EXPEDIA HOTEL-CLICK PREDICTIONS

Team OneHotEncoder

DISCUSSION OUTLINE

- Business Problem
- Methodology
  - Data Exploration
  - Data Cleaning
  - Data Preparation
- Modeling
- Model Evaluation
- Deployment

MEET OUR TEAM

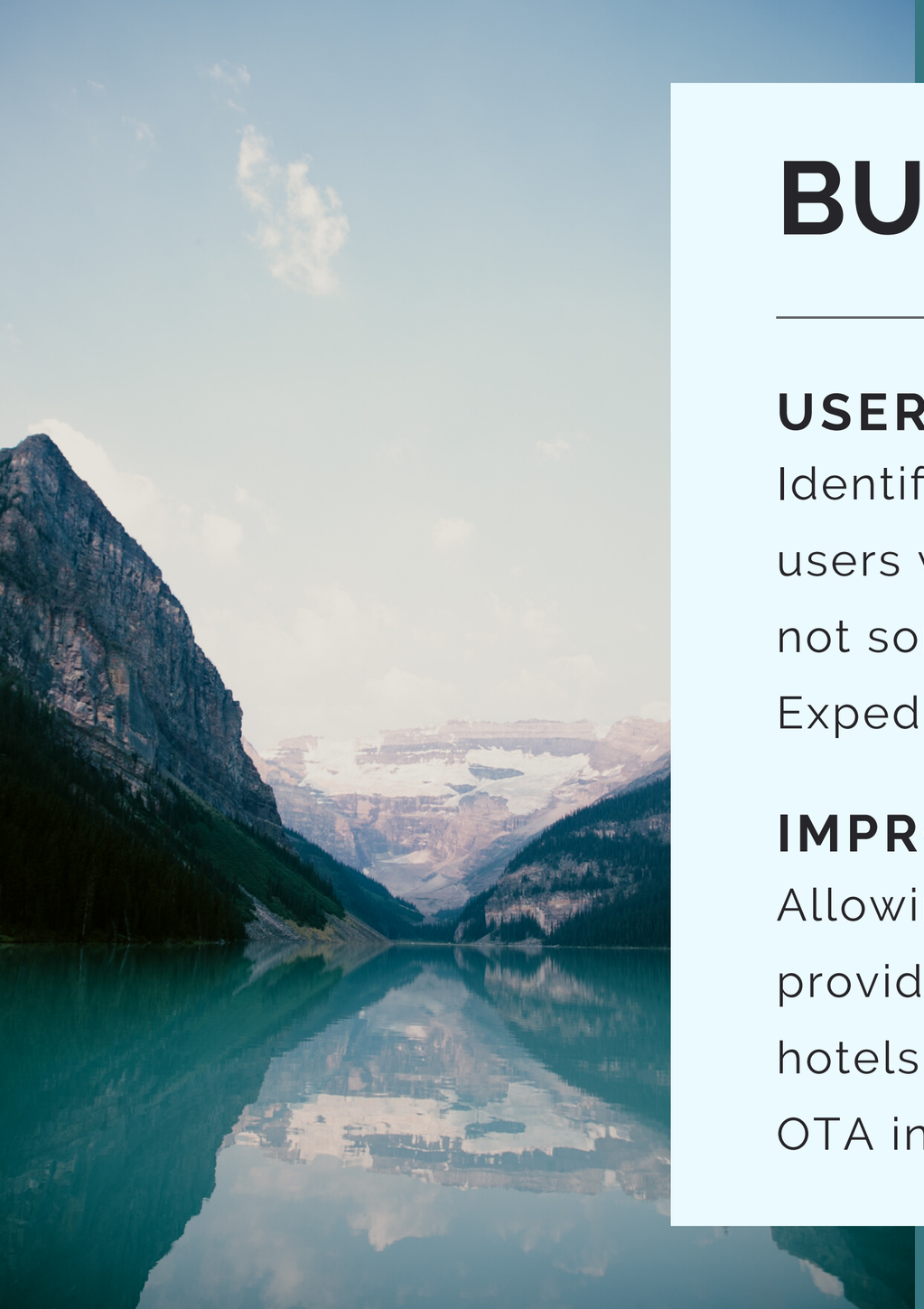
MSBA's Travel Experts

NEHA BANSAL

NIRJA MISTRY

CASSIO SALGE

GRACE ZENG



BUSINESS PROBLEM

USER PREFERENCES FOR HOTELS

Identifying the most important features according to users when choosing hotels and predicting whether or not someone will click on a particular hotel listed on Expedia

IMPROVED ENGAGEMENT WITH CUSTOMERS

Allowing Expedia to understand customer needs and provide the most competitive rates and selection of hotels, thereby, improving sales and market share in the OTA industry

THE DATA

Kaggle Dataset

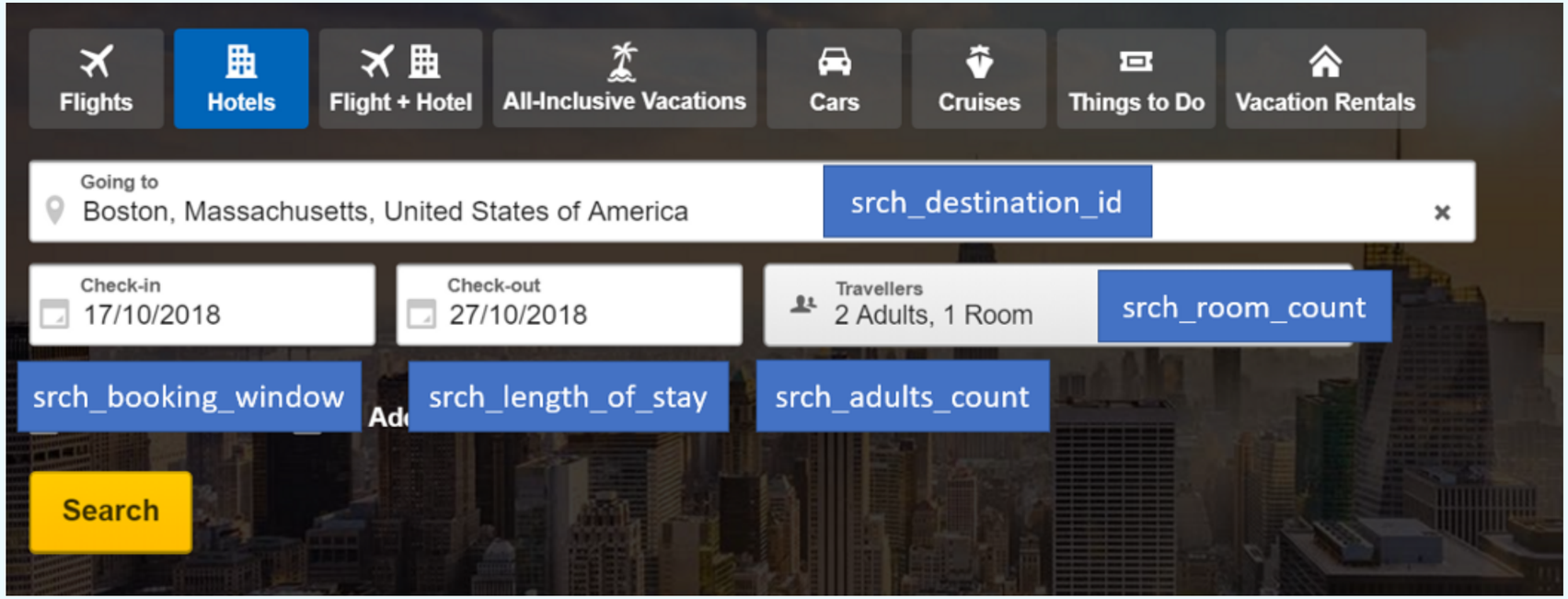
Training set ~10mi rows

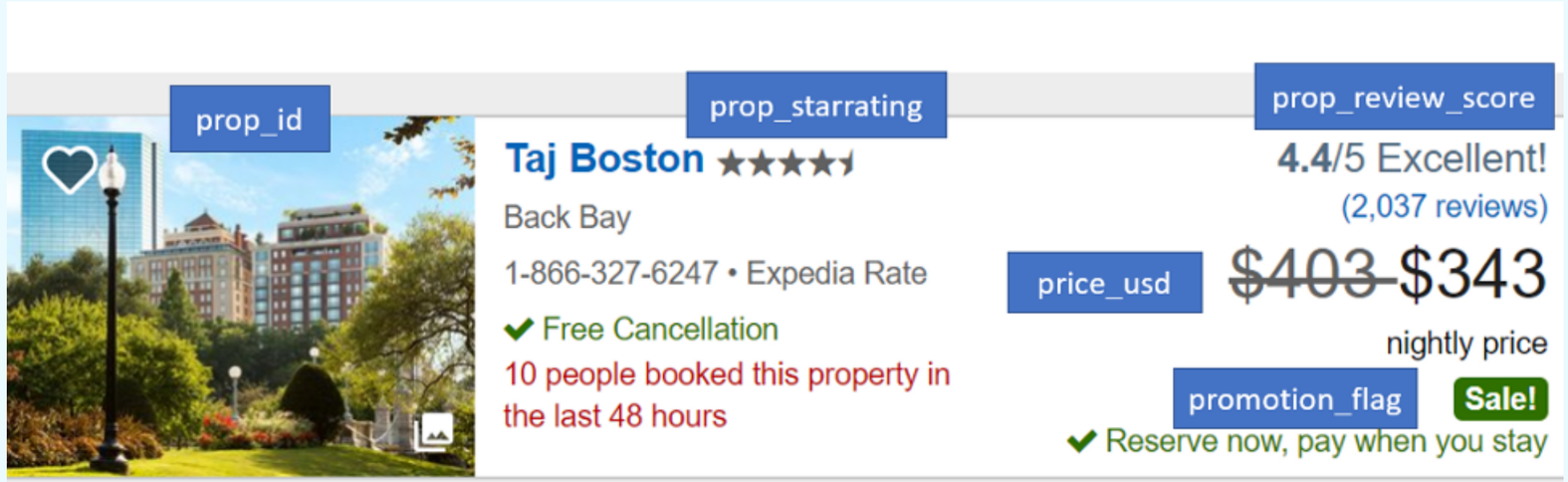
Test set ~6mi rows

53 columns of hotel attributes as feature variables

PREDICT

whether someone will click on a hotel or not





Summary Statistics

Class Distribution

Feature Distribution

Multicollinearity amongst variables

Feature Engineering

Click-through rate
Hotel Popularity
Competitor

Class Imbalance
Imputation
Remove NA

Models

Logistic Regression
Neural Net
Support Vector
Random Forest
XGBoost

Prediction

Whether or not someone will click on a hotel

F1-Score
AUC

Evaluation

Most informative attributes

DATA EXPLORATION

Summary Statistics

- price outlier

Distributions for each feature

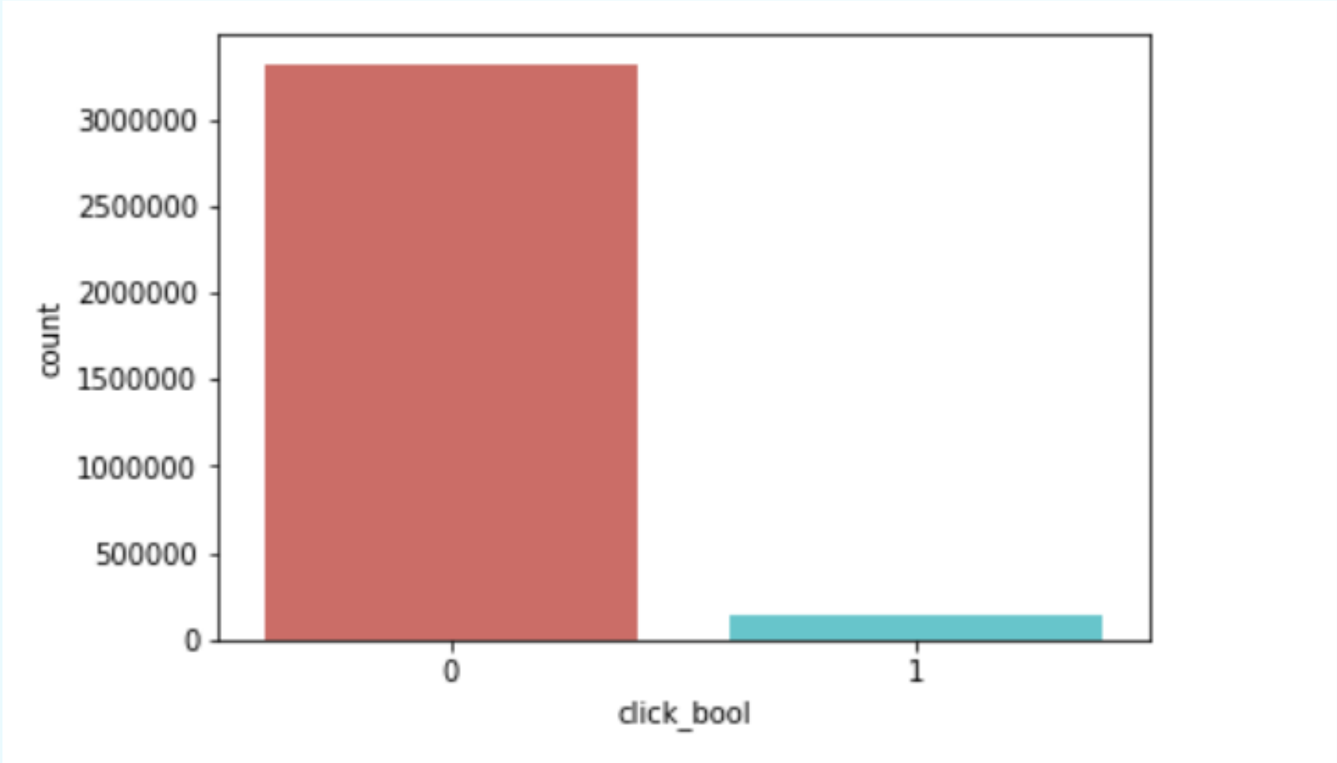
- class imbalance

Missing Variables

- some 90% missing data some 30% missing data

Multicollinearity

- heatmap to view coliniearity



DATA CLEANING

Dealing with missing values

- Remove columns with > 90% missing data
- Imputation for columns ~30% missing data

Price Range for Hotels

- Find the outliers
- Replace with upper bound of 99 percentile

Downsampling of the Data

- Split into dataframes based on class
- Match sample sizes

FEATURE ENGINEERING AND SELECTION

CLICK-THROUGH RATE

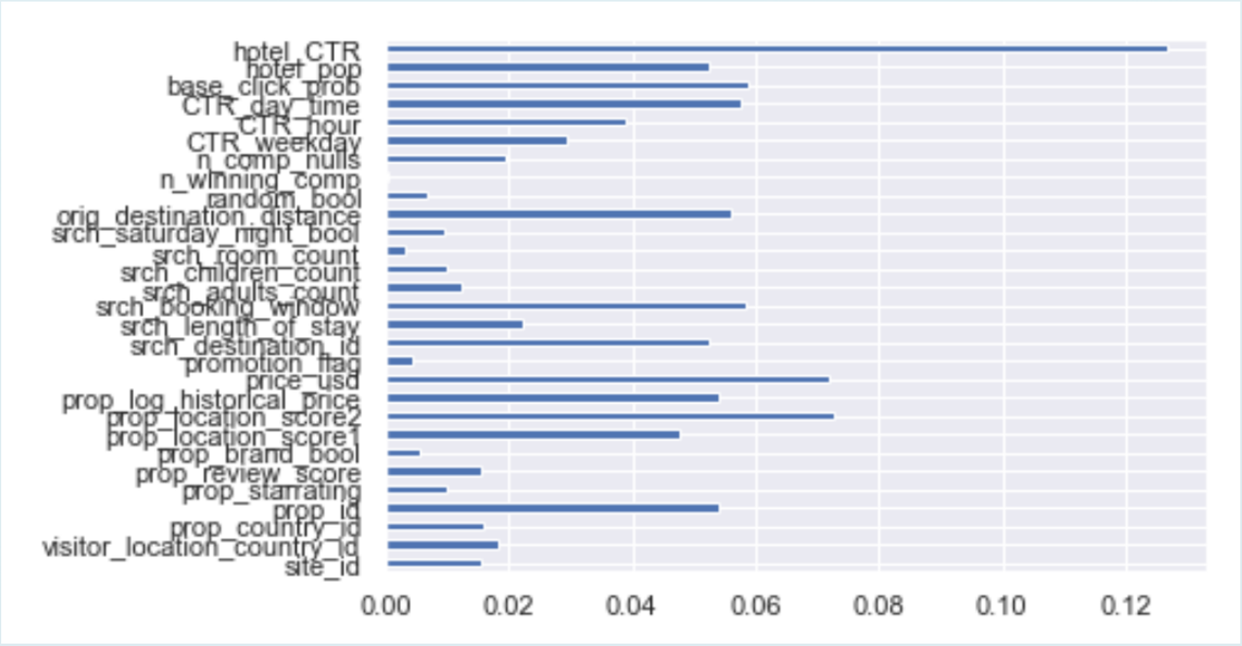
COMPETITOR

BASE CLICK PROBABILITY

HOTEL POPULARITY

MEASURES FOR FEATURE SELECTION

- Pearson Correlation
- VIF
- Lasso Regression



MODELING

5 models to see which yields the best results

Forward Selection

Hyperparameter Tuning - GridSearch

Stratified k-Fold crossvalidation

Choose the best Model

MODELING

- Logistic Regression
- **XGBoost**
- Neural Network
- Random Forest
- Support Vector Machine

MODEL EVALUATION

F1-Score

Area Under the Curve (AUC)

Model	Motivation	F-1 Score	AUC
Logistic Regression	Simple model that can handle nonlinear decision boundaries	0.62	0.65
Neural Network	Capture non-linearity within dataset	N/A	0.50
Support Vector Machine	Works well with large datasets but requires a lot of computational power	0.68	0.61
Random Forest	Aggregating decision trees to limit overfitting	0.71	0.69
XGBoost	Performs really well but can be difficult to tune	0.74	0.79

RECOMMENDATIONS

3 main features that stood out

Hotel Quality

Hotel Popularity

Hotel Price

MODEL DEPLOYMENT

IMPACT

- Engage with hotels that have high CTR and promote them on the platform
- Personalize experience for independent v/s big brand hotels

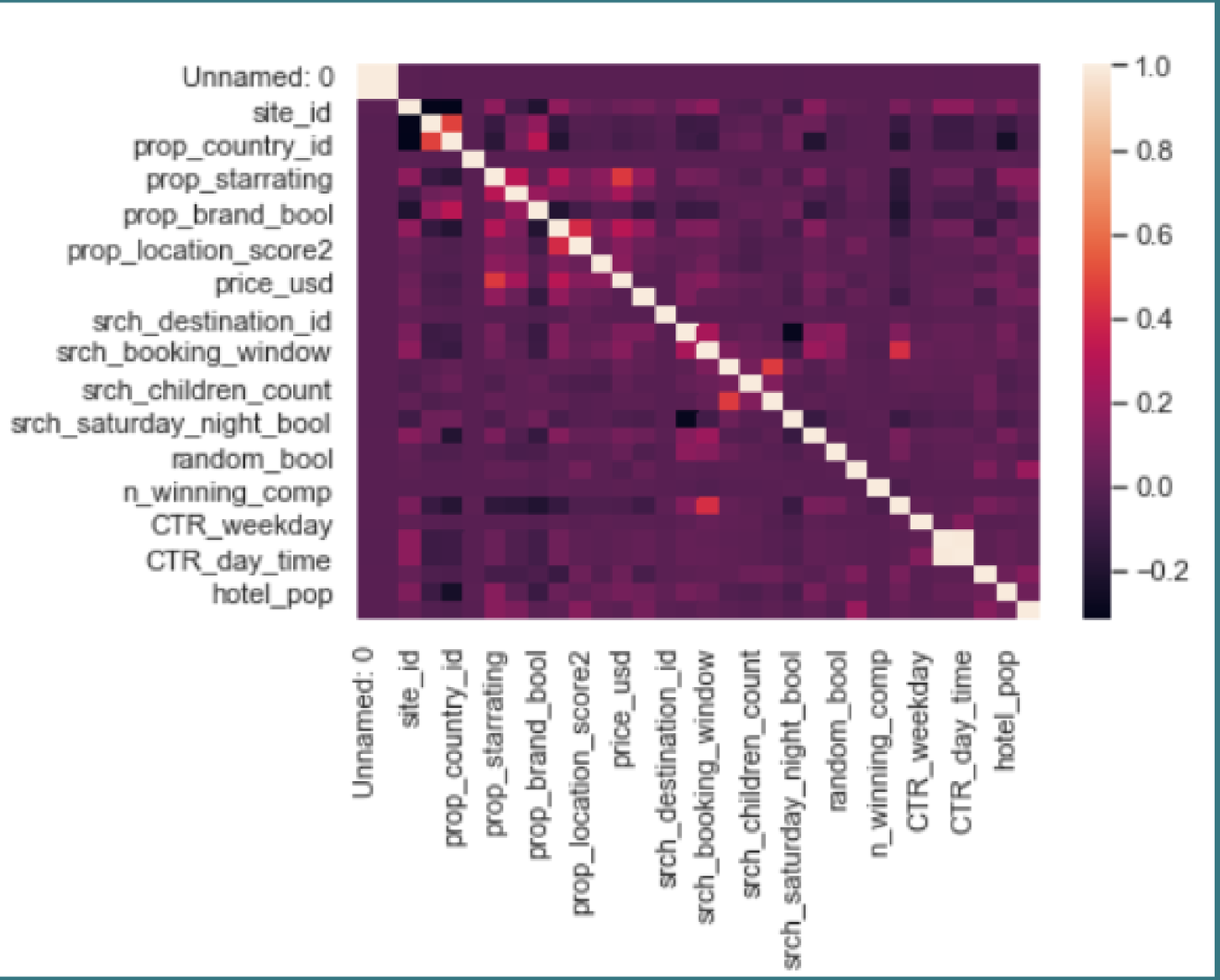
IMPROVEMENTS

- Revenue per click using booking data and a cost-benefit matrix


QUESTIONS?

APPENDIX

HEAT MAP FOR MULTICOLLINEARITY



FEATURES

comp1: comp1_rate * comp1_inv; it represents the competition result between Expedia and competitor 1. The value of -1 indicates the competitor 1 is winning, where competitor 1 has a lower price than Expeida's result and has availability of equivalent room type. Comp 2 to comp 8 are the corresponding columns for competitor 2 to 8.

n_winning_comp: the number of competitors winning in the competition. In another word, the number of value -1 among the columns of "comp1" to "comp8".

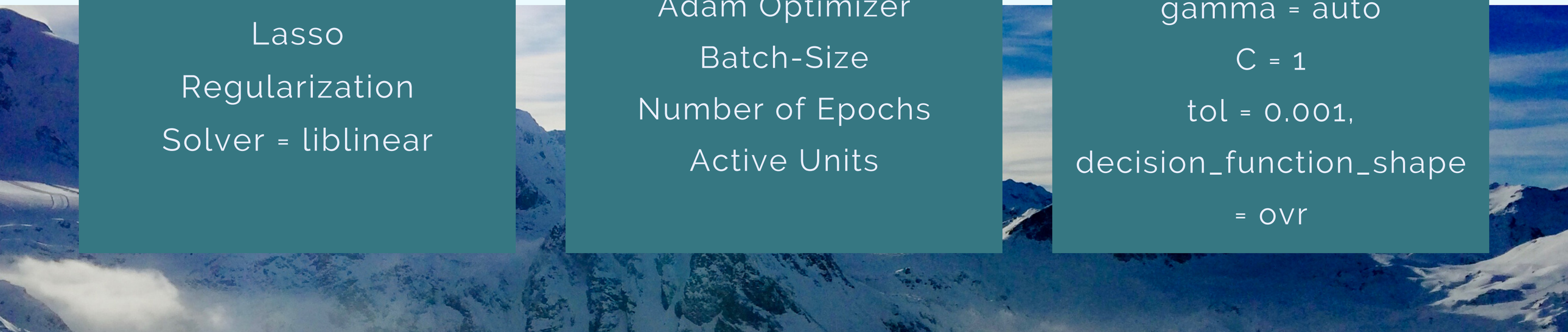
n_comp_nulls: the number of nulls among the columns of "comp1" to "comp8".

CTR_day_time: the product of CTR_weekday and CRT_hour. It refers to the clicking trend of a specific hour of a specific weekday.

base_click_prob: The base click probability of each result in a search. 1/srch_count. When there are only a few results in a search, the probability that a result gets clicked is higher.

hotel_pop: Total number of results of Hotel X within Country X / Total number of results within Country X

MODEL PARAMETERS



LOGISTIC REGRESSION

C=1

Lasso

Regularization

Solver = liblinear

NEURAL NET

3 hidden layers - relu, sigmoid

Adam Optimizer

Batch-Size

Number of Epochs

Active Units

SUPPORT VECTOR MACHINE

kernel = rbf

class_weight = balanced

gamma = auto

C = 1

tol = 0.001,

decision_function_shape = ovr

MODEL PARAMETERS

RANDOM FOREST

criterion = entropy

max_depth = 10

max_features = log2

n_estimators = 100

XGBOOST

objective = binary:logistic

booster = gbtree

eta = 0.1

max_depth = 5

subsample = 0.8

colsample_bytree = 0.8

eval_metric = logloss