Guided Capstone Project Report

A. <u>Objective:</u>

To determine if Big Mountain Resort (Montana) can increase daily ski lift ticket prices based on market comparisons and relative positioning

---

B. <u>Assumptions and Exclusions:</u>

B1. Key Assumptions →

- Prices are set by a free market (consumer demand versus available supply)
- State population data was extracted from Wikipedia.org; after inspection, the data appears to be consistent with census data
- I don't account for annualized visitor data or operating costs for each resort as that data wasn't available
- I don't account for demographic differences and household gross income differences between states/regions

B2. Key Exclusions →

- Dropped rows of resort data where the ticket price was not included after ensuring the ancillary data was not relevant for this analysis
- Dropped the weekday prices column and centralized the analysis around the weekend prices, as Montana's weekday/weekend prices are the same for all resorts
- After the holistic inspection of the dataset, it made sense to drop some numerical data deemed non-essential (no statistical influence) or incomplete for this analysis (e.g. FastEight)

C. <u>High-level steps for the Analysis:</u>

C1. Data Collection and Wrangling →

- Imported two separate datasets and inspected values
  - 1) comprehensive US ski resort dataset (ski_data, **Figs. A-C**) that incorporated 26 different, independent features
  - 2) state population and acreage data (state_summary, **Fig. D**) in order to account for density metrics
- Accounted for missing values and erroneous/extreme data points after visualizing distributions of each variable

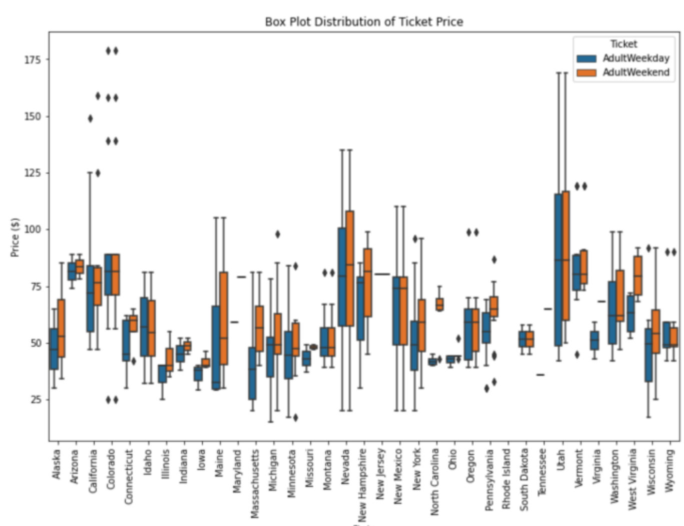

Fig. A – Ticket Price per State Box Plot (dependent variable)

	count	mean	std	min	25%	50%	75%	max
summit_elev	330.0	4591.818182	3735.535934	315.0	1403.75	3127.5	7806.00	13487.0
vertical_drop	330.0	1215.427273	947.864557	60.0	461.25	964.5	1800.00	4425.0
base_elev	330.0	3374.000000	3117.121621	70.0	869.00	1561.5	6325.25	10800.0
trams	330.0	0.172727	0.559946	0.0	0.00	0.0	0.00	4.0
fastEight	164.0	0.006098	0.078087	0.0	0.00	0.0	0.00	1.0
fastSixes	330.0	0.184848	0.651685	0.0	0.00	0.0	0.00	6.0
fastQuads	330.0	1.018182	2.198294	0.0	0.00	0.0	1.00	15.0
quad	330.0	0.933333	1.312245	0.0	0.00	0.0	1.00	8.0
triple	330.0	1.500000	1.619130	0.0	0.00	1.0	2.00	8.0
double	330.0	1.833333	1.815028	0.0	1.00	1.0	3.00	14.0
surface	330.0	2.621212	2.059636	0.0	1.00	2.0	3.00	15.0
total_chairs	330.0	8.266667	5.798683	0.0	5.00	7.0	10.00	41.0
Runs	326.0	48.214724	46.364077	3.0	19.00	33.0	60.00	341.0
TerrainParks	279.0	2.820789	2.008113	1.0	1.00	2.0	4.00	14.0
LongestRun_mi	325.0	1.433231	1.156171	0.0	0.50	1.0	2.00	6.0
SkiableTerrain_ac	327.0	739.801223	1816.167441	8.0	85.00	200.0	690.00	26819.0
Snow Making_ac	284.0	174.873239	261.336125	2.0	50.00	100.0	200.50	3379.0
daysOpenLastYear	279.0	115.103943	35.063251	3.0	97.00	114.0	135.00	305.0
yearsOpen	329.0	63.656535	109.429928	6.0	50.00	58.0	69.00	2019.0
averageSnowfall	316.0	185.316456	136.356842	18.0	69.00	150.0	300.00	669.0
AdultWeekday	276.0	57.916957	26.140126	15.0	40.00	50.0	71.00	179.0
AdultWeekend	279.0	64.166810	24.554584	17.0	47.00	60.0	77.50	179.0
projectedDaysOpen	283.0	120.053004	31.045963	30.0	100.00	120.0	139.50	305.0
NightSkiing_ac	187.0	100.395722	105.169620	2.0	40.00	72.0	114.00	650.0

