Guided Capstone Project Report

By - Vidushi Raval

Big Mountain Resort Report

Big Mountain ski Resort has access to 105 trails and every year about 350,000 people ski or snowboard at Big Mountain. This mountain serviced 11 lifts, 2 T-bars, and 1 magic carpet for novice skiers with the longest run is 3.3 miles (name = Hellre) in length.

The base elevation is 4,464 ft and the summit is 6,817 ft with a vertical drop of 2,353 ft.
The resort wanted to take data driven decisions on better value for their ticket price.

The resort has invested in new infrastructure but is unsure how to align this with their ticket pricing structure. This initiative is critical for ensuring that the resort can cover its operating costs and improve profitability, while also delivering a competitive advantage in the market.

There are three stakeholders to provide key insights.

1. Alesha Eisen - Database Manager
2. Jimmy Blackburn - Director of Operations
3. Executive Leadership Team

The main data source for this analysis is the CSV file provided by the Database Manager, containing data from 330 resorts.

This data includes pricing, resort features (e.g., lifts, runs, vertical drop), and other key performance metrics. The analysis will focus on identifying which amenities are most influential in determining pricing, both at Big Mountain Resort and at comparable resorts.

Problem Statement:

Big Mountain Resort needs to optimize its ticket pricing strategy to **increase revenue by at least 10%** while maintaining visitor numbers and customer satisfaction **by the end of the ski season**. The current pricing model is based on market averages, which may not fully capture the value of the resort's unique amenities, including a newly installed chairlift that added **$1.54 million** in operational costs.

To achieve this, we will **analyze data from 330 comparable resorts** to identify the most influential factors affecting pricing and develop a **data-driven pricing model** that better reflects the resort's value. The success of this initiative will be measured by improved revenue projections, competitive pricing alignment, and actionable recommendations for cost optimization.

Data Wrangling:
The purpose of this data science project is to come up with a pricing model for ski resort tickets in our market segment.

Big Mountain suspects it may not be maximizing its returns, relative to its position in the market. It also does not have a strong sense of what facilities matter most to visitors, particularly which ones they're most likely to pay more for.
This project aims to build a predictive model for ticket price based on a number of facilities, or properties, boasted by resorts (*at the resorts).* This model will be used to provide guidance for Big Mountain's pricing and future facility investment plans.

This dataset focuses on ski resorts, including Big Mountain Resort in Montana for the Data Wrangling.

```
#Call the info method on ski_data to see a summary of the data
ski_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 330 entries, 0 to 329
Data columns (total 27 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Name               330 non-null    object 
 1   Region             330 non-null    object 
 2   state              330 non-null    object 
 3   summit_elev        330 non-null    int64  
 4   vertical_drop      330 non-null    int64  
 5   base_elev          330 non-null    int64  
 6   trams              330 non-null    int64  
 7   fastEight          164 non-null    float64
 8   fastSixes          330 non-null    int64  
 9   fastQuads          330 non-null    int64  
 10  quad               330 non-null    int64  
 11  triple             330 non-null    int64  
 12  double             330 non-null    int64  
 13  surface            330 non-null    int64  
 14  total_chairs       330 non-null    int64  
 15  Runs               326 non-null    float64
 16  TerrainParks       279 non-null    float64
 17  LongestRun_mi      325 non-null    float64
 18  SkiableTerrain_ac  327 non-null    float64
 19  Snow Making_ac     284 non-null    float64
 20  daysOpenLastYear   279 non-null    float64
 21  yearsOpen          329 non-null    float64
 22  averageSnowfall    316 non-null    float64
 23  AdultWeekday       276 non-null    float64
 24  AdultWeekend       279 non-null    float64
 25  projectedDaysOpen  283 non-null    float64
 26  NightSkiing_ac     187 non-null    float64
dtypes: float64(13), int64(11), object(3)
memory usage: 69.7+ KB
```

`AdultWeekday` is the price of an adult weekday ticket. `AdultWeekend` is the price of an adult weekend ticket. The other columns are potential features.

Figure: Original Dataset with 330 rows and 27 columns

Original Dataset: Contained 330 rows and 27 columns, covering details like ticket prices, skiable terrain, and lifts. The "fastEight" column was removed due to excessive missing values (only 164 non-null entries).

The "AdultWeekday" column was also dropped, as it was not present in the final dataset.

53 rows were removed, primarily due to missing ticket prices, resulting in a final dataset of 277 rows and 25 columns.

2.6.3.1 Unique Resort Names

In [35]:
```
#Code task 7#
#Use pandas' Series method `value_counts` to find any duplicated resort names
ski_data['Name'].value_counts().head()
```

Out[35]:
```
Name
Crystal Mountain    2
Alyeska Resort      1
Brandywine          1
Boston Mills        1
Alpine Valley       1
Name: count, dtype: int64
```

You have a duplicated resort name: Crystal Mountain.

Q: 1 Is this resort duplicated if you take into account Region and/or state as well?

In [37]:
```
#Code task 8#
#Concatenate the string columns 'Name' and 'Region' and count the values again (as above)
(ski_data['Name'] + ', ' + ski_data['Region']).value_counts().head()
```

Out[37]:
```
Alyeska Resort, Alaska    1
Snow Trails, Ohio         1
Brandywine, Ohio          1
Boston Mills, Ohio        1
Alpine Valley, Ohio       1
Name: count, dtype: int64
```

Figure: Find out duplicate resort names

Duplicate Resort Names: "Crystal Mountain" appeared twice, but analysis confirmed they were distinct resorts in different states.

```
Name: count, dtype: int64
```

In [41]:
```
#**NB** because you know `value_counts()` sorts descending, you can use the `head()` method and know the rest of the counts
```

A: 1 No

In [43]:
```
ski_data[ski_data['Name'] == 'Crystal Mountain']
```

Out[43]:

	Name	Region	state	summit_elev	vertical_drop	base_elev	trams	fastEight	fastSixes	fastQuads	...	LongestRun_m
104	Crystal Mountain	Michigan	Michigan	1132	375	757	0	0.0	0	1	...	0.
295	Crystal Mountain	Washington	Washington	7012	3100	4400	1	NaN	2	2	...	2.

2 rows × 27 columns

So there are two Crystal Mountain resorts, but they are clearly two different resorts in two different states. This is a powerful signal that you have unique records on each row.

2.6.3.2 Region And State

Figure: Crystal Mountain Resort in two different State and Region

Columns like "TerrainParks," "LongestRun_mi," and "Snow Making_ac" had missing values, which were addressed through imputation or row removal.

Target Feature: The dataset aims to predict ticket prices, with "AdultWeekend" as the key variable.

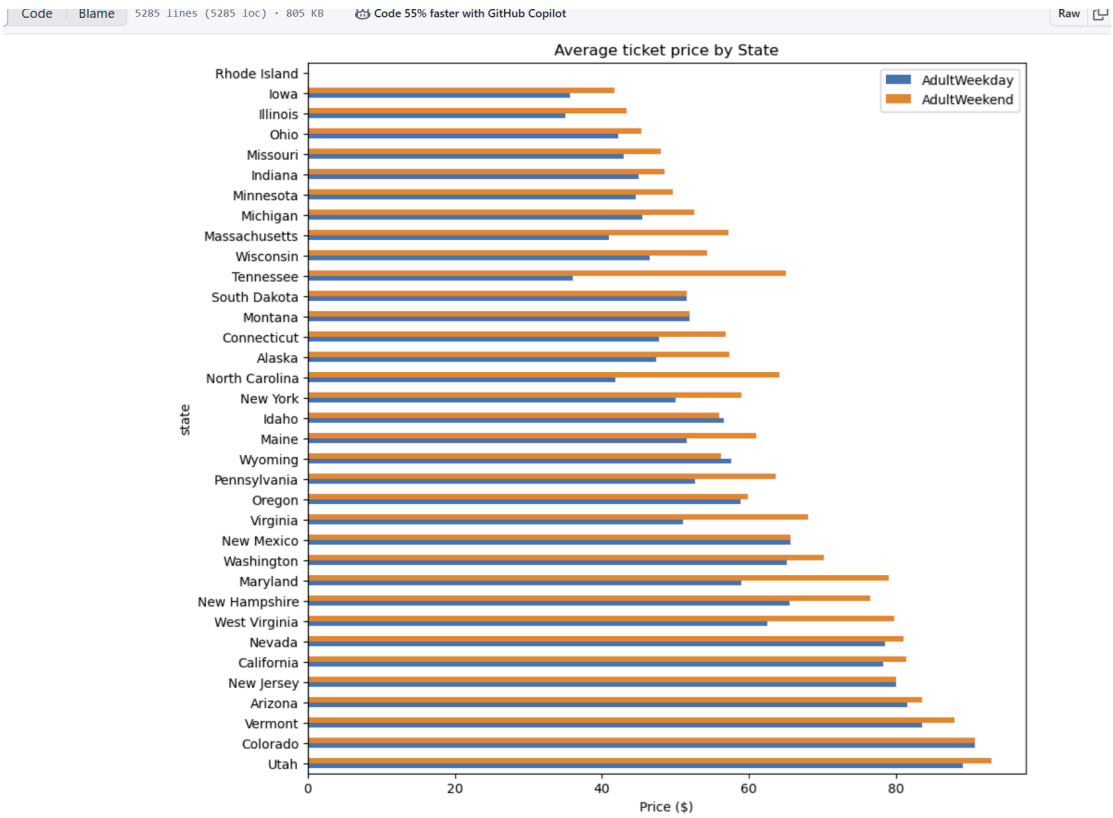

Figure: Average Ticket price by State

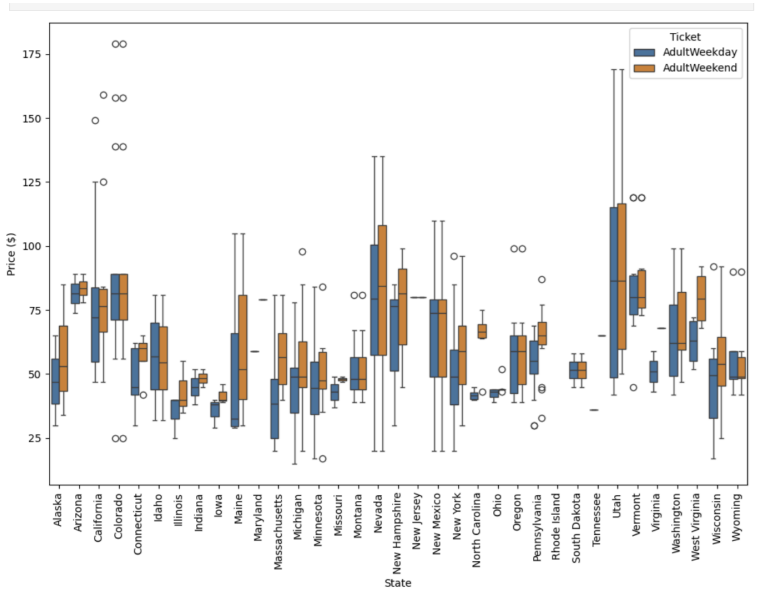

Figure: Expensive ticket prices in California, Colorado, and Utah

California, Utah, and Nevada showed the largest differences between weekend and weekday ticket prices. Ticket Price Gaps: 17% of resorts lacked ticket price data.

