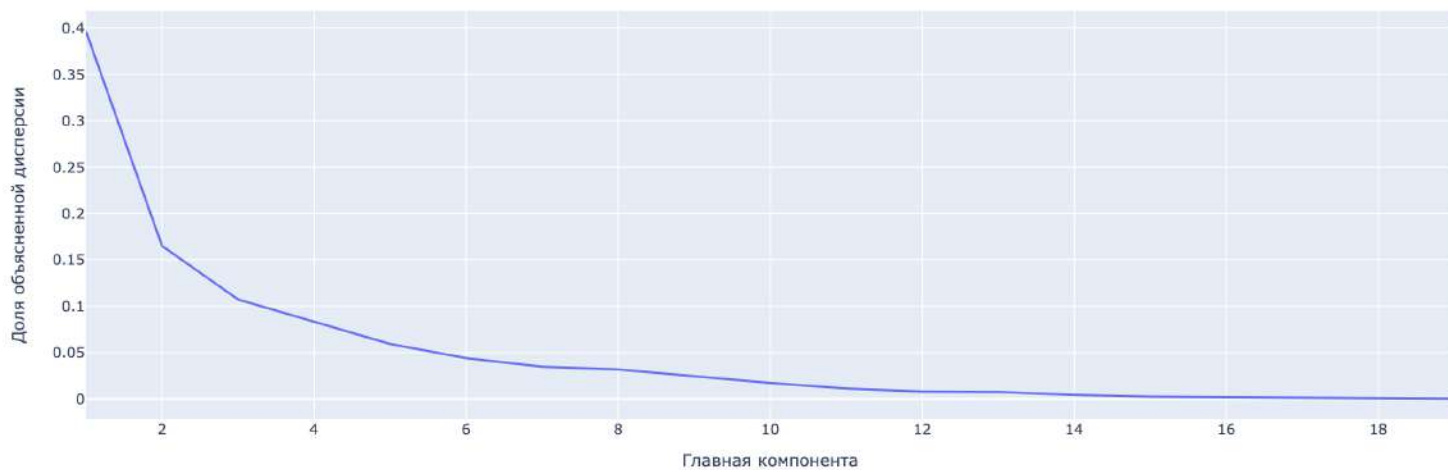


```
pca = PCA()
pca.fit(df_preprocessed)
fig = px.line(x=range(1, len(pca.explained_variance_ratio_) + 1), y=pca.explained_variance_ratio_,
              labels={'x': 'Главная компонента', 'y': 'Доля объясненной дисперсии'},
              title='Доля объясненной дисперсии PCA')
fig.show()
```

Доля объясненной дисперсии PCA



```
df_numeric = df.select_dtypes(include=[np.number])
corr = df_numeric.corr()
fig = px.imshow(corr, x=corr.columns, y=corr.columns, title='Карта корреляции', labels=dict(color="Корреляция"))
fig.show()
```