Fig. B – Statistical Summary of all Numerical Data for the Ski_Data Dataset

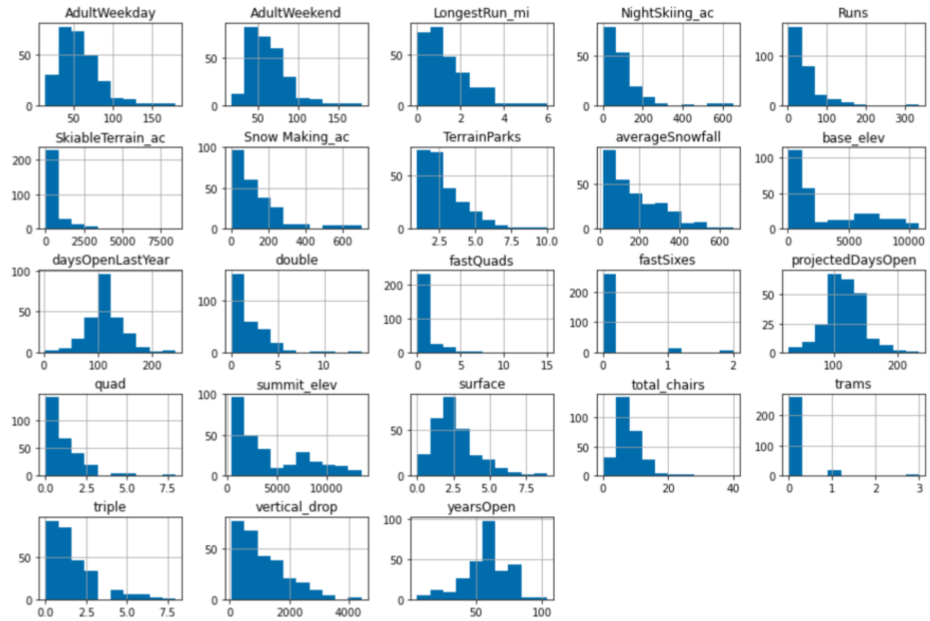

Fig C – Distributions of all Variables within the Ski_Data Dataset

```
state_summary.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   state                        35 non-null     object
 1   resorts_per_state            35 non-null     int64
 2   state_total_skiable_area_ac  35 non-null     float64
 3   state_total_days_open        35 non-null     float64
 4   state_total_terrain_parks    35 non-null     float64
 5   state_total_nightskiing_ac   35 non-null     float64
 6   state_population             35 non-null     int64
 7   state_area_sq_miles          35 non-null     int64
dtypes: float64(4), int64(3), object(1)
memory usage: 2.3+ KB
```

```
state_summary.head()
```

	state	resorts_per_state	state_total_skiable_area_ac	state_total_days_open	state_total_terrain_parks	state_total_nightskiing_ac	state_population	state_a
0	Alaska	3	2280.0	345.0	4.0	580.0	731545	
1	Arizona	2	1577.0	237.0	6.0	80.0	7278717	
2	California	21	25948.0	2738.0	81.0	587.0	39512223	
3	Colorado	22	43682.0	3258.0	74.0	428.0	5758736	
4	Connecticut	5	358.0	353.0	10.0	256.0	3565278	

