TripleTen Project 9

July 28, 2024

1 Sprint 9 Project

2 Introduction

As a data scientist at OilyGiant mining company, I have been tasked with finding the optimal location to develop a new oil well that will maximize profits for the company. To do this, I will analyze oil well data from three different regions, build a linear regression model to predict the volume of reserves in new wells, and use the model to estimate the potential profits and risks for each region.

The project will involve several key steps: 1. Collecting and preparing oil well data for each region, including oil quality and volume of reserves 2. Building and evaluating a linear regression model for each region using a train/test split 3. Calculating the minimum volume of reserves needed for a profitable new well 4. Selecting the best oil wells in each region based on the model's predictions 5. Estimating the total profit for the top well sites in each region 6. Assessing the potential profit and loss risks for each region using the bootstrapping statistical technique 7. Recommending the optimal region for development based on which one offers the highest estimated profit with a risk of loss below 2.5%

By carefully analyzing the data, building predictive models, and weighing potential risks and rewards, this project aims to provide a data-driven recommendation for where OilyGiant can develop a new oil well to maximize returns on its investment. The datasets, code, and detailed findings from each stage of the analysis are presented in the following sections.

2.1 Preparation of Data

2.1.1 Description of the Data

Geological exploration data for the three regions are stored in files:

`geo_data_0.csv`

`geo_data_1.csv`

`geo_data_2.csv`

`id` — unique oil well identifier

`f0`, `f1`, `f2` — three features of points (their specific meaning is unimportant, but the features themselves are significant)

`product` — volume of reserves in the oil well (thousand barrels).

2.1.2 Data Conditions

Only linear regression is suitable for model training (the rest are not sufficiently predictable). When exploring the region, a study of 500 points is carried with picking the best 200 points for the profit calculation.

The budget for development of 200 oil wells is 100 USD million.

One barrel of raw materials brings 4.5 USD of revenue The revenue from one unit of product is 4,500 dollars (volume of reserves is in thousand barrels).

After the risk evaluation, keep only the regions with the risk of losses lower than 2.5%. From the ones that fit the criteria, the region with the highest average profit should be selected.

The data is synthetic: contract details and well characteristics are not disclosed.

2.1.3 Intialization

[1]:
```
# Loading all the libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.utils import resample
import numpy as np
```

2.1.4 Load Data

[2]:
```
#Load tables into dataframes

#geo_data_0.csv
geo_data_0 = pd.read_csv('/datasets/geo_data_0.csv')

#geo_data_1.csv
geo_data_1 = pd.read_csv('/datasets/geo_data_1.csv')

#geo_data_2.csv
geo_data_2 = pd.read_csv('/datasets/geo_data_2.csv')
```

2.1.5 General Information

geo_data_0

[3]:
```
geo_data_0.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   id       100000 non-null  object
 1   f0       100000 non-null  float64
```

```
 2   f1       100000 non-null  float64
 3   f2       100000 non-null  float64
 4   product  100000 non-null  float64
dtypes: float64(4), object(1)
memory usage: 3.8+ MB
```

[4]:
```
display(geo_data_0)
```

```
          id        f0        f1        f2     product
0      txEyH  0.705745 -0.497823  1.221170  105.280062
1      2acmU  1.334711 -0.340164  4.365080   73.037750
2      409Wp  1.022732  0.151990  1.419926   85.265647
3      iJLyR -0.032172  0.139033  2.978566  168.620776
4      Xdl7t  1.988431  0.155413  4.751769  154.036647
…      …       …        …        …          …
99995  DLsed  0.971957  0.370953  6.075346  110.744026
99996  QKivN  1.392429 -0.382606  1.273912  122.346843
99997  3rnvd  1.029585  0.018787 -1.348308   64.375443
99998  7kl59  0.998163 -0.528582  1.583869   74.040764
99999  1CWhH  1.764754 -0.266417  5.722849  149.633246

[100000 rows x 5 columns]
```