Карта корреляции

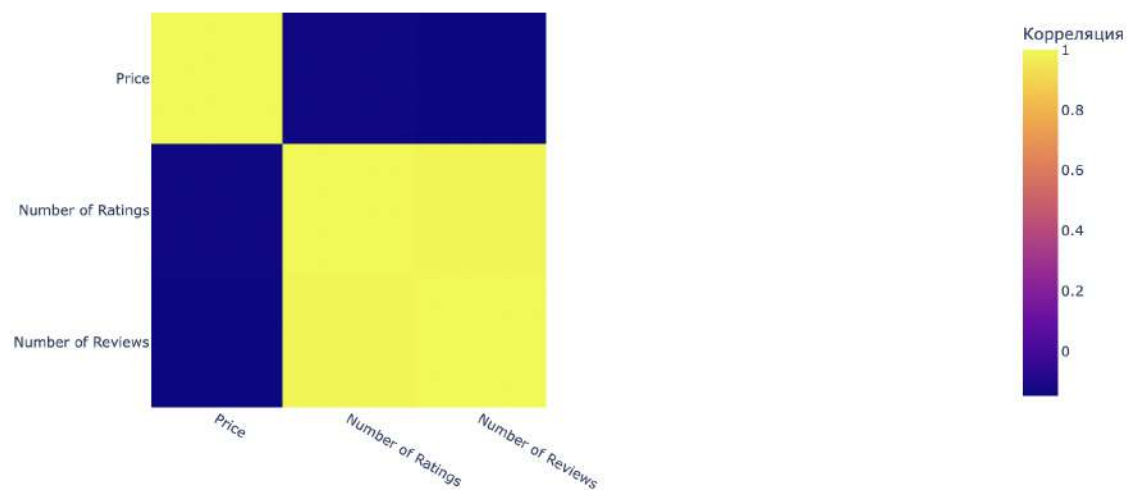


Диаграмма рассеяния: Number of Reviews vs Number of Reviews

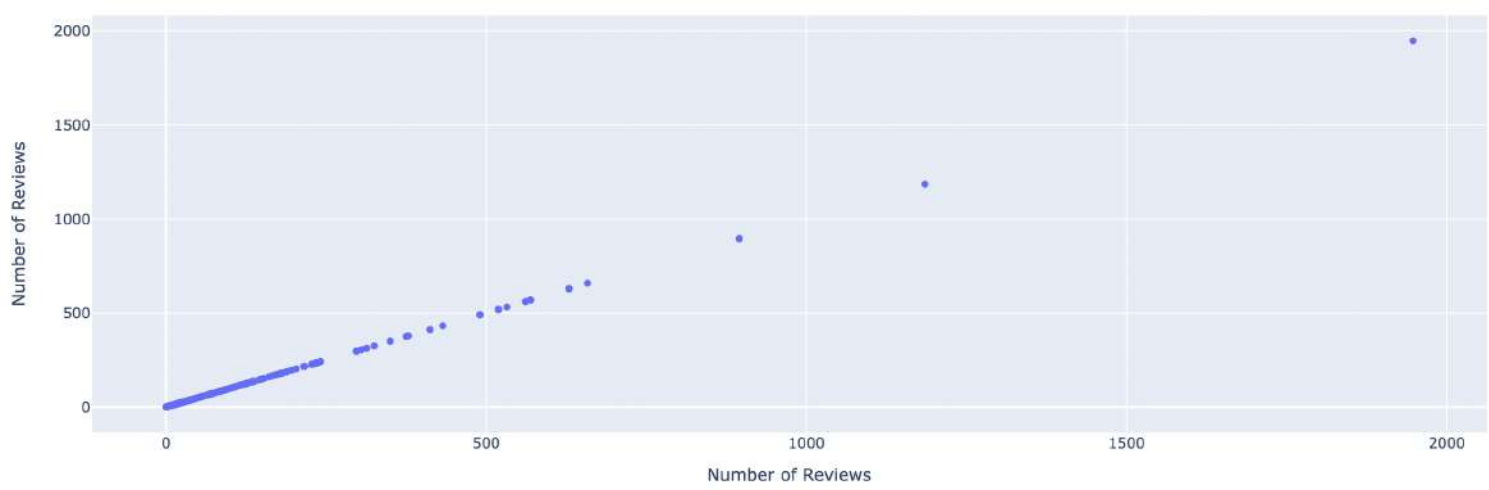


