Problem Statement:

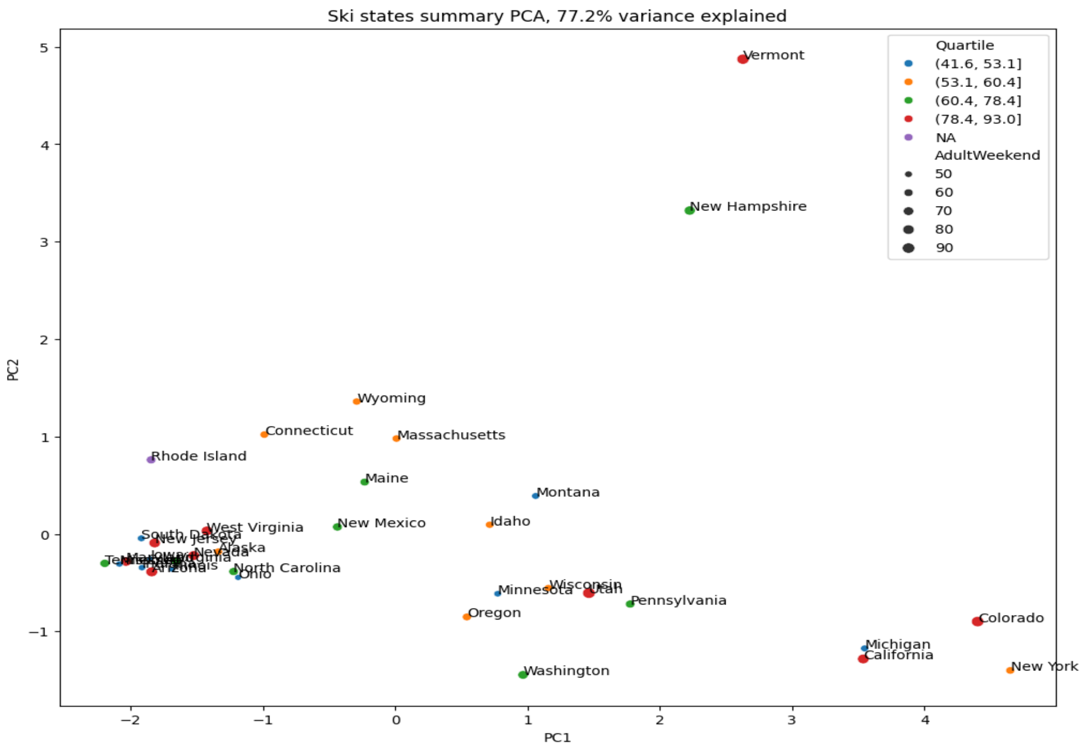
How can Big Mountain Resort (BM) support its increasing operational costs, $1,540,000 this season, by adjusting its operational facets such as snow cover, number of chairlifts, or ticket prices

Data Wrangling:

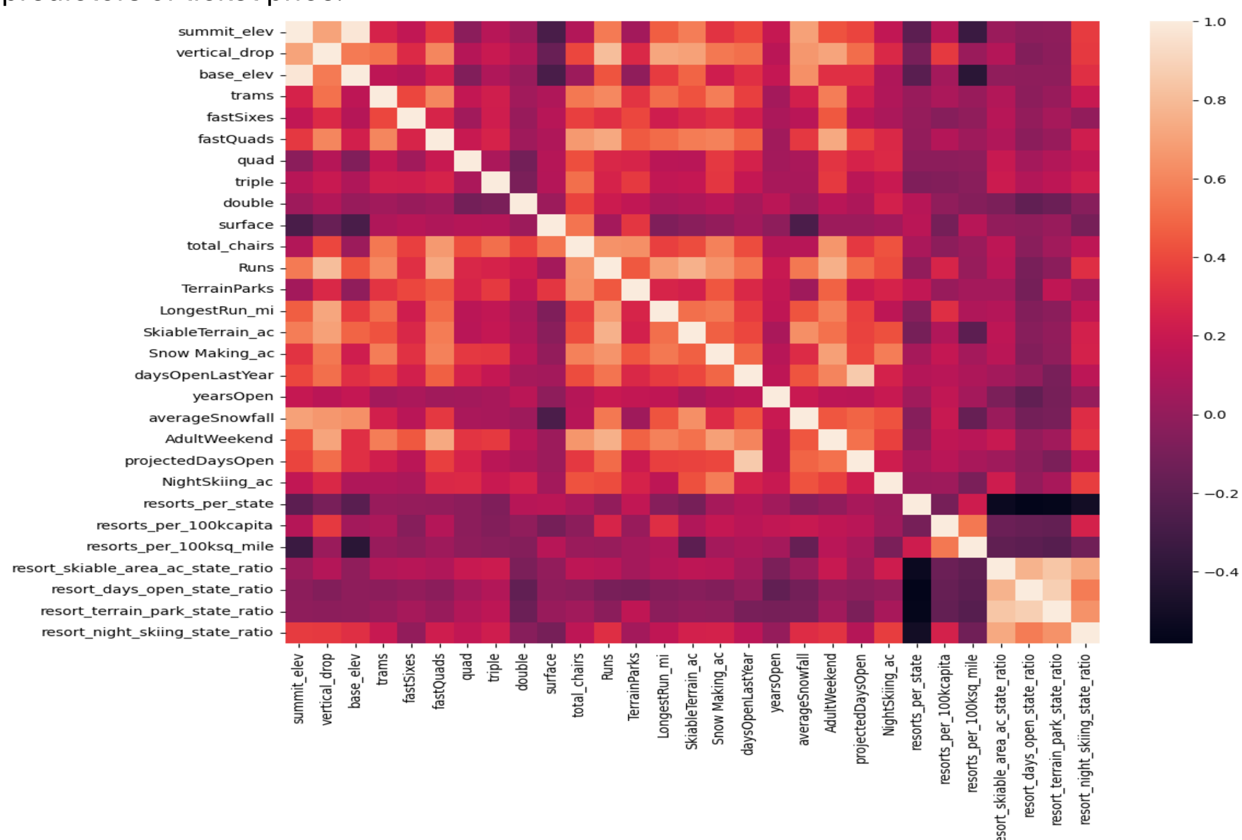
Using data extraction and manipulation methods, we downloaded the ski data, removed missing values, duplicate resort names, corrected for improbable values and removed columns that had mostly bad or missing data and rows with missing relevant data (ticket prices). Looking at distribution of data for each variable helped us in identifying potential bad data and outliers. Additional data sources, such as state population and area information were explored for better judgement.
We decided our target feature to be '***AdultWeekend'*** prices due to it having a greater number of relevant values, compared to 'AdultWeekday'. Other parameters such as variation were similar between the two. After doing this, we had a clean dataset, which we saved, that we could use in our next step: EDA.

Exploratory Data Analysis (EDA):

Here we further explored the ski resort data as well as the relationship between the ski data and USA states data. The first step is to explore how to extract meaningful information from the states data that's relevant to the ski data. To do this, we used a combination of histograms and reasoning to create density variables that are relevant to skiing but at the scope of the state. We explored feature engineering using PCA to account for the high dimensionality of our data, and found the first two features to explain most of the variance in the data. We also looked at the distribution of prices w.r.t. the two components for the various states and saw no obvious patterns in price w.r.t. state.



Therefore we treat all states equally in our analysis.
Finally we explored the relationships between the different attributes to see potential predictors of ticket price.



Pre-Processing and Model Development:

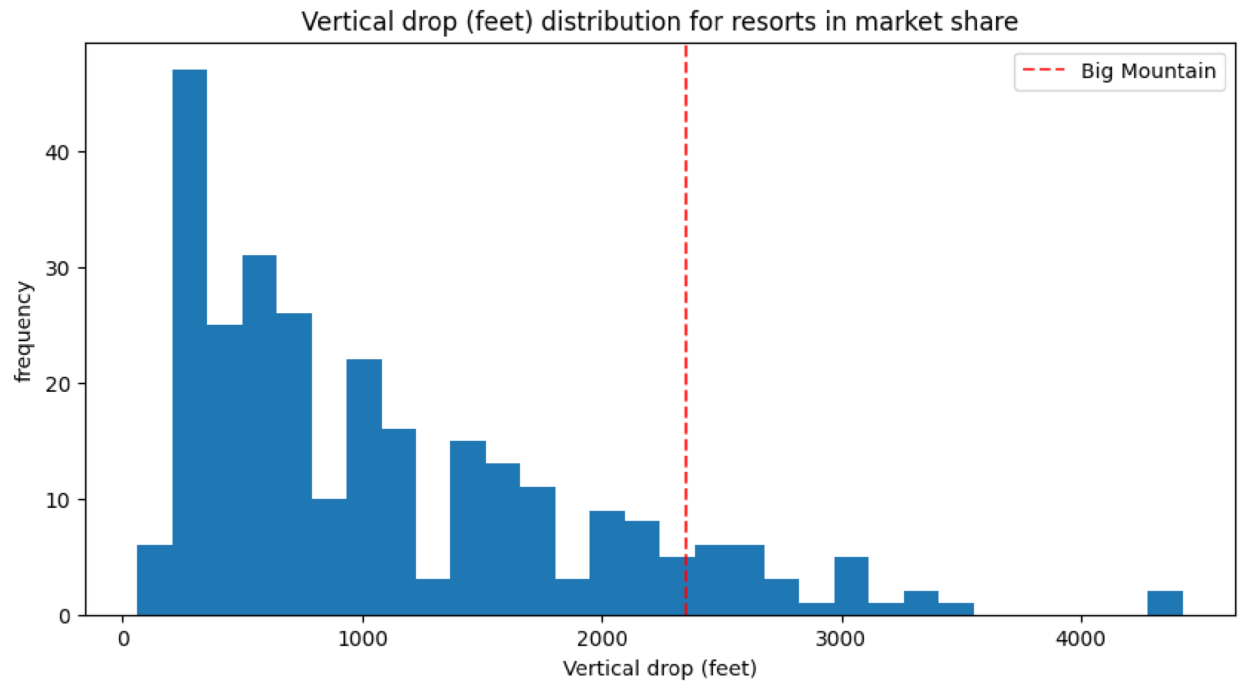
We loaded the data and excluded Big Mountain to train and evaluate on remaining data. Using various metrics (MAE, MSE and RMSE), We established a baseline by simply guessing the average price, which achieved an R-squared score of 0 on both the training and test sets. This demonstrated the importance of building models that outperform this basic approach. We then built a linear regression model, exploring different imputation strategies and feature selection methods. A SelectKBest with k=8 features yielded the best performance, achieving an R-squared score of approximately 0.7 on the test set.

Feature	Corr. to Price
vertical_drop	10.767857
Snow Making_ac	6.290074
total_chairs	5.794156
fastQuads	5.745626
Runs	5.370555
LongestRun_mi	0.181814
trams	-4.142024
SkiableTerrain_ac	-5.24978
LongestRun_mi	0.181814
trams	-4.142024
SkiableTerrain_ac	-5.24978

While this model outperformed the baseline, cross-validation results suggested a mean absolute error of around $10.50, with some variability. Finally, we experimented with a random forest regressor. The best performing model achieved a test R-squared around .7. Cross-validation results indicated a lower mean absolute error of approximately $9.50, with less variability compared to the linear regression model. Considering the slightly lower cross-validation MAE and reduced variability, we have chosen the random forest regressor as the final model for further business modeling. This model provides a robust and reliable prediction of ticket prices, potentially leading to more informed business decisions.

