ML Final Project - Hotel bookings

Jingxuan Xu, 804226

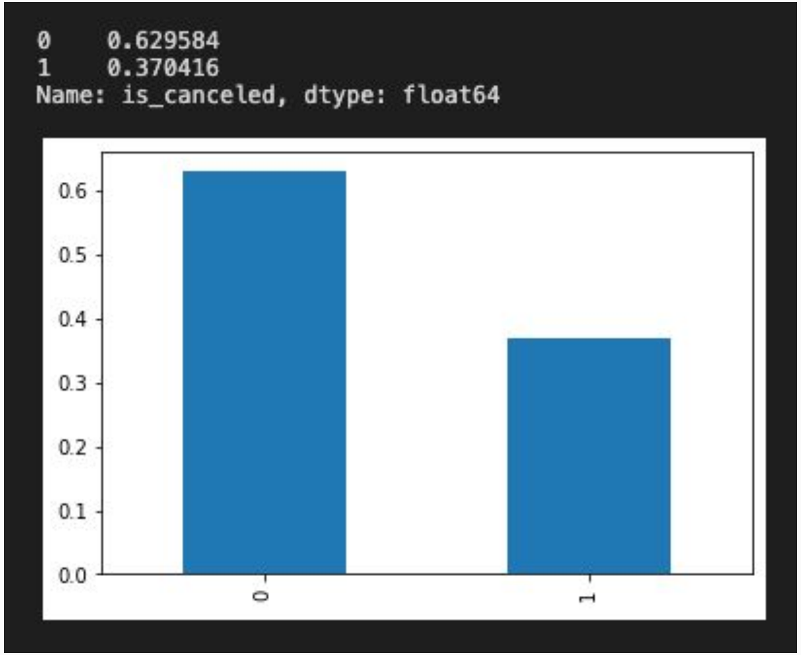
The Problem

- Predict whether a hotel booking will be canceled
- Using the Hotel bookings dataset from Kaggle containing booking information for a city hotel and a resort hotel
- Motivation: help hotel anticipate cancelation to maximize occupancy and thus profit

Evaluation

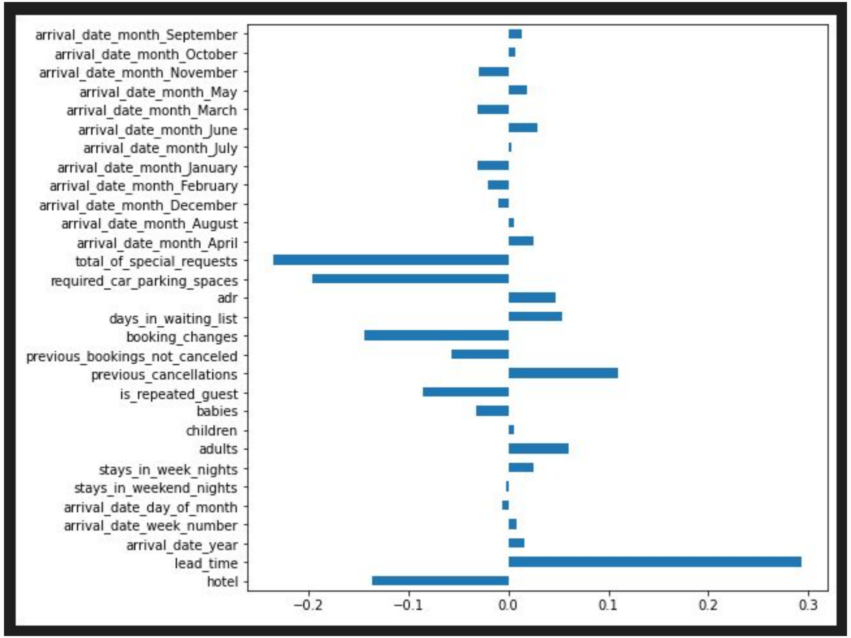
- As this is a classification problem that is pretty balanced in its labels, we use accuracy as our evaluation metric
- The data is randomly shuffled and then split into a training set of 10% and a testing set of 2%

