Guided Capstone Step 6

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.

In this capstone we are going to give detailed information on important topics of:

- Problem statement
- Data Wrangling
- Exploratory Data Analysis
- Pre-Processing and Training Data
- Model Pre-processing
- Algorithms used to build the model with evaluation metric
- Winning model and scenario modelling
- Pricing recommendation
- Conclusion
- Future scope of work

Problem statement

The problem statement looks as below:

What opportunities exist for Big Mountain Resort to recoup the increased operation cost of $1,540,000 while keeping the profit margin of at least 9.2% for the next year?

The operation cost has been increased by $1,54 million for installing a new chair lift to distribute visitors across the mountain. For this reason, the Management has expressed a desire change resort's pricing strategy to capitalizing on its facilities as much as they could. In particular, the Executive Team is looking for ways to compensate the increased cost to keep its profit margin minimum at 9.2 %. Therefore, adding an additional cost on ticket pricing will support with an increased operational costs which was the result of installed an additional chair lift to help increase the distribution of visitors across the mountain.

Data Wrangling

After uploading the data we realized that "`fastEight`" has the most missing values, at just over 50%. Unfortunately, you see you're also missing quite a few of your desired target quantity, the ticket price, which is missing 15-16% of values. "`AdultWeekday`"is missing in a few more records than "`AdultWeekend`"

```
missing = pd.concat([ski_data.isnull().sum(), 100 * ski_data.isnull().mean()], axis=1)
missing.columns=['count', '%']
missing.sort_values(by='count', ascending = False)
```

	count	%
fastEight	166	50.303030
NightSkiing_ac	143	43.333333
AdultWeekday	54	16.363636
AdultWeekend	51	15.454545
daysOpenLastYear	51	15.454545
TerrainParks	51	15.454545
projectedDaysOpen	47	14.242424
Snow Making_ac	46	13.939394
averageSnowfall	14	4.242424
LongestRun_mi	5	1.515152
Runs	4	1.212121
SkiableTerrain_ac	3	0.909091
yearsOpen	1	0.303030
total_chairs	0	0.000000
Name	0	0.000000
Region	0	0.000000
double	0	0.000000

The original number consisted of 330 rows, 27 columns. Our resort 'Big Mountain Resort' was presented on 151 row. Our resort had no missing values. Columns which had missing values were removed. Importantly, the key columns 'AdultWeekday' & 'AdultWeekend' had missing values of 16.36% and 15.45% respectively. The most missing values were found in `fastEight' with 50.3%. Found that more than 82% of the resorts have no missing values.

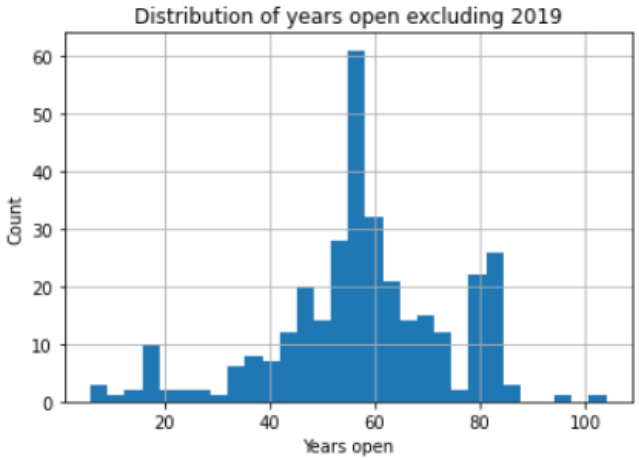
```
missing_price = ski_data[['AdultWeekend', 'AdultWeekday']].isnull().sum(axis=1)
missing_price.value_counts()/len(missing_price) * 100
```

```
0    82.317073
2    14.329268
1     3.353659
```

