```python
import pandas as pd

file_path = 'investmentsutf8.csv'  # замените на актуальный путь к файлу
data = pd.read_csv(file_path)

# Вывод всех колонок датасета
print("Колонки в датасете:")
print(data.info())
```

```
Колонки в датасете:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54294 entries, 0 to 54293
Data columns (total 39 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   permalink             49438 non-null  object
 1   name                  49437 non-null  object
 2   homepage_url          45989 non-null  object
 3   category_list         45477 non-null  object
 4    market               45470 non-null  object
 5    funding_total_usd    49438 non-null  object
 6   status                48124 non-null  object
 7   country_code          44165 non-null  object
 8   state_code            30161 non-null  object
 9   region                44165 non-null  object
 10  city                  43322 non-null  object
 11  funding_rounds        49438 non-null  float64
 12  founded_at            38554 non-null  object
 13  founded_month         38482 non-null  object
 14  founded_quarter       38482 non-null  object
 15  founded_year          38482 non-null  float64
 16  first_funding_at      49438 non-null  object
 17  last_funding_at       49438 non-null  object
 18  seed                  49438 non-null  float64
 19  venture               49438 non-null  float64
 20  equity_crowdfunding   49438 non-null  float64
 21  undisclosed           49438 non-null  float64
 22  convertible_note      49438 non-null  float64
```

```
 23  debt_financing         49438 non-null  float64
 24  angel                  49438 non-null  float64
 25  grant                  49438 non-null  float64
 26  private_equity         49438 non-null  float64
 27  post_ipo_equity        49438 non-null  float64
 28  post_ipo_debt          49438 non-null  float64
 29  secondary_market       49438 non-null  float64
 30  product_crowdfunding   49438 non-null  float64
 31  round_A                49438 non-null  float64
 32  round_B                49438 non-null  float64
 33  round_C                49438 non-null  float64
 34  round_D                49438 non-null  float64
 35  round_E                49438 non-null  float64
 36  round_F                49438 non-null  float64
 37  round_G                49438 non-null  float64
 38  round_H                49438 non-null  float64
dtypes: float64(23), object(16)
memory usage: 16.2+ MB
None
```

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
```

```python
# Загрузка датасета
df = pd.read_csv('investmentsutf8.csv', encoding='utf-8')