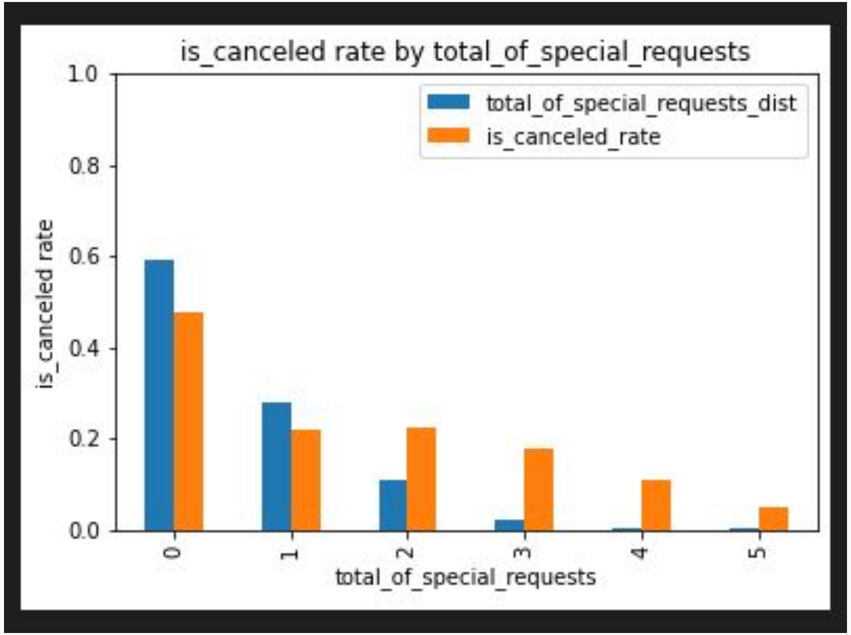
Data description



```
0    0.629584
1    0.370416
Name: is_canceled, dtype: float64
```

- 119,390 examples
- train: 11,939 examples (10%)
- test: 2,388 examples (2%)
- In total 30 features including is_canceled
- some features are missing values like the children feature
- Labels are pretty balanced, is_canceled for example 37% true vs 63% false (graph)





Left graph shows correlation of all features to is_canceled, for example the more total_of_special_requests the less likely a cancelation (right graph)

Data engineering

- Remove features that are not informative: country, agent, company, market_segment, distribution_channel, reserved_room_type, assigned_room_type, deposit_type, customer_type
- Added new features through categorical values to 1-hot (example arrival_date_month_April)
- Replaced missing values in children feature with the children feature median label (value)

Algorithm performance

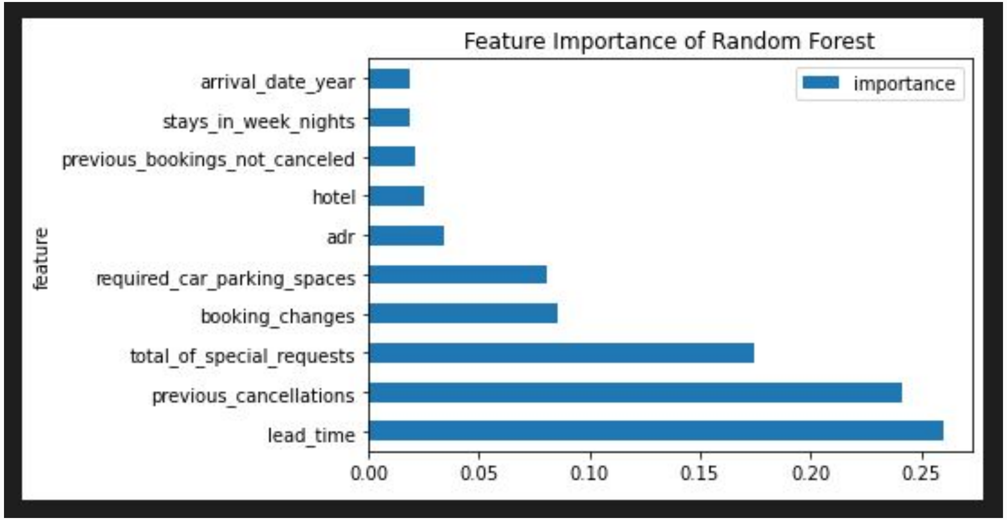
- *Classification Baseline on Testing: 63.2%*
- *Regression Baseline on Testing: MSE: 23.4%*