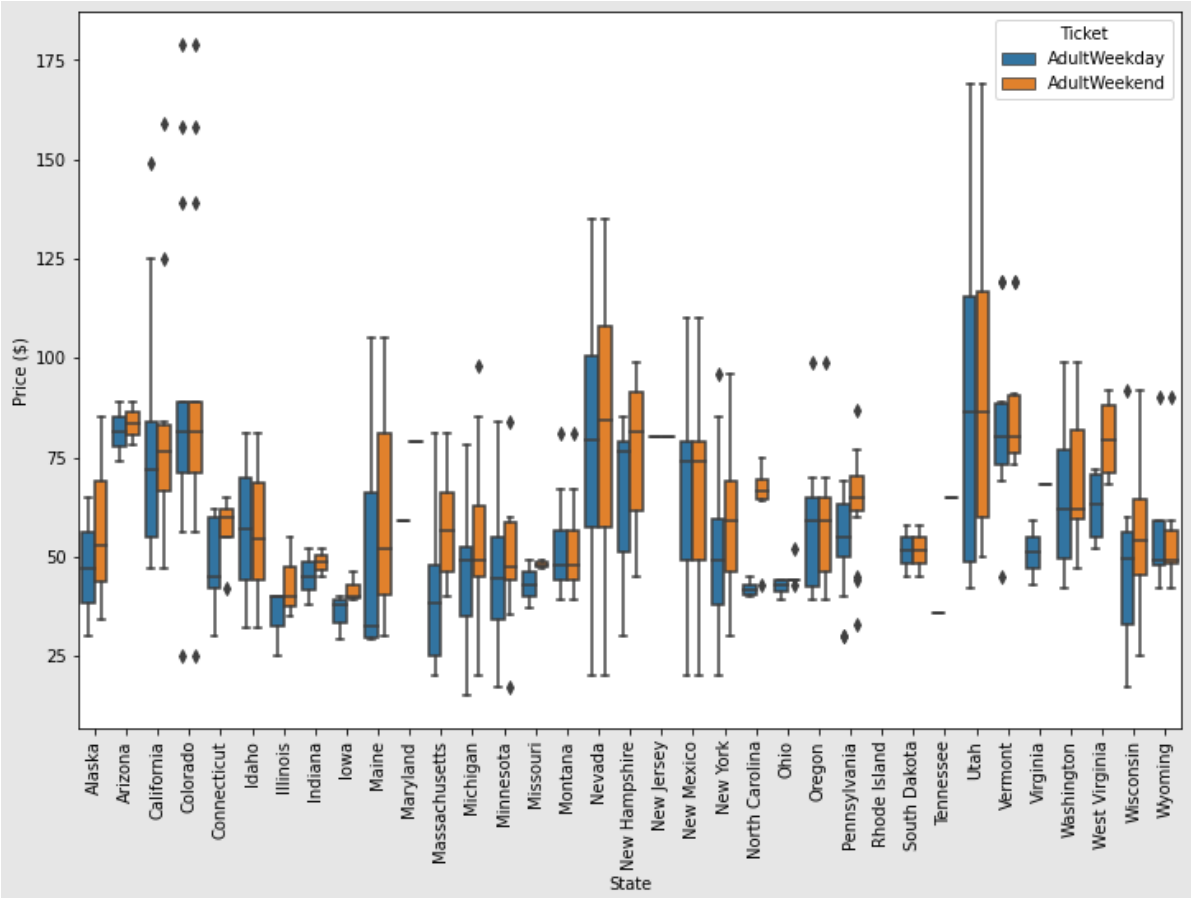
Regarding the 'AdultWeekend' & 'AdultWeekday'(two important columns) we found that 3%of the resorts with missing values in one column and 14% with 2 columns. We dropped the records for these columns where we had both missing columns.

For 'Silverton Mountain' we found a mistake in skiable area (which was = 26819) and we corrected after doing research that the skiable area = 1819.

Pine Knob Ski Resort was down as having been open for 2019 years, which is an obvious wrong information, and we kept the data without this resort.



Distribution of weekday and weekend price by state



The above chart represent the price variation of weekdays and weekends.

Relatively expensive ticket prices in California, Colorado, and Utah. The majority of others from 25 to 100 dollars. Some States show more variability than others. Nevada and Utah, on the other hand, show the most range in prices. Some States, like North Carolina and Virginia, have weekend prices higher than weekday.