Диаграмма рассеяния: Number of Reviews vs Number of Ratings

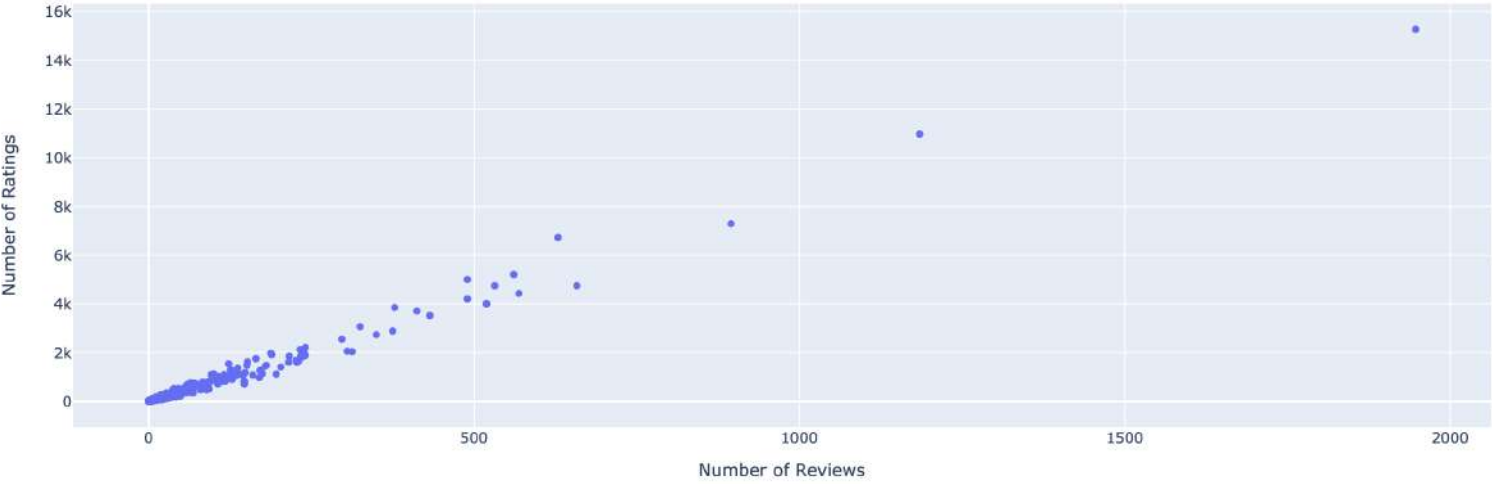


Диаграмма рассеяния: Number of Reviews vs Price

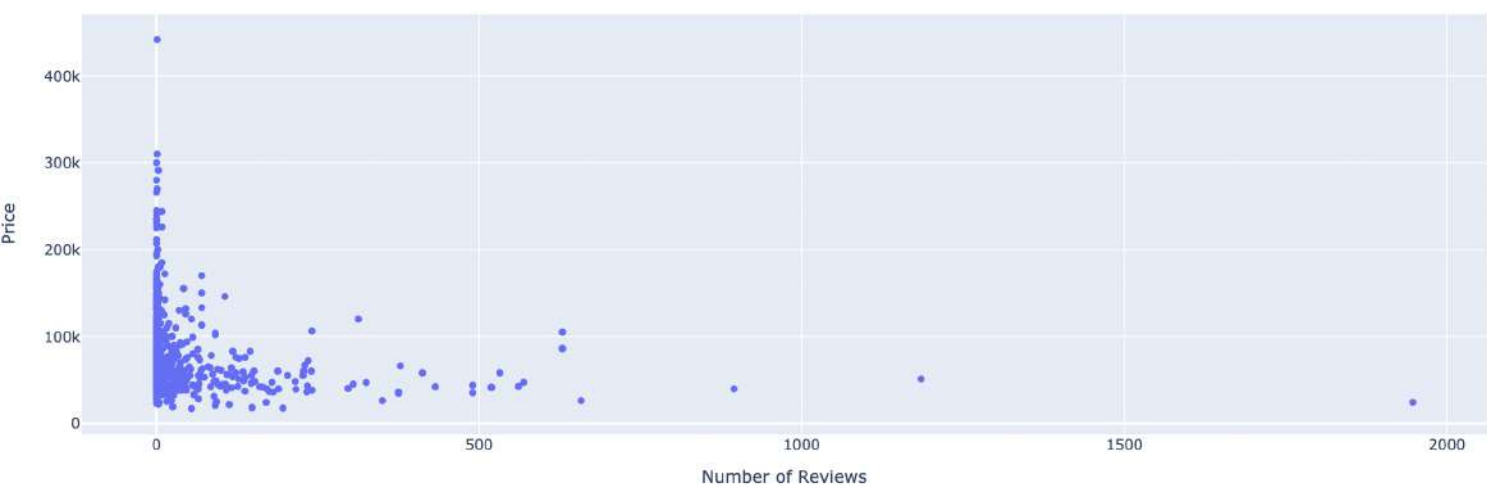