[5]:
```
geo_data_0.describe()
```

[5]:

	f0	f1	f2	product
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	0.500419	0.250143	2.502647	92.500000
std	0.871832	0.504433	3.248248	44.288691
min	-1.408605	-0.848218	-12.088328	0.000000
25%	-0.072580	-0.200881	0.287748	56.497507
50%	0.502360	0.250252	2.515969	91.849972
75%	1.073581	0.700646	4.715088	128.564089
max	2.362331	1.343769	16.003790	185.364347

geo_data_1

[6]:
```
geo_data_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   id       100000 non-null  object
 1   f0       100000 non-null  float64
 2   f1       100000 non-null  float64
 3   f2       100000 non-null  float64
 4   product  100000 non-null  float64
```

```
dtypes: float64(4), object(1)
memory usage: 3.8+ MB
```

[7]:
```
display(geo_data_1)
```

```
          id         f0         f1        f2     product
0      kBEdx -15.001348  -8.276000 -0.005876    3.179103
1      62mP7  14.272088  -3.475083  0.999183   26.953261
2      vyE1P   6.263187  -5.948386  5.001160  134.766305
3      KcrkZ -13.081196 -11.506057  4.999415  137.945408
4      AHL4O  12.702195  -8.147433  5.004363  134.766305
…      …        …        …        …          …
99995  QywKC   9.535637  -6.878139  1.998296   53.906522
99996  ptvty -10.160631 -12.558096  5.005581  137.945408
99997  09gWa  -7.378891  -3.084104  4.998651  137.945408
99998  rqwUm   0.665714  -6.152593  1.000146   30.132364
99999  relB0  -3.426139  -7.794274 -0.003299    3.179103

[100000 rows x 5 columns]
```

[8]:
```
geo_data_1.describe()
```

[8]:

	f0	f1	f2	product
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	1.141296	-4.796579	2.494541	68.825000
std	8.965932	5.119872	1.703572	45.944423
min	-31.609576	-26.358598	-0.018144	0.000000
25%	-6.298551	-8.267985	1.000021	26.953261
50%	1.153055	-4.813172	2.011479	57.085625
75%	8.621015	-1.332816	3.999904	107.813044
max	29.421755	18.734063	5.019721	137.945408

geo_data_2

[9]:
```
geo_data_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 5 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   id       100000 non-null  object
 1   f0       100000 non-null  float64
 2   f1       100000 non-null  float64
 3   f2       100000 non-null  float64
 4   product  100000 non-null  float64
dtypes: float64(4), object(1)
memory usage: 3.8+ MB
```

[10]:
```
display(geo_data_2)
```

	id	f0	f1	f2	product
0	fwXo0	-1.146987	0.963328	-0.828965	27.758673
1	WJtFt	0.262778	0.269839	-2.530187	56.069697
2	ovLUW	0.194587	0.289035	-5.586433	62.871910
3	q6cA6	2.236060	-0.553760	0.930038	114.572842
4	WPMUX	-0.515993	1.716266	5.899011	149.600746
…	…	…	…	…	…
99995	4GxBu	-1.777037	1.125220	6.263374	172.327046
99996	YKFjq	-1.261523	-0.894828	2.524545	138.748846
99997	tKPY3	-1.199934	-2.957637	5.219411	157.080080
99998	nmxp2	-2.419896	2.417221	-5.548444	51.795253
99999	V9kWn	-2.551421	-2.025625	6.090891	102.775767

[100000 rows x 5 columns]

[11]:
```
geo_data_2.describe()
```

[11]:

	f0	f1	f2	product
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	0.002023	-0.002081	2.495128	95.000000
std	1.732045	1.730417	3.473445	44.749921
min	-8.760004	-7.084020	-11.970335	0.000000
25%	-1.162288	-1.174820	0.130359	59.450441
50%	0.009424	-0.009482	2.484236	94.925613
75%	1.158535	1.163678	4.858794	130.595027
max	7.238262	7.844801	16.739402	190.029838

2.1.6 Fix the Data

For the three tables there are no null values and all of the columns have the correct data type.

I will do my due dilligence and check every table for duplicates.

