**Q: 1** Write a summary of the work in this notebook. Capture the fact that you gained a baseline idea of performance by simply taking the average price and how well that did. Then highlight that you built a linear model and the features that found. Comment on the estimate of its performance from cross-validation and whether its performance on the test split was consistent with this estimate. Also highlight that a random forest regressor was tried, what preprocessing steps were found to be best, and again what its estimated performance via cross-validation was and whether its performance on the test set was consistent with that. State which model you have decided to use going forwards and why. This summary should provide a quick overview for someone wanting to know quickly why the given model was chosen for the next part of the business problem to help guide important business decisions.

* Load Data
* Remove our mountain from dataset
* Remaining data was broken into test (30%) and train (70%) data
  + The Y sets include only the AdultWeekend column (ticket price), which is the variable we are trying to predict
  + The X sets include all other columns which we are using to predict the ticket price
* The columns for Name, state and Region were removed from the X sets as they are not useful in predicting ticket price (no correlation between name and price, etc.). They also are non-numeric values while all remaining columns are numeric
* Scikit learns’ DummyRegressor was used to create a predictor that always outputs the mean of the Y training data. This is used as a baseline comparison against future models to confirm they are better than less complex methods
* We reviewed R^2 and determined that the R^2 value going from our y\_train to y\_test was a negative value meaning the Dummy model did a worse job of limiting variance from the mean on the test set than it did on the training set. This is understandable as it was based on the training sets mean value not the test sets. The R^2 metric is good for expressing the amount of variance from the mean that the model reduced, but it does not express whether the prediction was accurate or not.
* The Mean Absolute Error (MAE) amd the Mean Squared Error (MSE) were reviewed as metrics to express how well the model predicts the dependent variable vs. the actual values. Each looks at the average difference between the observed and predicted values with with one squaring the values and the other taking the absolute value of them.
* Scikit learn has built in functions for each of these metrics and others. It is important to properly enter the function arguments to get the proper output.
* We filled in missing data in the training and test X sets with first the median and then the mean. The predicted values were not substantially different between the two methods, though that could change as we eliminate columns from future models
* Scikit learn’s methods can be piped together to make the trial and error process of model building move faster. We piped together SimpleImputer (fills missing values), StandardScaler (centers data around zero or mean and scales it to unit variance which some algorithms require), and LinearRegression (fits and predicts a least squares linear regression).
* We created a new pipe that added scikit learns SelectKBest. This method selects the top K values based upon an inputted scoring function that determines how correlated each variable is with the predicted variable. We used f\_regression which is simply linear regression. We ran predications and reviewed performance metrics for 10 15 k values and compared the results of performance on our test set. This is not advised as it can lead to overfitting.
* We explored cross\_validate which allows us to compare the effectiveness of different parameters of the model to determine which k’s are best to use and how many. Cross\_validate splits the training data into as many sections as there are parameters. Trains it self on all but one of the sections and uses the last as a test. This allows the us to not use the actual test set to refine our model as this would lead to overfitting.
* GridSearchCV was used on our pipe to determine the best value of k to use. This method iterates through each value of k and records the metrics into a dictionary. It was found that 8 was the best value for k as it has the highest R^2 value as well as the lowest variance. Note that this is based on only comparing against the sectioned training data.
* Based on the metrics stored by GridSearchCV the following are the 8 variables that best predict the ticket price, some being negatively and some being positively correlated:
  + vertical\_drop 10.767857
  + trams -4.142024
  + total\_chairs 5.745626
  + fastQuads 5.794156
  + Snow Making\_ac 5.370555
  + SkiableTerrain\_ac 0.181814
  + Runs -5.249780
  + LongestRun\_mi 6.290074
* They note that the skiable terrain should be negative, why is mine not? Looked back thru but didn’t see anything that would have caused this. Is this a product of the random sectioning being slightly different?
* We created a RandomForestRegressor to give a comparison to our linear regression model. This model performed slightly better than our linear regression model.
* We used the randomforest model to compare the following options:
  + Number of trees?
  + Scaling vs. not
  + Median vs. mean fill
* Based on the Gridsearch of the randomforest model we found that 69 trees, the median and not scaling were the best parameters to use.
* The top four variables used to train the RandomForest model match the top four variables to train the Liner Regression model, though in a different order. This is hopeful as it further validates those variables as important features to base our final model on.
* We ran a final comparison of the two models and based on the Mean Absolute Error as our deciding metric the Random Forest model performed better and is what we will move forward with. We did a check using learning\_curve to determine if additional data would increase the accuracy of our model. It was found that the current amount of data is sufficient as we see limited returns after adding in data about more than 50 mountains.