Hotel Booking Cancellation Prediction

EAS 506

Bharatwaj Majji (50442312)
Jayanth Puthineedi (50442725)
Vishnu Bhadramraju (50441735)

University at Buffalo The State University of New York

Objective Statement

- Finding the minimalist model to determine hotel booking cancellation information from Hotel Booking Demand Data.
- Analyze outlier existence, feature importance and dimensionality into the model.
- Confirm most important features that determine the cancellation status of the hotel booking.

Data Set

- Hotel Booking Demand Data
- 31 Features: hotel, lead time, guests, company, car parking spaces, meal etc.
- 0.1 million records in the dataset.

Data Visualisation and Pre-Analysis

- Handled missing values
- Replaced missing values with average mean and most frequent values.
- Analysed pairwise association between highly correlated features.
- Combined multiple features to get a new feature.
- Removed redundant columns.

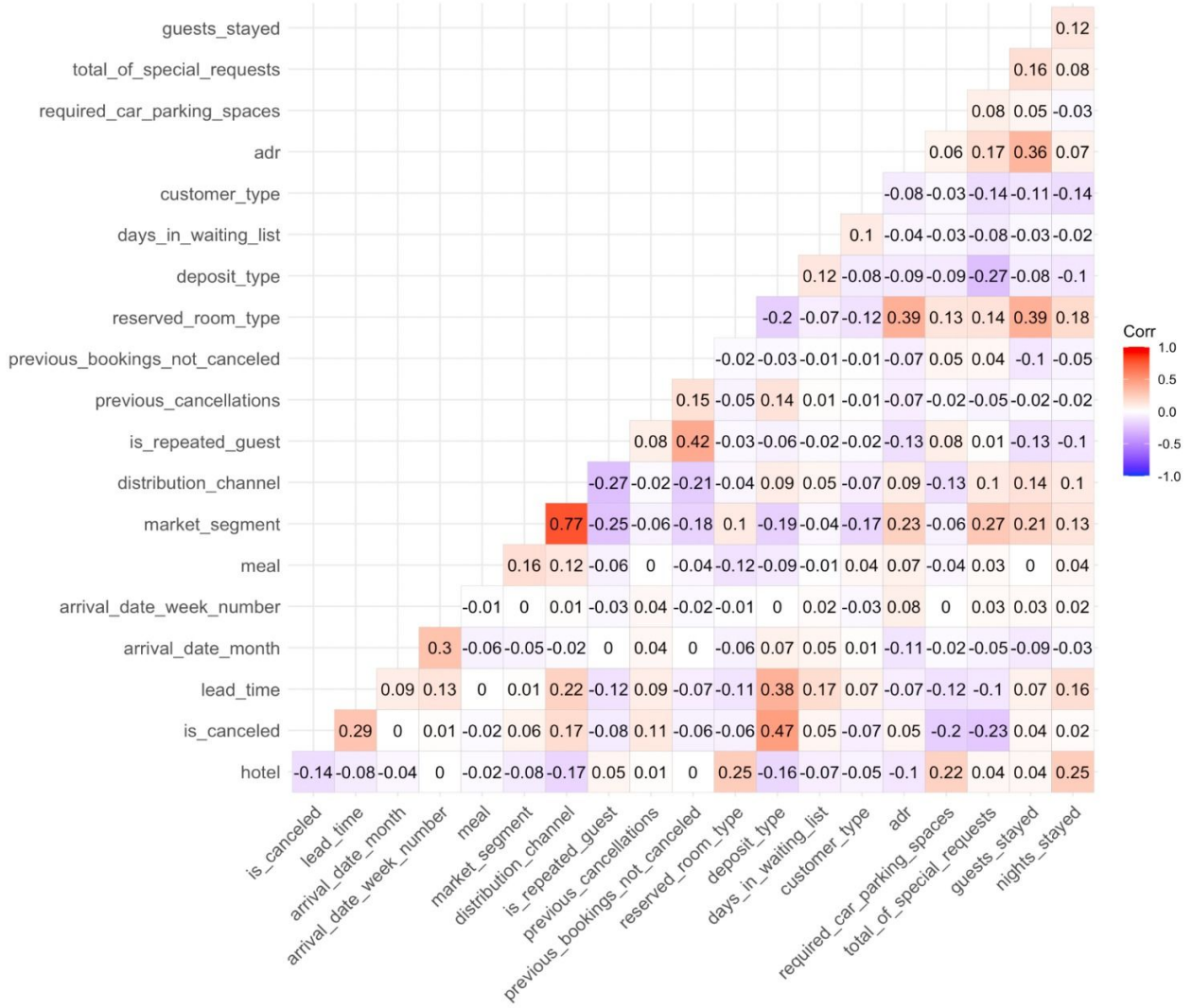
Highly Correlated Numerical Features

Highest correlated features:

1. Lead Time
