1. The Target Value ($13.22 Increase)

I ran the GridSearchCV to find the best Random Forest model. When that model is applied to Big Mountain's specific features, it consistently predicts a "market-supported" price significantly higher than the actual $81.00. That $13.22 figure comes from calculating the difference between the model's prediction and reality.

2. The Impact of Improvements

I ran used the predict_increase function.

- I saw that adding **Vertical Drop** and **Snow Making Acreage** moved the needle the most.
- I saw that adding a **Chair Lift** had a positive but smaller impact on the *price*, but I then used math to see if that price bump covered the **$1.5M operating cost** you provided.

3. The Risk of Run Closures

I used list comprehensions to map out what happens to revenue when I close runs.

- The code showed a **negative slope**: as runs_closed increased, price_deltas and revenue_deltas went down.
- This is why the report warns that "saving money" by closing runs might actually lose more money in ticket value.

4. Why the Model Price is "Higher"

This is based on the **Feature Importance** I calculated. The model saw that Big Mountain is in the top tier for "Total Chairs," "Vertical Drop," and "Runs." Because the model "learned" from other resorts that having those high-end stats usually means a price closer to $100, it flagged Big Mountain as being underpriced for its weight class.

5. Deployment Recommendation

The suggestion to use a **Dashboard** comes from the complexity of the predict_increase function. Since we had to write code every time I wanted to change a parameter (like adding 0.2 miles to a run), it's logical to conclude that a non-coder would need a visual interface to do the same thing.