Machine Learning Group Assignment

Angelo Barisano, Millie Zhu, JeanLuc Oudshoorn, Jaimy Lai

Objective

The objective is to help a hotel **predict whether a hotel booking will be canceled or not.**
The owner of the hotel is hoping for an accuracy of at least **85%.** Besides the overall performance, they have expressed an interest in the ability of the model to **detect canceled bookings.**

Why a Machine-Learning Task?

- Data Availability
- Complexity & Nonlinearities

Advantages

With this information, the hotel can optimize e.g. its planning and resources

- Better insights into the necessary staffing within a certain time period
- Optimize the inventories and costs
- In months with more cancellations, the hotel can e.g. do more marketing activities

Data

The dataset that will be used in this project consists of a training and test dataset. The training dataset includes 24,035 bookings and the test set contains 16,025 bookings and is collected throughout the same time. The dataset consists of 26 variables which consist of numerical and categorical variables and includes the target variable: is_cancelled.

Exploratory data analysis

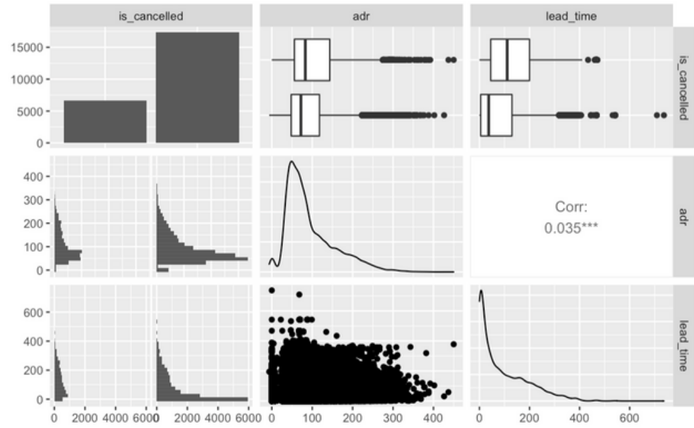
Performed EDA in order to understand the data and avoid poblems in modelling

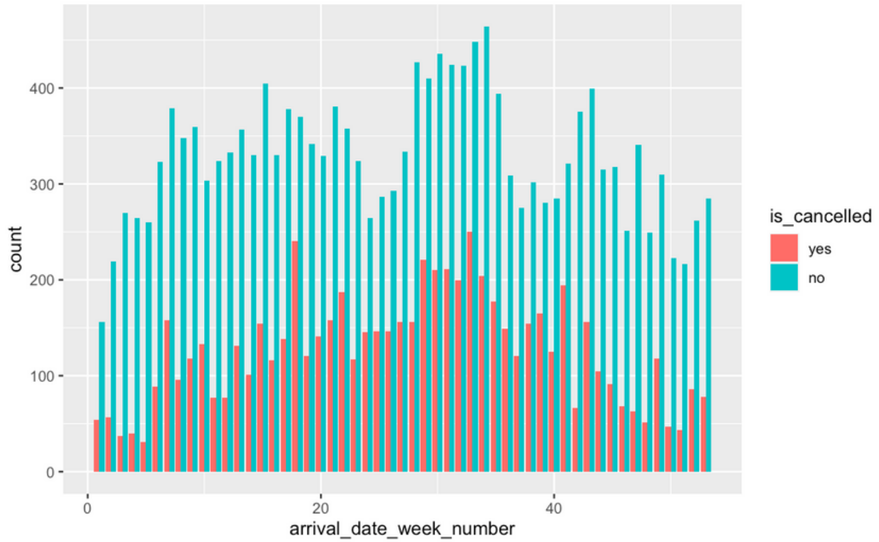
- Summary statistics
  - summary table
  - skimr: summary statistics
- Graphs
  - ggplot: see distributions, observe data imbalance, etc
  - pairplots: understand relationships between variables, etc.

Interesting finds examples

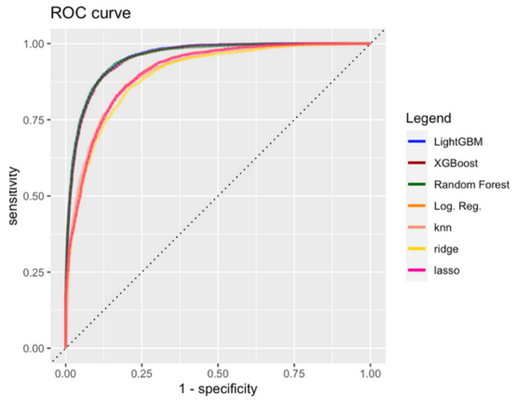
- People with booked parking spaces are not likely to cancel
- No missing values
- Most customers are from Portugal or Britain
- Arrival date week has a 53rd week
- Most big group bookings will be cancelled