Диаграмма рассеяния: Number of Ratings vs Number of Reviews

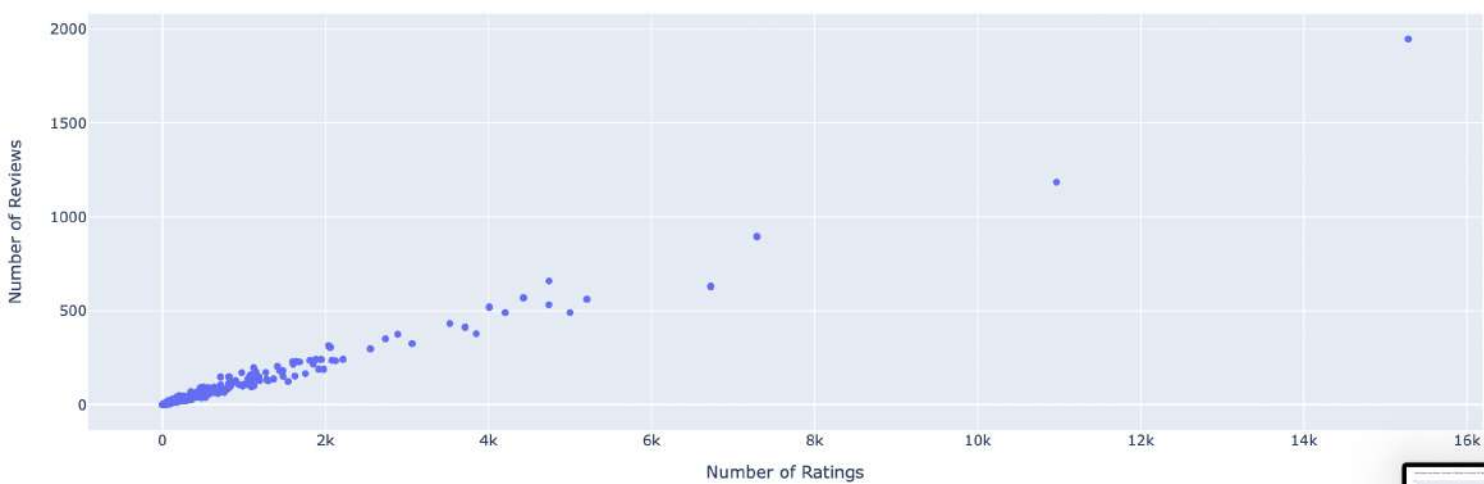


Диаграмма рассеяния: Number of Ratings vs Number of Ratings

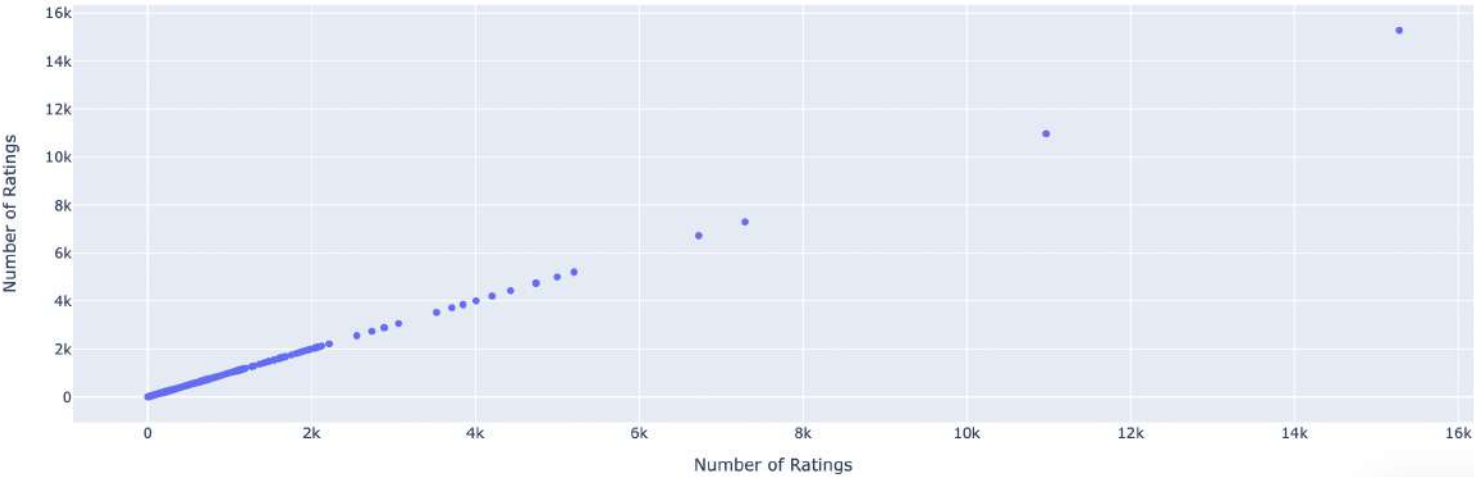