In Montana we found that the average ticket price is the same ($81) during weekends and weekdays and are cheaper in comparison with others. Most states have different price variations between Weekends and Weekdays. In majority of them weekends prices are higher than weekdays.

Exploratory Data Analysis

We started by checking the total state area, and our state, Montana, comes in at third largest.

```
1 state_summary_newind.state_area_sq_miles.sort_values(ascending=False).head()
```

```
state
Alaska        665384
California    163695
Montana       147040
New Mexico    121590
Arizona       113990
```

Then we calculated *Total state population* and found that California dominates the state population figures despite coming in second behind Alaska in size (by a long way). The resort's state of Montana was in the top five for size, but doesn't figure in the most populous states. Thus our state is less densely populated.

```
1 state_summary_newind.state_population.sort_values(ascending=False).head()
```

```
state
California      39512223
New York        19453561
Pennsylvania    12801989
Illinois        12671821
Ohio            11689100
```

In total skiable area New York State may have the most resorts, but they don't account for the most skiing area. In fact, New York doesn't even make it into the top five of skiable area. Good old Montana makes it into the top five, though. You may start to think that New York has more, smaller resorts, whereas Montana has fewer, larger resorts. Colorado seems to have a name for skiing; it's in the top five for resorts and in top place for total skiable area.

```
1 state_summary_newind.state_total_skiable_area_ac.sort_values(ascending=False).head()
```

```
state
Colorado      43682.0
Utah          30508.0
California    25948.0
Montana       21410.0
Idaho         16396.0
```

Then, we decided to calculate ‘Resort dencity’. Montana is 4th in resorts per 100k capital.

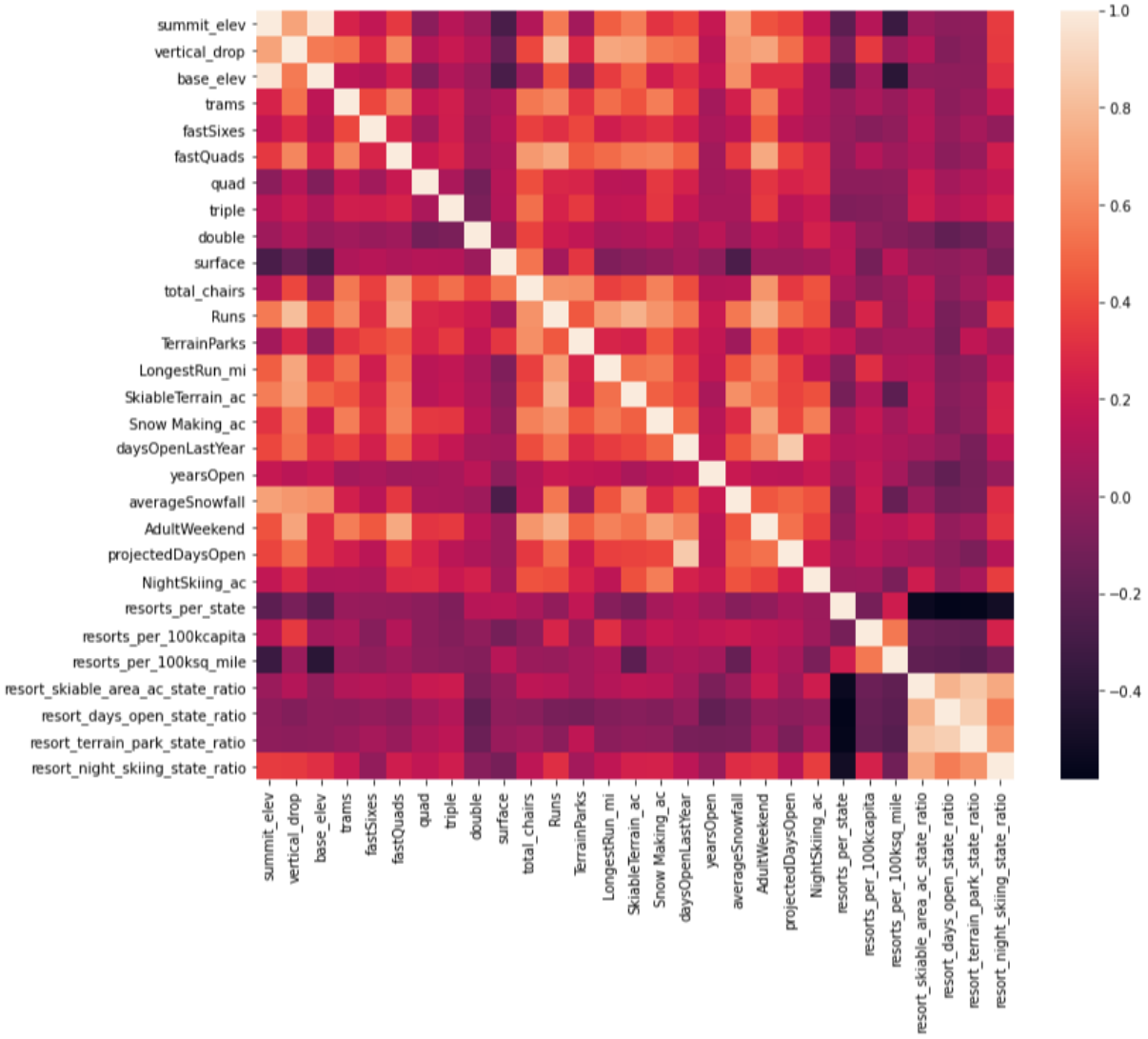
After scaling the data and exploring trends in features using PCA techniques we checked mean and standard deviation.