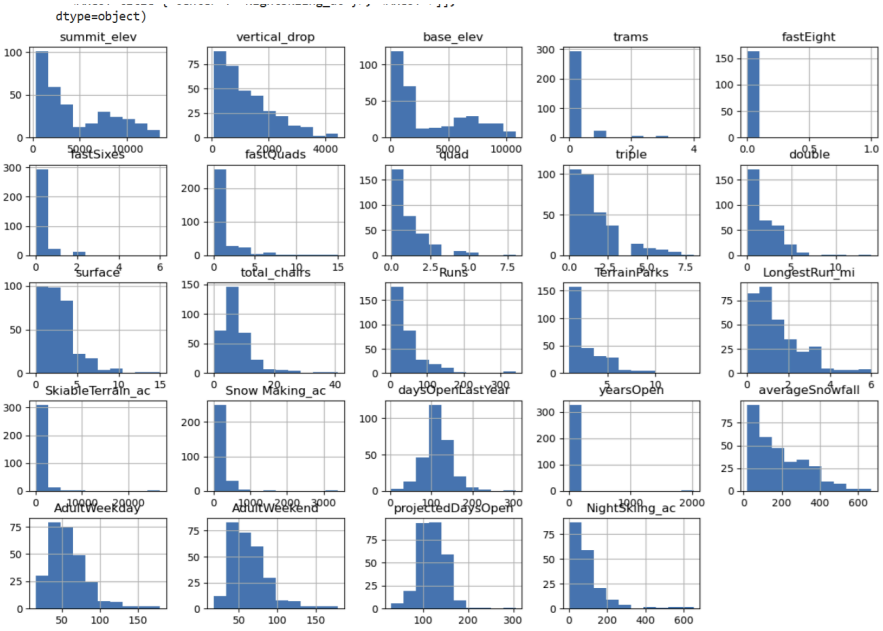

Figure: Distributions Of Feature Values

Features that have possible cause for concern about are:

- SkiableTerrain_ac because values are clustered down the low end,
- Snow Making_ac for the same reason,
- fastEight because all but one value is 0 so it has very little variance, and half the values are missing,
- fastSixes raises an amber flag; it has more variability, but still mostly 0,
- trams also may get an amber flag for the same reason,
- yearsOpen because most values are low but it has a maximum of 2019, which strongly suggests someone recorded the calendar year rather than number of years.

Silverton Mountain Resort in region Colorado, state Colorado has an incredibly large skiable terrain area.

Out[75]:

	39
Name	Silverton Mountain
Region	Colorado
state	Colorado
summit_elev	13487
vertical_drop	3087
base_elev	10400
trams	0
fastEight	0.0
fastSixes	0
fastQuads	0
quad	0
triple	0
double	1
surface	0
total_chairs	1
Runs	NaN
TerrainParks	NaN
LongestRun_mi	1.5
SkiableTerrain_ac	26819.0
Snow Making_ac	NaN
daysOpenLastYear	175.0
yearsOpen	17.0
averageSnowfall	400.0
AdultWeekday	79.0
AdultWeekend	79.0
projectedDaysOpen	181.0
NightSkiing_ac	NaN

Figure: Silver Mountain Resort (Colorado)

Silverton Mountain Resort (Colorado) had an exceptionally large skiable area (verified at 1819 acres). Other outliers were found in Skiable Terrain, Snow Making, fastEight, and fastSixes columns.

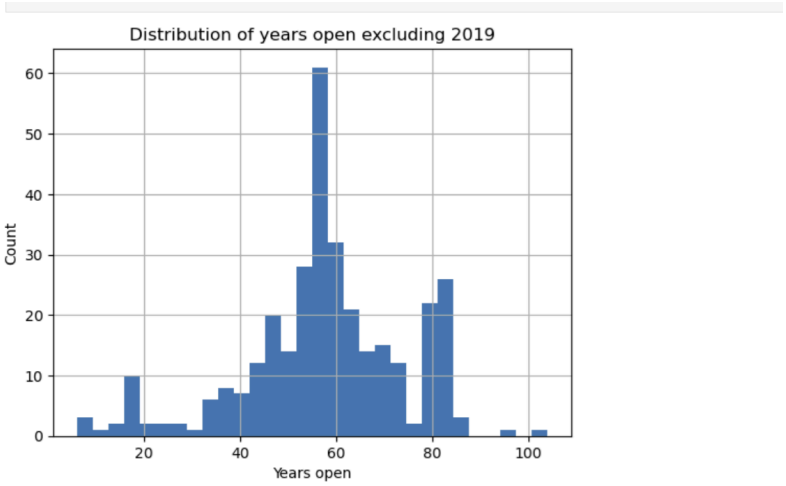

Figure: Distribution of years excluding 2019

The above distribution of years seems entirely plausible, including the 104 year value. You can certainly state that no resort will have been open for 2019 years! It likely means the resort opened in 2019. It could also mean the resort is due to open in 2019. You don't know when these data were gathered!

Some states, including California, Nevada, Oregon, and Utah, had inconsistencies between state and region classifications. Big Mountain Resort (Montana): The resort is located in a state ranking 11th in the number of ski resorts.

Visualization & Transformation: Used Matplotlib to visualize price variations across states.

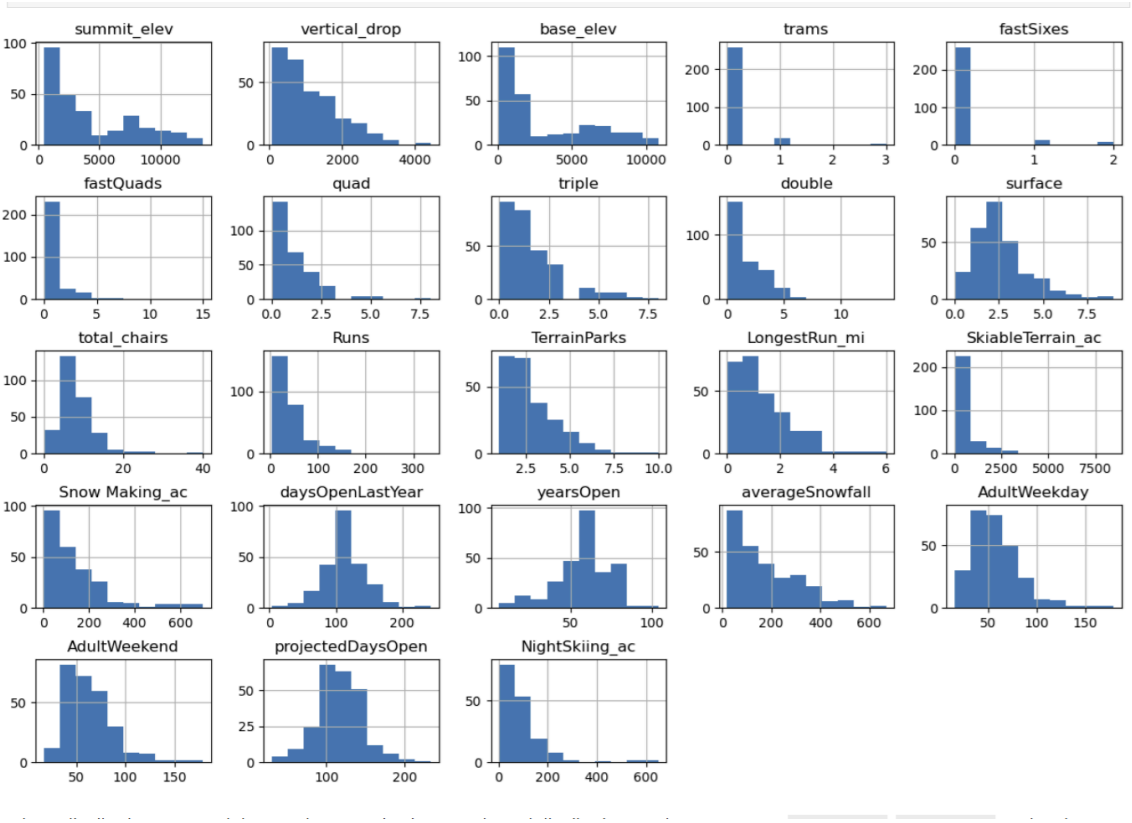

Figure: Distributions

Transformed ticket price data into a single column, introducing a new categorical column for ticket type using the melt function.