Диаграмма рассеяния: Number of Ratings vs Price

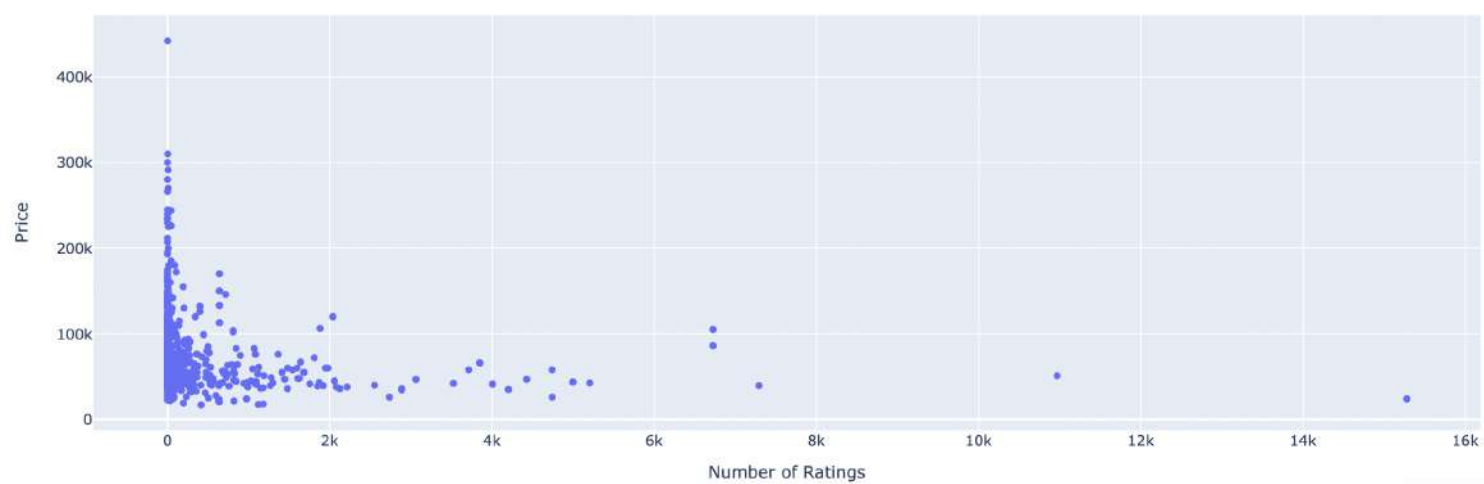


Диаграмма рассеяния: Price vs Number of Reviews

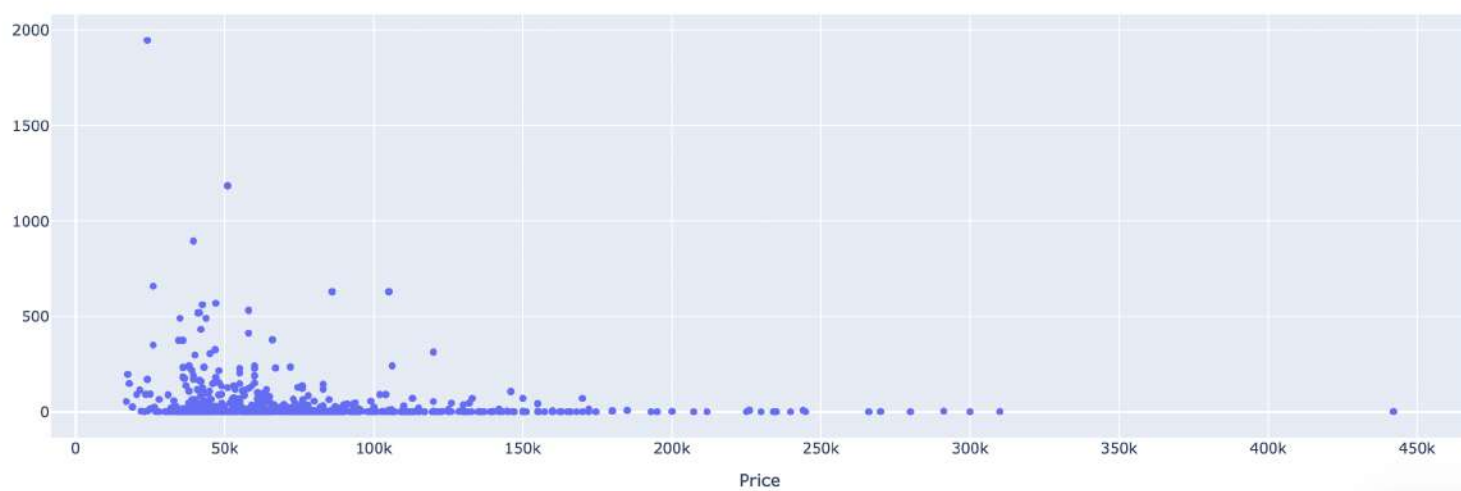