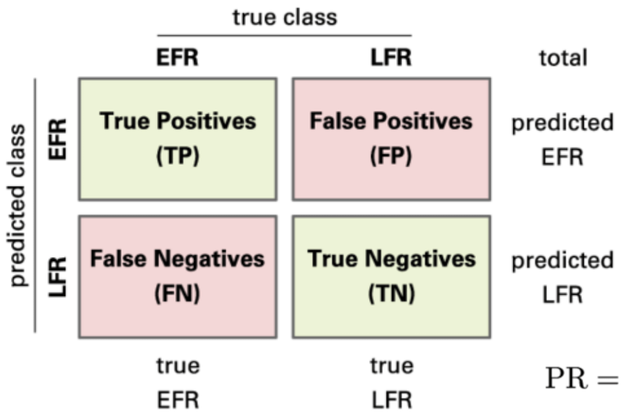
Statistic	N	Mean	St. Dev.	Min	Max
lead_time	24,035	92.740	97.601	0	737
arrival_date_year	24,035	2,016.123	0.720	2,015	2,017
arrival_date_week_number	24,035	27.117	13.968	1	53
arrival_date_day_of_month	24,035	15.790	8.895	1	31
stays_in_weekend_nights	24,035	1.195	1.149	0	16
stays_in_week_nights	24,035	3.132	2.479	0	40
adults	24,035	1.868	0.754	0	55
children	24,035	0.129	0.442	0	3
babies	24,035	0.014	0.120	0	2
is_repeated_guest	24,035	0.045	0.206	0	1
previous_cancellations	24,035	0.104	1.357	0	26
previous_bookings_not_cancelled	24,035	0.148	0.999	0	30
booking_changes	24,035	0.284	0.707	0	16
days_in_waiting_list	24,035	0.524	7.370	0	185
adr	24,035	94.575	61.169	-6.380	450.000
required_car_parking_spaces	24,035	0.138	0.347	0	3
total_of_special_requests	24,035	0.618	0.813	0	5





Metrics





$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$
$$\mathrm{RE} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$$
$$\mathrm{CA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$$
$$\mathrm{F_1} = \frac{2\mathrm{TP}}{2\mathrm{TP+FP+FN}}$$

- **Stakeholder Demands & Context**
  - Requested 85% Accuracy & focus on cancellation detection
    - Risks of Missing Cancellations (False Negatives) - Stakeholder focus
      - Overstaffing & missed last-minute marketing potential, and overall revenue lost
    - Risks of Misclassifying Cancellations (False Positives) - potentially neglected
      - Understaffing & overbooking

→ Task: Rebalance Stakeholders' expectation metric penalizing a complete focus on cancellation by the model

- **Central Evaluation Metrics**
  - **F1 (Tuning Metric):** Harmonic mean of the models' Precision & Recall - Balance False Cancellation (FP) & False Non-Cancellations (FN)
    - Advantage: Accuracy only considers correct predictions and F1 penalizes excessive focus on Recall
    - F1 reflects both the stakeholder's demands and also the best choice from an engineering perspective
  - **AUC-ROC:** (Complementary) A threshold (for classification) independent classifying metric; any value above 0.5 represents that the model is an improvement over a random classifier
    - Advantage: Provides summary for comparing models regardless of class imbalance (unequal number of outcomes) & versatility
  - **Accuracy:** Proportion of correct predictions made by the classifier out of all the predictions made
    - Advantage: Simple metric & required by Stakeholders

Preprocessing

- **Impactful Variable Preparation**
  - "country" - Null-Value conversions & Lump small-frequency (200) countries together in "Other"
  - "adr" (average daily rate) - delete (impossible) negative values but keep "zero" values as possible gifts - 1 dropped
  - Remove "no-adult-bookings" - 252 dropped
  - Remove potentially risky variables (information leakage potential & uninformative variables) - see update_role()

→ General investigation for possible information leakage variables (derivates of outcome/ measured post hoc)

- **Feature Engineering (No Free Lunch Theorem)**
  - "got_assigned_room" - At the time of booking, did the customer receive the booked room; no information leakage
  - "week_day_of_week" - Distinguishing between weekdays and weekend days
  - "total_visitors" - Deriving total visitors to reduce noise