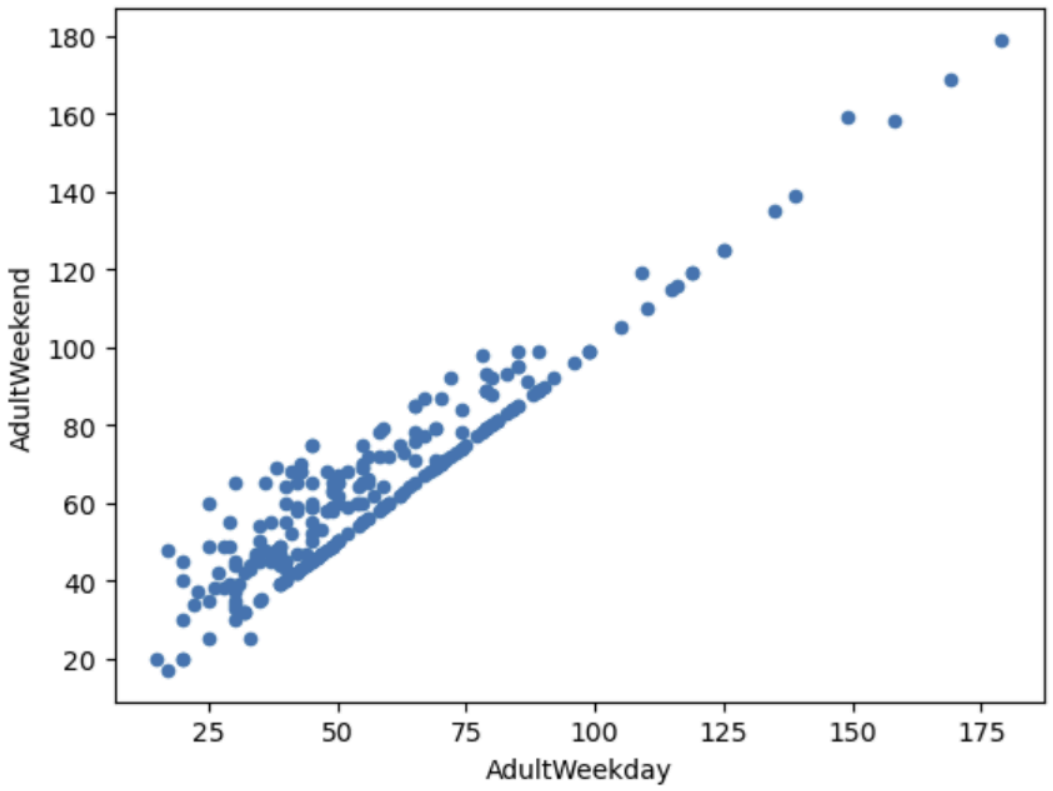

Figure: a scatterplot (kind='scatter') with 'AdultWeekday' on the x-axis and #'AdultWeekend' on the y-axis

A scatterplot of weekend vs. weekday ticket prices showed that tickets under $100 had the most significant price variability.

This summary provides a structured overview of the dataset, its cleaning process, and key insights derived from the analysis.

Exploratory data Analysis

Here, I have performed EDA on two datasets: ski_data(individual resort data) and state_summary(statewide data).We have a total 25 columns in ‘ski_data’ and 8 columns in ‘state_wide’ summary data.

3.4.1 Ski data

In [139…
```
ski_data = pd.read_csv('../data/ski_data_cleaned.csv')
```

In [141…
```
ski_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 277 entries, 0 to 276
Data columns (total 25 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Name               277 non-null    object 
 1   Region             277 non-null    object 
 2   state              277 non-null    object 
 3   summit_elev        277 non-null    int64  
 4   vertical_drop      277 non-null    int64  
 5   base_elev          277 non-null    int64  
 6   trams              277 non-null    int64  
 7   fastSixes          277 non-null    int64  
 8   fastQuads          277 non-null    int64  
 9   quad               277 non-null    int64  
 10  triple             277 non-null    int64  
 11  double             277 non-null    int64  
 12  surface            277 non-null    int64  
 13  total_chairs       277 non-null    int64  
 14  Runs               274 non-null    float64
 15  TerrainParks       233 non-null    float64
 16  LongestRun_mi      272 non-null    float64
 17  SkiableTerrain_ac  275 non-null    float64
 18  Snow Making_ac     240 non-null    float64
 19  daysOpenLastYear   233 non-null    float64
 20  yearsOpen          277 non-null    float64
 21  averageSnowfall    268 non-null    float64
 22  AdultWeekend       277 non-null    float64
 23  projectedDaysOpen  236 non-null    float64
 24  NightSkiing_ac     163 non-null    float64
dtypes: float64(11), int64(11), object(3)
memory usage: 54.2+ KB
```

Figure: Ski Data with 25 columns

The exploratory data analysis examined numerical and categorical features related to ski resorts, including ticket prices, vertical drop, skiable acreage, snowfall, and state-wide attributes like population and total ski days.

3.5.1.4 Total skiable area

In [159…
```
state_summary_newind.state_total_skiable_area_ac.sort_values(ascending=False).head()
```

Out[159…
```
state
Colorado      43682.0
Utah          30508.0
California    25948.0
Montana       21410.0
Idaho         16396.0
Name: state_total_skiable_area_ac, dtype: float64
```

Figure: Colorado has most resorts

New York state may have the most resorts, but they don't account for the most skiing area.

3.5.1.5 Total night skiing area

```
state_summary_newind.state_total_nightskiing_ac.sort_values(ascending=False).head()
```

```
state
New York        2836.0
Washington      1997.0
Michigan        1946.0
Pennsylvania    1528.0
Oregon          1127.0
Name: state_total_nightskiing_ac, dtype: float64
```

Figure: New York dominates the area of skiing available at night.

Resort specific numerical features were Ticket price, vertical drop, skiable acreage, snowfall, snow-making acres, number of runs/lifts, summit & base elevation, days open, night skiing acres. State wide numeric features are Population, area, resorts per state, total skiable area, total night skiing acres, total days open.

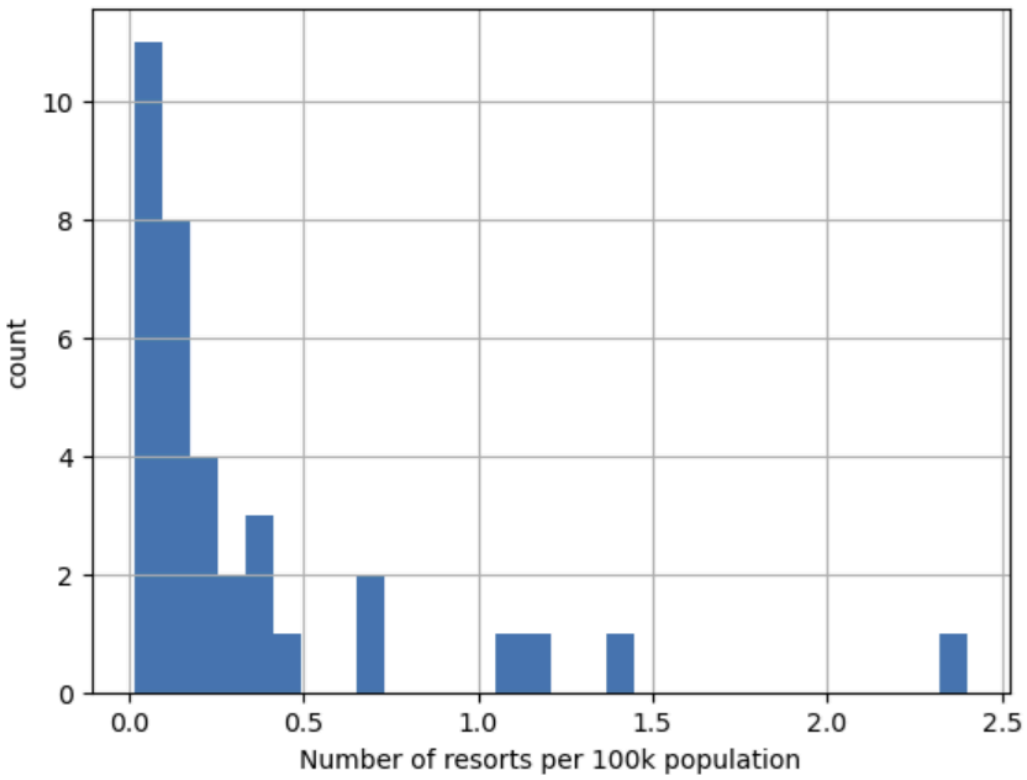

Figure: Number of resorts per 100k population

State is a categorical feature in State location and presence of quad lifts, high-speed lifts are resort specific categorial features.

Analyzing the number of resorts per 100k population and per 100k square miles can help eliminate the influence of larger states and provide more comparable data.

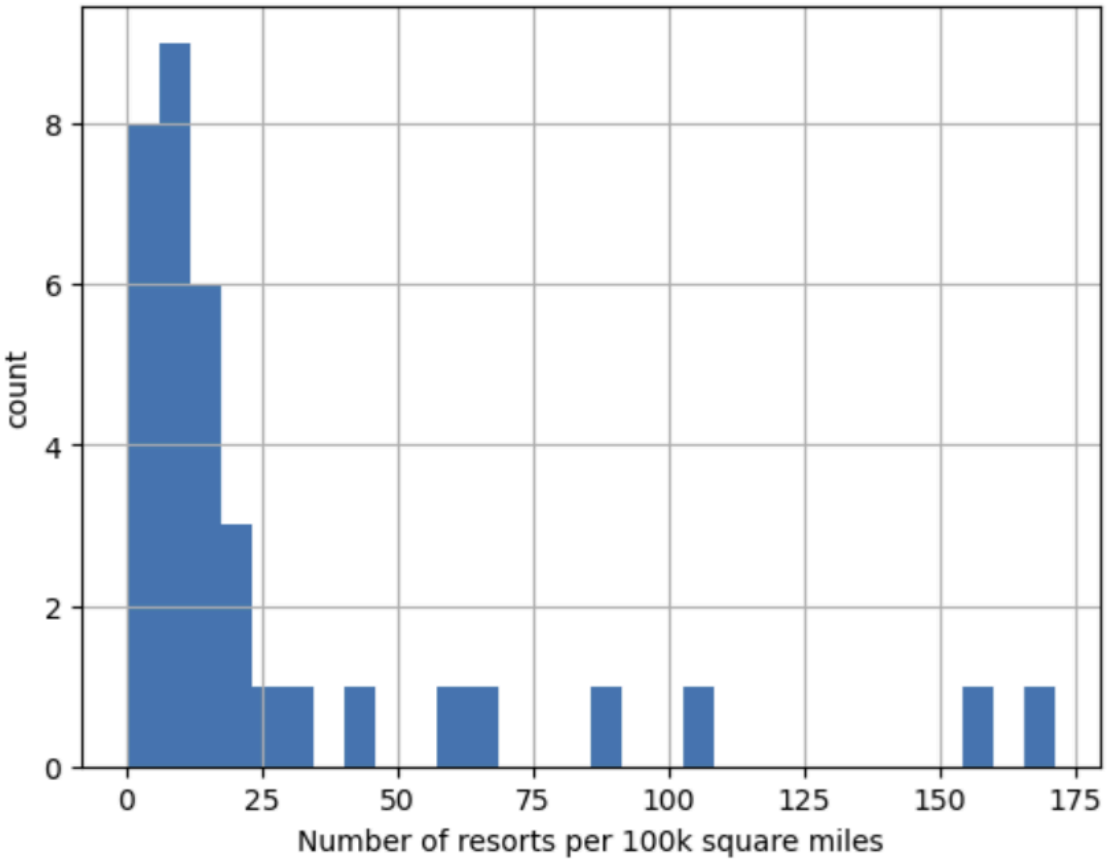

Figure: Number of resort per 100k square miles

We have calculated adult weekend ticket prices by state. The analysis suggested a relationship between state and ticket price.