Диаграмма рассеяния: Price vs Number of Ratings

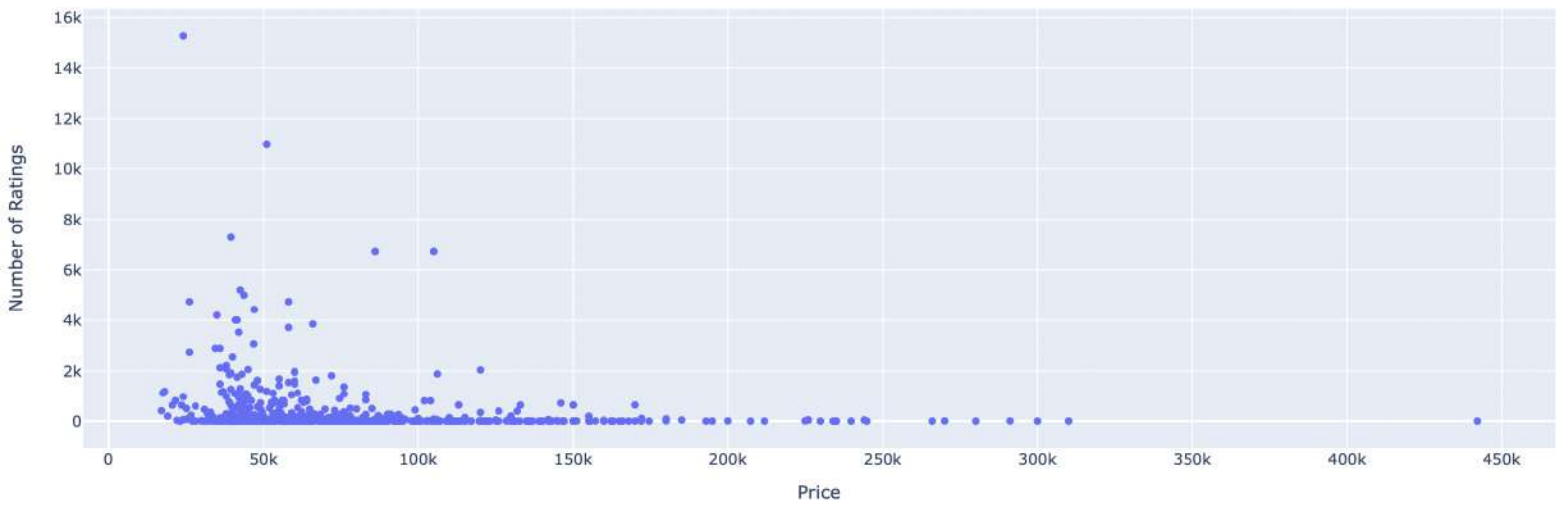
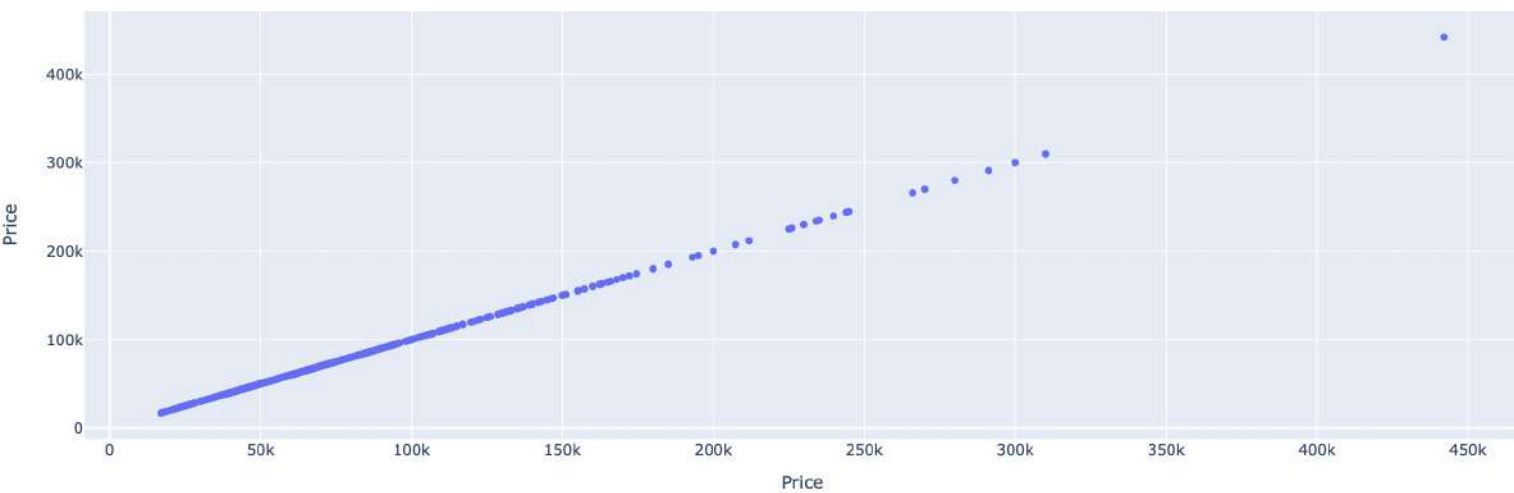
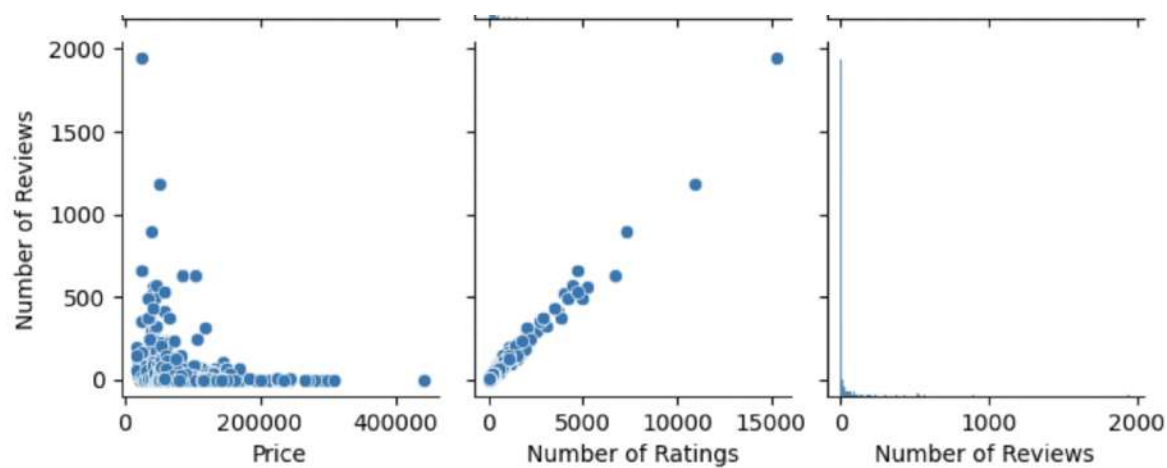