- **Workflow**
  - Step_novel - new factor levels for test_set
  - Step_harmonic - for date-related features
  - Step_impute_mode - impute country
  - Step_normalize - boosting, regression, KNN
  - Step_dummy - for categorical variables
  - **NO DOWNSAMPLING - improve performance**

```
## Define Recipe
```{r}
recipe_prelim <- recipe(is_cancelled ~ ., data = bookings_train) |>
    step_novel('assigned_room_type') |>          # Account for new factor levels
  update_role( "date_col", "arrival_date_month", "babies",  "arrival_date_day_of_month", "days_in_waiting_list", new_role = "metadata") |>
  step_harmonic('arrival_date_week_number',frequency=1,cycle_size=53, role='predictor') |>     # More realistic representation of time
  step_harmonic('arrival_date_month_numb',frequency=1,cycle_size=12, role='predictor') |>
  step_harmonic('arrival_date_day_of_week',frequency=1,cycle_size=7, role='predictor') |>
  step_impute_mode("country") |>                 # Fill NA values
  step_normalize(all_numeric_predictors())|>     # Normalization for faster convergence
  step_dummy(all_nominal_predictors())           # Dummy encode all remaining categorical variables
```
```

Logistic, ridge, lasso

Method

- Logistic regression as the baseline model: a linear model that uses a logistic function to model the probability of an event occurring
- Ridge: add a new term to the logistic function to reduce the variance of the model
- Lasso: add a new term to the logistic function to select a subset of the most important predictors

Goal

- Provide a benchmark to measure the performance of more complex models

Advantage

- Simple to understand, computationally efficient
- Help prevent overfitting

K-nearest neighbors (KNN)

Method

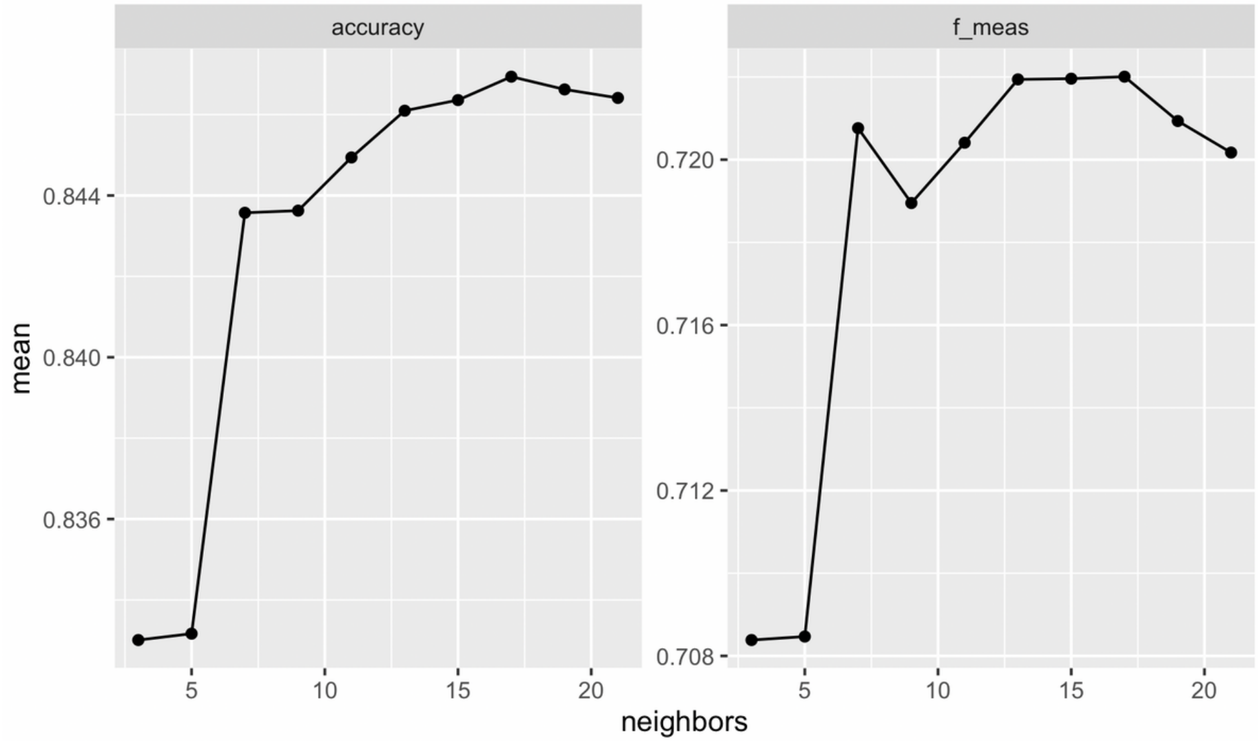
- Find the k nearest data points to a given data point
- Use their class labels on the cancellation to make a prediction for the given data point
- Does not make strong assumptions on any functional form from the training data