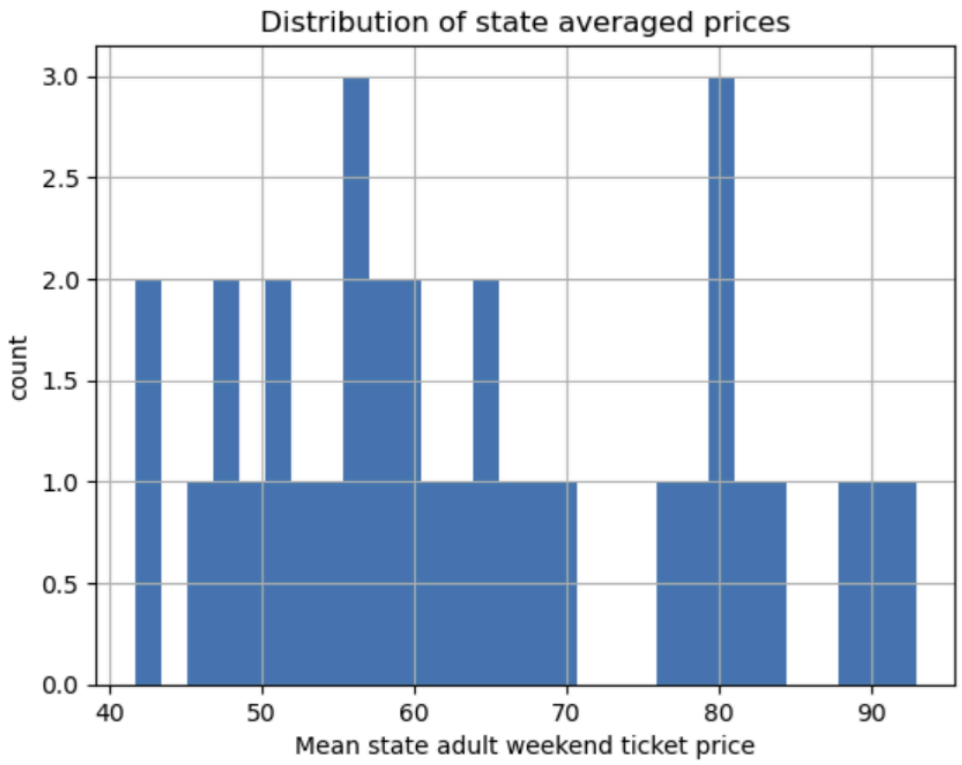

Figure: Mean State Adult Weekend Ticket Price

AdultWeekend ticket price has been chosen as the target to avoid high cardinality.

In the northern region, night skiing was more common and had higher average ticket prices. States with more resorts had longer seasons and a wider range of ticket prices. There is a variation in ticket prices by state.

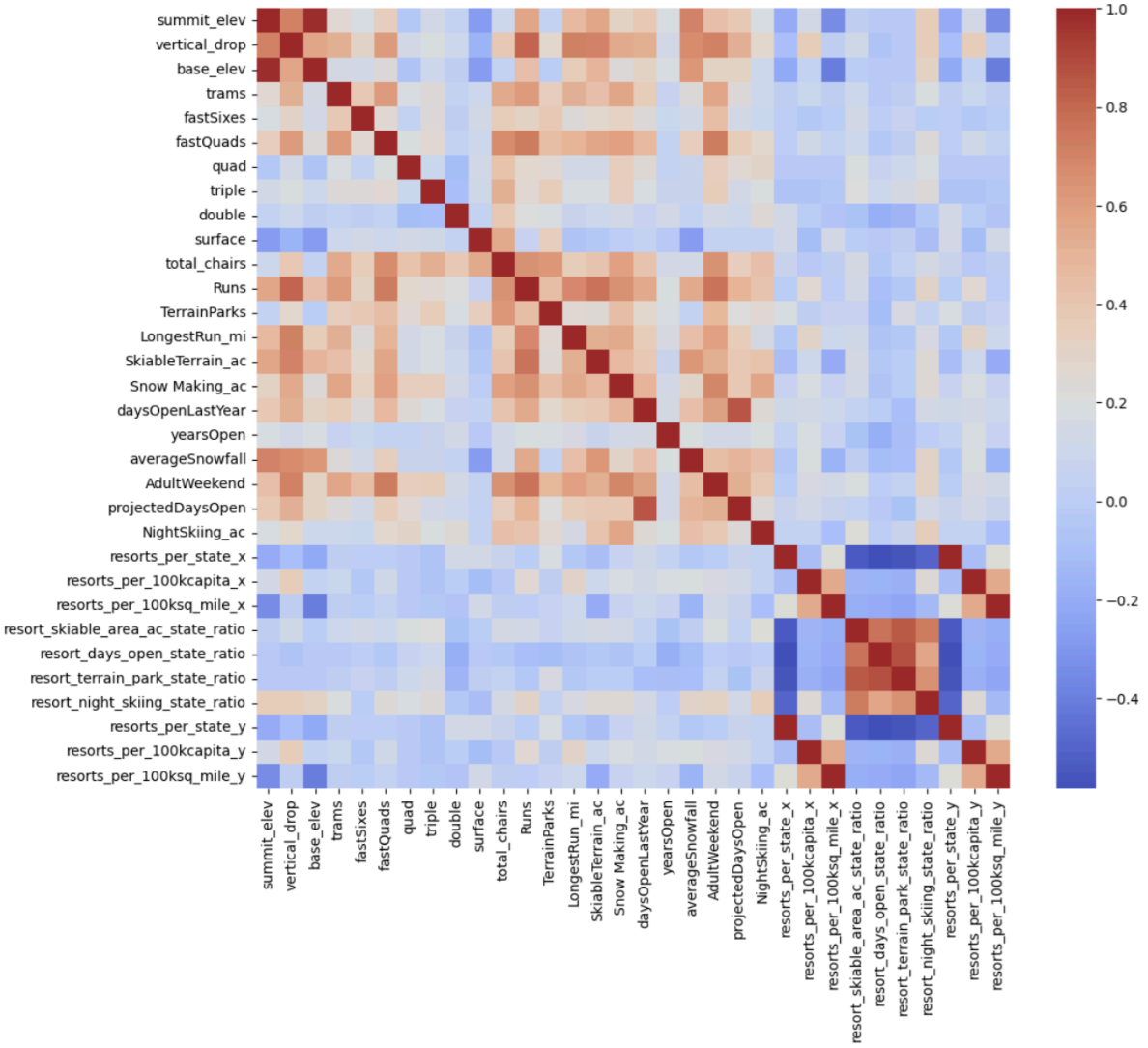

Figure: a seaborn heatmap of correlations in ski_data

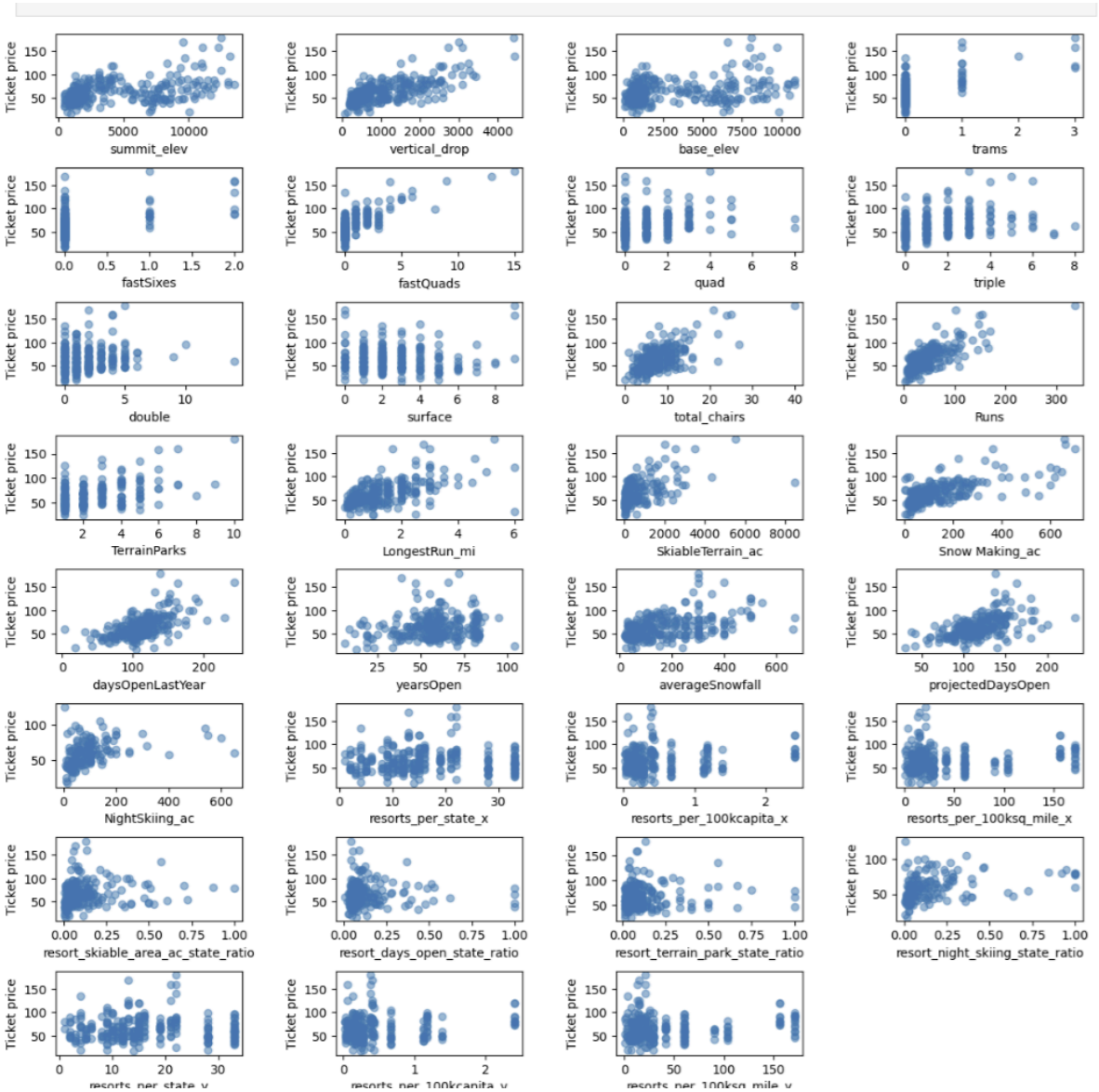

Figure: ***scatterplots of ticket prices against desired columns***

One data point from Rhode Island was excluded early on due to NaN values. A correlation heat map was generated to identify relationships between the variables, revealing strong correlations between summit and base elevations, as well as between night skiing and the number of resorts per capita.