2. ADR
3. Previous Cancellations
4. Days in Waiting List

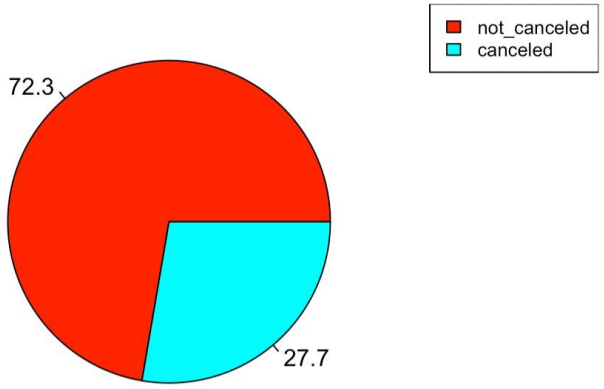
Negatively correlated features:

5. Total Special Requests
6. Required Parking Spaces
7. Booking Changes

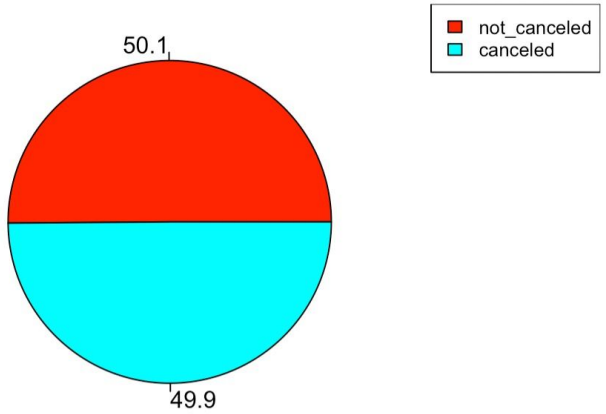


PairWise Associations:

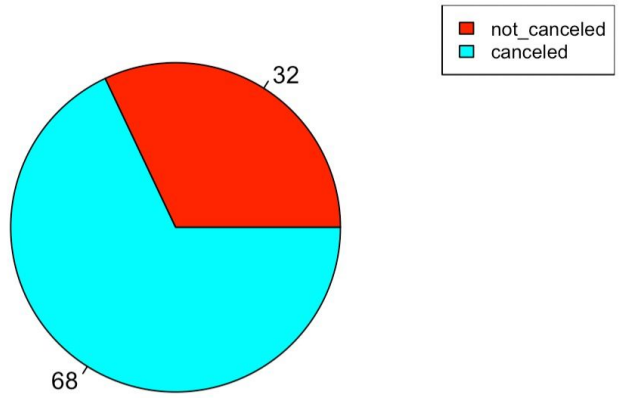
Lead Time vs Cancelations:



Lead Time <**100** days



Lead Time >=**100** days & < **365** days



Lead Time >= **365** days

From this we can understand that as lead_time increases the chances of booking cancellation as well increases.

Previous Cancellations vs Cancelations:

Never Previously Cancelled Bookings	**33.9%**
Previously cancelled only once	**94.4%**
Previously cancelled more than 10 times	**99.3%**

As the number of previous cancellations increases the chances of booking cancellations as well increases.

Total of Special Requests vs Cancelations (highly -vely correlated feature):

Even though it is negatively correlated feature, as the number of special requests increases the booking cancelation percentage decreases.

Parking Spaces vs Cancelations (highly -vely correlated feature):

required_car_parking_spaces <int>	V1 <int>
0	67750
1	7383
2	28
3	3
8	2

Non Cancelled Bookings

required_car_parking_spaces <int>	V1 <int>
0	44224

Cancelled Bookings

From this we can understand the model can tune in such a way that if the number of required spaces is zero the booking can be canceled which is not the case ideally. So, we can ignore this feature while modeling.

Hotel vs Cancelations:

Description: df [12 × 2]

	resort_cancel <dbl>	city_cancel <dbl>
January	0.1481988	0.3966809
February	0.2562037	0.3828802
March	0.2287170	0.3694642
April	0.2934331	0.4632353
May	0.2877213	0.4437561
June	0.3307061	0.4469217
July	0.3140171	0.4087537
August	0.3344912	0.4009796
September	0.3236808	0.4202703
October	0.2751055	0.4297173
November	0.1891670	0.3812256
December	0.2382931	0.4211036

From the above stats we can understand that wrt month city hotels has more booking cancellations compared to resort hotels according to arrival months.

Data Cleaning

- Drop unwanted columns.
- Encode categorical features.
- Standard scale the numerical features.

Drop Columns:

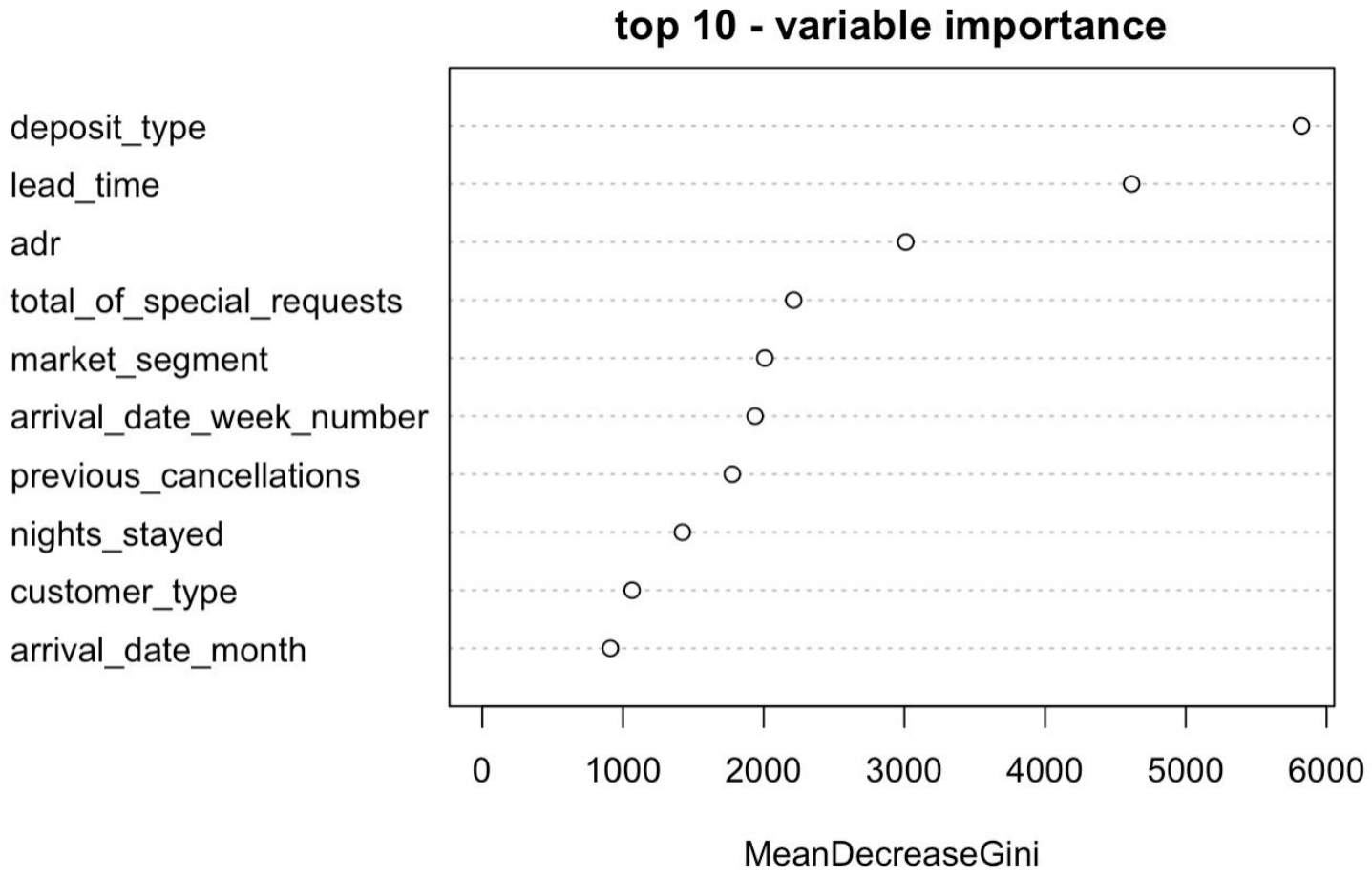
- Numerical Columns:
  - agent & company => These columns are uninformative since they contain discrete codes for the agents and company using which the booking is made.
  - booking_changes => Could constantly change over time and has no much effect on the predictor.
  - arrival_date_day_of_month & arrival_date_year => Prevents the model from generalizing, since we have arrival_week information that would be sufficient.
- Categorical Columns:
  - reservation_status => It has values Check-Out, Cancelled and No-Show which means not-canceled and canceled considering this feature can cause the model to overfit.
  - reservation_status_date => Date when the reservation_status is last changed this is not relevant.
  - assigned_room_type => This is irrelevant and more over reserved_room_type makes more sense since the booking can be canceled only before checking-in which means room is assigned.
  - country => There are many countries and not uniformly distributed so there are higher chances that this model can prevent the model from generalising.

Modeling

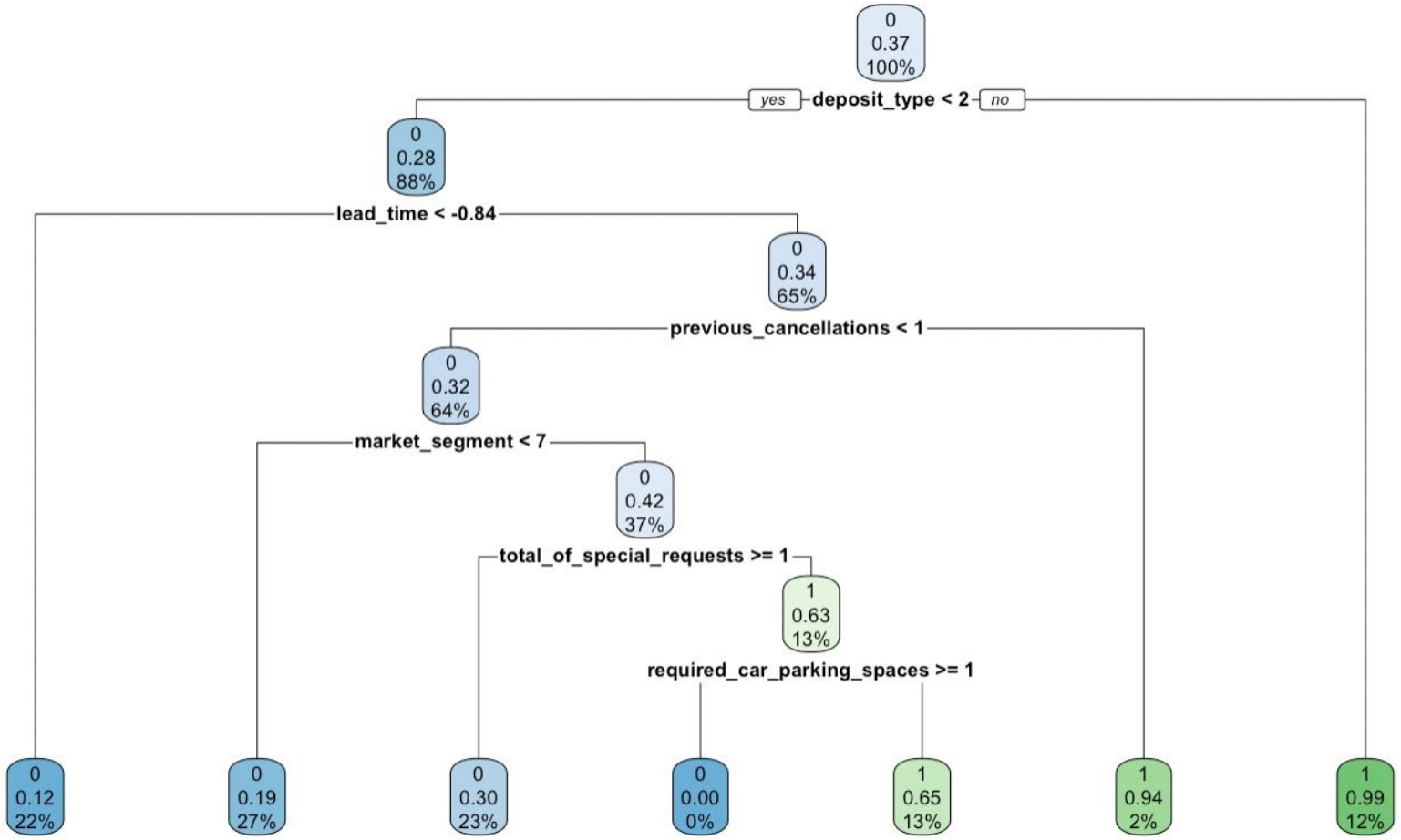
- Logistic Regression
- KNN (K-Nearest Neighbors)
- Decision Tree
- Random Forest
- ADABoost

Cross Validation is as well applied on Logistic Regression, KNN and Decision Tree models to get the actually accuracy of the model.

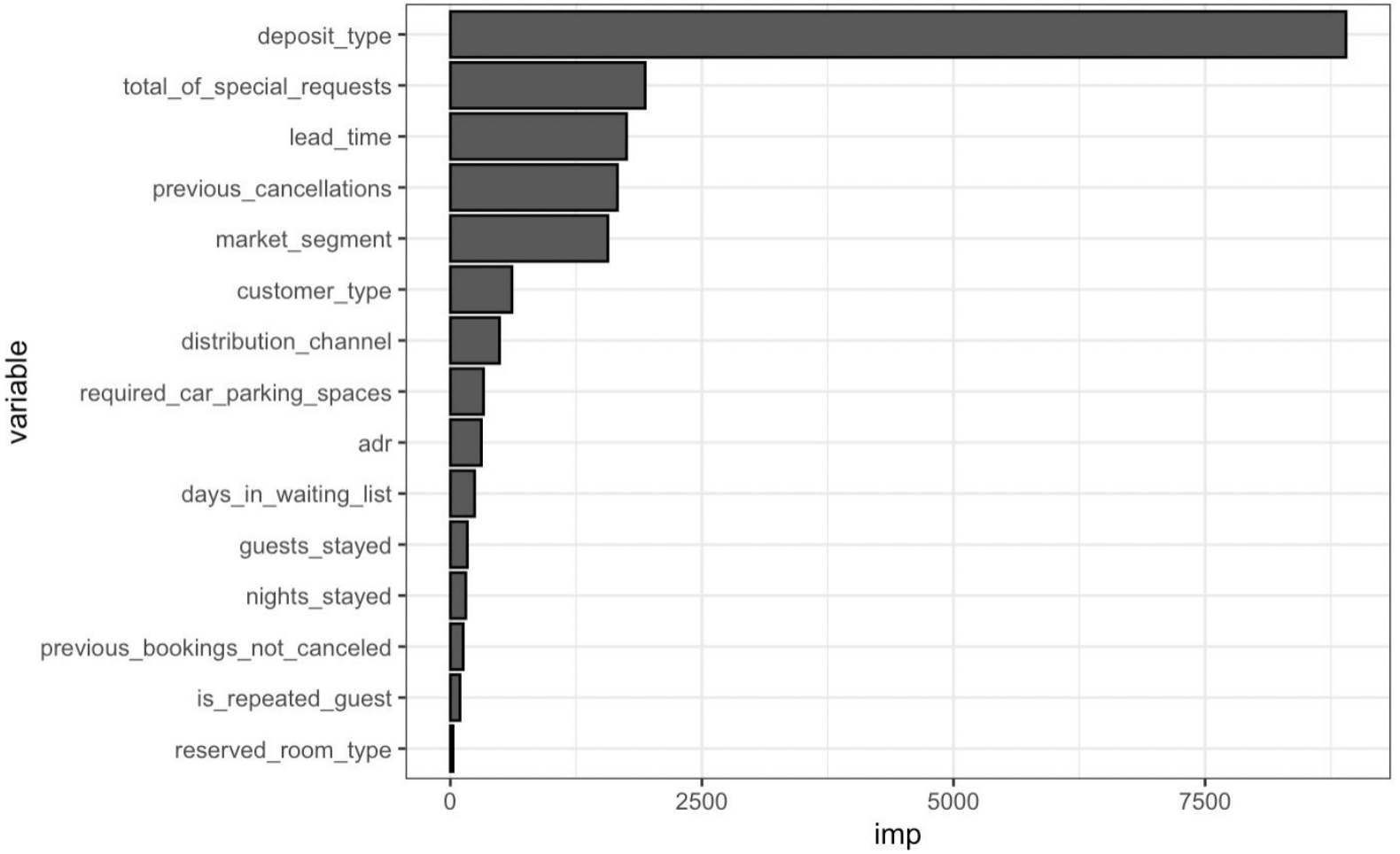
Random Forest



Decision Tree



Decision Tree Variable Importance



Adaboost

	var <chr>	rel. <dbl>
deposit_type	deposit_type	53.9776556
lead_time	lead_time	11.9996973
total_of_special_requests	total_of_special_requests	8.2236348
market_segment	market_segment	6.8892176
previous_cancellations	previous_cancellations	6.8076200
required_car_parking_spaces	required_car_parking_spaces	5.5535621
customer_type	customer_type	2.3249470
adr	adr	1.6872330
previous_bookings_not_canceled	previous_bookings_not_canceled	1.3319520
nights_stayed	nights_stayed	0.4867930
arrival_date_month	arrival_date_month	0.2354475
arrival_date_week_number	arrival_date_week_number	0.1288771
meal	meal	0.1221856
guests_stayed	guests_stayed	0.0931385
reserved_room_type	reserved_room_type	0.0651408
days_in_waiting_list	days_in_waiting_list	0.0645454
distribution_channel	distribution_channel	0.0083517
hotel	hotel	0.0000000
is_repeated_guest	is_repeated_guest	0.0000000

19 rows

Feature Selection

- Considering feature importance from different machine learning models(random forest, decision tree, adaboost) the following 5 features are selected
  - Deposit Type
  - Total Special Requests
  - Lead Time
  - ADR
  - Market Segment

Results

Final Decision:

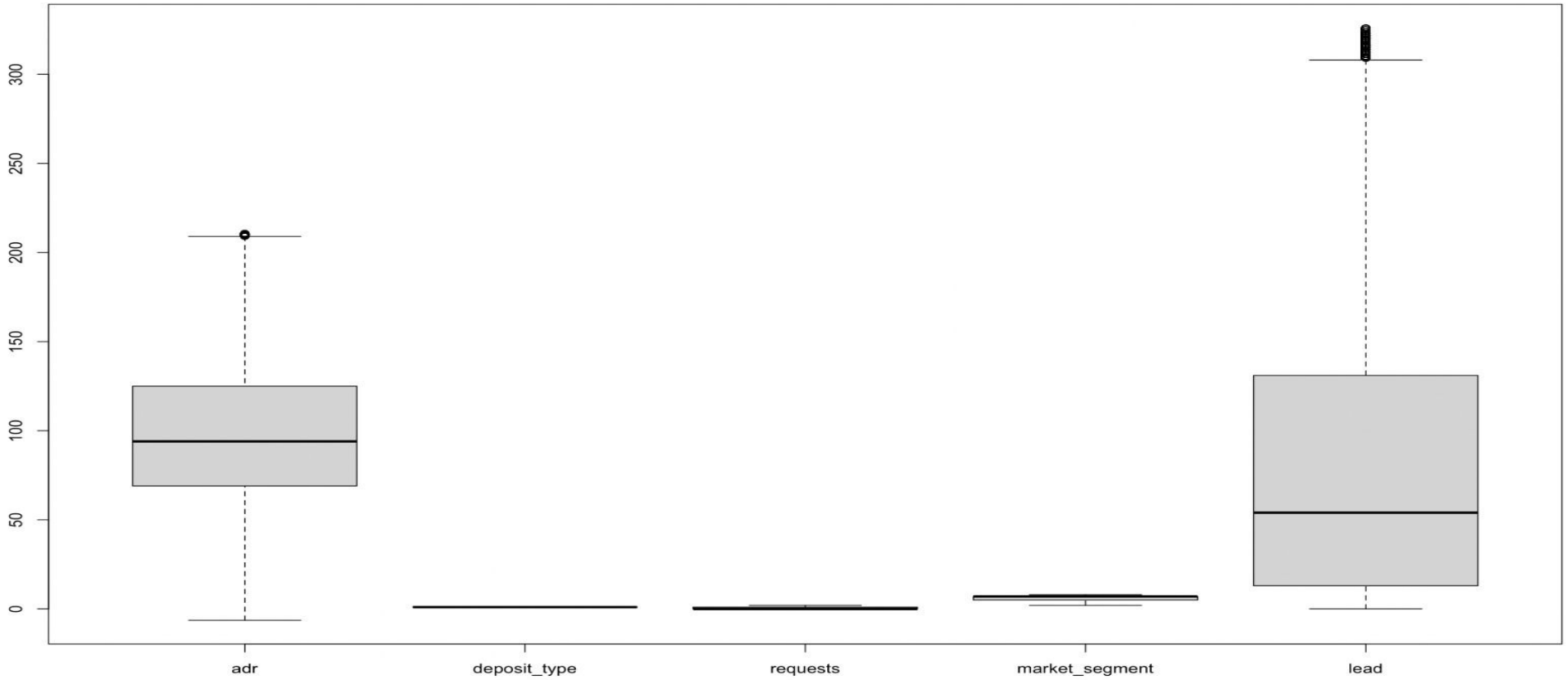
We have considered 5 features with higher importance.

We can observe no or less difference between without and with feature selection.

So there is some scope to generalize the model, which means we can go ahead **with feature selection**.

Classifications Models	Without feature selection	With feature selection
Logistic Regression	Train: 79.4% Test: 79.4% (CV)	Train: 76.7% Test: 76.8% (CV)
KNN	Train:80.3% Test: 79.7% (CV)	Train: 79.9% Test: 79.8% (CV)
Decision Tree	**Train: 80.6%** **Test: 80.6%(CV)**	**Train: 79.4%** **Test: 79.1%** (CV)
Random Forest	**Train: 90.5%** **Test: 85.2%**	**Train: 82.3%** **Test: 82.2%**
Adaboost	Train: 81.3% Test: 81.1%	Train: 80.3% Test: 79.6%