[12]:
```
# Get the number of duplicate rows

num_duplicates_0 = geo_data_0.duplicated().sum()

print('num_duplicates_0:',num_duplicates_0)

num_duplicates_1 = geo_data_1.duplicated().sum()

print('num_duplicates_1:',num_duplicates_1)

num_duplicates_2 = geo_data_2.duplicated().sum()

print('num_duplicates_2:',num_duplicates_2)
```

```
num_duplicates_0: 0
num_duplicates_1: 0
num_duplicates_2: 0
```

I will drop the column `id` from all three tables because it will interfere with the model's ability to work effectively

[13]:
```
geo_data_0 = geo_data_0.drop('id', axis=1)

geo_data_1 = geo_data_1.drop('id', axis=1)

geo_data_2 = geo_data_2.drop('id', axis=1)
```

2.2 Train and test the model for each region:

[14]:
```
def train_and_evaluate_model(data):
    # 2.1. Split the data into a training set and validation set at a ratio of␣
 ↪75:25
    X = data.drop('product', axis=1)
    y = data['product']
    X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.25,␣
 ↪random_state=42)

    # 2.2. Train the model and make predictions for the validation set
    model = LinearRegression()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_valid)

    # 2.3. Save the predictions and correct answers for the validation set
    validation_data = X_valid.copy()
    validation_data['actual'] = y_valid
    validation_data['predicted'] = y_pred

    # 2.4. Print the average volume of predicted reserves and model RMSE
    avg_predicted_reserves = y_pred.mean()
    rmse = mean_squared_error(y_valid, y_pred, squared=False)
    print(f"Average volume of predicted reserves: {avg_predicted_reserves:.2f}")
    print(f"Model RMSE: {rmse:.2f}")

    return validation_data, rmse

# Analyze each region
for region, data in zip(['Region 0', 'Region 1', 'Region 2'], [geo_data_0,␣
 ↪geo_data_1, geo_data_2]):
    print(f"\n{region}:")
    validation_data, rmse = train_and_evaluate_model(data)

    # 2.5. Analyze the results
```

```
    print(f"Validation data sample:")
    print(validation_data.head())
```

```
Region 0:
Average volume of predicted reserves: 92.40
Model RMSE: 37.76
Validation data sample:
             f0        f1        f2      actual   predicted
75721  0.599283 -0.557623  2.121187  122.073350  101.901017
80184  0.739017 -0.463156 -1.347584   48.738540   78.217774
19864  1.422743 -0.534917  3.718798  131.338088  115.266901
76699  1.580244 -0.238458  2.805149   88.327757  105.618618
92991  0.918974  0.023961  2.598575   36.959266   97.980185

Region 1:
Average volume of predicted reserves: 68.71
Model RMSE: 0.89
Validation data sample:
              f0        f1        f2      actual   predicted
75721   6.078076  0.084568  0.002957    0.000000    0.844738
80184  16.462386  2.712946  1.993030   53.906522   52.921612
19864   7.051898  0.766983  4.990194  134.766305  135.110385
76699  -0.240045 -0.380804  3.999693  107.813044  109.494863
92991  13.350111 -8.558281  0.002010    0.000000   -0.047292

Region 2:
Average volume of predicted reserves: 94.77
Model RMSE: 40.15
Validation data sample:
             f0        f1        f2      actual   predicted
75721  2.111118 -1.679773  3.112240  117.441301   98.301916
80184  0.734759  0.747788  3.670879   47.841249  101.592461
19864 -2.513109  0.844631 -4.922889   45.883483   52.449099
76699 -2.035301 -1.522988  5.072839  139.014608  109.922127
92991  2.744145  1.429952 -1.372661   84.004276   72.411847
```

2.2.1 Conclusion:

Based on the results for each region, we can make the following observations:

1. Region 0:
   - The average volume of predicted reserves is 92.40.
   - The model's RMSE is 37.76, indicating a relatively high prediction error.
   - The validation data sample shows that the model's predictions are not very close to the actual values.
2. Region 1:
   - The average volume of predicted reserves is 68.71.

- The model's RMSE is 0.89, which is significantly lower compared to Region 0, suggesting better prediction accuracy.
- The validation data sample demonstrates that the model's predictions are much closer to the actual values.

3. Region 2:
   - The average volume of predicted reserves is 94.77.
   - The model's RMSE is 40.15, indicating a high prediction error, similar to Region 0.
   - The validation data sample reveals that the model's predictions deviate considerably from the actual values.

In conclusion, among the three regions, the linear regression model performs best for Region 1, as evidenced by the lowest RMSE value of 0.89. This suggests that the model is able to predict the volume of reserves in Region 1 with higher accuracy compared to the other two regions. Region 0 and Region 2 have significantly higher RMSE values, indicating that the model's predictions are less reliable for these regions.