Fig. D – State_Summary Dataset Excerpt

C2. Exploratory Data Analysis →

- Prior to combining that state_summary and ski_resort datasets, it was necessary to complete a parts component analysis (PCA) on the state_summary dataset in order to determine which states demonstrate similar statistical characteristics when considering population and density metrics
    - As you will see below, we have some clustering in the bottom left quadrant with dispersion becoming more pronounced as you move to the right and up
    - This step prompts additional questions regarding the outliers (New York, Vermont, New Hampshire, Colorado)
    - PC1 and PC2 account for ~80% of the statistical variance when comparing state-level data

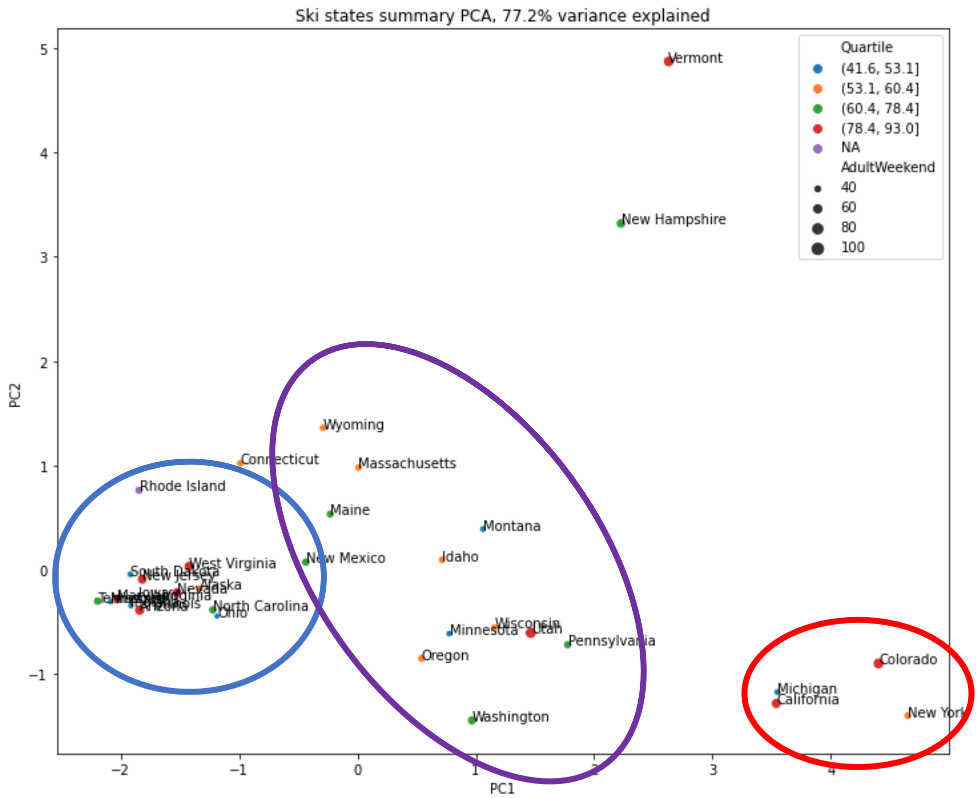

Fig. E – PCA analysis of state_summary data

- After inspecting the data more closely, it appears as though the resorts_per_100ksq_mile and resorts_per_100kcapita are the features most influencing this PCA analysis
    - Vermont and New Hampshire have large values for resorts_per_100ksq_mile in absolute terms and Vermont also has a large value for resorts_per_100kcapita while New York's value is low
- Based on the PCA analysis, it doesn't make sense to treat certain states/regions differently
- Now that we have a more robust dataset, it is time to understand the correlation between all features and determine if multicollinearity is an issue; **Fig. F** showcases the correlation matrix for all features/variables

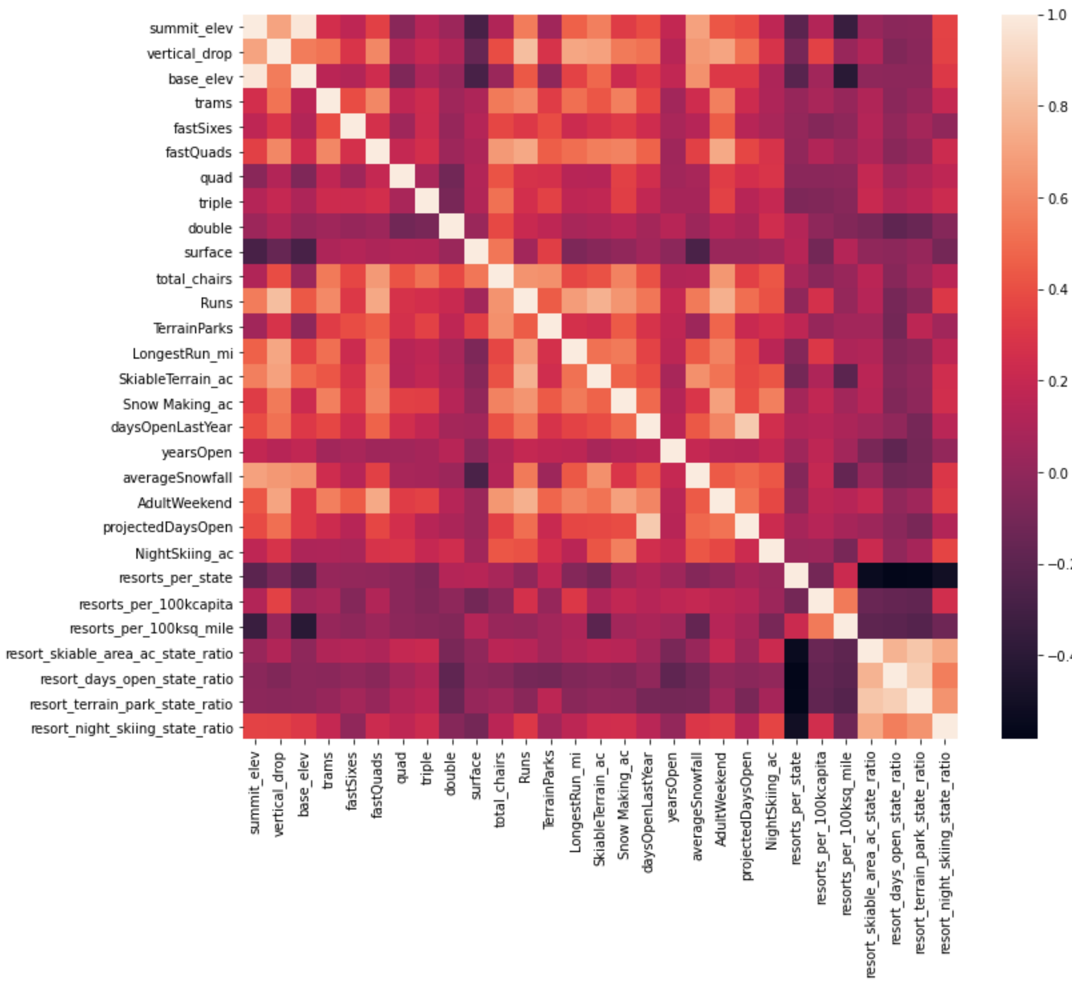

Fig. F – Correlation Matrix for all Features

- Based on the above matrix, the variables that are most positively correlated with ticket price are: vertical_drop, fastQuads, runs, total_chairs and snow_making; it is important to note that correlation doesn't necessarily mean causation

C3. Preprocessing, Training and Optimizing the Predictive Models →

- Before building and training, I needed to compartmentalize the data
  - I split the ski_resort dataset into training (70%) and testing (30%) datasets and imputed missing values with the median of each feature
  - This training/testing partition was necessary to limit overfitting and increase the accuracy of my models
  - The test dataset was only used after the initial model was trained, refined and optimized using the training dataset
  - Key metrics for determining accuracy and relevance/performance of each model:
    - Coefficient of determination (Squared)
    - Mean absolute error (MAE)
    - Mean squared error (MSE)
- Initially, I built a simple linear regression model
  - Cross-validation and SelectBestK techniques were used to optimize the linear regression model

- **Fig. G** showcases the k scores; from the graph we can see that the model is optimal when considering the 8 most influential features; beyond this, the model begins to become erratic and overfitting is a real problem

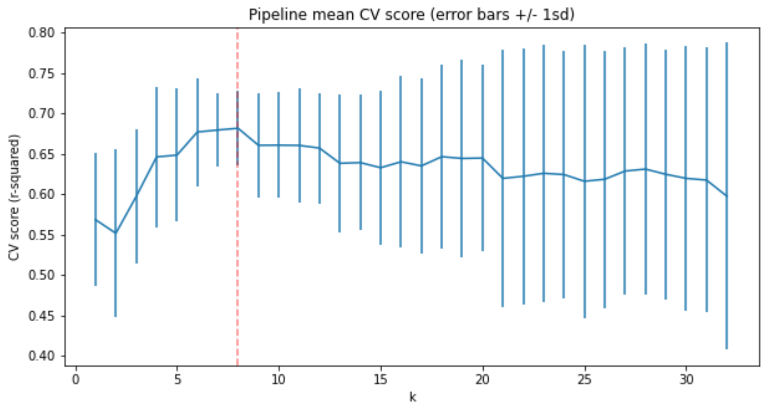

Fig. G – Understanding the # of features to include in the model (SelectKBest technique)

```
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      -5.249780
dtype: float64
```

Fig. H – 8 Features and their Coefficients

- For the second model, I used the Random Forest Regressor package
  - After completing the cross-validation step, the best model included features that weren't scaled, imputed missing values with the median of each feature and used 69 random forest regressors
- Based on the Random Forest model, the most important features are shown below in **Fig. I**

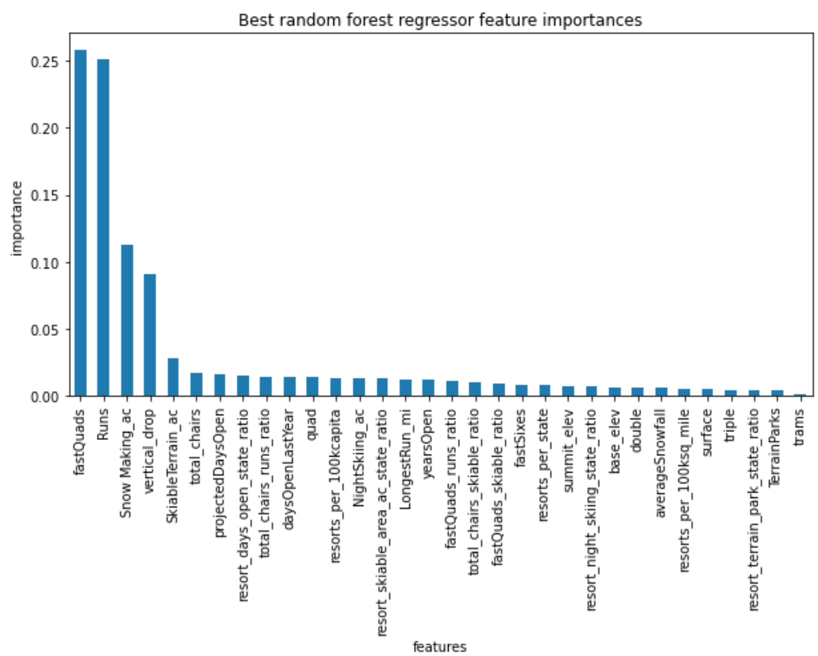

Fig. I – Random Forest Feature Importance

- When comparing the two models, the Random Forest Regressor was more accurate and also exhibited less variability
  - For the linear regression model, the MAE = $10.50 +- $1.62
  - For the random forest model, the MAE = $9.65 +- $1.35

C4. Modelling Scenarios - Determining the true value for a Big Mountain Ski Lift Ticket →

- After running the random forest model on the test dataset, Big Mountain's modelled price came out to $95.87
    - when compared to their current price of $81 and after accounting for an MAE of $10.39, the model implied that Big Mountain could possibly raise their ticket price without adversely affecting their current market share
- Does the modelled price make sense? Where does Big Mountain fit into the market landscape? Let's see below

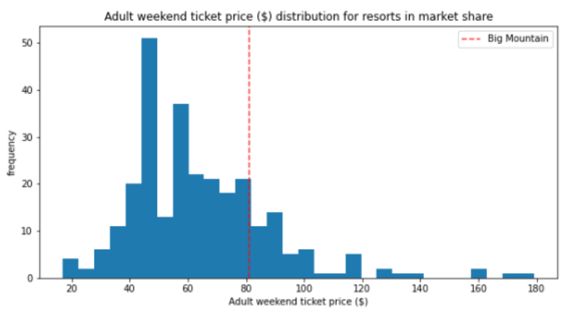

Fig. J – Ticket Prices (All Resorts)