# Удаление пробелов из названий столбцов
df.columns = df.columns.str.strip()

# Просмотр первых строк
```

```
display(df.head())
```

	permalink	name	homepage_url	category_list
0	/organization/waywire	#waywire	http://www.waywire.com	|Entertainmer
1	/organization/tv-communications	&TV Communications	http://enjoyandtv.com	|Games|
2	/organization/rock-your-paper	'Rock' Your Paper	http://www.rockyourpaper.org	|Publishing|Ec
3	/organization/in-touch-network	(In)Touch Network	http://www.InTouchNetwork.com	|Electronics|G
4	/organization/r-ranch-and-mine	-R- Ranch and Mine	NaN	|Tourism|Ente

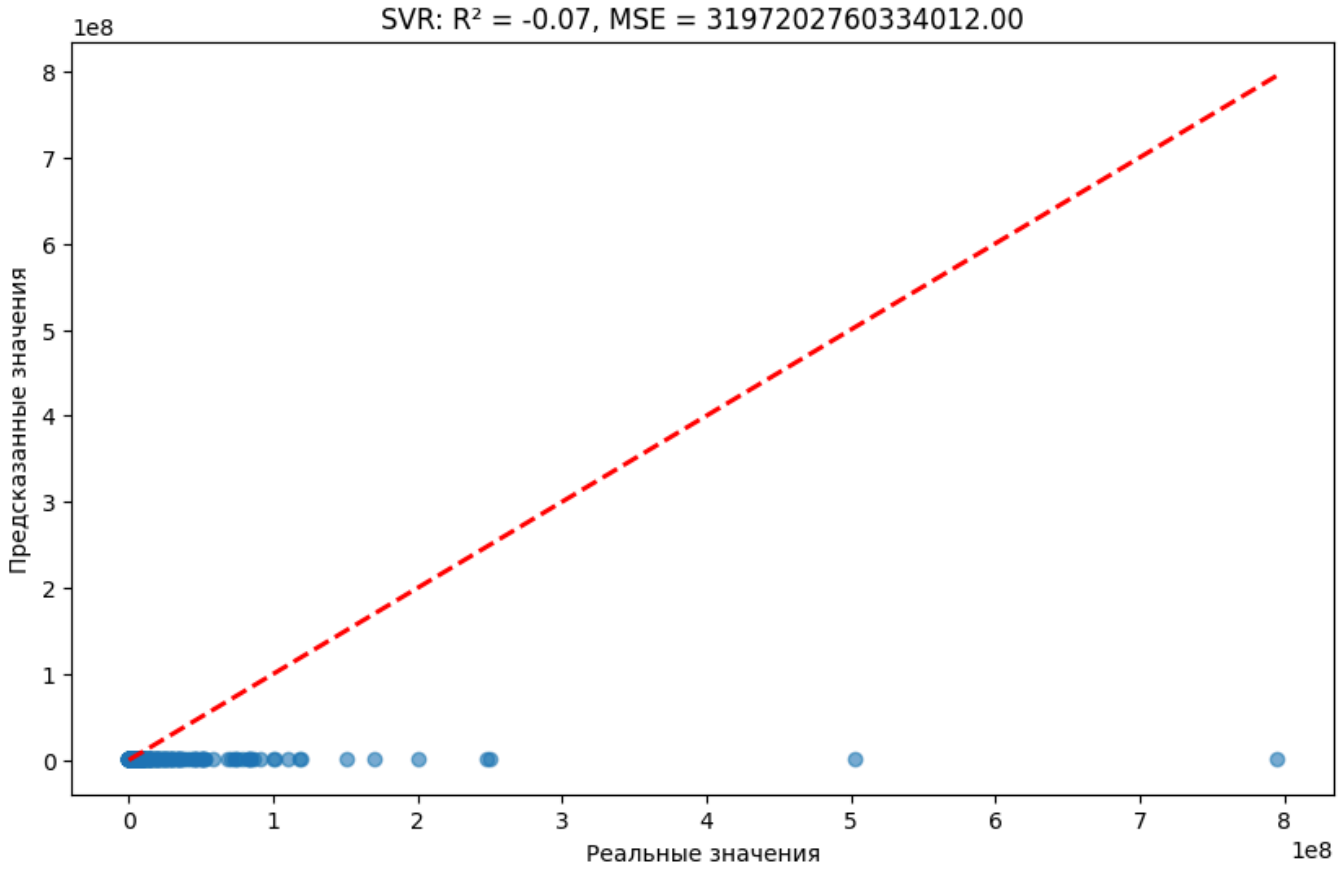
5 rows × 39 columns

```
# Удалим неинформативные поля
drop_cols = ['permalink', 'name', 'homepage_url', 'first_funding_at',
'last_funding_at',
             'founded_at', 'founded_month', 'founded_quarter', 'city',
'region']
df.drop(columns=drop_cols, inplace=True)

# Очистка целевого признака (удалим $, запятые, 'nan', '-', заменим на NaN)
df['funding_total_usd'] = df['funding_total_usd'].replace('[\$,-, ]', '',
regex=True)
df['funding_total_usd'] = pd.to_numeric(df['funding_total_usd'],
errors='coerce')

# Преобразование категориальных признаков
categorical_cols = ['category_list', 'market', 'status', 'country_code',
'state_code']
for col in categorical_cols:
    df[col] = df[col].astype('category')
    df[col] = df[col].cat.codes.replace(-1, np.nan)

# Заполнение пропусков медианой
imputer = SimpleImputer(strategy='median')
df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns)