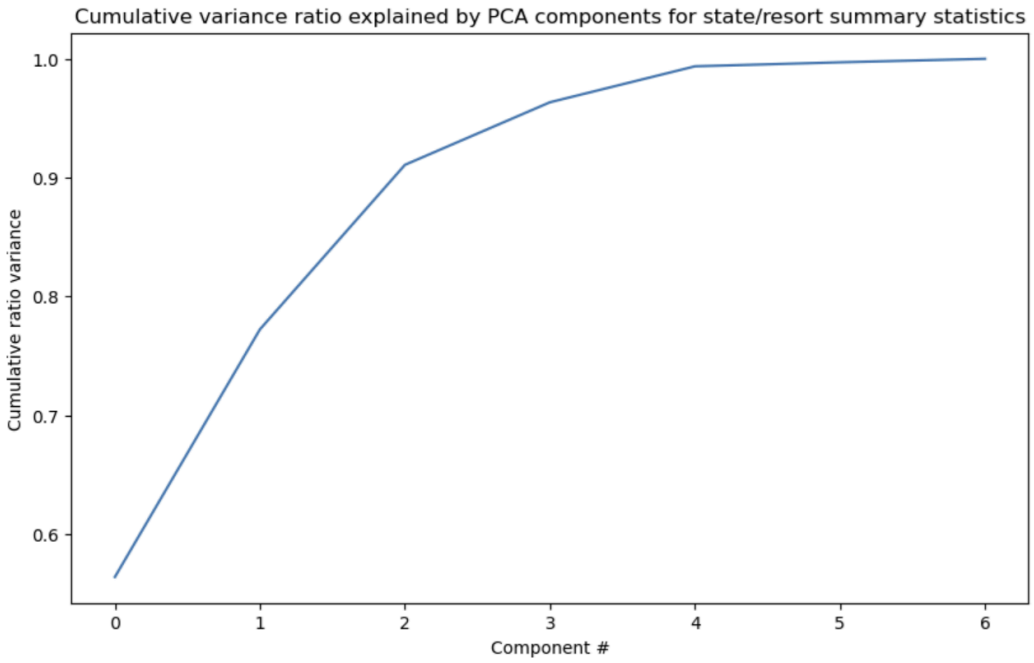

Figure: ***line plot to visualize the cumulative explained variance ratio with number of components, xlabel to 'Component #', the ylabel to 'Cumulative ratio variance', and the title to 'Cumulative variance ratio explained by PCA components for state/resort summary statistics'***

Correlation analysis shows that vertical drop, number of runs, how many days resorts are open makes a bigger impact on ticket prices than state wide attributes.

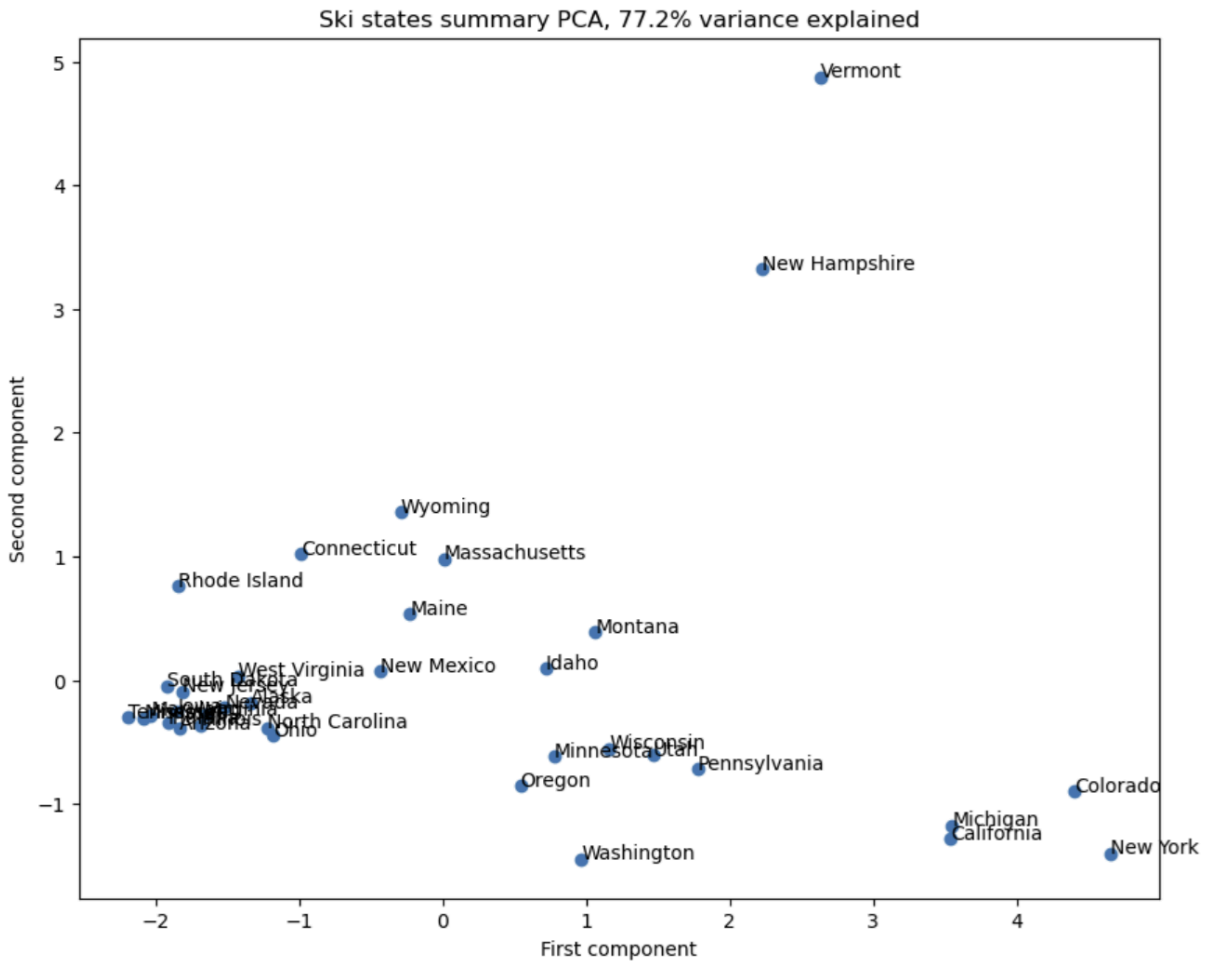

Figure: Ski states summary PCA

Histogram gives insight into how pricing differs across states and resorts. Scatter Plots shows plots to explore relationships between ticket prices and other variables.

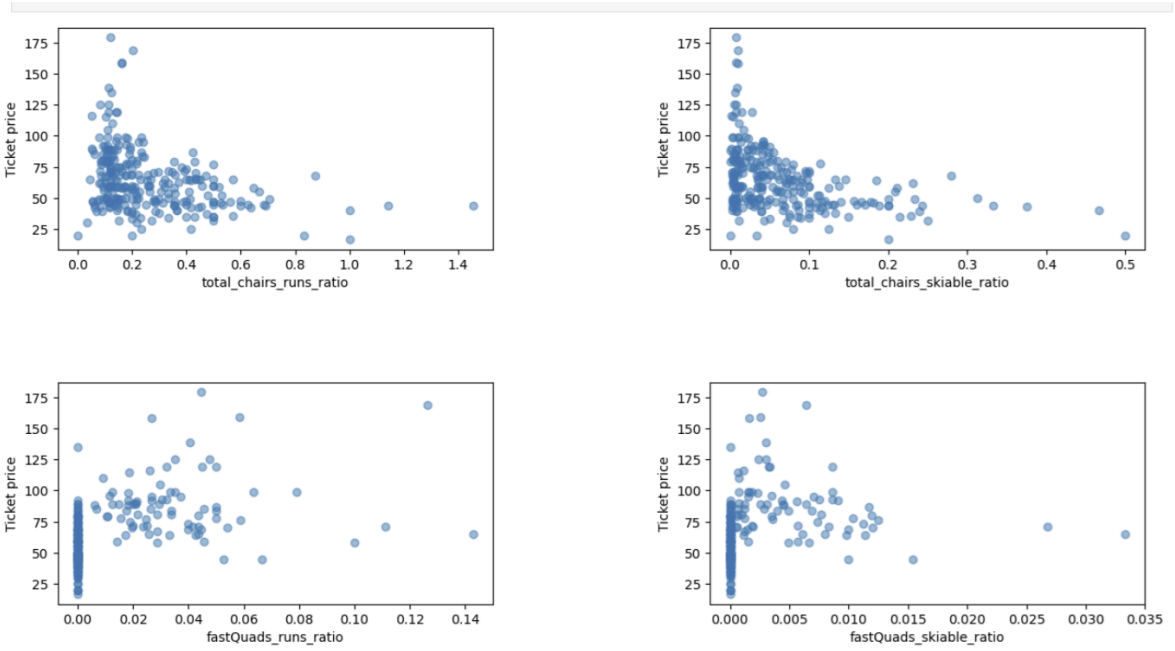

Figure: Total chairs to run ratio

TO figure out how easily a resort can transport people around. We have the numbers of various chairs, and the number of runs. So the ratio of chairs to runs would inform you how easily, and so quickly, people could get to their next ski slope!

