CS 6220 Final Project: Predicting Firms' Financial Risk

Data Cleaning, EDA, and Baseline Model

Group Members: Jiajun Fang, Yini Li

In [1]:
```
# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
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.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
import warnings
warnings.filterwarnings('ignore')
```

Load Parquet Files

In [2]:
```
# Step 1: Import required libraries
import pandas as pd
import os
```

In [3]:
```
# Step 2: Check if the Parquet files exist in the current directory
train_file = "20231124_Financial_Risk_Project_train (1).parquet"
test_file = "20231124_Financial_Risk_Project_test_public (1).parquet"

print("Train file exists:", os.path.exists(train_file))
print("Test file exists:", os.path.exists(test_file))
```

```
Train file exists: True
Test file exists: True
```

In [8]:
```
# Step 3: Load the parquet files
train_df = pd.read_parquet(train_file)
test_df = pd.read_parquet(test_file)

print("Train shape:", train_df.shape)
print("Test shape:", test_df.shape)

train_df.head()
```

```
Train shape: (415, 73)
Test shape: (104, 71)
```

Out[8]:

ticker	url	call_transcript	VWAP	exchangeCountry	securityType	CIK	name	securityID	incorporationCountry	exchangeName	...
FICO	https://seekingalpha.com/article/4649507-fair-...	Fair Isaac Corporation (FICO) Q4 2023 Earnings...	700.83	USA	Common or ordinary	0000814547	Fair Isaac Corporation	138240101	USA	New York Stock Exchange	...
RXRX	https://seekingalpha.com/earnings/earnings-cal...	Earnings Call Transcripts | Seeking Alpha\r\n...	6.67	USA	Common or ordinary	0001601830	Recursion Pharmaceuticals Inc	384740101	USA	Nasdaq Stock Market	...
BLDR	https://seekingalpha.com/article/4645938-build...	Builders FirstSource, Inc. (BLDR) Q3 2023 Earn...	88.34	USA	Common or ordinary	0001316835	Builders FirstSource Inc	1630360101	USA	New York Stock Exchange	...
JKHY	https://seekingalpha.com/article/4649242-jack-...	Jack Henry & Associates, Inc. (JKHY) Q1 2024 E...	150.47	USA	Common or ordinary	0000779152	Henry (Jack) & Associates Inc	118110101	USA	Nasdaq Stock Market	...
OC	https://seekingalpha.com/article/4643409-owens-...	Owens Corning (OC) Q3 2023 Earnings Call Trans...	95.47	USA	Common or ordinary	0001370946	Owens Corning	82140601	USA	New York Stock Exchange	...

5 rows × 73 columns

In [ ]:
```
# Use the already loaded parquet data
df = train_df.copy()
print(f"Dataset shape: {df.shape}")
df.head()
```

```
Dataset shape: (415, 73)
```

Out [ ]:

ticker	url	call_transcript	VWAP	exchangeCountry	securityType	CIK	name	securityID	incorporationCountry	exchangeName	...
FICO	https://seekingalpha.com/article/4649507-fair-...	Fair Isaac Corporation (FICO) Q4 2023 Earnings...	700.83	USA	Common or ordinary	0000814547	Fair Isaac Corporation	138240101	USA	New York Stock Exchange	...
RXRX	https://seekingalpha.com/earnings/earnings-cal...	Earnings Call Transcripts | Seeking Alpha\r\n...	6.67	USA	Common or ordinary	0001601830	Recursion Pharmaceuticals Inc	384740101	USA	Nasdaq Stock Market	...
BLDR	https://seekingalpha.com/article/4645938-build...	Builders FirstSource, Inc. (BLDR) Q3 2023 Earn...	88.34	USA	Common or ordinary	0001316835	Builders FirstSource Inc	1630360101	USA	New York Stock Exchange	...
JKHY	https://seekingalpha.com/article/4649242-jack-...	Jack Henry & Associates, Inc. (JKHY) Q1 2024 E...	150.47	USA	Common or ordinary	0000779152	Henry (Jack) & Associates Inc	118110101	USA	Nasdaq Stock Market	...
OC	https://seekingalpha.com/article/4643409-owens-...	Owens Corning (OC) Q3 2023 Earnings Call Trans...	95.47	USA	Common or ordinary	0001370946	Owens Corning	82140601	USA	New York Stock Exchange	...