2.3 Prepare for profit calculation

[15]:
```
# 3.1. Store all key values for calculations in separate variables
oil_price = 4.5  # price of oil per barrel in dollars
budget = 100000000  # budget for oil well development in dollars
num_wells = 200  # number of oil wells to be developed
revenue_per_unit = oil_price * 1000  # revenue per unit of oil (1 unit = 1000␣
 ↪barrels)

# 3.2. Calculate the volume of reserves sufficient for developing a new well␣
 ↪without losses
min_volume_reserve = budget / (num_wells * revenue_per_unit)

print(f"Minimum volume of reserves required for each well to avoid losses:␣
 ↪{min_volume_reserve:.2f} thousand barrels")

# Compare the minimum volume with the average volume of reserves in each region
region_sufficient = []
for region, data in zip(['Region 0', 'Region 1', 'Region 2'], [geo_data_0,␣
 ↪geo_data_1, geo_data_2]):
    avg_reserves = data['product'].mean()
    print(f"\n{region}:")
    print(f"Average volume of reserves: {avg_reserves:.2f} thousand barrels")
    if avg_reserves >= min_volume_reserve:
        print(f"The average volume of reserves in {region} is sufficient for␣
 ↪developing a new well without losses.")
        region_sufficient.append(region)
    else:
        print(f"The average volume of reserves in {region} is not sufficient␣
 ↪for developing a new well without losses.")
```

```
# 3.3. Provide the findings about the preparation for profit calculation step
print("\nFindings:")
print(f"- The minimum volume of reserves required for each well to avoid losses␣
 ↪is {min_volume_reserve:.2f} thousand barrels.")
print("- Based on the comparison with the average volume of reserves in each␣
 ↪region:")
for region in ['Region 0', 'Region 1', 'Region 2']:
    if region in region_sufficient:
        print(f"  - {region}: Sufficient reserves for profitable well␣
 ↪development.")
    else:
        print(f"  - {region}: Insufficient reserves for profitable well␣
 ↪development.")
if len(region_sufficient) > 0:
    print(f"- {' and '.join(region_sufficient)} have a higher potential for␣
 ↪profitable well development.")
else:
    print("- None of the regions have sufficient reserves for profitable well␣
 ↪development.")
```

```
Minimum volume of reserves required for each well to avoid losses: 111.11
thousand barrels

Region 0:
Average volume of reserves: 92.50 thousand barrels
The average volume of reserves in Region 0 is not sufficient for developing a
new well without losses.

Region 1:
Average volume of reserves: 68.83 thousand barrels
The average volume of reserves in Region 1 is not sufficient for developing a
new well without losses.

Region 2:
Average volume of reserves: 95.00 thousand barrels
The average volume of reserves in Region 2 is not sufficient for developing a
new well without losses.

Findings:
- The minimum volume of reserves required for each well to avoid losses is
111.11 thousand barrels.
- Based on the comparison with the average volume of reserves in each region:
  - Region 0: Insufficient reserves for profitable well development.
  - Region 1: Insufficient reserves for profitable well development.
  - Region 2: Insufficient reserves for profitable well development.
- None of the regions have sufficient reserves for profitable well development.
```

2.3.1 Conclusion:

Based on the analysis performed in this section, we can conclude that none of the three regions under consideration have sufficient reserves for profitable well development. The key findings are as follows:

The minimum volume of reserves required for each well to avoid losses is calculated to be 111.11 thousand barrels. This value is derived from the given budget of $100,000,000, the number of wells to be developed (200), and the revenue per unit of oil ($4,500 per 1,000 barrels). The average volume of reserves in each region is compared against the minimum required volume:

Region 0 has an average volume of 92.50 thousand barrels Region 1 has an average volume of 68.83 thousand barrels Region 2 has an average volume of 95.00 thousand barrels

All three regions have average reserves below the minimum required volume of 111.11 thousand barrels. Consequently, none of the regions - Region 0, Region 1, or Region 2 - have sufficient reserves to ensure profitable well development. Developing wells in these regions with the given budget and the expected revenue per unit of oil would likely result in financial losses.