It's interesting that vertical drop and projected days open have a high correlation with ticket prices.

Additionally, infrastructure factors like quad lifts and snow making also seem to impact pricing. The correlation with snowmaking and runs indicates that resorts with better snow conditions and more varied terrain may charge more for access. It also appears that having no fast quads may limit the ticket price, but if your resort covers a wide area then getting a small number of fast quads may be beneficial to ticket price.

Model Preprocessing with feature engineering

Here I have performed preliminary assessments of data quality. Big Mountain is the resort so I have separated it from the rest of the data to use later. To establish a baseline for performance, the initial approach was simply taking the average price.

A linear regression model was developed next. Feature selection identified key predictors that contributed significantly to housing prices. I see that a simple linear regression model explains over 80% of the variance on the train set and over 70% on the test set. I have calculated mean and median values for imputing missing values. Then scale the data and assess the model performance.

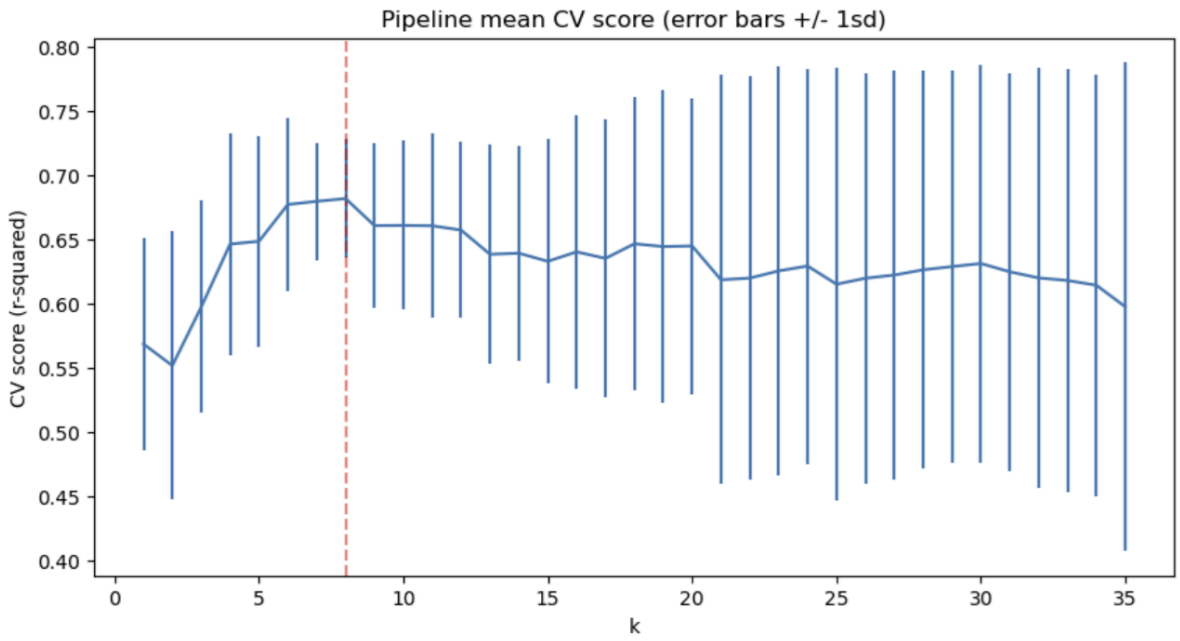

Figure: Pipeline mean CV score

Then I defined the pipeline, called fit method and made a prediction. I have suspected that the model was overfitting so I have refined the Linear Model.

Then I have defined a new pipeline to select a different number of features. After that I have also accessed the performance using cross-validation. Based on the pipeline cv score, I found out that a good value for k is 8.

Based on the result, vertical drop was the biggest positive feature.

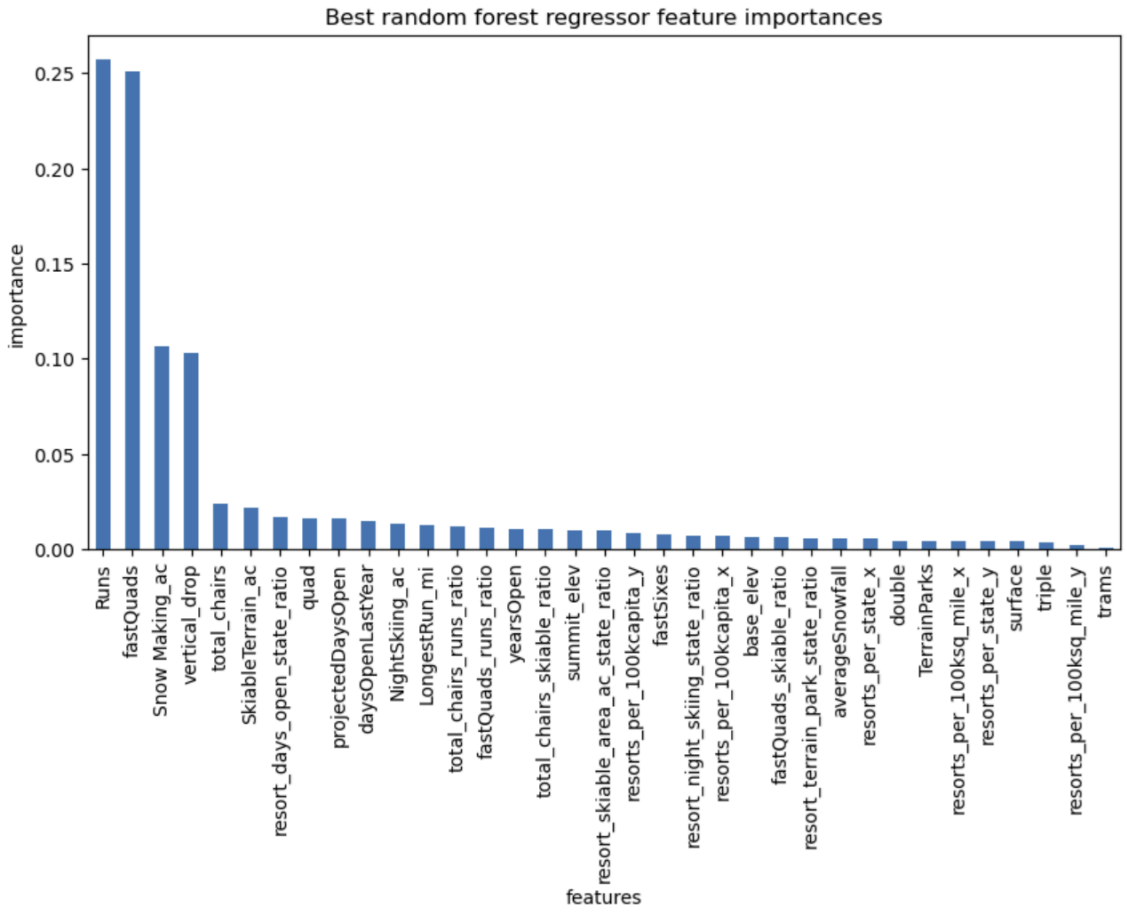

***Figure:** bar plot of the random forest's feature importances*

The random forest model has a lower cross-validation mean absolute error by almost $1. It also exhibits less variability. Verifying performance on the test set produces performance consistent with the cross-validation results.

4.11.1 Linear regression model performance

In [283…
```
# 'neg_mean_absolute_error' uses the (negative of) the mean absolute error
lr_neg_mae = cross_validate(lr_grid_cv.best_estimator_, X_train, y_train,
                            scoring='neg_mean_absolute_error', cv=5, n_jobs=-1)
```

In [285…
```
lr_mae_mean = np.mean(-1 * lr_neg_mae['test_score'])
lr_mae_std = np.std(-1 * lr_neg_mae['test_score'])
lr_mae_mean, lr_mae_std
```

Out[285…
```
(10.499032338015294, 1.6220608976799664)
```

In [287…
```
mean_absolute_error(y_test, lr_grid_cv.best_estimator_.predict(X_test))
```

Out[287…
```
11.793465668669326
```

4.11.2 Random forest regression model performance

In [289…
```
rf_neg_mae = cross_validate(rf_grid_cv.best_estimator_, X_train, y_train,
                            scoring='neg_mean_absolute_error', cv=5, n_jobs=-1)
```

In [291…
```
rf_mae_mean = np.mean(-1 * rf_neg_mae['test_score'])
rf_mae_std = np.std(-1 * rf_neg_mae['test_score'])
rf_mae_mean, rf_mae_std
```

Out[291…
```
(9.721783475783477, 1.362257714837129)
```

In [293…
```
mean_absolute_error(y_test, rf_grid_cv.best_estimator_.predict(X_test))
```

Out[293…
```
9.418440428380189
```

Figure: Linear regression model performance and Random Forest regression model performance

Given its superior performance and consistency between cross-validation and test results, the Random Forest Regressor was chosen for the next stage of analysis.