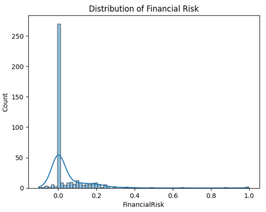
5 rows × 73 columns

Data Cleaning

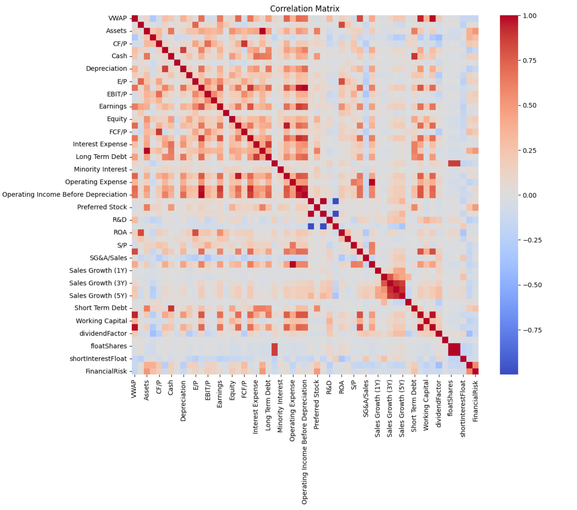
In [9]:
```
# Drop irrelevant features
irrelevant_cols = ['url', 'exchangeCountry', 'CIK', 'securityID',
                   'incorporationCountry', 'exchangeName', 'exchangeID', 'businessDescription']
df.drop(columns=irrelevant_cols, inplace=True, errors='ignore')

# Drop columns with more than 20% missing values
missing_threshold = 0.2
df.dropna(axis=1, thresh=(1 - missing_threshold) * len(df), inplace=True)

# Separate numerical and categorical columns
num_cols = df.select_dtypes(include=['float64', 'int64']).columns.tolist()
cat_cols = df.select_dtypes(include=['object']).columns.tolist()
```

Exploratory Data Analysis (EDA)

In [10]:
```
# Check distribution of target
sns.histplot(df['FinancialRisk'], kde=True)
plt.title('Distribution of Financial Risk')
plt.show()

# Log-transform target
df['FinancialRisk_log'] = np.log1p(df['FinancialRisk'])
```



In [11]:
```
# Correlation matrix
plt.figure(figsize=(12, 10))
corr = df[num_cols].corr()
sns.heatmap(corr, cmap='coolwarm', center=0)
plt.title('Correlation Matrix')
plt.show()
```



Baseline Model: Linear Regression

In [31]:
```
# Baseline Model: Linear Regression
selected_features = ['Assets', 'Cash', 'Debt/Equity', 'B/P', 'SG&A/Sales', 'Long Liabilities']
print('Columns in DataFrame:', df.columns.tolist())
missing_features = [col for col in selected_features if col not in df.columns]
if missing_features:
    print('The following features are missing from the data:', missing_features)
else:
    print('All selected features are present in the data.')
if 'FinancialRisk_log' not in df.columns:
    print('FinancialRisk_log column is missing. Please make sure the previous cell has been executed.')
else:
    print('FinancialRisk_log column exists.')
print('Missing value count for selected features:')
print(df[selected_features].isnull().sum())
# Convert selected features to numeric type
for col in selected_features:
    df[col] = pd.to_numeric(df[col], errors='coerce')
# Remove all rows with missing values in selected features or target
df_clean = df.dropna(subset=selected_features + ['FinancialRisk_log']).copy()
print('Shape after dropping missing:', df_clean.shape)
# Continue with modeling
X = df_clean[selected_features]
y = df_clean['FinancialRisk_log']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, X.columns)
    ])
from sklearn.linear_model import LinearRegression
model = Pipeline(steps=[('preprocessor', preprocessor),
                        ('regressor', LinearRegression())])
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
from sklearn.metrics import mean_squared_error, r2_score
r2 = r2_score(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred) ** 0.5
print(f'R² Score: {r2:.2f}')
print(f'RMSE: {rmse:.2f}')
```

```
Columns in DataFrame: ['call_transcript', 'VWAP', 'securityType', 'name', 'Accrual Ratio', 'Assets', 'B/P', 'CF/P', 'Capital Expenditure', 'Cash', 'Debt/Equity', 'Depreciation', 'Dividend', 'E/P', 'EBIT', 'EBIT/P', 'EBIT/TEV', 'Earnings', 'Earnings Growth (1Y)', 'Equity', 'FCF', 'FCF/P', 'Income Tax', 'Interest Expense', 'Long Liabilities', 'Long Term Debt', 'Market Cap', 'Minority Interest', 'Operating Cash Flow', 'Operating Expense', 'Operating Income', 'Operating Income Before Depreciation', 'Operating Margin', 'Preferred Stock', 'Profit Margin', 'R&D', 'R&D/Sales', 'ROA', 'ROE', 'S/P', 'SG&A', 'SG&A/Sales', 'Sales', 'Sales Growth (1Y)', 'Sales Growth (2Y)', 'Sales Growth (3Y)', 'Sales Growth (4Y)', 'Sales Growth (5Y)', 'Sales Variability', 'Short Term Debt', 'TEV', 'Working Capital', 'close', 'dividendFactor', 'floatShares', 'outstandingShares', 'shortInterestFloat', 'FinancialSector', 'FinancialRisk', 'FinancialRisk_log']
All selected features are present in the data.
FinancialRisk_log column exists.
Missing value count for selected features:
Assets                0
Cash                  0
Debt/Equity          18
B/P                   0
SG&A/Sales            7
Long Liabilities      0
dtype: int64
Shape after dropping missing: (390, 61)
R² Score: -0.08
RMSE: 0.08
```