In [2]:

```
pip install numpy pandas matplotlib seaborn scikit-learn
```

```
Requirement already satisfied: numpy in c:\users\dgous\anaconda3\lib\site-pac
kages (1.24.3)
Requirement already satisfied: pandas in c:\users\dgous\anaconda3\lib\site-pa
ckages (1.5.3)
Requirement already satisfied: matplotlib in c:\users\dgous\anaconda3\lib\sit
e-packages (3.7.1)
Requirement already satisfied: seaborn in c:\users\dgous\anaconda3\lib\site-p
ackages (0.12.2)
Requirement already satisfied: scikit-learn in c:\users\dgous\anaconda3\lib\s
ite-packages (1.4.1.post1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\dgous\anaco
nda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\dgous\anaconda3\lib\s
ite-packages (from pandas) (2022.7)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dgous\anaconda3\l
ib\site-packages (from matplotlib) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\dgous\anaconda3\lib\s
ite-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dgous\anaconda3
\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dgous\anaconda3
\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\dgous\anaconda3\li
b\site-packages (from matplotlib) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dgous\anaconda3\lib
\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\dgous\anaconda3\l
ib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: scipy>=1.6.0 in c:\users\dgous\anaconda3\lib\s
ite-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\dgous\anaconda3\lib
\site-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dgous\anacond
a3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\dgous\anaconda3\lib\site-
packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

In [6]:

```
# Set random seed for reproducibility
np.random.seed(42)

# Define the number of samples
num_samples = 200

# Generate synthetic formulation data
granulation_method = np.random.choice([0, 1], num_samples)  # 0 = Dry, 1 = Wet
compression_force = np.random.uniform(5, 50, num_samples)  # Compression force
binder_ratio = np.random.uniform(1, 10, num_samples)  # Binder ratio in %

# Batch Failure Rate Calculation (Logarithmic Influence)
batch_failure_rate = 30 - 5 * granulation_method - 0.3 * np.log(compression_fc
batch_failure_rate = np.clip(batch_failure_rate, 0, None)  # Ensure non-negati

# Create DataFrame
df_granulation = pd.DataFrame({
    "Granulation Method (Wet=1, Dry=0)": granulation_method,
    "Compression Force (kN)": compression_force,
    "Binder Ratio (%)": binder_ratio,
    "Batch Failure Rate (%)": batch_failure_rate
})

# Display first few rows
print(df_granulation.head())
```

```
   Granulation Method (Wet=1, Dry=0)  Compression Force (kN)  \
0                                  0                6.414313
1                                  1               33.638469
2                                  0               19.146019
3                                  0               27.885681
4                                  0               45.840491

   Binder Ratio (%)  Batch Failure Rate (%)
0          1.465135               29.020301
1          5.782192               22.311904
2          5.865716               26.526546
3          6.736869               24.609472
4          7.534822               23.914965
```

In [7]:

```
# Log transformation for compression force and binder ratio
df_granulation["log(Compression Force)"] = np.log(df_granulation["Compression
df_granulation["log(Binder Ratio)"] = np.log(df_granulation["Binder Ratio (%)"

# Define Features and Target
X_granulation = df_granulation[["Granulation Method (Wet=1, Dry=0)", "log(Comp
y_granulation = df_granulation["Batch Failure Rate (%)"]
```

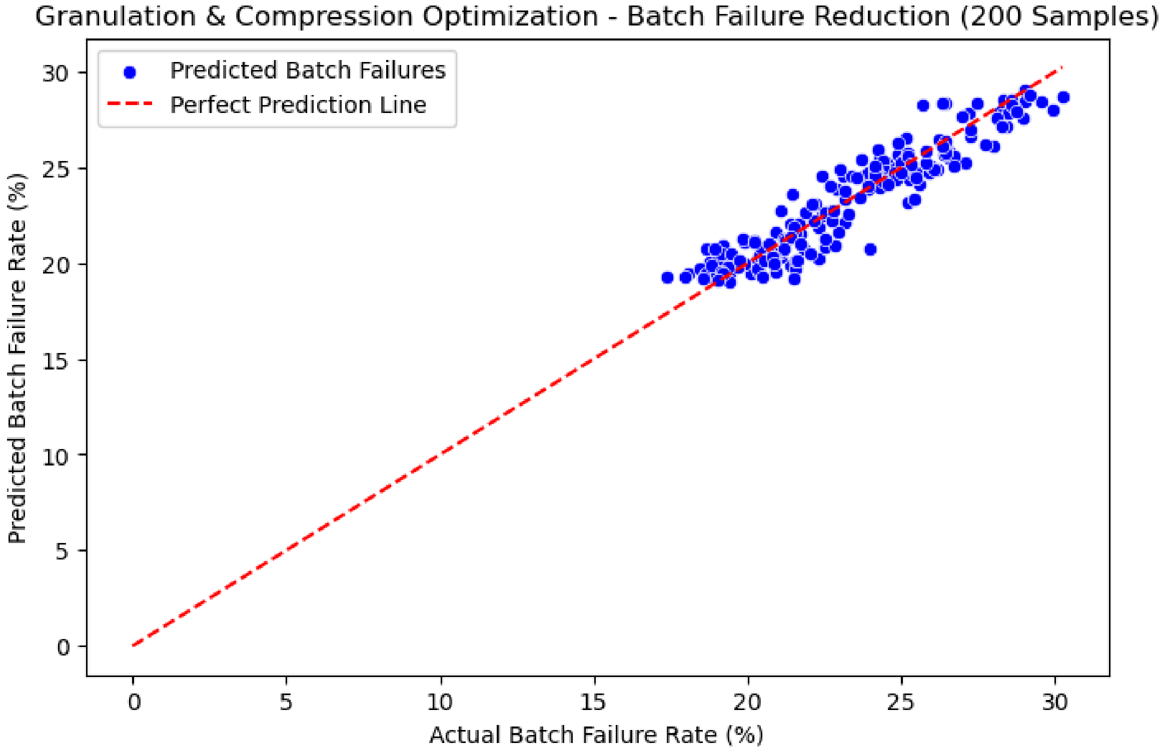
In [8]:

```
# Train a linear regression model
reg_model_granulation = LinearRegression()
reg_model_granulation.fit(X_granulation, y_granulation)

# Predict batch failure rates
batch_failure_pred = reg_model_granulation.predict(X_granulation)

print("Model trained successfully!")
```

```
Model trained successfully!
```

In [10]:

```
plt.figure(figsize=(8, 5))
sns.scatterplot(x=batch_failure_rate, y=batch_failure_pred, color="blue", labe
plt.plot([0, max(batch_failure_rate)], [0, max(batch_failure_rate)], color="re
plt.xlabel("Actual Batch Failure Rate (%)")
plt.ylabel("Predicted Batch Failure Rate (%)")
plt.title("Granulation & Compression Optimization - Batch Failure Reduction (2
plt.legend()
plt.show()
```



In [ ]: