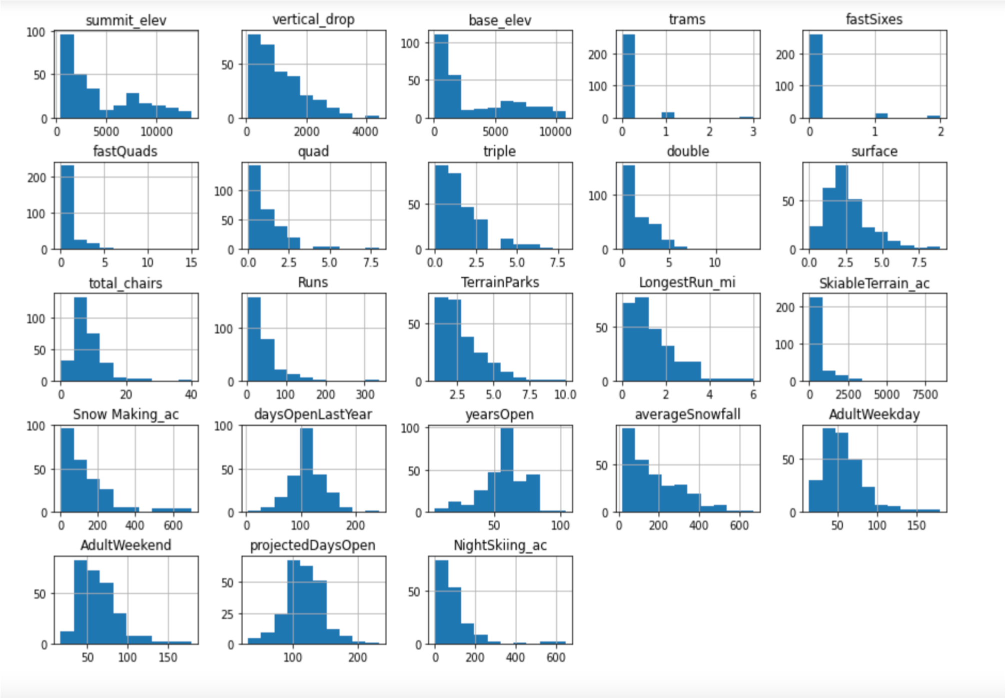
Guided Capstone Project Report

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. There were also some obvious issues with some of the other features in the data that, for example, led to one column being completely dropped, a data error corrected, and some other rows dropped. In the training section we used cross-validation and the random forest with GridSearch algorithm to identify the best algorithm and best parameters. At the end, we took our model for ski resort ticket price and leveraged it to gain some insights into what price Big Mountain's facilities might actually support as well as explore the sensitivity of changes to various resort parameters.

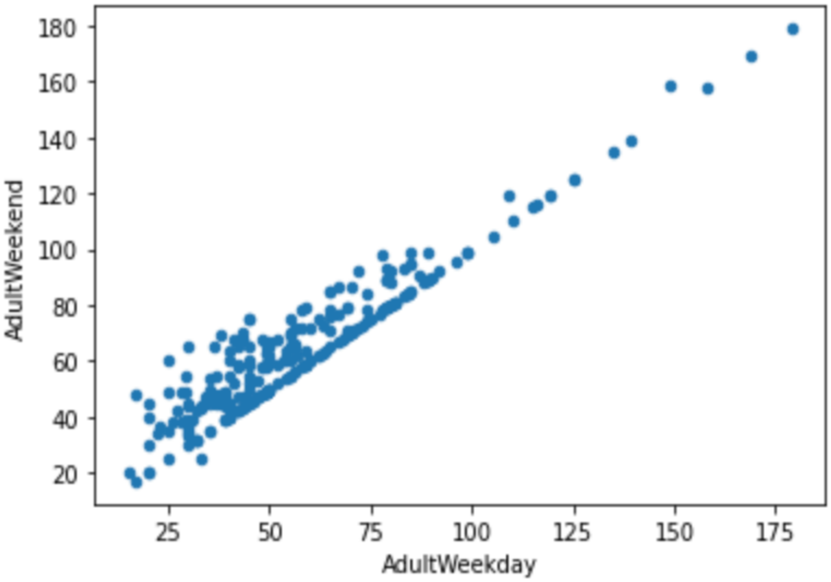
The data we started with contained 330 entries from 0 to 329 and 27 columns, but with a number of missing values that led to several rows being dropped completely. Unfortunately we had also missed quite a few of our desired target quantity, the ticket price AdultWeekday and AdultWeekend, which is missing 15-16% of values.