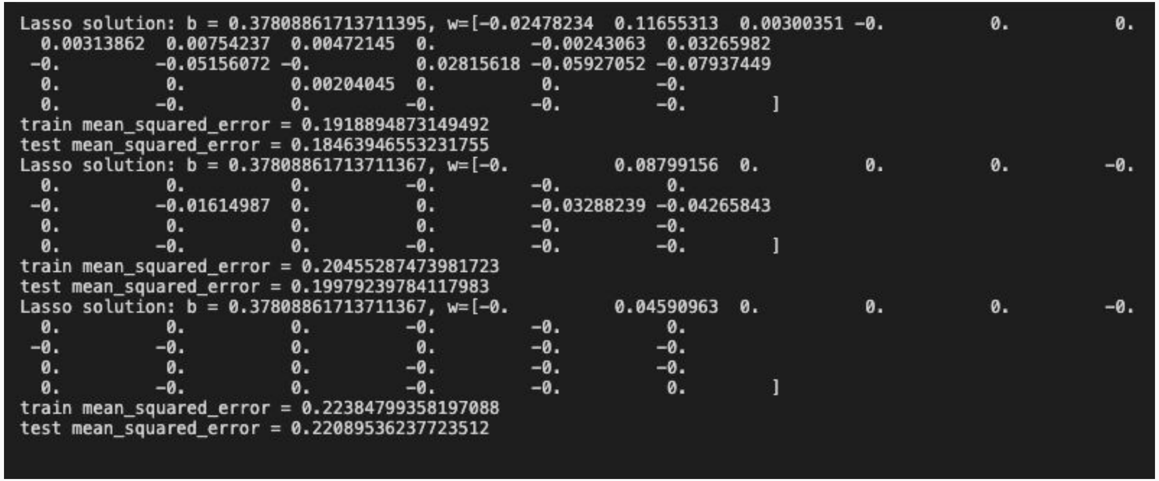
- KNN (k=2): 74.4%
- Scaled values KNN (k=2): 76.9%
- Decision Tree (max_depth=4): 75%
- Random Forest (max_depth=4): 74.8%
- **Ada Boost** (max_depth=8): 78.6%
- Lasso Regression (alpha=0.5): MSE: 18.5%

