final_summary_analysis

February 13, 2025

[103]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import r2_score, mean_squared_error,␣
↪root_mean_squared_error, mean_absolute_error

sns.set_theme()
```

1 Final Summary

[96]:
```
data = pd.read_csv("../data/final_summary.csv")
data.describe()
```

[96]:

	Prediction	Target	Residual	Difference%
count	762.000000	762.000000	762.000000	762.000000
mean	15013.973739	16680.517441	1666.543702	33.654993
std	11663.728696	15936.152341	7873.292132	59.048843
min	1304.924179	800.000000	-31500.695009	0.060727
25%	7346.587111	7225.000000	-1808.301467	9.869726
50%	11375.998551	11500.000000	303.593068	21.808475
75%	19013.148937	19875.000000	3153.794625	37.821843
max	72362.597035	100000.000000	68954.744641	916.207230

1.0.1 Regression Metrics

[106]:
```
r2 = r2_score(data['Target'], data['Prediction'])
mse = mean_squared_error(data['Target'], data['Prediction'])
rmse = root_mean_squared_error(data['Target'], data['Prediction'])
mae = mean_absolute_error(data['Target'], data['Prediction'])

print(f"R^2 Score: {r2}")
print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
print(f"MAE: {mae}")
```

```
R^2 Score: 0.7449617897235532
MSE: 64684746.867536046
```

```
RMSE: 8042.682815300877
MAE: 4534.720900974607
```

R^2 Score R^2 tells us how well the model explains the variance in the target values. Since our score is 0.7449, it means that 74.49% of the variance in the target varible is explained by the model. It's not perfect, but it's fairly a strong fit.

MSE & RMSE Mean Squared Error is the avg squared difference between the predicted and actual values. Higher => larger errors, However, it's hard to interpret because it's squared and the unit is not the same as the target variable ($ for price in our case). Root Mean Squared Error is the square root of MSE, making it easier to interpret because it's in the smae unit as the target variable. Our RMSE value is 8042.68, meaning that on avverage, the prediction deviates from the actual values by $8042.68.

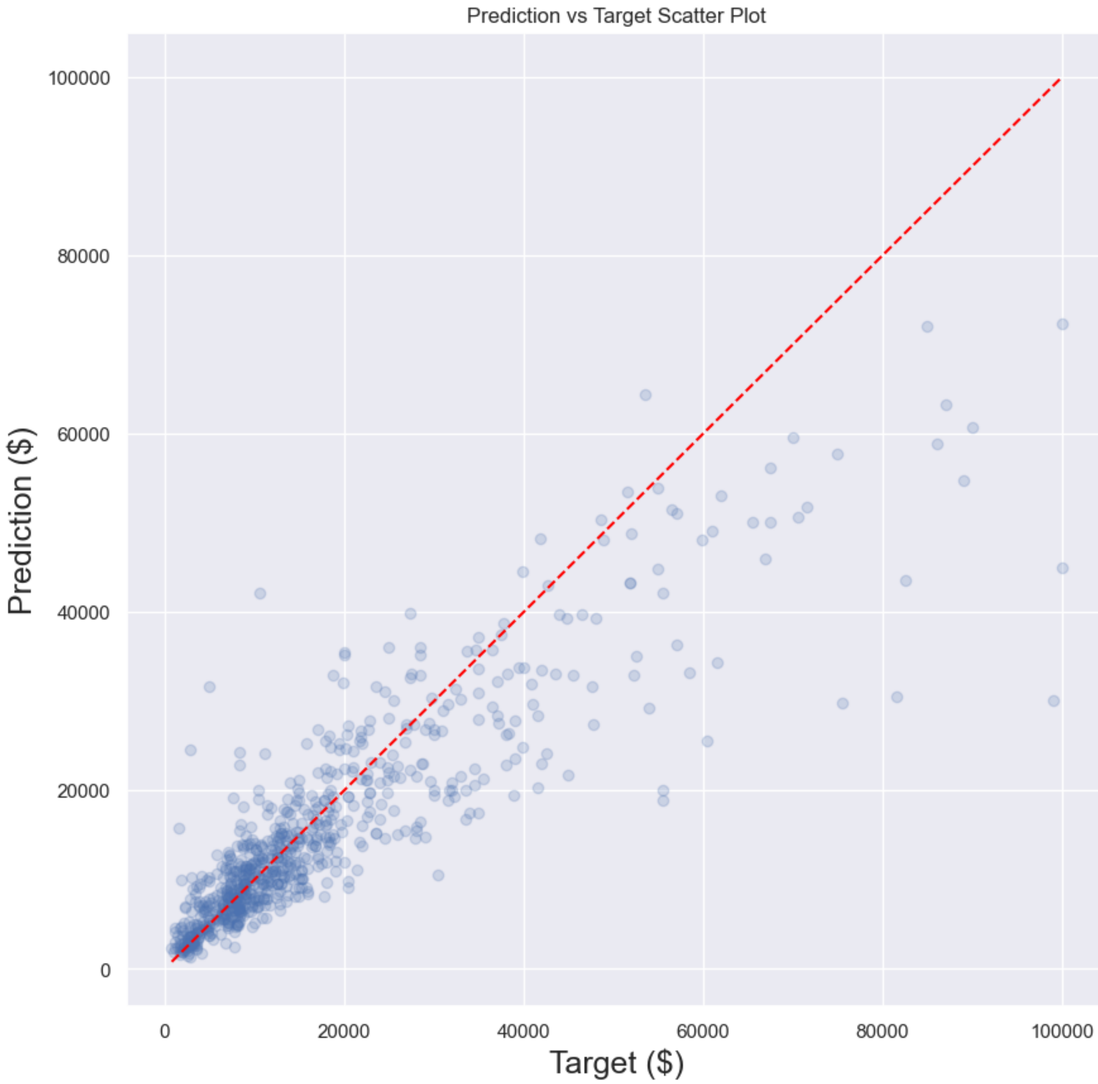
1.0.2 MAE

Mean Absolute Error is the abs avg error between the predcition and the actual value. Unlike MSE, it doesn't square the errors => less sensitive to outliers. MAE is always < RMSE becuase it never squares the errors at the first place. The difference between RMSE and MAE is that RMSE penalizes larger errors more than MAE by squaring the differneces. In MAE they're all considered equal (= no modification to the differences). So if the difference between RMSE and MAE are huge it tells us that some predictions were really off. Our MAE is $4534.62 which is half of RMSE, showing that we have outliers that are very off.

1.0.3 Prediction vs Target Value Scatter Plot

[97]:
```python
plt.figure(figsize=(10,10))
plt.scatter(data['Target'], data['Prediction'], alpha=0.2)