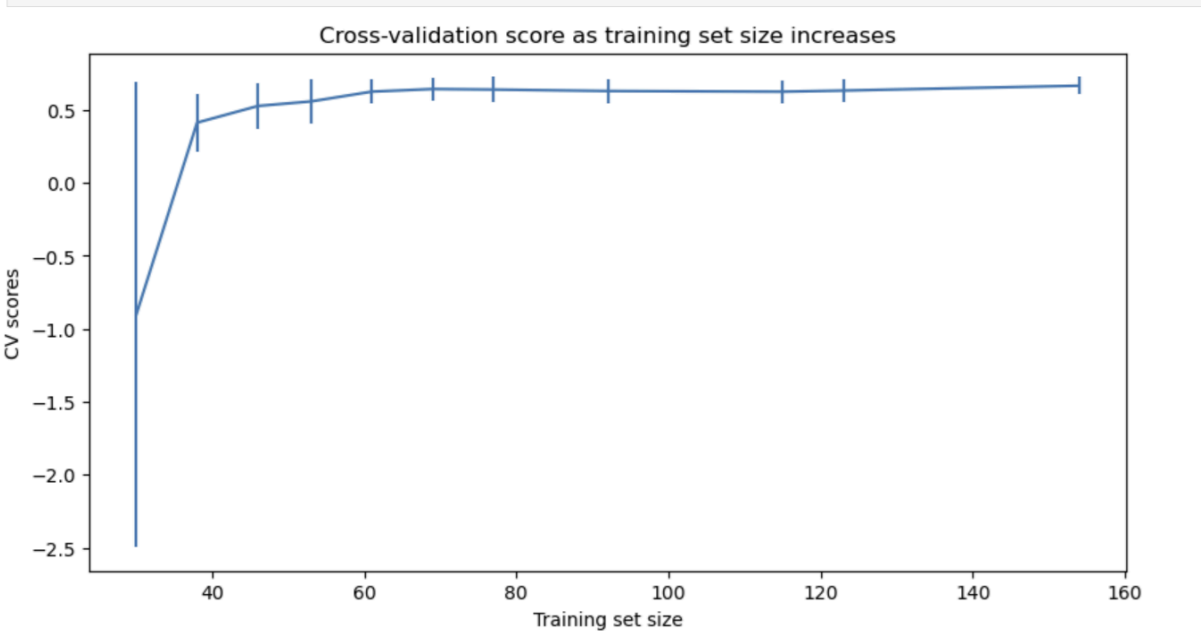

Figure: Cross validation score as training set size increases

This shows that you seem to have plenty of data. There's an initial rapid improvement in model scores as one would expect, but it's essentially levelled off by around a sample size of 40-50.

This model's ability to capture non-linear relationships and interactions makes it a more suitable choice for guiding business decisions based on housing price predictions.

Algorithms used to build the model with evaluation metric:

Big mountain resort current actual adult weekend ticket price is $ 81.00 and modelled price is 97.96. That is a potential price increase of $16.96.

This modeling analysis shows that ticket price is currently underpricing.

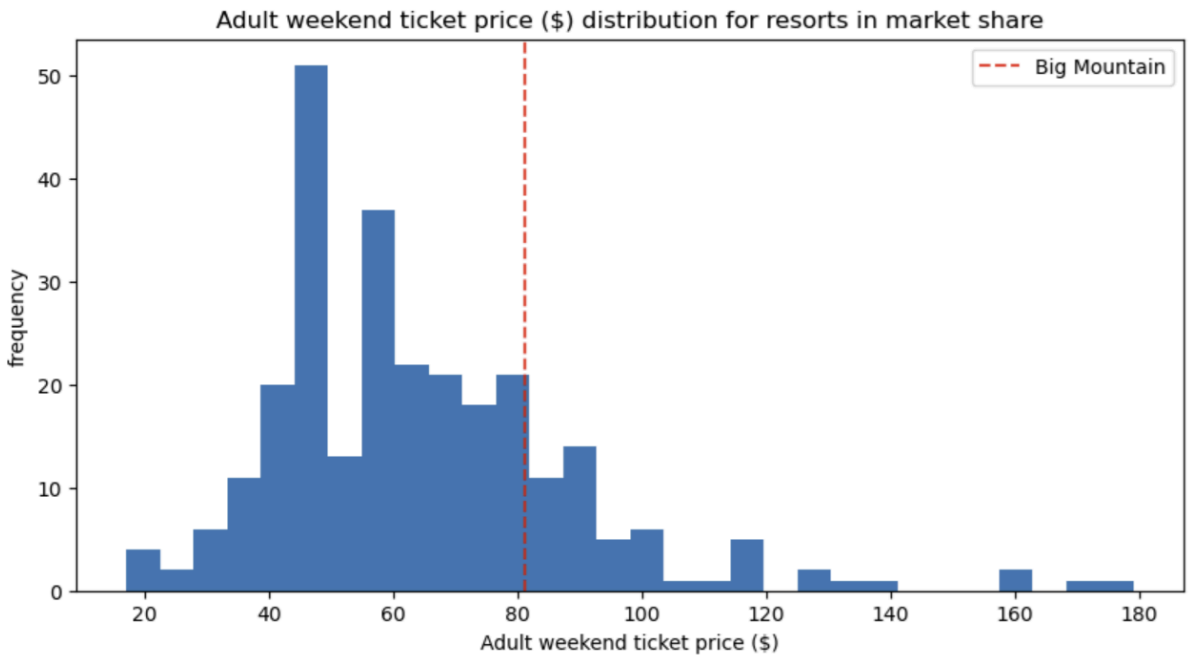

Figure: Adult weekend ticket price distribution for resorts

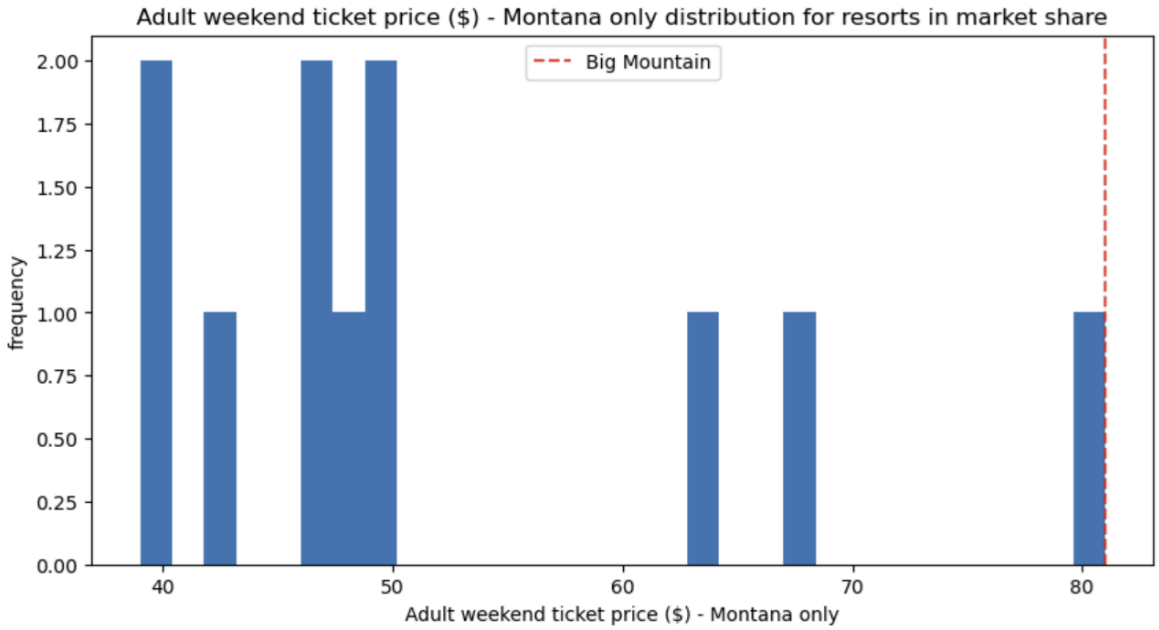

Figure: Adult Weekend ticket price distribution for Montana Resort only

Leadership wanted to train their model and wanted to make data driven business decisions to predict Big Mountain's ticket price. They also wanted to check competitors' prices and calculate a ticket price based on competitor pricing and facility offering.

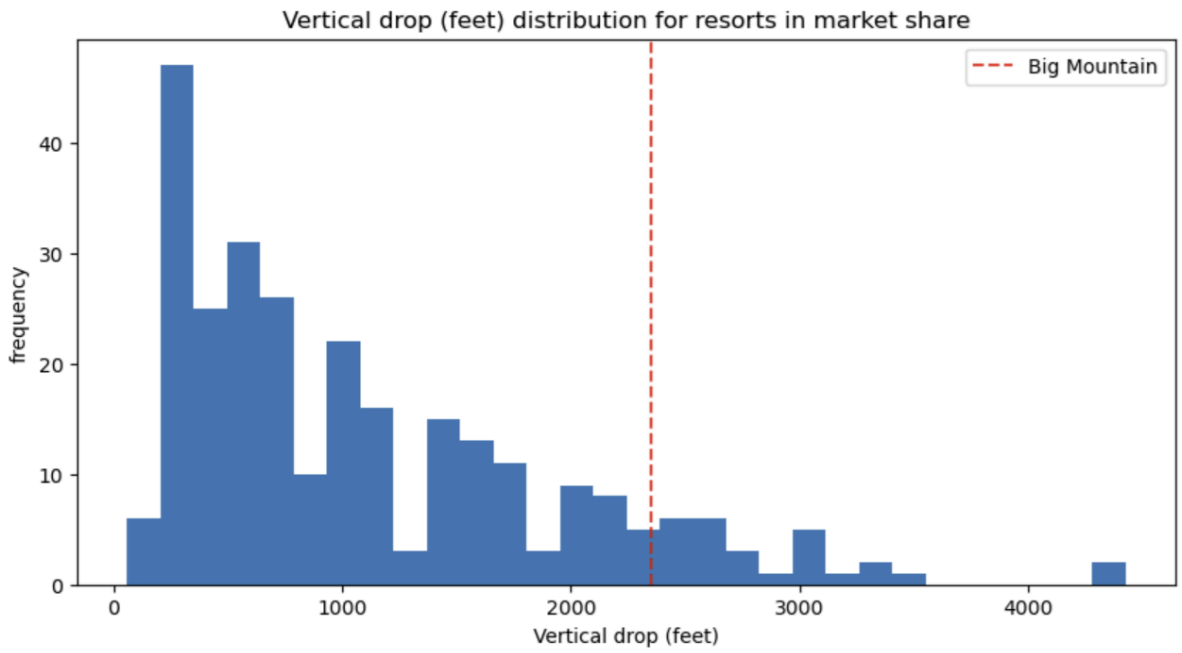

Figure: Vertical drop (feet) distribution for resorts

Big Mountain is doing well for vertical drop, but there are still quite a few resorts with a greater drop.