# Удалим строки с NaN в целевом признаке (если остались)
```

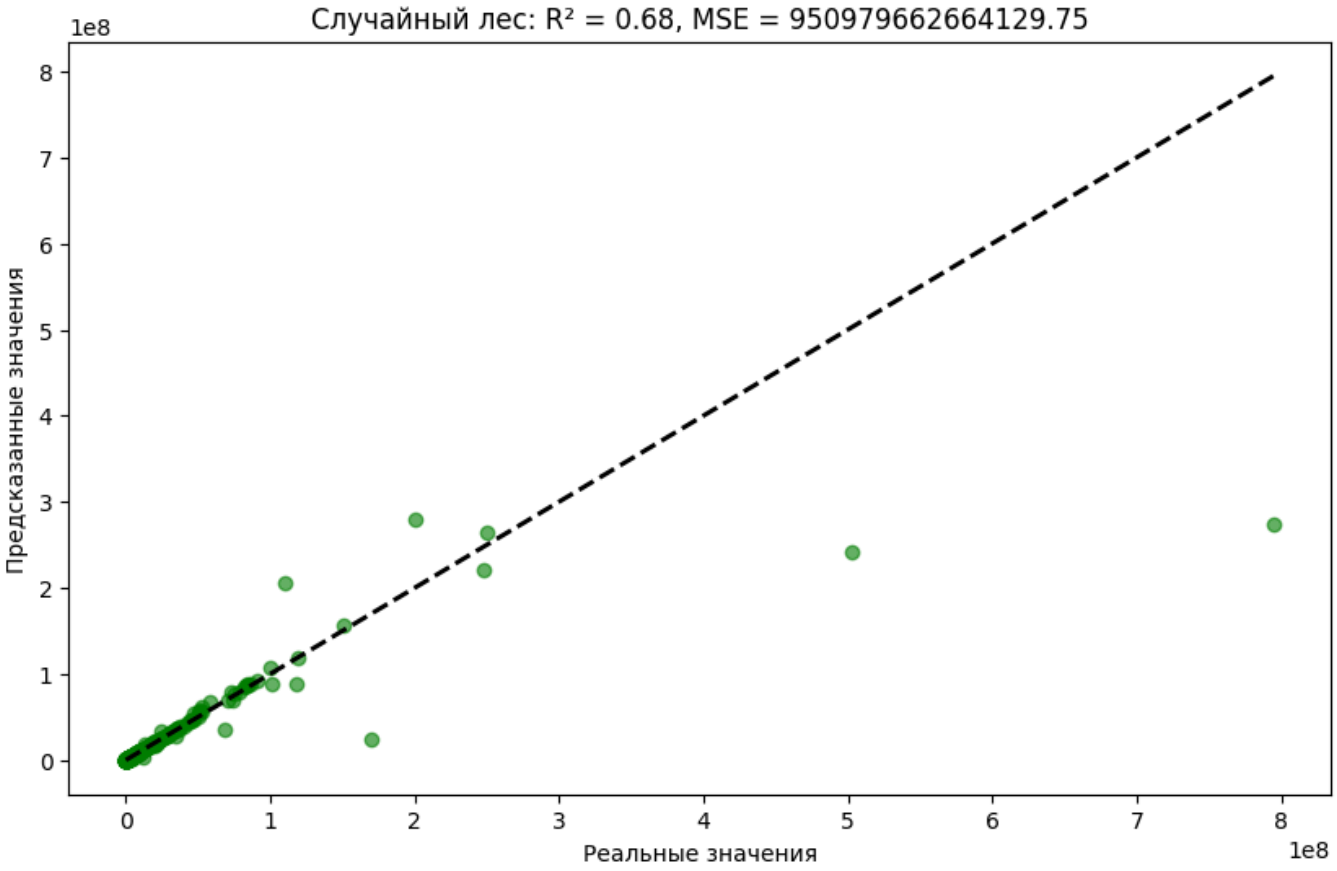
```
df_imputed = df_imputed[df_imputed['funding_total_usd'].notna()]
```

```
X = df_imputed.drop('funding_total_usd', axis=1)
y = df_imputed['funding_total_usd']

# Ограничим объём для ускорения (например, 2000 строк)
X = X.iloc[:2000]
y = y.iloc[:2000]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Масштабирование признаков
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