plt.plot([data['Target'].min(), data['Target'].max()],
         [data['Target'].min(), data['Target'].max()],
          color='red', linestyle='dashed')

plt.xlabel('Target ($)', size=18)
plt.ylabel('Prediction ($)', size=18)

plt.title('Prediction vs Target Scatter Plot')

print([data['Target'].min(), data['Target'].max()])
print([data['Target'].min(), data['Target'].max()])

plt.show()
```

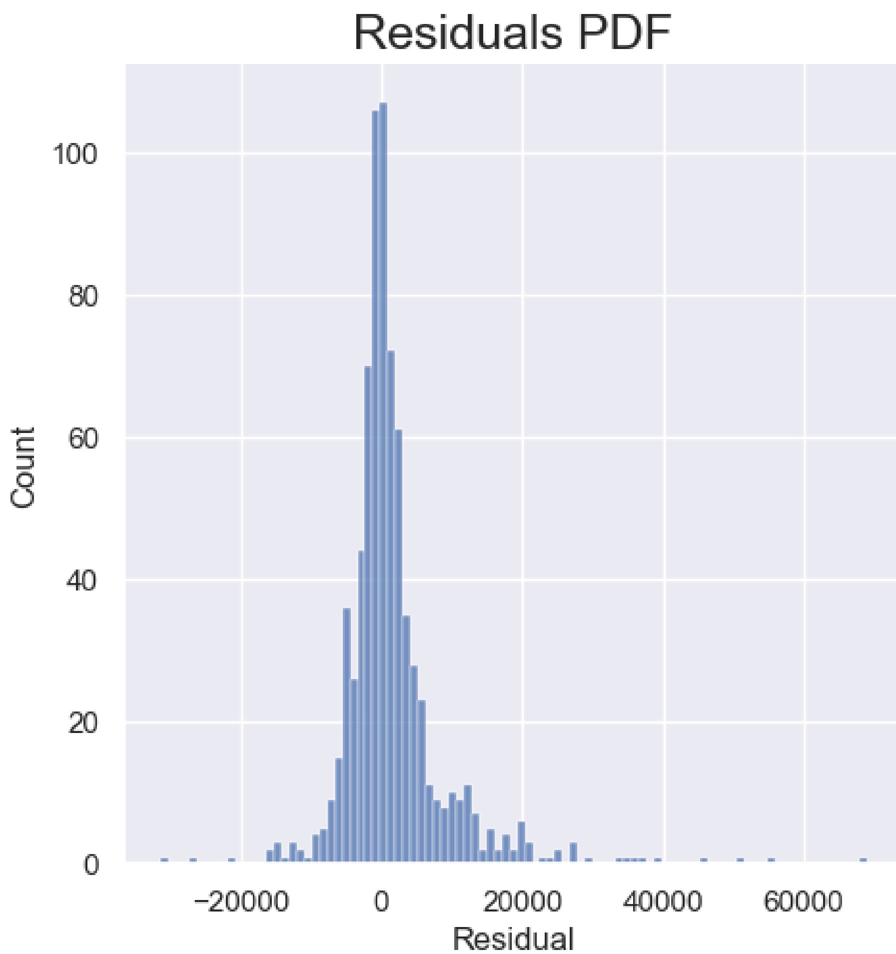
```
[799.9999999999999, 100000.0]
[799.9999999999999, 100000.0]
```



1.0.4 Residual Plot

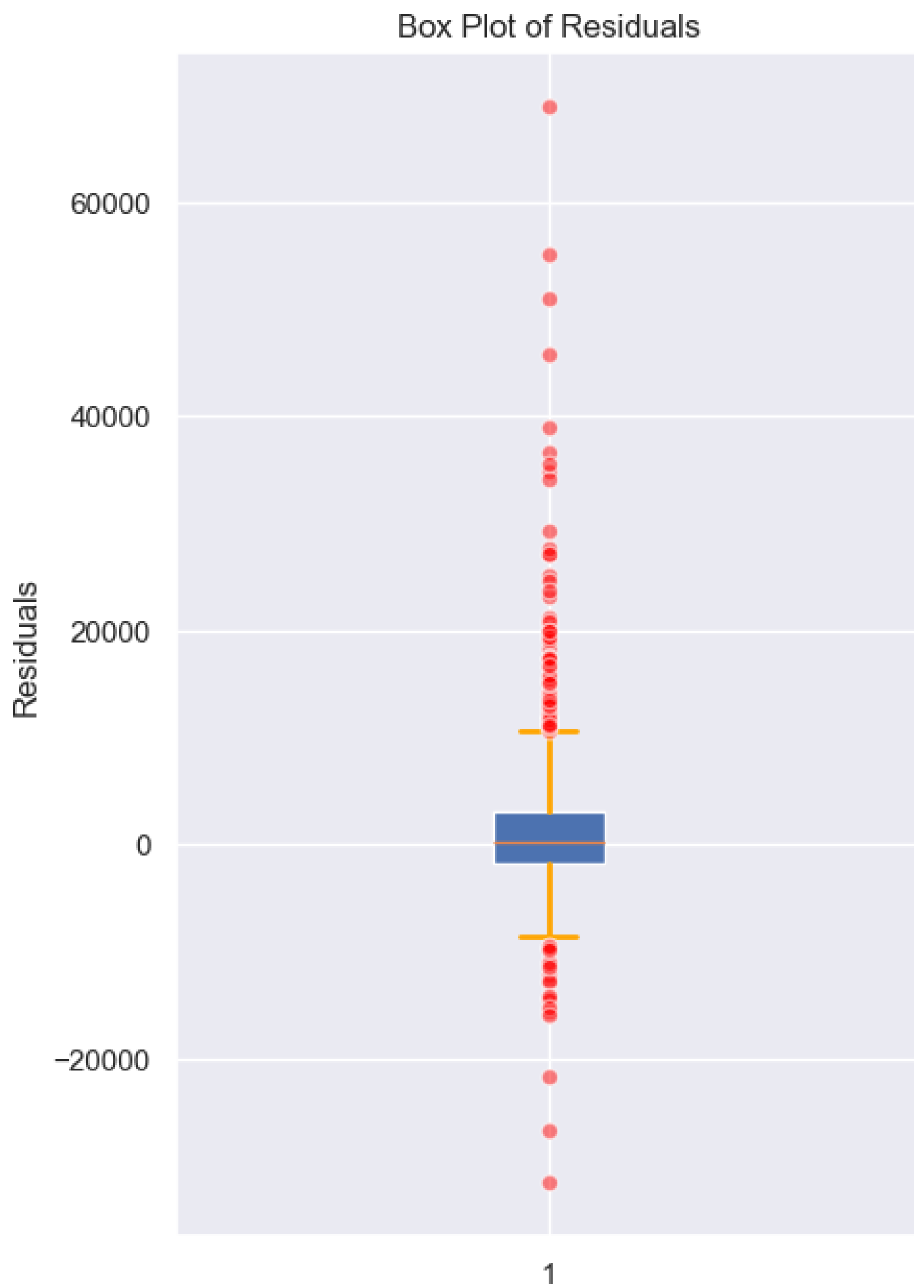
[98]:
```
sns.displot(data['Residual'])
plt.title("Residuals PDF", size=18)
```

[98]: Text(0.5, 1.0, 'Residuals PDF')



1.0.5 Residuals Box Plot

[100]:
```
plt.figure(figsize=(5,8))
plt.boxplot(data["Residual"], vert=True, patch_artist=True,␣
 ↪flierprops={'markerfacecolor': 'red', 'alpha': 0.5}, whiskerprops={'color':␣
 ↪'orange', 'linewidth': 2},   #  (Whiskers)
            capprops={'color': 'orange', 'linewidth': 2})

plt.ylabel("Residuals")
plt.title("Residuals Box Plot")

plt.show()
```


Box Plot of Residuals
60000
40000
20000
0
−20000
Residuals
1