The heat map gave us a lot of multicollinearity with your new ratio features; they are negatively correlated with the number of resorts in each state. An interesting observation in this region of the heatmap is that there is some positive correlation between the ratio of night skiing area with the number of resorts per capita. In other words, it seems that when resorts are more densely located with population, more night skiing is provided.

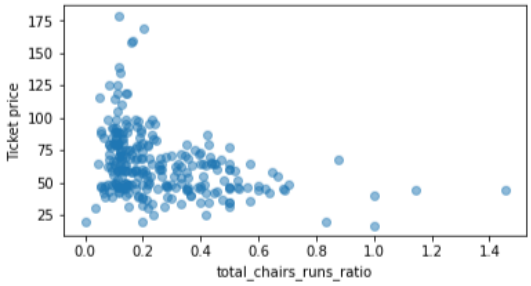
Turning your attention to your target feature, AdultWeekend ticket price, you see quite a few reasonable correlations. fastQuads stands out, along with Runs and Snow Making_ac. The last one is interesting. Visitors would seem to value more guaranteed snow, which would cost in terms of snow making equipment, which would drive prices and costs up. Of the new features, resort_night_skiing_state_ratio seems the most correlated with ticket price. If this is true, then perhaps seizing a greater share of night skiing capacity is positive for the price a resort can charge.

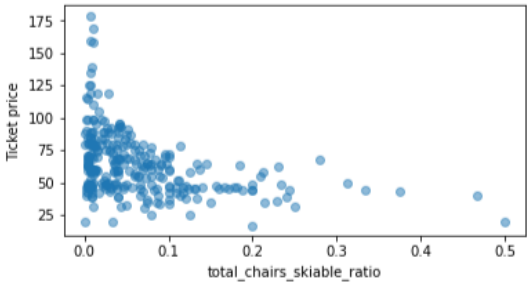
As well as “Runs”,” total_chairs” is quite well correlated with ticket price. This is plausible; the more runs you have, the more chairs you'd need to ferry people to them! Interestingly, they may count for more than the total

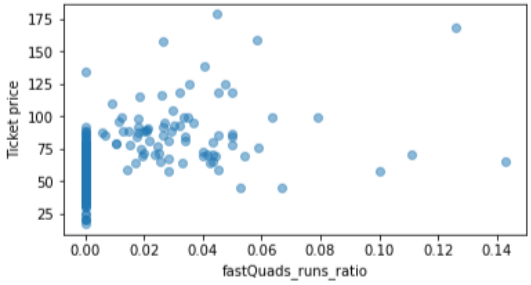
skiable terrain area. For sure, the total skiable terrain area is not as useful as the area with snow making. People seem to put more value in guaranteed snow cover rather than more variable terrain area. The vertical drop seems to be a selling point that raises ticket prices as well.

