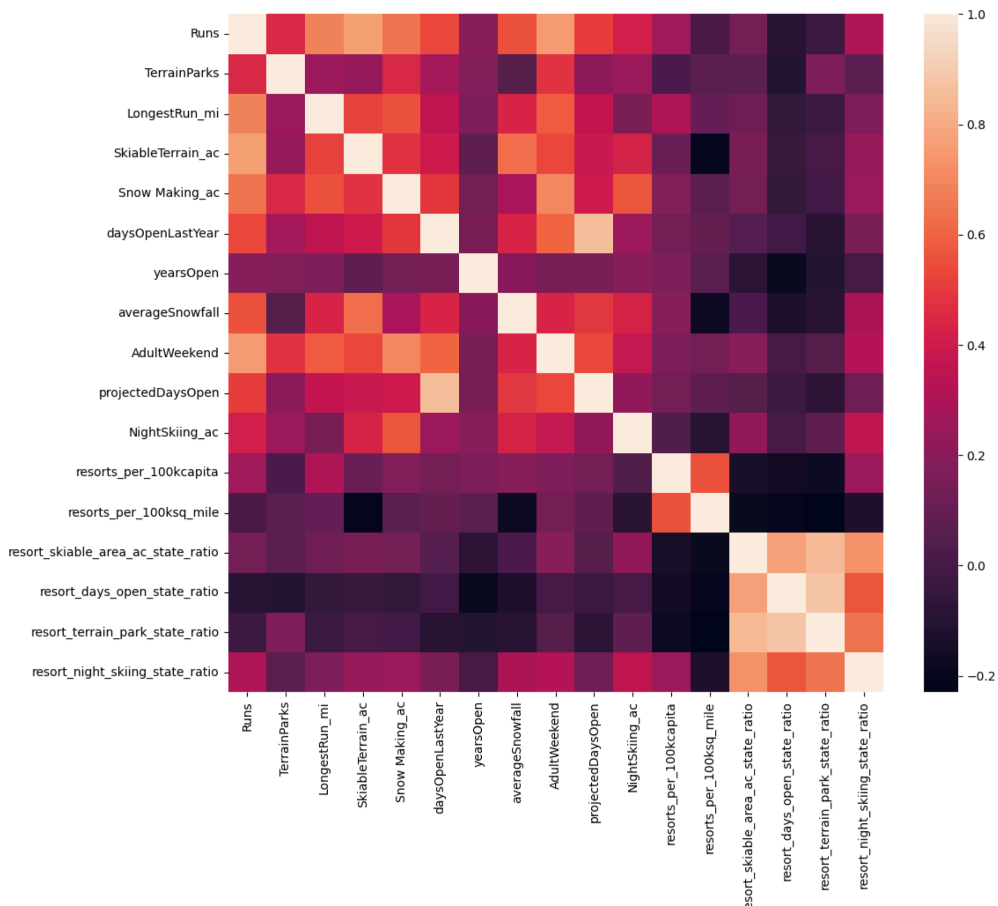
Big Mountain Report

Addressing the problem of Big Mountain trying to find a solution that increases revenue by increasing ticket prices, and reducing operational costs without compromising much. This report details a project focusing on optimizing ski resort ticket pricing using data analysis. The process began with rigorous data cleaning, including outlier removal and missing value treatment, to prepare a dataset fixated on ski resort attributes like location, size, and amenities.

The next phase involved Exploratory Data Analysis (EDA), using visualization tools such as bar graphs, datatables, and heatmaps to identify factors impacting resort pricing. This step was crucial as it gave a visualization of a base understanding of the data's underlying patterns and correlations. Here is the heatmap that was generated. The lighter the square, the more correlated it is.



Following EDA, the project moved into preprocessing and feature engineering. This stage involved transforming raw data into a model-friendly format, including normalizing data and encoding categorical variables. We tested two models: A linear Regression model, and a Random Forest Model. We first created a pipeline that filled in missing data with either the median or mean of the column, scaled data into units size in correspondence to a more accurate unit size. To prevent overfitting and to use the best features, we used a function called Grid_search_V, which would cross validate multiple versions of our dataset that would test up to k times of how many features we should include in our model, and which ones were the best. Various machine learning models were then explored and evaluated using metrics like Mean Absolute Error (MAE) and R-squared.After all of this, we ended up choosing the Random Forest Regressor, which provided more accurate predictions.

4.11.1 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
```

```
(10.499032338015294, 1.6220608976799658)
```

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

```
11.793465668669324
```

4.11.2 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
```

```
(9.644639167595688, 1.3528565172191818)
```

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

```
9.537730050637332
```

After finding the model, when it came to predicting the ticket price, it recommended a price of $95, with a margin of $10 to improve. We also created a function, where we can predict prices while making any hypothetical changes we want to make to our resort. After testing our function, we can conclude that we can reduce 1-5 runs in facilities to only lose $.75 in pricing, while increasing our ticket price to $95 to increase revenue. After this, I hope to move forward in other work. If you would like to test this yourself, you can just run this function to test any changes you are unsure would be beneficial to the resort.

```
Big Mountain Resort modelled price is $95.87, actual price is $81.00.
Even with the expected mean absolute error of $10.39, this suggests there is room for an increase.
```