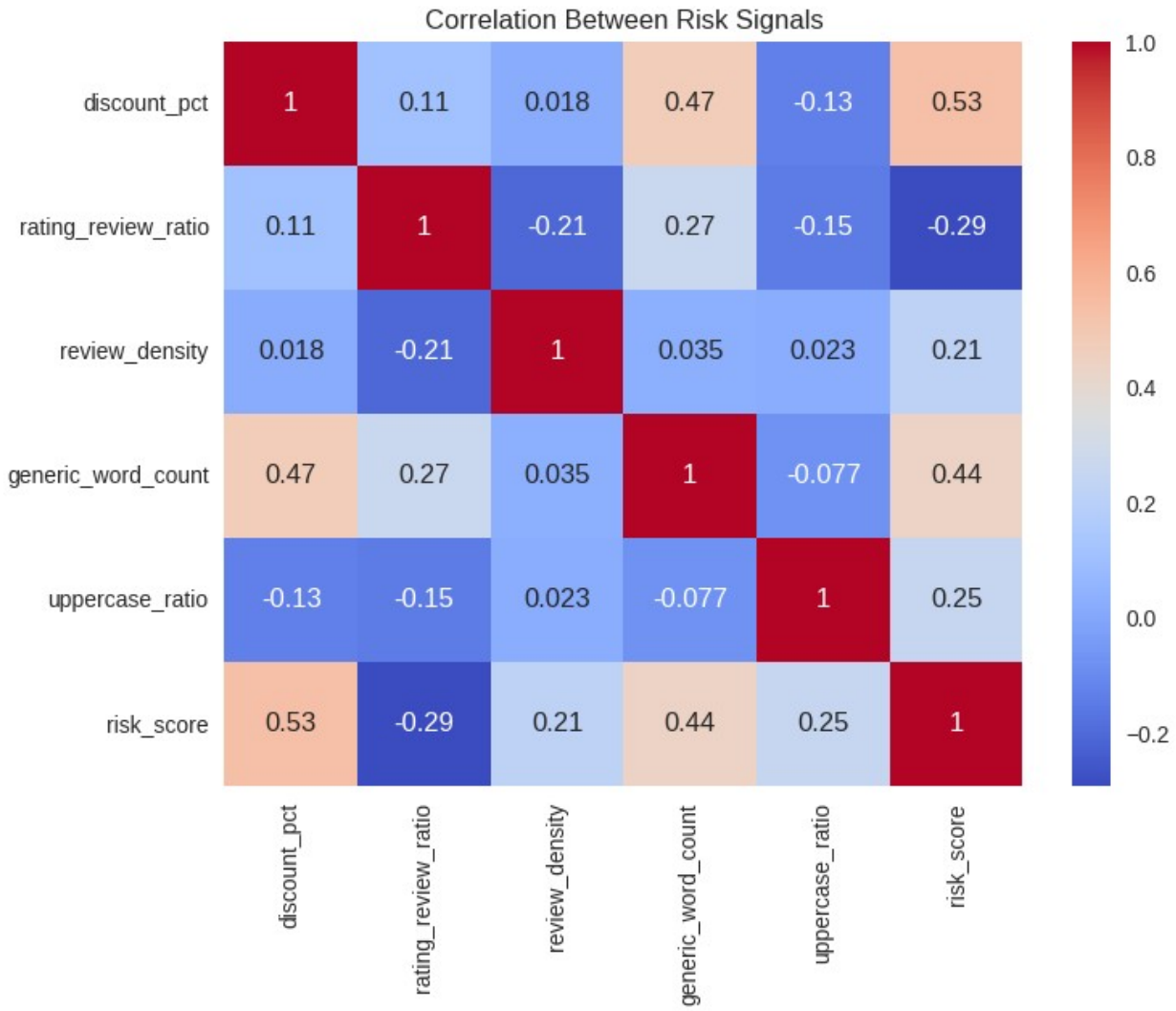

Price vs Rating
rating
price
category
bluetooth_headphones
power_bank
smart_watch
perfume
men_shoes
3.6
3.8
4.0
4.2
4.4
4.6
4.8
5.0
200
400
600
800
1000
1200




```
df["extreme_discount_flag"] = (df["discount_pct"] > 0.6).astype(int)
df["suspicious_trust_flag"] = (df["rating_review_ratio"] <
1.2).astype(int)
df["aggressive_listing_flag"] = (
    (df["generic_word_count"] >= 3) | (df["uppercase_ratio"] > 0.25)
).astype(int)

df["risk_score"] = (
    df["extreme_discount_flag"] +
    df["suspicious_trust_flag"] +
    df["aggressive_listing_flag"]
)

df["risk_level"] = pd.cut(
    df["risk_score"],
    bins=[-1,1,3,10],
    labels=["Low","Medium","High"]
)

plt.figure(figsize=(6,4))
df["risk_level"].value_counts().plot(kind="bar",
color=["green","orange","red"])
plt.title("Risk Level Distribution")
plt.show()

risk_features = [
    "discount_pct","rating_review_ratio","review_density",
    "generic_word_count","uppercase_ratio","risk_score"
]
```

```
plt.figure(figsize=(8,6))
sns.heatmap(df[risk_features].corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Between Risk Signals")
plt.show()
```






```
cluster_features = df[
    ["price","discount_pct","rating_review_ratio",
     "generic_word_count","uppercase_ratio"]
].fillna(0)

X = StandardScaler().fit_transform(cluster_features)
df["seller_cluster_id"] = KMeans(n_clusters=5,
random_state=42).fit_predict(X)

scaler = MinMaxScaler()
scale_cols = [
    "price","discount_pct","rating_review_ratio",
    "review_density","price_deviation_from_category"
]

df[scale_cols] = scaler.fit_transform(df[scale_cols])
```

```
final_file = "round2_processed_dataset.csv"
df.to_csv(final_file, index=False)
files.download(final_file)

print("□ FINAL DATASET SHAPE:", df.shape)
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
□ FINAL DATASET SHAPE: (90, 23)
```