Guided Capstone Project Report

1. ***Objectif of the project***

The Big Mountain Resort faced a high operating cost due to the installation of an additional chair lift to help increase visitors' distribution across the mountain.

Our data science project aims to identify the causes of the dysfunction and propose solutions to mitigate the high operating cost.

Our model will predict the AdultWeekend price: The cost of an adult weekend chairlift ticket. The management could forecast the year's annual revenue, knowing the predicted rates if they follow our recommendation.

1. ***The Dataset***

This project's data set is an aggregation of information about different companies in skiing industries across the United States of America. The data has three hundred thirty (330) rows and twenty-seven (27) columns.

*# determine the size of the dataframes*

df. Shape

(330, 27)

As we said in the first part, the AdultWeekend price is our response variable. The linear regression model that we will implement aims the predict that using the independent variables.

1. ***Data preprocessing***

The following features have missing values

1 fastEight 164 non-null float64

2 Runs 326 non-null float64

3 TerrainParks 279 non-null float64

4 LongestRun\_mi 325 non-null float64

5 SkiableTerrain\_ac 327 non-null float64

6 Snow Making\_ac 284 non-null float64

7 daysOpenLastYear 279 non-null float64

8 yearsOpen 329 non-null float64

9 averageSnowfall 316 non-null float64

10 AdultWeekday 276 non-null float64

11 AdultWeekend 279 non-null float64

12 projectedDaysOpen 283 non-null float64

13 NightSkiing\_ac 187 non-null float64

We filled out the missing value with 0 for variables where NaN means an absence of values. However, we use the mean to fill out the missing values in the variable ***AdultWeekday, AdultWeekend***

df.fastEight.fillna(0, inplace= **True**)

df['AdultWeekday'].fillna((df['AdultWeekday'].mean()), inplace=**True**)

We examine the relationship between the response variable AdultWeekend and the other features. To do so, we use Sklearn Mutual\_Info\_regression to display the importance of each variable in predicting the response feature.

**from** **sklearn.feature\_selection** **import** mutual\_info\_regression

MI = mutual\_info\_regression( X\_train, y\_train)

MI = pd.Series(MI)

MI.index= X\_train. columns

MI.sort\_values(ascending=**False** )

A close up of a logo

Description automatically generated

Using the Sklearn SelectKBest module, we select below the fifteen most critical independent variables of our model.

**from** **sklearn.feature\_selection** **import** SelectKBest

sel\_= SelectKBest(mutual\_info\_regression, k=15).fit(X\_train, y\_train)

X\_train. columns[sel\_.get\_support()]

Features Coefficient

AdultWeekday **1.270598**

vertical\_drop  **0.263500**

Runs **0.227629**

SkiableTerrain\_ac **0.152941**

TerrainParks **0.146766**

clusters **0.141109**

Snow Making\_ac **0.120680**

summit\_elev **0.119183**

daysOpenLastYear **0.100439**

projectedDaysOpen **0.083342**

Vermont **0.070889**

yearsOpen **0.066736**

fastQuads **0.063204**

total\_chairs **0.062888**

LongestRun\_mi **0.0569**

1. ***Evaluation of the model***

We choose two metrics to evaluate the model's performance: The Explained Variance Scores and the Mean Absolute Error.

The  **Explained variance** (also called explained variation) is used to measure the discrepancy between a model and actual data. In other words, it's the part of the model's total [variance](https://www.statisticshowto.com/probability-and-statistics/variance/)that is explained by factors that are present and isn't due to [error variance](https://www.statisticshowto.com/residual-variance/).

**from** **sklearn.metrics** **import** explained\_variance\_score, mean\_absolute\_error

print ("Explained\_variance\_score:", explained\_variance\_score(y\_test, y\_pred))

Explained\_variance\_score: **0.753255510586298**

Our model is able to explain 75.32% of the variation in AldultWeekend price.