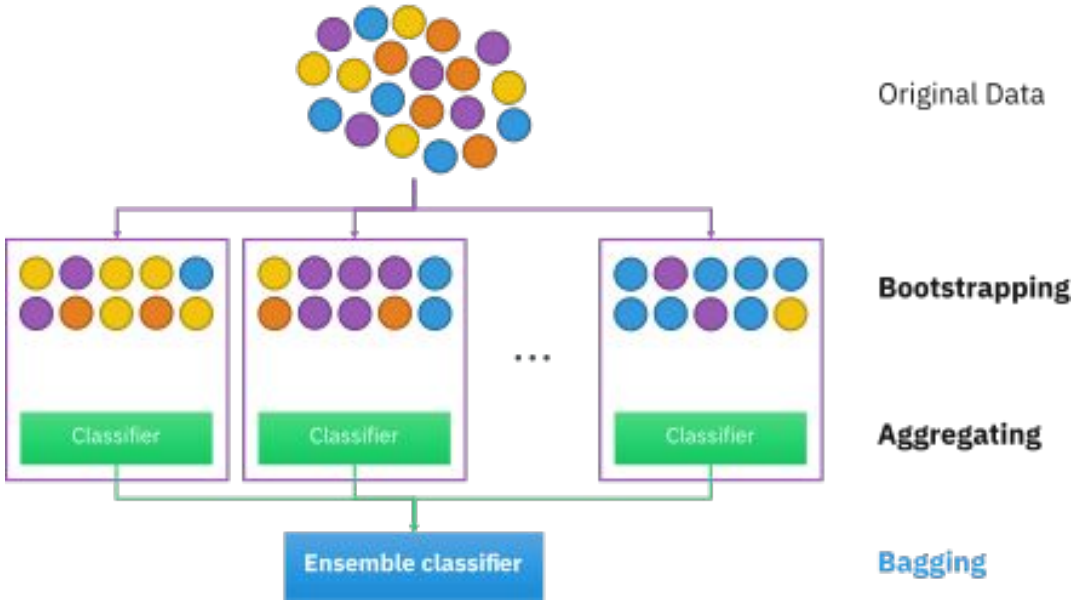
Outliers on Features Selected:

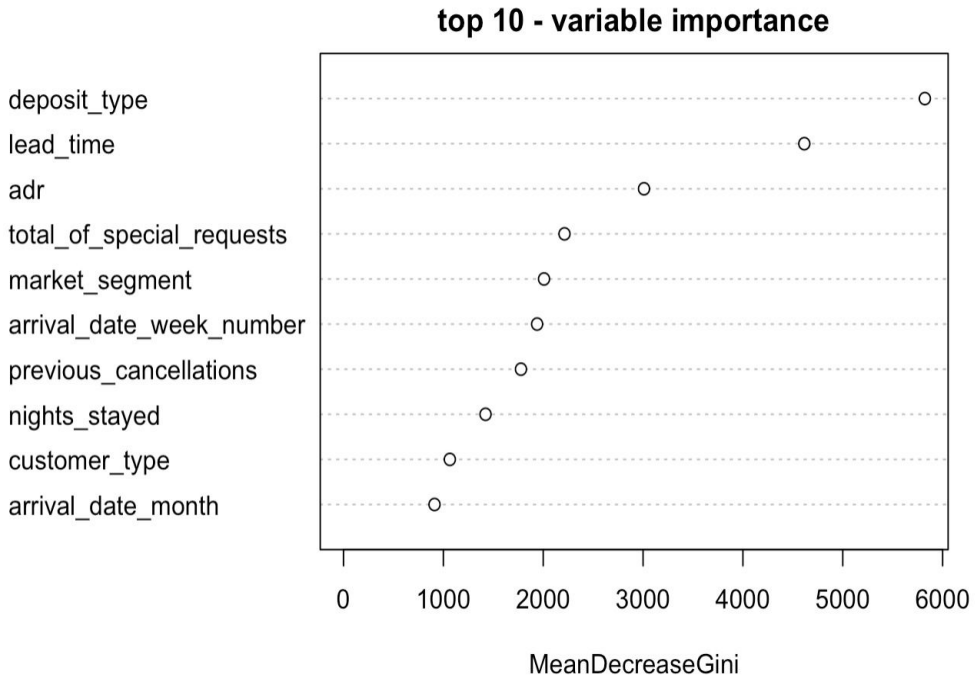


From this we can understand there are no outliers for deposit_type, requests, market_segment and there is few to no outliers on adr and there is few outliers on lead_time.

Final Model Decision: Random Forest With Feature Selection

Model	Without Outliers	With Outliers
Random Forest	Train: 80% Test: 80.08%	Train: 82.3% Test:82.2%





Conclusion

- Reduction of features maintains the accuracy of the model wrt all features and helps to generalise the model for new dataset.
- Unsure of removing outliers due to its influence on the model
  - Unable to determine if the outliers are actual outliers even if they fall out of the IQR
- From the best model, the features deposit_type, lead_time, adr, special_requests and market_segment are the most influential features.

Dataset Reference:

References:

Dataset - https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand?datasetId=511638

Github - https://github.com/jayanthjay12/EAS506_StatisticalDataMining1_Team5

Thank You