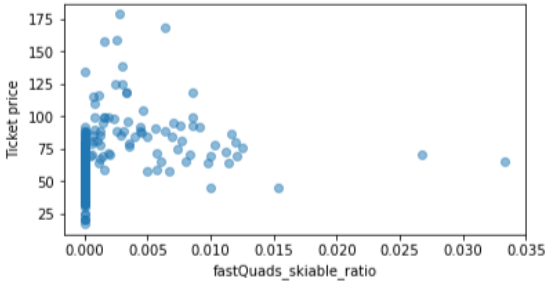


We note strong correlations between "*summit elevation*" vs "*vertical drop*", "*number of fast quads*" vs "*total number*" of "*chairs*", and "*number of runs*" vs "*a resort's ticket prices*". In future steps, modelling these features with state labels as indices will hopefully provide useful information.









At first these relationships are quite counterintuitive. It seems that the more chairs a resort has to move people around, relative to the number of runs, ticket price rapidly plummets and stays low. What we may be seeing

here is an exclusive vs. mass market resort effect; if you don't have so many chairs, you can charge more for your tickets, although with fewer chairs you're inevitably going to be able to serve fewer visitors. Your price per visitor is high but your number of visitors may be low. Something very useful that's missing from the data is the number of visitors per year.

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.

Pre-Processing and Training Data

We decided to split our data into Train/Test sets. By partitioning the data into training and testing splits, without letting a model (or missing-value imputation) learn anything about the test split, we would have a somewhat independent assessment of how our model might perform in the future.

```
len(ski_data) * .7, len(ski_data) * .3
```

```
(193.2, 82.8)
```

```
X_train, X_test, y_train, y_test = train_test_split(ski_data.drop(columns='AdultWeekend'),
                                                    ski_data.AdultWeekend, test_size=0.3,
                                                    random_state=47)
```

```
X_train.shape, X_test.shape
```

```
((193, 35), (83, 35))
```

```
y_train.shape, y_test.shape
```

```
((193,), (83,))
```

We started by seeing how good the mean is as a predictor, and `sklearn`'s `DummyRegressor` does this:

```
dumb_reg = DummyRegressor(strategy='mean')
dumb_reg.fit(X_train, y_train)
dumb_reg.constant_
```

```
array([[63.81108808]])
```

Then, we moved into calculating R-squared, Mean absolute error and Mean squared error.

R-squared

```
r2_score(y_train, y_tr_pred), r2_score(y_test, y_te_pred)
```

```
(0.0, -0.0031235200417913944)
```

Mean absolute error

```
mean_absolute_error(y_train, y_tr_pred), mean_absolute_error(y_test, y_te_pred)
```

```
(17.92346371714677, 19.136142081278486)
```