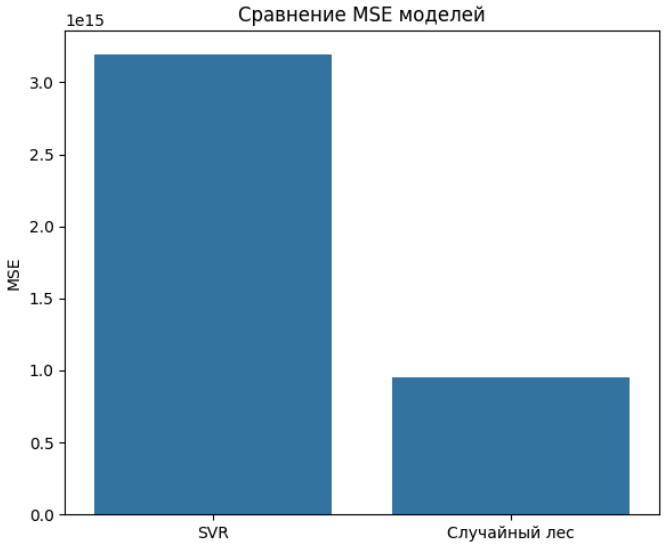
```
svr = SVR(kernel='rbf', C=100, epsilon=1.0)
svr.fit(X_train_scaled, y_train)
y_pred_svr = svr.predict(X_test_scaled)

mse_svr = mean_squared_error(y_test, y_pred_svr)
r2_svr = r2_score(y_test, y_pred_svr)

plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred_svr, alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--',
lw=2)
plt.xlabel('Реальные значения')
plt.ylabel('Предсказанные значения')
plt.title(f'SVR: R² = {r2_svr:.2f}, MSE = {mse_svr:.2f}')
plt.show()
```

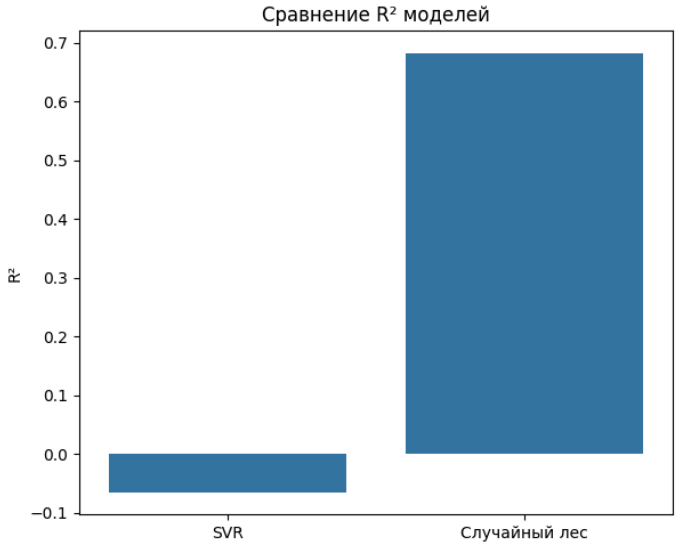



```
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)

mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred_rf, alpha=0.6, color='green')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--',
lw=2)
plt.xlabel('Реальные значения')
plt.ylabel('Предсказанные значения')
plt.title(f'Случайный лес: R² = {r2_rf:.2f}, MSE = {mse_rf:.2f}')
plt.show()
```




```
models = ['SVR', 'Случайный лес']
mse_scores = [mse_svr, mse_rf]
r2_scores = [r2_svr, r2_rf]

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.barplot(x=models, y=mse_scores)
plt.title('Сравнение MSE моделей')
plt.ylabel('MSE')

plt.subplot(1, 2, 2)
sns.barplot(x=models, y=r2_scores)
plt.title('Сравнение R² моделей')
plt.ylabel('R²')

plt.tight_layout()
plt.show()
```


Сравнение MSE моделей
1e15
3.0
2.5
2.0
1.5
1.0
0.5
0.0
MSE
SVR
Случайный лес


Сравнение R² моделей
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
R²
SVR
Случайный лес