2.4 Write a function to calculate profit from a set of selected oil wells and model predictions

[25]:
```
def train_and_evaluate_model(data):
    # Split the data
    features = data[['f0', 'f1', 'f2']]
    target = data['product']
    X_train, X_val, y_train, y_val = train_test_split(features, target,␣
 ↪test_size=0.25, random_state=42)

    # Train the model
    model = LinearRegression()
    model.fit(X_train, y_train)

    # Make predictions
    predictions = model.predict(X_val)

    # Calculate RMSE
    rmse = mean_squared_error(y_val, predictions, squared=False)

    # Create validation dataset
    validation_data = X_val.copy()
    validation_data['product'] = y_val
    validation_data['predicted'] = predictions

    return validation_data, rmse

validation_data_list = []
rmse_list = []
```

```
# Analyze each region
for region, data in zip(['Region 0', 'Region 1', 'Region 2'], [geo_data_0,␣
 ↪geo_data_1, geo_data_2]):
    print(f"\n{region}:")
    validation_data, rmse = train_and_evaluate_model(data)
    validation_data_list.append(validation_data)
    rmse_list.append(rmse)

    # 2.5. Analyze the results
    print(f"Validation data sample:")
    print(validation_data.head())
    print(f"Average predicted volume of reserves: {validation_data['predicted'].
 ↪mean():.2f}")
    print(f"RMSE: {rmse:.2f}")

def calculate_profit(data, num_wells, revenue_per_unit, development_budget):
    # 4.1. Pick the wells with the highest values of predictions
    data = data.sort_values(by='predicted', ascending=False).head(num_wells)

    # 4.2. Summarize the target volume of reserves in accordance with these␣
 ↪predictions
    total_reserves = data['product'].sum()

    # Calculate the profit
    revenue = total_reserves * revenue_per_unit
    profit = revenue - development_budget

    return total_reserves, profit

# Add the 'predicted' column to the original DataFrames
geo_data_0['predicted'] = validation_data_list[0]['predicted']
geo_data_1['predicted'] = validation_data_list[1]['predicted']
geo_data_2['predicted'] = validation_data_list[2]['predicted']

# Calculate profit for each region
regions = ['Region 0', 'Region 1', 'Region 2']
profit_data = []

num_wells = 200
revenue_per_unit = 4500  # $4500 per thousand barrels
budget = 100_000_000  # $100 million

for region, data in zip(regions, [geo_data_0, geo_data_1, geo_data_2]):
    total_reserves, profit = calculate_profit(data, num_wells,␣
 ↪revenue_per_unit, budget)
    profit_data.append({'Region': region, 'Total Reserves': total_reserves,␣
 ↪'Profit': profit})
```

```python
# Create a DataFrame with profit data
profit_df = pd.DataFrame(profit_data)

# 4.3. Provide findings
print("Profit Calculation Results:")
print(profit_df)

most_profitable_region = profit_df.loc[profit_df['Profit'].idxmax(), 'Region']
max_profit = profit_df.loc[profit_df['Profit'].idxmax(), 'Profit']

print(f"\nMost Profitable Region: {most_profitable_region}")
print(f"Estimated Profit: ${max_profit:,.2f}")

if max_profit > 0:
    print(f"Recommendation: Develop oil wells in {most_profitable_region} for a␣
 ↪potential profit of ${max_profit:,.2f}.")
else:
    print("Recommendation: None of the regions are expected to be profitable.␣
 ↪Consider alternative investment opportunities.")
```

```
Region 0:
Validation data sample:
             f0        f1        f2     product   predicted
75721  0.599283 -0.557623  2.121187  122.073350  101.901017
80184  0.739017 -0.463156 -1.347584   48.738540   78.217774
19864  1.422743 -0.534917  3.718798  131.338088  115.266901
76699  1.580244 -0.238458  2.805149   88.327757  105.618618
92991  0.918974  0.023961  2.598575   36.959266   97.980185
Average predicted volume of reserves: 92.40
RMSE: 37.76

Region 1:
Validation data sample:
              f0        f1        f2     product   predicted
75721   6.078076  0.084568  0.002957    0.000000    0.844738
80184  16.462386  2.712946  1.993030   53.906522   52.921612
19864   7.051898  0.766983  4.990194  134.766305  135.110385
76699  -0.240045 -0.380804  3.999693  107.813044  109.494863
92991  13.350111 -8.558281  0.002010    0.000000   -0.047292
Average predicted volume of reserves: 68.71
RMSE: 0.89

Region 2:
Validation data sample:
             f0        f1        f2     product   predicted
75721  2.111118 -1.679773  3.112240  117.441301   98.301916
```

```
80184  0.734759  0.747788  3.670879   47.841249  101.592461
19864 -2.513109  0.844631 -4.922889   45.883483   52.449099
76699 -2.035301 -1.522988  5.072839  139.014608  109.922127
92991  2.744145  1.429952 -1.372661   84.004276   72.411847
Average predicted volume of reserves: 94.77
RMSE: 40.15
Profit Calculation Results:
     Region  Total Reserves        Profit
0  Region 0    29686.980254  3.359141e+07
1  Region 1    27589.081548  2.415087e+07
2  Region 2    27996.826132  2.598572e+07

Most Profitable Region: Region 0
Estimated Profit: $33,591,411.14
Recommendation: Develop oil wells in Region 0 for a potential profit of
$33,591,411.14.
```