Диаграмма рассеяния: Price vs Price

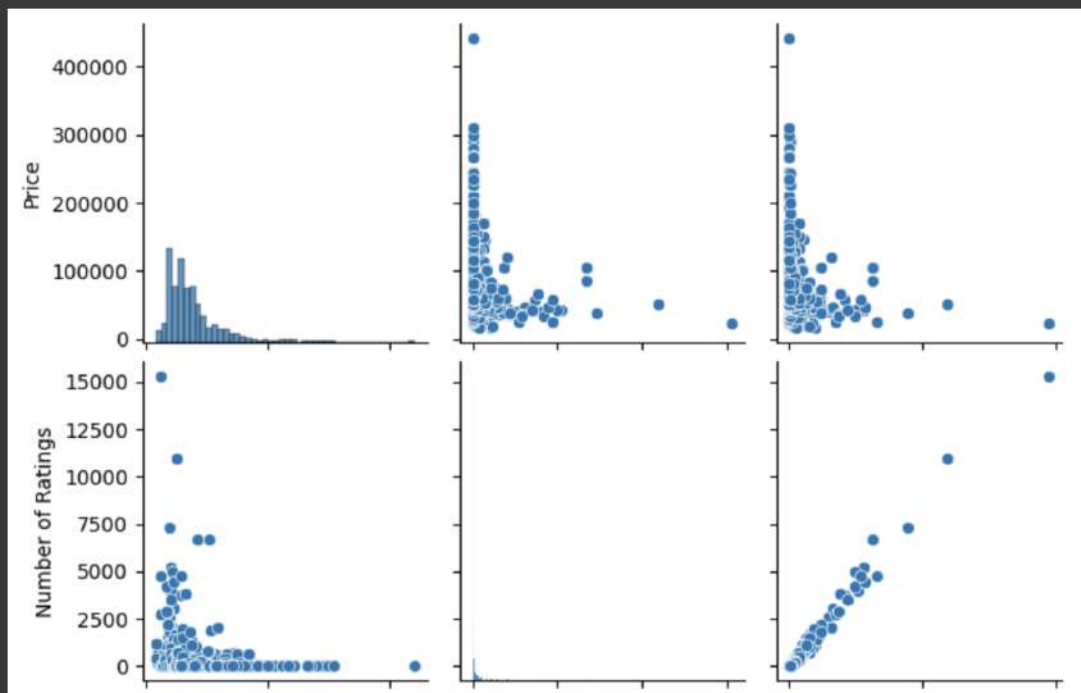


[8]

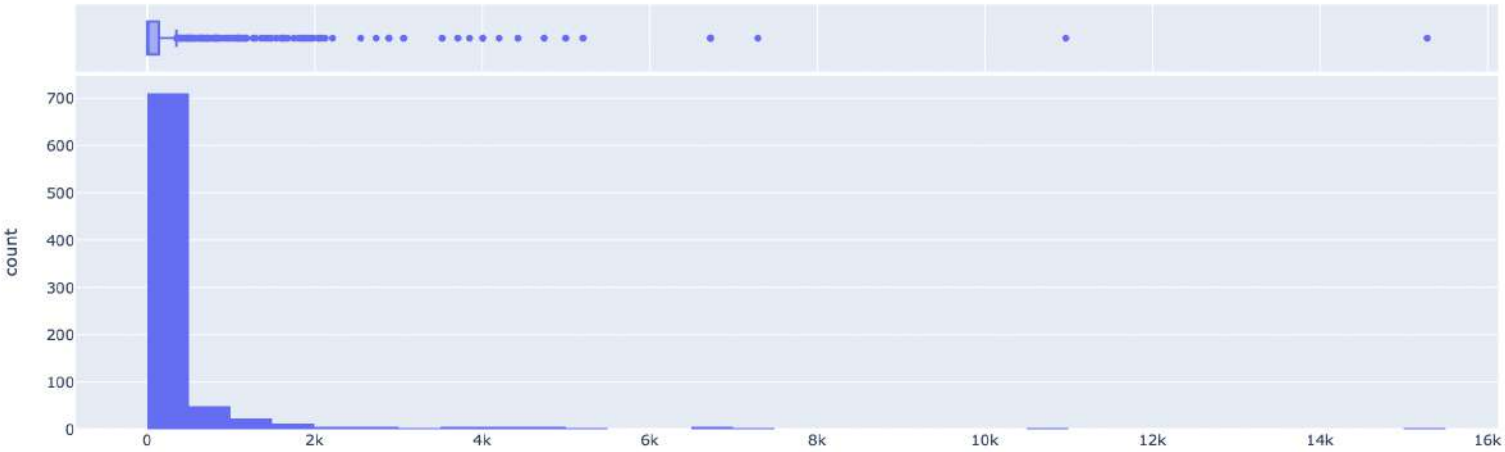


```
for col1 in df.select_dtypes(include=[np.number]).columns:
    for col2 in df.select_dtypes(include=[np.number]).columns:
        fig = px.scatter(df, x=col1, y=col2, title=f'Диаграмма рассеяния: {col1} vs {col2}')
        fig.show()
```

```
[8] sns.pairplot(df.select_dtypes(include=[np.number]))  
plt.show()
```