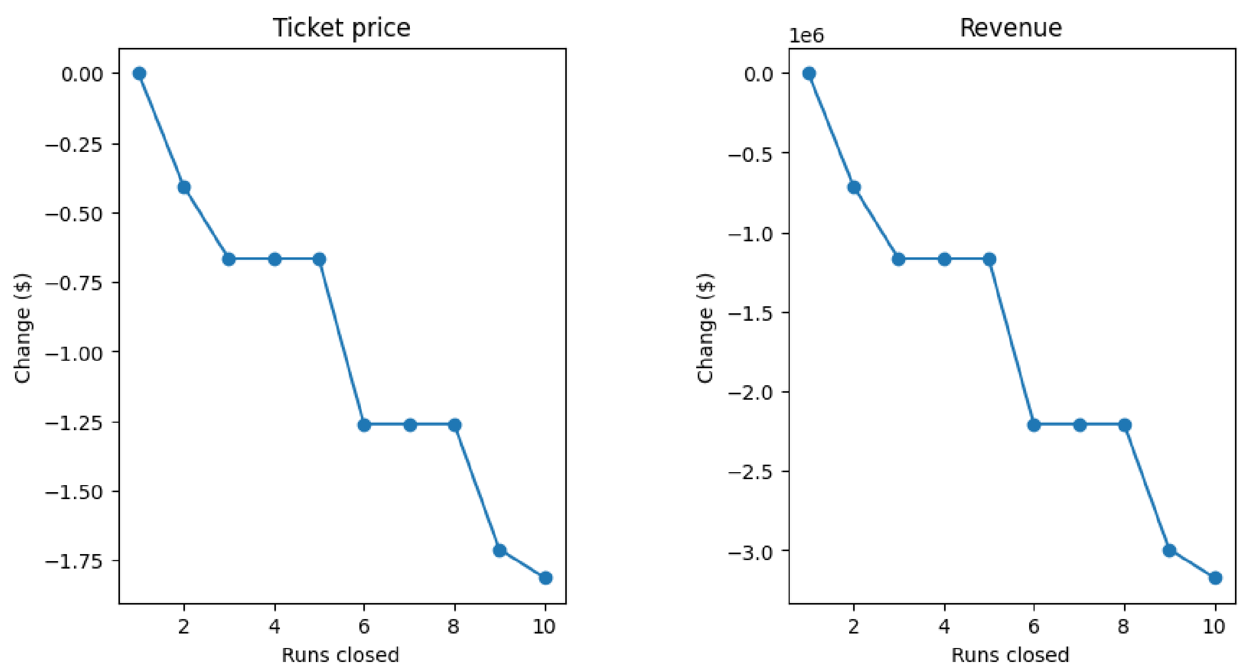
Modeling and Pricing:

Now we expand the training set to all the available data, trying to predict the price for BM. Our first prediction gives us an estimated price of $96 (but with a significant mae of 10.39), so there is room for some price increase. Next we explore where BM stands in regard to its facilities w.r.t. other providers. We do this by exploring histograms of the eight top facilities indicators we obtained previously using feature engineering and noting BM's position for each of them. For example,



The observations show that BM's facilities are competitive compared to other providers. Vertical drop is one area that has some providers offering being better than BM's. We then explored four avenues for cost reduction (assuming 350,000 visitors per season) by modeling the expected ticket price based on the parameters changed by each avenue:

- Closing down some of the most unused runs - Closing 10 runs can reduce ticket price by about 1.75, yielding a revenue drop of -3x10^6 USD



- Increase vertical run by 150 ft with the installation of an additional chair - Can support an increased ticket price by USD 1.99 and resulting revenue increase of USD 3474638
- In addition, add 2 acres of snow cover - No change

- Increase the longest run by .2 m but this requires an additional 4 ac. - No change
Pricing recommendations:
- Given that the cost of operating the additional chairlift is USD 1.5 million, we would still have to cover USD 1.15 M after increasing the ticket price. But this seems to be the best way to cover the additional costs
- We could temporarily partition the run into two parts, implementing a separate lift system for the additional 150 ft. We could entice users to agree for testing the additional drop using discounts or offers to other facilities. Could we go higher in price, upto USD 4 instead of just USD 2, based on our initial prediction of $96(-10 MAE)? When we test this additional price increase, does it discourage users?

Future Work:

One very useful information would be the number of visitors for each resort. This would help us make better estimates for how features impact the buyers' attitudes. For example, would lower ticket prices by eliminating some runs attract more customers?
Multi-collinearity could also be explored further - Do two or more parameters taken together have a larger impact on ticket price (for example, vertical drop AND skiiable area)?
We could use better models such as Neural networks to model such complex cases.
The model can be served using a custom function to test out different cases as required