2.4.1 Conclusion

Based on the profit calculation results, Region 0 emerges as the most promising region for oil well development. The analysis shows that developing the top 200 wells in Region 0 is expected to yield a total reserve volume of approximately 29,686.98 thousand barrels. With the given revenue per unit of oil and the development budget, this translates to an estimated profit of $33,591,411.14.

Region 2 ranks second in profitability, with a total reserve volume of 27,996.83 thousand barrels and an estimated profit of `$25,985,717.59`. Region 1 follows closely in third place, with a total reserve volume of 27,589.08 thousand barrels and an estimated profit of `$24,150,870.00`.

It's noteworthy that all three regions are projected to be profitable, which provides the company with multiple viable options for investment. The profitability ranking from highest to lowest is:

1. Region 0: $33,591,411.14
2. Region 2: $25,985,717.59
3. Region 1: $24,150,870.00

Considering this profitability analysis, the recommendation is to focus on developing oil wells in Region 0. This region offers the highest potential profit among the three regions evaluated, with an advantage of over $7.6 million compared to the next most profitable region.

However, the fact that all three regions show positive profit projections provides the company with flexibility in its development strategy. While Region 0 is the most attractive option, the company could consider diversifying its operations across multiple regions to mitigate risks, especially given the relatively close profit projections for Regions 1 and 2.

By prioritizing Region 0, the company can maximize its return on investment while keeping in mind the potential of the other regions for future development or risk diversification strategies. This approach allows for optimal resource allocation while maintaining options for expansion in other profitable areas.

2.5 Calculate risks and profit for each region

[26]:
```
def calculate_profit_distribution(validation_data, num_wells, revenue_per_unit,␣
 ↪development_budget, num_iterations=1000):
    profits = []
    for _ in range(num_iterations):
        # Sample 500 rows with replacement from the validation set
        resampled_data = resample(validation_data, replace=True, n_samples=500)
        total_reserves, profit = calculate_profit(resampled_data, num_wells,␣
 ↪revenue_per_unit, development_budget)
        profits.append(profit)
    return profits

# 5.1. Use the bootstrapping technique with 1000 samples to find the␣
 ↪distribution of profit
profit_distributions = {}
for region, validation_data in zip(regions, validation_data_list):
    profit_distributions[region] =␣
 ↪calculate_profit_distribution(validation_data, num_wells, revenue_per_unit,␣
 ↪budget)

# 5.2. Find average profit, 95% confidence interval and risk of losses
results = []
for region, profits in profit_distributions.items():
    avg_profit = np.mean(profits)
    ci_lower, ci_upper = np.percentile(profits, [2.5, 97.5])
    risk_of_loss = np.mean(np.array(profits) < 0) * 100

    results.append({
        'Region': region,
        'Average Profit': avg_profit,
        '95% CI Lower': ci_lower,
        '95% CI Upper': ci_upper,
        'Risk of Loss (%)': risk_of_loss
    })

# Create a DataFrame with the results
results_df = pd.DataFrame(results)

# 5.3. Provide findings
print("Risk and Profit Analysis Results:")
print(results_df)

profitable_regions = results_df[results_df['Risk of Loss (%)'] < 2.5]

if len(profitable_regions) > 0:
```

```
    best_region = profitable_regions.loc[profitable_regions['Average Profit'].
↪idxmax(), 'Region']
    avg_profit = profitable_regions.loc[profitable_regions['Average Profit'].
↪idxmax(), 'Average Profit']
    risk_of_loss = profitable_regions.loc[profitable_regions['Average Profit'].
↪idxmax(), 'Risk of Loss (%)']

    print(f"\nMost Promising Region: {best_region}")
    print(f"Average Profit: ${avg_profit:,.2f}")
    print(f"Risk of Loss: {risk_of_loss:.2f}%")

    print(f"\nRecommendation: Develop oil wells in {best_region} for an average␣
↪profit of ${avg_profit:,.2f} with a risk of loss of {risk_of_loss:.2f}%.")
else:
    print("\nRecommendation: None of the regions meet the acceptable risk␣
↪threshold of 2.5%. Consider alternative investment opportunities.")
```

```
Risk and Profit Analysis Results:
     Region  Average Profit  95% CI Lower  95% CI Upper  Risk of Loss (%)
0  Region 0    3.970389e+06 -1.701113e+06  9.022637e+06               6.5
1  Region 1    4.387155e+06  1.961251e+05  8.174389e+06               1.7
2  Region 2    3.619723e+06 -1.713178e+06  9.174340e+06               8.1

Most Promising Region: Region 1
Average Profit: $4,387,154.86
Risk of Loss: 1.70%

Recommendation: Develop oil wells in Region 1 for an average profit of
$4,387,154.86 with a risk of loss of 1.70%.
```