Mean squared error

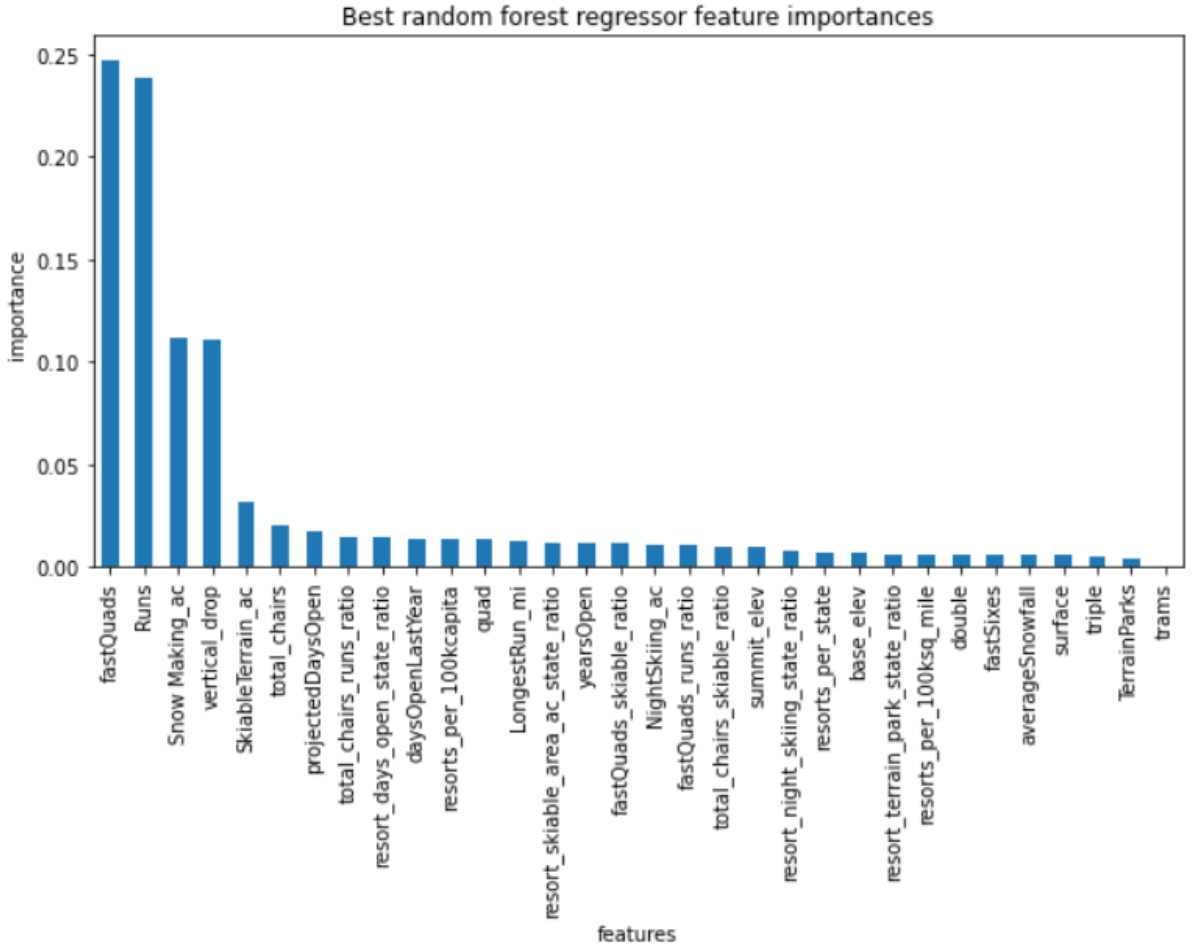
```
mean_squared_error(y_train, y_tr_pred), mean_squared_error(y_test, y_te_pred)
```

```
(614.1334096969046, 581.4365441953483)
```

We decided to use *Random Forest Model -* a model that can work very well in a lot of cases is the random forest. For regression, this is provided by `sklearn`'s `RandomForestRegressor` class.

After defining the “pipeline”, Fit and assess performance using cross-validation, using “Hyperparameter” search using “GridSearchCV” we created “Best random forest regressor feature importances” bar chart as below:

The below chart gave us the dominant top four features are: fastQuads, Runs, Snow Making_ac, vertical_drop.



Final Model Selection

Time to select your final model to use for further business modeling! It would be good to revisit the above model selection; there is undoubtedly more that could be done to explore possible “hyperparameters”. It would also be worthwhile to investigate removing the least useful features. Gathering or calculating, and storing, features adds business cost and dependencies, so if features genuinely are not needed they should be removed.

I.1 Linear regression model performance

```
# '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)
```

```
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
```

```
(10.499032338015297, 1.6220608976799646)
```

```
mean_absolute_error(y_test, lr_grid_cv.best_estimator_.predict(X_test))
```

```
11.793465668669327
```

I.2 Random forest regression model performance

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

```
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
```

```
(9.73870044534413, 1.234723025791237)
```

```
mean_absolute_error(y_test, rf_grid_cv.best_estimator_.predict(X_test))
```

```
9.490780722891566
```

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.