Algorithm introspection

1. What is the random forest feature importance?
2. What are the weights of the lasso coefficients?

   Measured for alpha=0.01, 0.05, 0.1

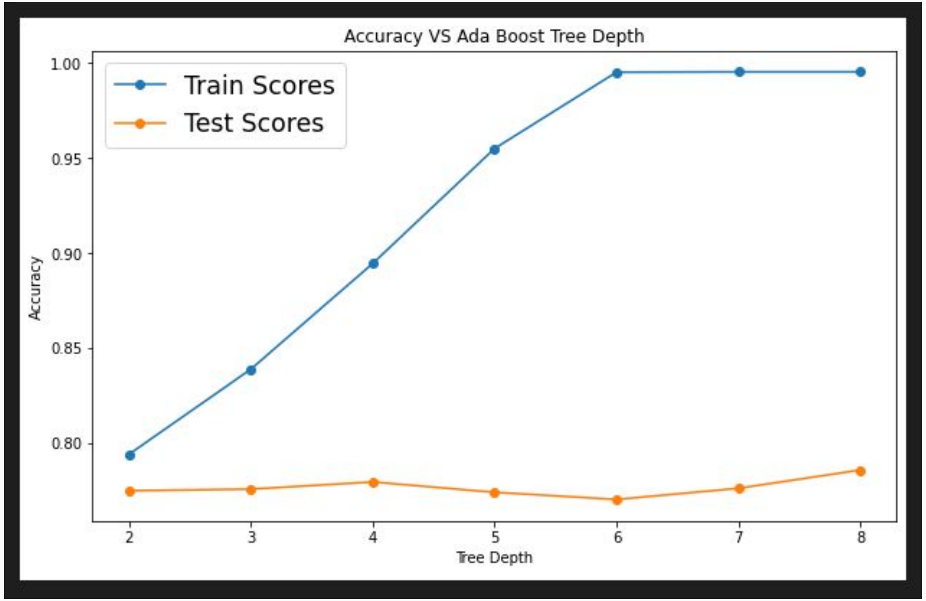




```
Lasso solution: b = 0.37808861713711395, w=[-0.02478234  0.11655313  0.00300351 -0.          0.          0.
  0.00313862  0.00754237  0.00472145  0.         -0.00243063  0.03265982
 -0.         -0.05156072 -0.          0.02815618 -0.05927052 -0.07937449
  0.          0.          0.00204045  0.          0.         -0.
  0.         -0.          0.         -0.         -0.         -0.        ]
train mean_squared_error = 0.1918894873149492
test mean_squared_error = 0.18463946553231755
Lasso solution: b = 0.37808861713711367, w=[-0.          0.08799156  0.          0.          0.         -0.
  0.          0.          0.         -0.         -0.          0.
 -0.         -0.01614987  0.          0.         -0.03288239 -0.04265843
  0.          0.          0.          0.         -0.         -0.
  0.         -0.          0.         -0.         -0.         -0.        ]
train mean_squared_error = 0.20455287473981723
test mean_squared_error = 0.19979239784117983
Lasso solution: b = 0.37808861713711367, w=[-0.          0.04590963  0.          0.          0.         -0.
  0.          0.          0.         -0.         -0.          0.
 -0.         -0.          0.          0.         -0.         -0.
  0.          0.          0.         -0.         -0.         -0.
  0.         -0.          0.         -0.         -0.          0.        ]
train mean_squared_error = 0.22384799358197088
test mean_squared_error = 0.22089536237723512
```

Hyperparameters

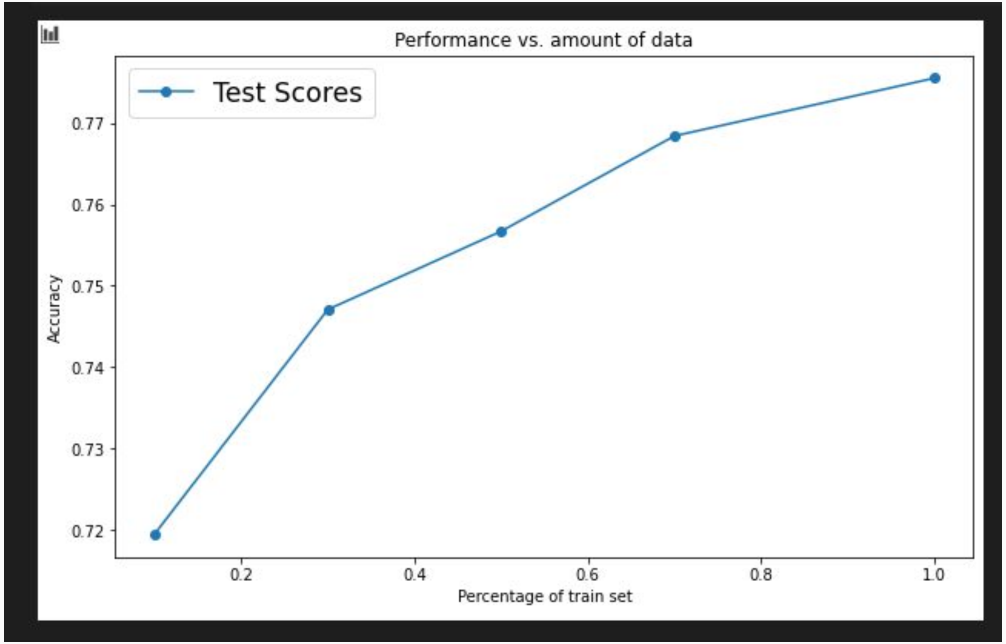
- hyper parameters of the Ada Boost
- Ada Boost is best performing algorithm
- Local maximum at depth=4
- Global maximum as far as we know at depth=8



Additional Analysis - Ada Boost

Performance vs. amount of data:

As maximum test set acc is only reached at 100% of the train set I recommend collecting more data


Performance vs. amount of data
Test Scores
Accuracy
Percentage of train set