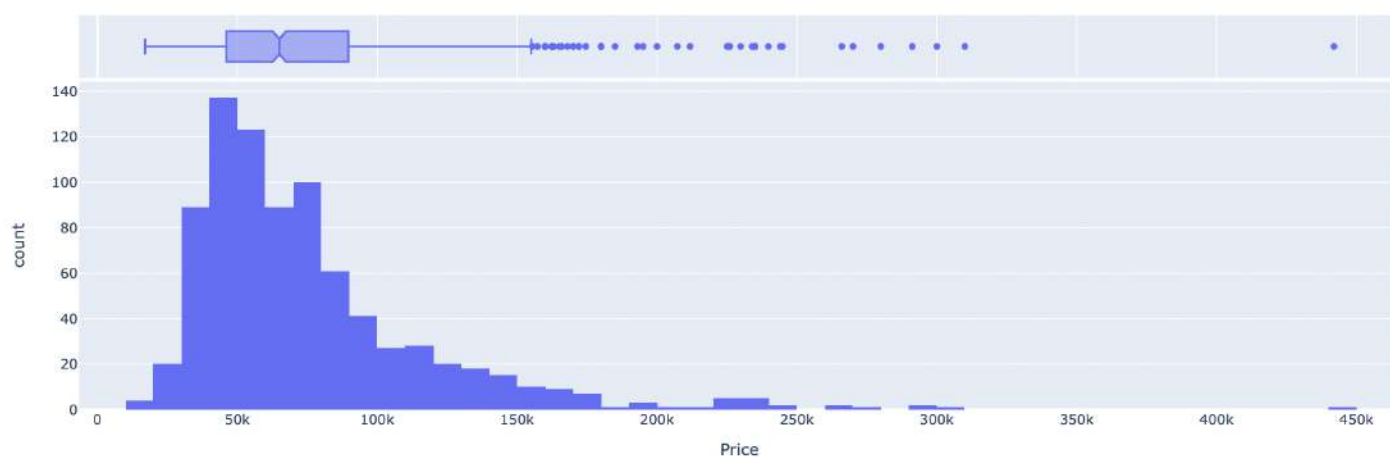
Распределение Number of Ratings



```
2 ✓  
0 сек.  
for col in df.select_dtypes(include=[np.number]).columns:  
    fig = px.histogram(df, x=col, marginal="box", nbins=50, title=f'Распределение {col}')  
    fig.show()
```



Распределение Price



iii

+

Код

+

Текст

✓

ОЗУ

Диск

↑

↓

✓

0 сек.

```
import pandas as pd
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import PCA
import numpy as np
import seaborn as sns
import sqlite3
import matplotlib.pyplot as plt
```

✓

0 сек.

```
[4] df = pd.read_csv('/content/laptopPrice.csv')
# Создание подключения к базе данных SQLite
conn = sqlite3.connect('/content/laptopPrice.db')

# Загрузка датафрейма в новую таблицу базы данных SQLite
df.to_sql('laptop_price', conn, if_exists='replace', index=False)

# Проверка, что таблица была успешно создана
cursor = conn.cursor()
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")

<sqlite3.Cursor at 0x7f49f9ab95c0>
```

✓

0 сек.

```
[6] query = "SELECT * FROM laptop_price"
df = pd.read_sql_query(query, conn)
df.head()
```

<>

brand processor_brand processor_name processor_garntn ram_gb ram_type ssd hdd os os_bit graphic_card_gb weight warranty Touchscreen msoffic

+ Код + Текст

Диск

```
df = pd.get_dummies(df, drop_first=True)
y = df['Price']
X = df.drop('Price', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model_lr = LinearRegression()
model_lr.fit(X_train, y_train)
predictions_lr = model_lr.predict(X_test)
rmse_lr = np.sqrt(mean_squared_error(y_test, predictions_lr))
r2_lr = r2_score(y_test, predictions_lr)

model_rf = RandomForestRegressor()
model_rf.fit(X_train, y_train)
predictions_rf = model_rf.predict(X_test)
rmse_rf = np.sqrt(mean_squared_error(y_test, predictions_rf))
r2_rf = r2_score(y_test, predictions_rf)
```

[15] model_metrics = {'Линейная регрессия': [rmse_lr, r2_lr], 'Случайный лес': [rmse_rf, r2_rf]}

```
for model, metrics in model_metrics.items():
    print(f'Метрики для {model}:')
    print('Среднеквадратическое отклонение: ', metrics[0])
    print('R2-оценка: ', metrics[1])
```

Метрики для Линейная регрессия:
Среднеквадратическое отклонение: 24008.542545320597
R2-оценка: 0.7042241051300424
Метрики для Случайный лес:
Среднеквадратическое отклонение: 22218.94616697833
R2-оценка: 0.7466749733460987