Model Preprocessing

We can now use our model to gain insight into what Big Mountain's ideal ticket price could/should be, and how that might change under various scenarios.

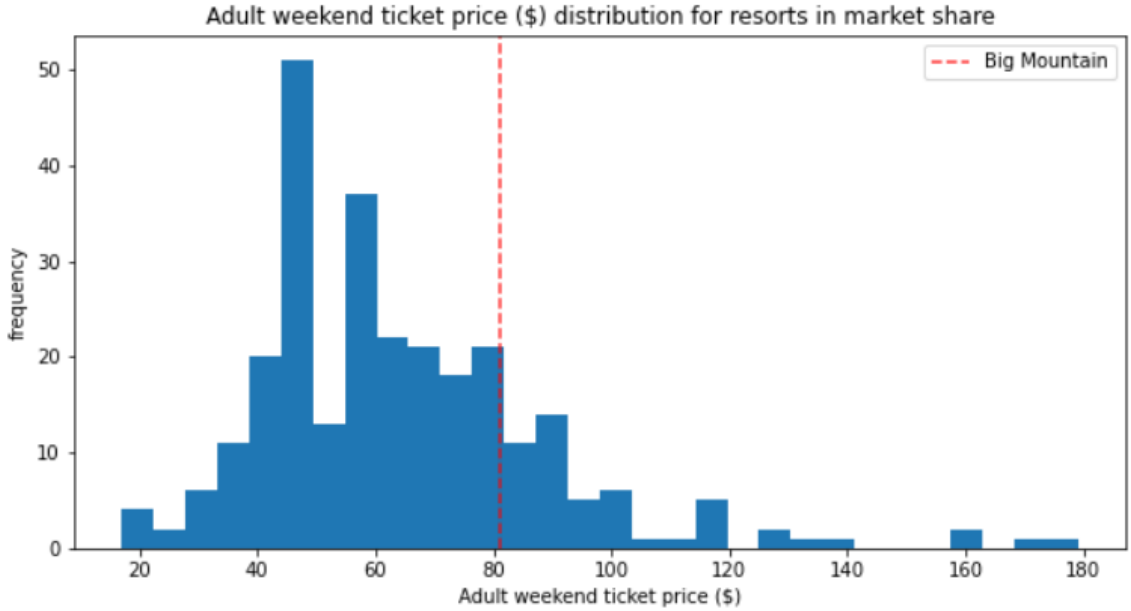
We calculated that:

```
Big Mountain Resort modelled price is $93.93, actual price is $81.00.
Even with the expected mean absolute error of $10.26, this suggests there
is room for an increase.
```

Features that came up as important in the modeling (not just our final, random forest model) included:

- vertical_drop
- Snow Making_ac
- total_chairs
- fastQuads
- Runs
- LongestRun_mi
- trams
- SkiableTerrain_ac

Algorithms used to build the model with evaluation metric.



- Big Mountain is doing well for vertical drop, but there are still quite a few resorts with a greater drop.
- Big Mountain is very high up the league table of snow making area.
- Big Mountain has amongst the highest number of total chairs, resorts with more appear to be outliers.
- Most resorts have no fast quads. Big Mountain has 3, which puts it high up that league table. There are some values much higher, but they are rare.
- Big Mountain compares well for the number of runs. There are some resorts with more, but not many.
- Big Mountain has one of the longest runs. Although it is just over half the length of the longest, the longer ones are rare.
- The vast majority of resorts, such as Big Mountain, have no trams.
- Big Mountain is amongst the resorts with the largest amount of skiable terrain.