Data wrangling

We dropped all rows with no price data an review the important distributions using histograms:



By the end of this capstone we used ski_data's plot() method to create a scatter plot with AdultWeekday on the x-axis and AdultWeekend on the y-axis.



We can observe there is a clear line where weekend and weekday prices are equal. Weekend prices being higher than weekday prices seem restricted to sub $100 resorts.

Exploratory Data Analysis

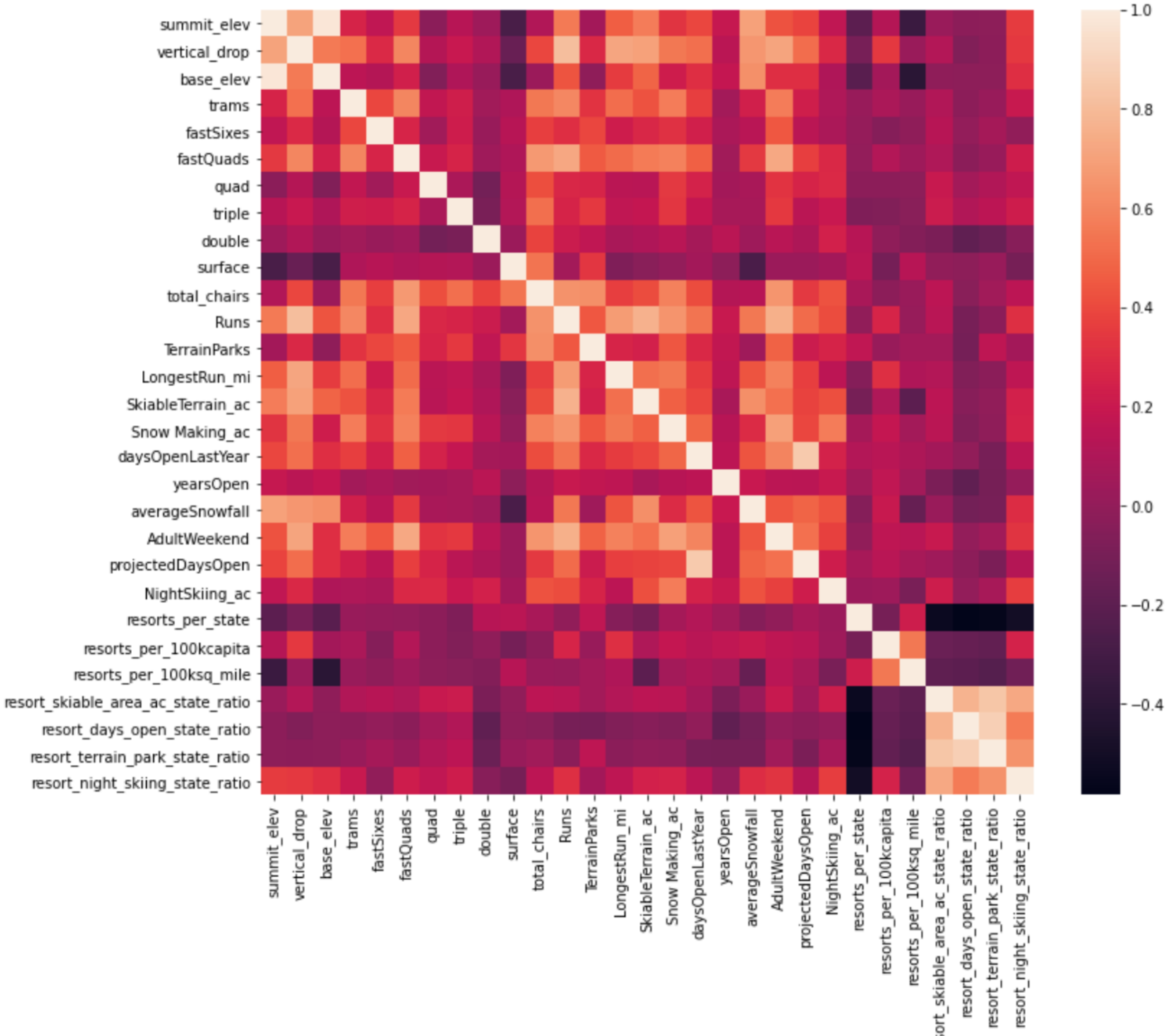
At this capstone, we had a firm idea of what our data science problem is and had the data we believe could help solve it. The business problem was a general one of modeling resort revenue. The data you started with contained some ticket price values, but with a number of missing values that led to several rows being dropped completely.

In this section we focussed on exploring many features in turn and found various trends and also explored how the original features contribute to these derived features.