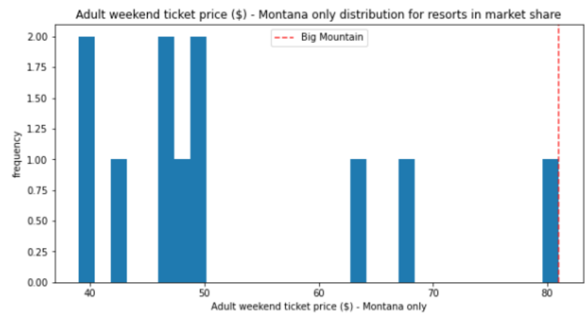

Fig. K – Ticket Prices (Montana Resorts)

- Big Mountain is above the median ticket price when considering all resorts in the US and is also the most expensive in Montana

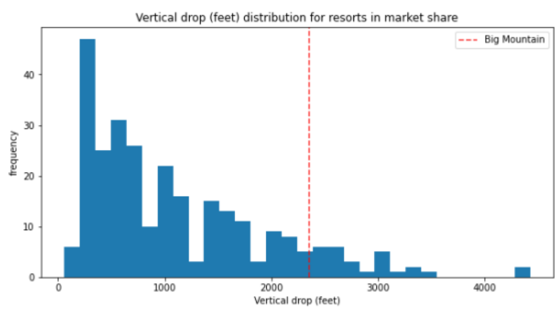

Fig. L – Vertical Drop

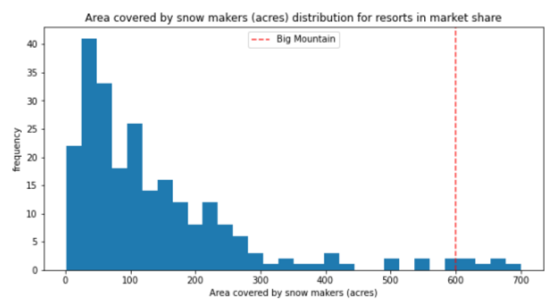

Fig. M – Snow Making Area

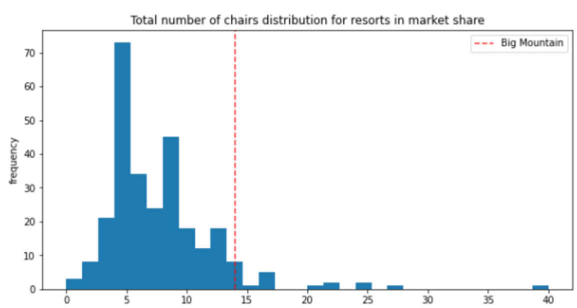

Fig. N – # of Chairs

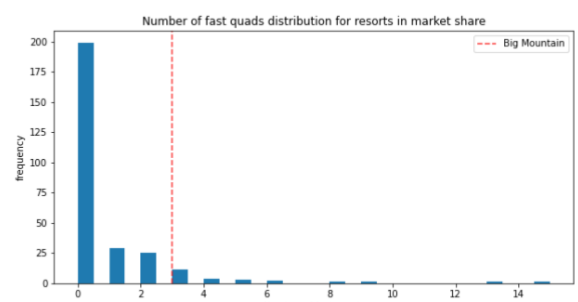

Fig. O – # of Fast Quads

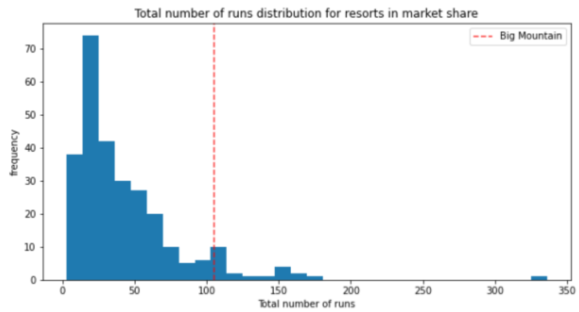

Fig. P – # of Runs

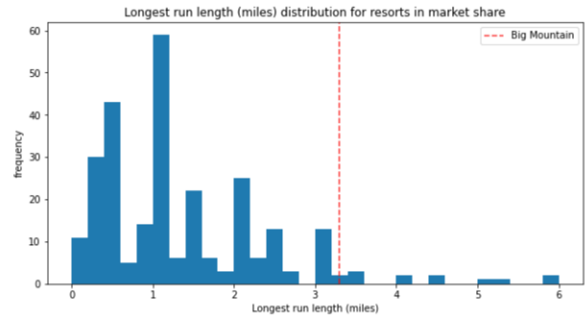

Fig. Q – Longest Run Length

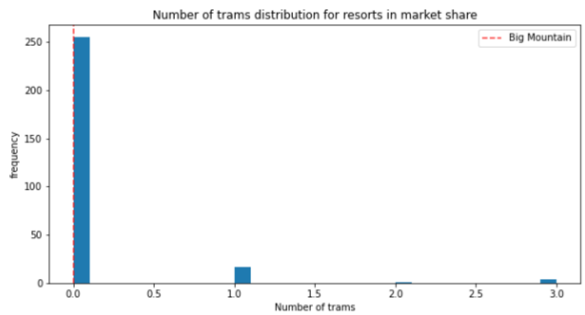

Fig. R – # of Trams

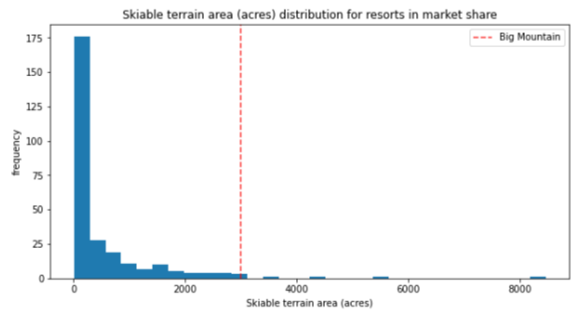

Fig. S – Total Skiable Terrain Area

- Now, it makes sense to test out some scenarios; for each scenario, I used the assumption that Big Mountain expects 350,000 visitors this year
- Scenario 1 – Close up to 10 of the least used runs
  - As you will be able to see from the figures below, closing down 3 runs would result in a $.70 reduction in ticket price
  - Interestingly enough, closing down 1 run wouldn't affect the ticket price per the model

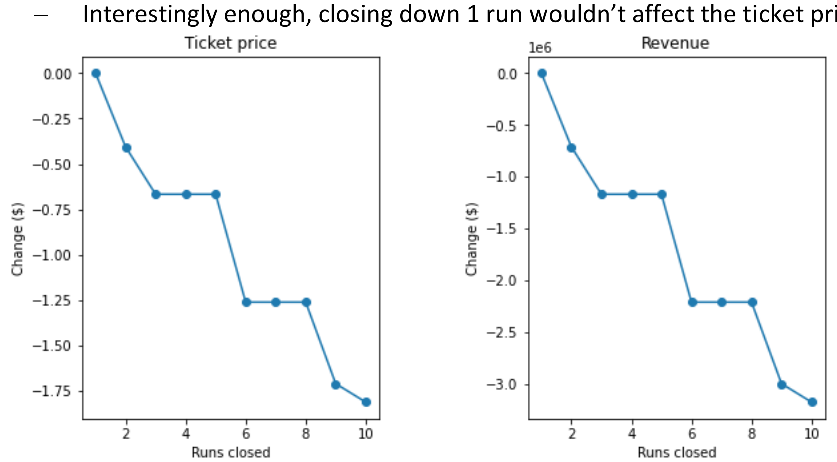

Fig. T – Modeling Ticket Price based on # of Runs Closed

- Scenario 2 – adding a run, increasing the vertical drop by 150 ft and installing an additional chair lift
  - After running the model, this scenario supported a ticket price increase of $1.99
- Scenario 3 – replication of scenario 2 + adding 2 acres of snow making
  - Again, the model suggests the resort could increase their ticket price by $1.99
- Scenario 4 – increasing the longest run by .2 miles + adding 4 acres of snow making capability
  - The model suggested that the ticket price remain the same

D. Concluding Thoughts and Recommendations:

- Based on the proposed model and scenarios, I believe that Big Mountain is underestimating their position in the marketplace by $5-10
- To better understand the decision to price the median ticket price at $81, I would need additional insight from the business and leadership team (operating parameters, immediate/regional competition, gross household income, etc.)
- At the very least, the model suggests that the leadership team should:
  - review their current operating parameters and general layout (e.g. what runs add the most value)
  - review their current pricing strategy
  - better understand their overall position in the market
  - create a pragmatic plan that supports ticket price appreciation