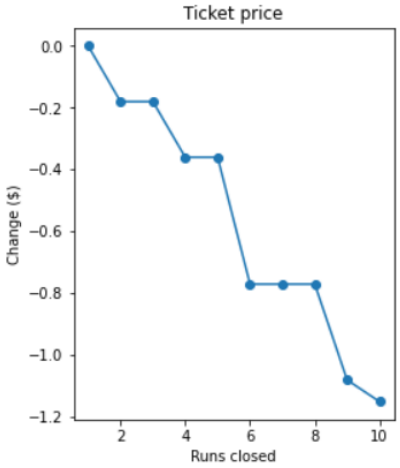
Winning model and scenario modelling

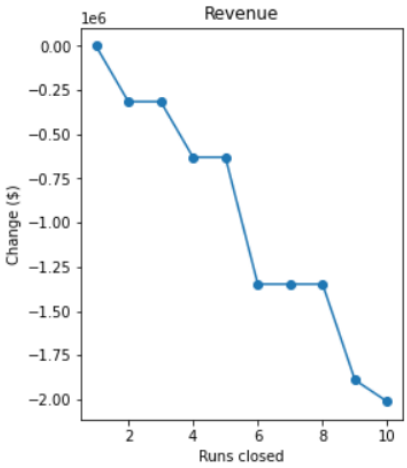
Big Mountain Resort has been reviewing potential scenarios for either cutting costs or increasing revenue (from ticket prices). Ticket price is not determined by any set of parameters; the resort is free to set whatever price it likes. However, the resort operates within a market where people pay more for certain facilities, and less for others. Being able to sense how facilities support a given ticket price is valuable business intelligence. This is where the utility of our model comes in.

We had some scenarios:

- Scenario 1

Close up to 10 of the least used runs. The number of runs is the only parameter varying. The model says closing one run makes no difference. Closing 2 and 3 successively reduces support for ticket price and so revenue. If Big Mountain closes down 3 runs, it seems they may as well close down 4 or 5 as there is no further loss in ticket price. Increasing the closures down to 6 or more leads to a large drop.





- Scenario 2

In this scenario, Big Mountain is adding a run, increasing the vertical drop by 150 feet, and installing an additional chair lift. This scenario increases support for ticket price by $11.22.

```
This scenario increases support for ticket price by $11.22
Over the season, this could be expected to amount to $19635175
```

- Scenario 3

In this scenario, you are repeating the previous one but adding 2 acres of snow making. This scenario increases support for ticket price by $12.27.

```
This scenario increases support for ticket price by $12.27
Over the season, this could be expected to amount to $21472675
```

- Scenario 4

This scenario calls for increasing the longest run by 0.2 miles and guaranteeing its snow coverage by adding 4 acres of snow making capability.

```
#Code task 6#
#Predict the increase from adding 0.2 miles to `LongestRun_mi` and 4 to `Snow Making_ac`
predict_increase(['LongestRun_mi', 'Snow Making_ac'], [0.2, 4])
```

```
0.0
```

No difference whatsoever. Although the longest run feature was used in the linear model, the random forest model (the one we chose because of its better performance) only has longest run way down in the feature importance list.

Pricing recommendation

Assuming that adult ticket is currently priced at $81 and average visitors ski for 5 days and 350,000 visitors in the coming season, the model predicts $19635175 increase in revenue from raising the ticket price by $11.22 by adding a run, increasing the vertical drop by 150 feet, and installing an additional chair lift.

Conclusion

The model does not take into account the operational costs and expenditure for adding increasing the vertical drop by 150 feet, and installing an additional chair lift. We only know that additional chair lift would increase the operating cost by $1.54 million. The model says closing one run makes no difference. Closing 2-3 will reduce the ticket price and revenue. If they close 3, then they can close 4 and 5 runs as there will no loss on ticket pricing. The significant drop will be if 6 runs will be closed.

Future scope of work.

After Calculating Expected Big Mountain Ticket Price from The Model, Our model suggests that the modelled price should be *$93.93, $94* if we round it. Even with the expected mean absolute error of $10.26, this suggests there is room for an increase. Meaning that we could increase the price for $10-15, which will support with expenditure for maintenance, operation and other costs. Also, importantly could help us to improve the facility.

Increased price will attract less people, but those who value the quality and better facility. This makes easier to maintain the resort and improve the services of the facility.

Consideration:

Would be great if we could survey existing and potential customers regarding the price change for making the resort better and adding more features to it.