Based on the risk and profit analysis results, we can conclude the following:

Region 1 emerges as the most promising area for oil well development. It offers the highest average profit of $4,387,154.86 and, crucially, has the lowest risk of loss at 1.70%. This risk level falls below the 2.5% threshold set in the project conditions, making it the only region that meets this important criterion.

While Region 0 and Region 2 show potential for higher profits in their upper confidence intervals, their risk profiles are concerning. Region 0 has a 6.5% risk of loss, and Region 2 has an even higher 8.1% risk, both exceeding the acceptable threshold.

The 95% confidence intervals provide additional insights: - Region 1: `$196,125` to `$8,174,389`

- Region 0: `-$1,701,113` to `$9,022,637`
- Region 2: `-$1,713,178` to `$9,174,340`

Although Regions 0 and 2 have wider ranges that include higher potential profits, they also extend into negative territories, reflecting their higher risk.

Given these results, the recommendation is to focus oil well development efforts on Region 1.

This strategy balances the potential for profit with a manageable level of risk, aligning with the company's risk tolerance as specified in the project conditions. By choosing Region 1, the company can expect an average profit of over $4.3 million while maintaining a low 1.70% chance of incurring losses.

This approach provides the best combination of financial opportunity and risk mitigation among the three regions analyzed, making it the most prudent choice for investment.

3 Conlcusion

Based on the comprehensive analysis conducted throughout this project, we can draw the following conclusions:

1. Model Performance: Among the three regions analyzed, Region 1 demonstrated the best performance in terms of prediction accuracy, with the lowest RMSE of 0.89. This suggests that our linear regression model is most reliable for predicting oil reserves in Region 1.
2. Profit Potential: Initially, when considering average reserves, none of the regions appeared profitable. However, by using our predictive model to select the most promising wells, all three regions showed potential for profit.
3. Risk and Profit Analysis: After applying bootstrapping techniques to assess risk:
    - Region 1 emerged as the most promising, with an average profit of $4,387,154.86 and the lowest risk of loss at 1.70%.
    - Region 0 showed higher potential profits but also higher risk, with a 6.5% chance of loss.
    - Region 2 had the highest risk at 8.1%, despite showing potential for high profits.
4. Final Recommendation: Based on the project's risk tolerance threshold of 2.5%, Region 1 is the only area that meets this criterion. It offers the best balance of profit potential and risk mitigation.

In conclusion, we recommend that OilyGiant focus its oil well development efforts on Region 1. This strategy is expected to yield an average profit of over $4.3 million while maintaining a low 1.70% risk of loss. This approach aligns with the company's risk tolerance and offers the most prudent investment opportunity among the three regions analyzed.

It's important to note that while Regions 0 and 2 show potential for higher profits, their risk levels exceed the company's acceptable threshold. However, these regions could be considered for future exploration or development if the company's risk tolerance changes or if additional risk mitigation strategies can be implemented.

Moving forward, OilyGiant should: 1. Prioritize development in Region 1 2. Continue refining the predictive model, especially for Region 1 3. Consider further geological studies in Regions 0 and 2 to potentially reduce risk in these areas for future development

This data-driven approach should position OilyGiant to maximize its return on investment while managing risk appropriately in its oil well development strategy.