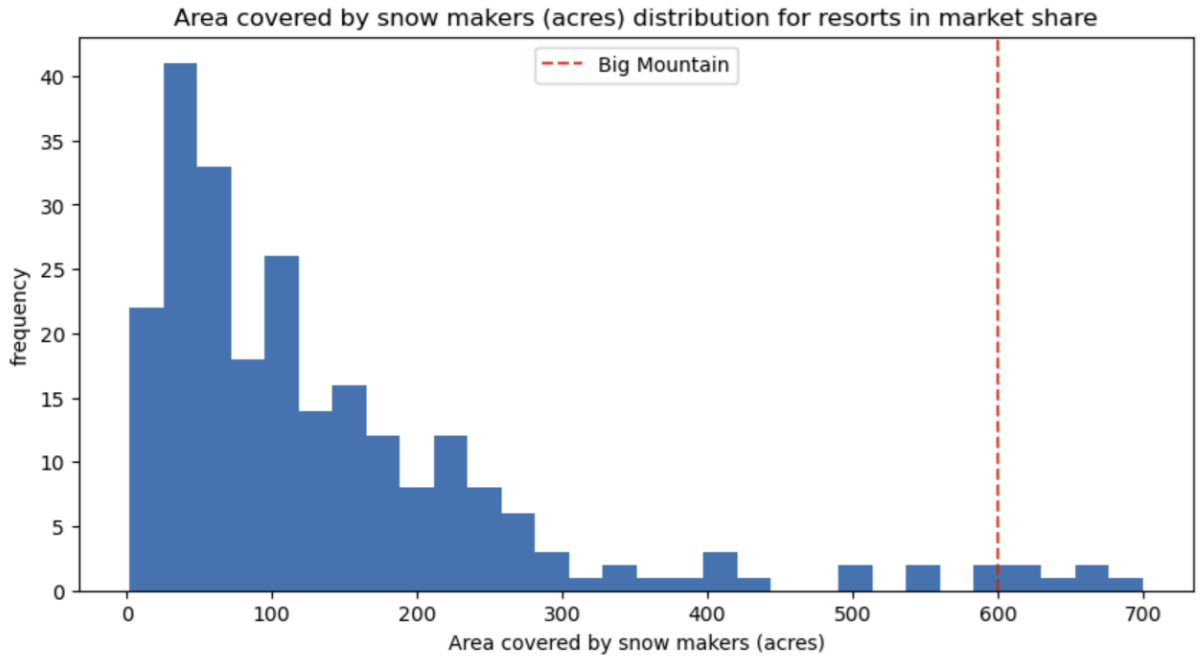

Figure: Area covered by snow makers (acres) distribution for resorts

Big Mountain is very high up the league table of snowmaking.

The additional operating cost per visitor should be analyzed based on ticket sales.

If each visitor buys 5 day tickets on average, the cost impact per ticket should be estimated. If increasing the price by approximately $17.00 covers the additional cost while maintaining market competitiveness, it should be considered.

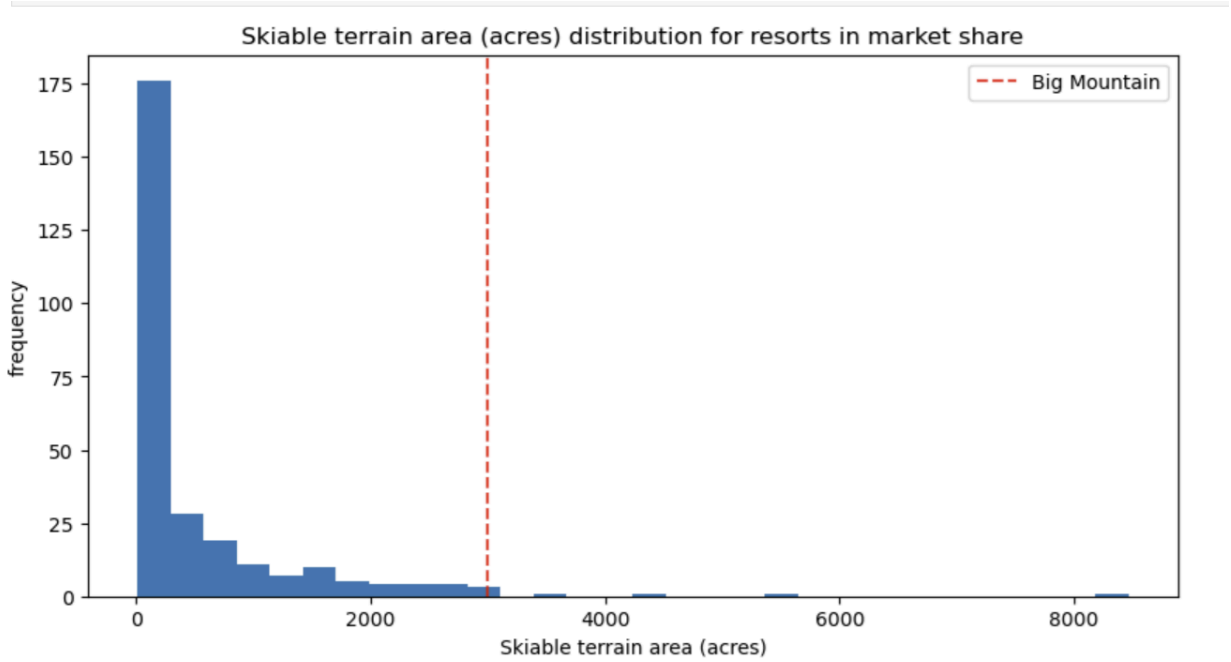

Figure: Skiable terrain area distribution for resorts in market share

Big Mountain is amongst the resorts with the largest amount of skiable terrain.

We should experiment with seasonal closures for low-traffic runs and also monitor visitor satisfaction and revenue impact before making permanent changes.

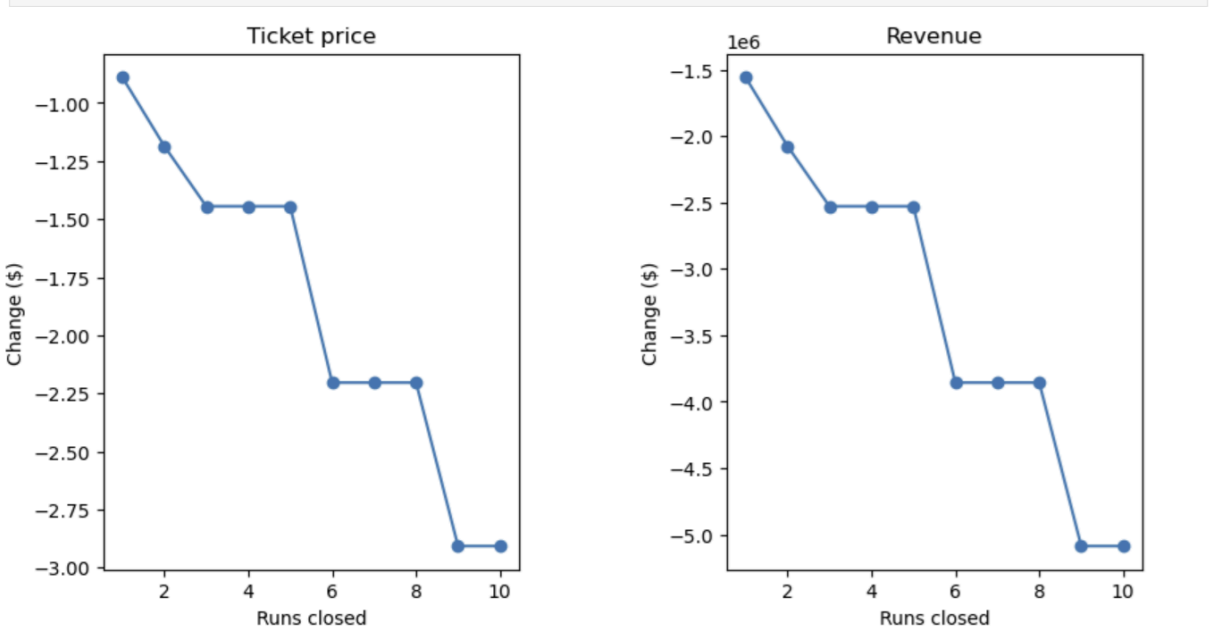

Figure: Two plots, side by side, for the predicted ticket price change (delta) for each condition (number of runs closed) in the scenario and the associated predicted revenue, change on the assumption that each of the expected visitors buys 5 tickets

Pricing recommendation

We took only price data for ticket price from our dataset. We should consider detailed operating cost breakdowns e.g., staff wages, maintenance cost. We should understand profit margins, that would make price increase recommendations more precise.

Big Mountain performs well in key facility metrics such as vertical drop, number of chairs, and skiable terrain.

The model assumes competitors' prices optimally, which may not always be true. Conducting a competitive pricing analysis with executive input would provide better insights.

Ticket price is not determined by any set of parameters; the resort is free to set whatever price it likes.

The business has shortlisted some options:

1. Permanently closing down up to 10 of the least used runs. This doesn't impact any other resort statistics.
2. Increase the vertical drop by adding a run to a point 150 feet lower down but requiring the installation of an additional chair lift to bring skiers back up, without additional snow making coverage
3. Same as number 2, but adding 2 acres of snow making cover
4. Increase the longest run by 0.2 mile to boast 3.5 miles length, requiring an additional snow making coverage of 4 acres

Conclusion:

The conclusion of the Big Mountain Resort case study suggests that the current pricing model is underpricing ticket sales, and there's potential for a price increase of approximately $16.96 per ticket without negatively impacting customer satisfaction.

The analysis revealed that the resort's vertical drop, skiable terrain, and snow-making capacity are strong factors that influence ticket prices. However, it is recommended that further steps be taken to refine the pricing strategy, including conducting competitive pricing analyses, experimenting with seasonal closures for low-traffic runs, and testing the impact of different pricing scenarios.

We need to develop an interactive dashboard where analysts can test different price and cost scenarios.

Future scope of work:

We should consider further improvements and further analysis. To avoid demand drops, we should consider price sensitivity testing.

If the model is deemed useful, executives may want to explore different pricing and facility scenarios.

To make it scalable, a self-service tool for business analysts could be built.

To make the model more accessible, we need to develop an interactive dashboard where analysts can test different price and cost scenarios. We also need to integrate it with financial projections to make data-backed pricing decisions.

We should experiment with seasonal closures for low-traffic runs and also monitor visitor satisfaction and revenue impact before making permanent changes.