The basic steps in this process are:

1. scale the data (important here because our features are heterogenous)
2. fit the PCA transformation (learn the transformation from the data)
3. apply the transformation to the data to create the derived features
4. (optionally) use the derived features to look for patterns in the data and explore the coefficients

At the end we came up with a feature correlation heatmap and scatter plot, which were a great way to gain a high level view of relationships amongst the features.



Pre-Processing and Training Data

In this notebook we started to build machine learning models. Our first model is a baseline performance comparator for any subsequent model. We then built up the process of efficiently and robustly creating and assessing models against it. We partition the data into training and testing splits.

First, we wanted to see how good the mean is as a predictor (estimated ticket price by simply using a known average), how closely does this match, or explain, the actual values? There are many ways of assessing how good one set of values agrees with another, which brought us to the subject of metrics such as R-squared or coefficient of determination, mean absolute errors and mean squared error.

We defined the pipeline to assess performance using cross-validation. which will perform the fitting as part of the process. This uses the default settings for the random forest so we then proceed to investigate some different hyperparameters.

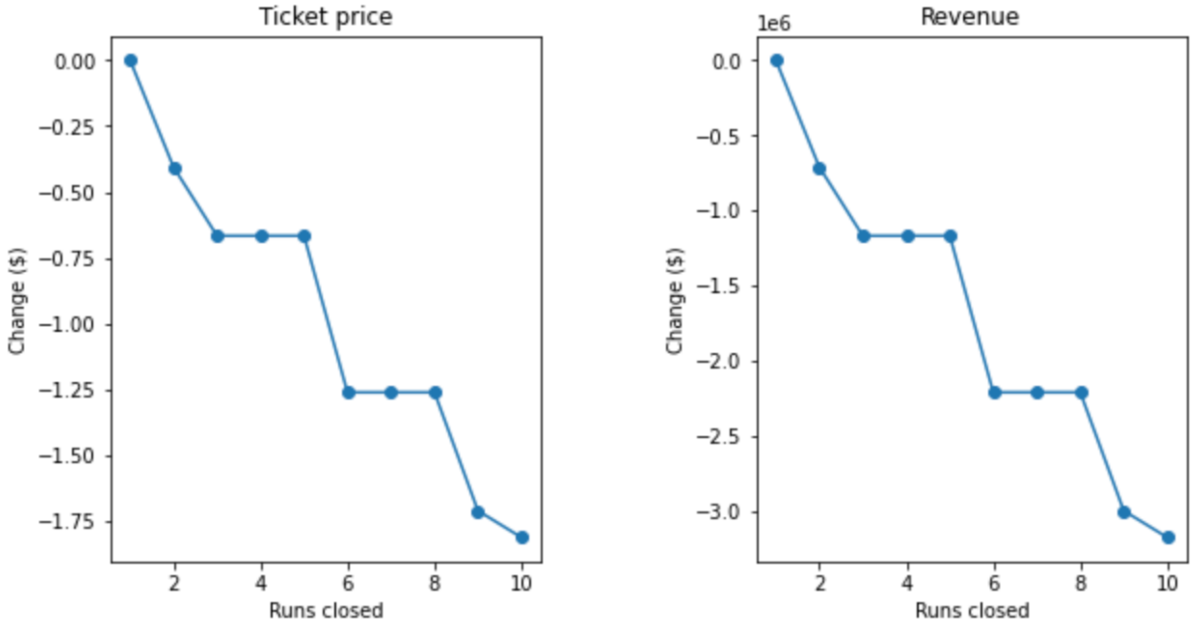
After many comparason and testing the models we conclued that the random forest model has a lower cross-validation mean absolute error by almost $1. It also exhibits less variability. Verifying performance on the test set produces performance consistent with the cross-validation results.

Modeling

In this part of capstone, we took our model for ski resort ticket price and leveraged it to gain some insights into what price Big Mountain's facilities might actually support as well as explore the sensitivity of changes to various resort parameters. That this relies on the implicit assumption that all other resorts are largely setting prices based on how much people value certain facilities. Essentially this assumes prices are set by a free market.
We used our model to gain insight into what Big Mountain's ideal ticket price could be, and how that might change under various 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's 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 $1.99
Over the season, this could be expected to amount to $3474638

Scenario 3

In this scenario, we are repeating the previous one but adding 2 acres of snow making.
This scenario increases support for ticket price by $1.99
Over the season, this could be expected to amount to $3474638

Such a small increase in the snow making area makes no difference!

Scenario 4

This scenario calls for increasing the longest run by .2 miles and guaranteeing its snow coverage by adding 4 acres of snow making capability.

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 the longest run way down in the feature importance list.