Goal

- Flexibly classify a given data point

Advantage

- Simple and fast
- robust to meaningless information in the data



Random forest

Method

- Based on decision trees
- Trains every tree on a bootstrapped version of the training set
- Only considers a set number of predictors to split on at each node
- Averages predictions over all trees in the final model

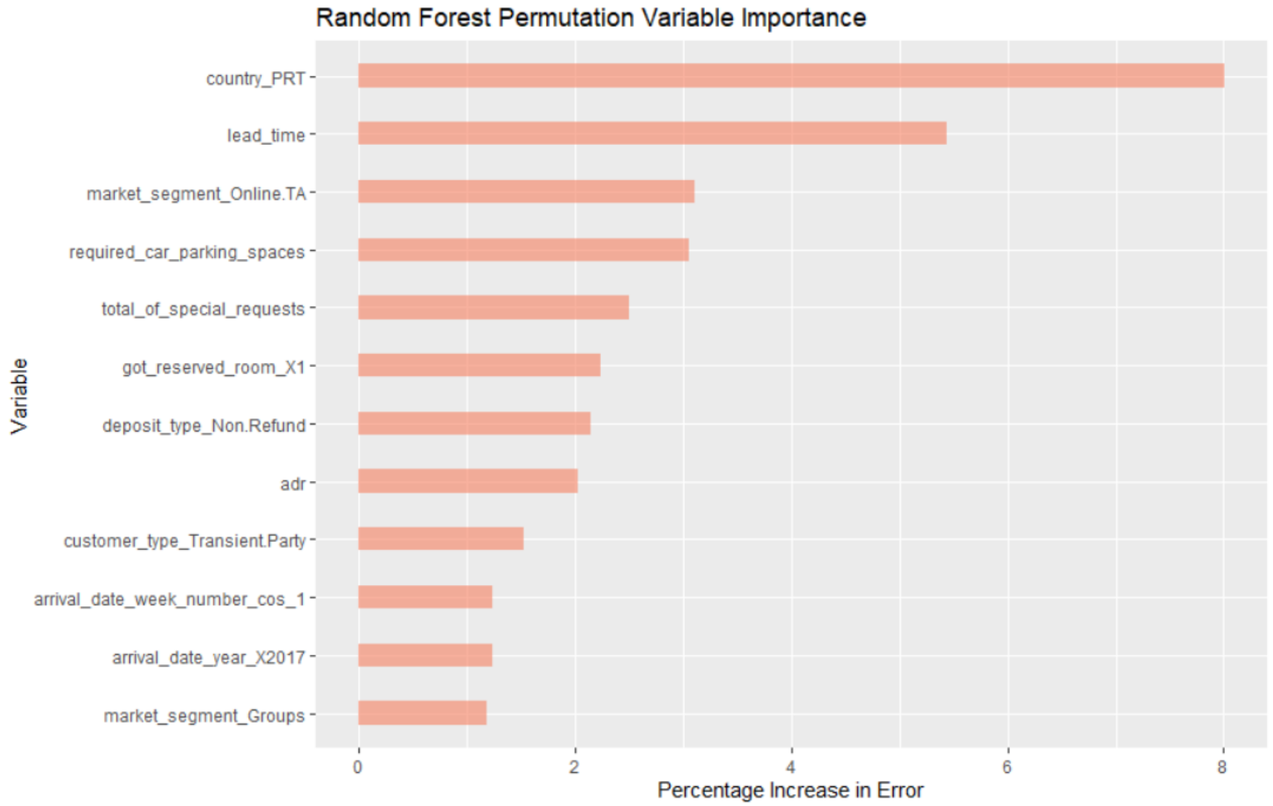
Goal

Reduce variance while keeping bias low by constructing many uncorrelated trees

Advantages:

- More stable and accurate predictions than simple decision trees
- Less sensitive to overfitting and changes in the data
- Requires little hyperparameter tuning (good out-of-the-box performance)
- Natively captures interaction effects between variables

On this dataset: Random Forest achieves the highest precision but recall is lacking which leads to a lower F1-score than other models. The overall accuracy and area under the ROC-curve are also slightly lower.



Boosting

Method

- A single decision tree is fit and predicts the dataset
- Another tree is fit subsequently on the prediction residuals of the previous tree
- Base prediction is updated by adding the predictions of each subsequent tree
- This process is iterated many times

Goal

Make increasingly accurate predictions by identifying and iteratively building trees that specialize in predicting hard cases

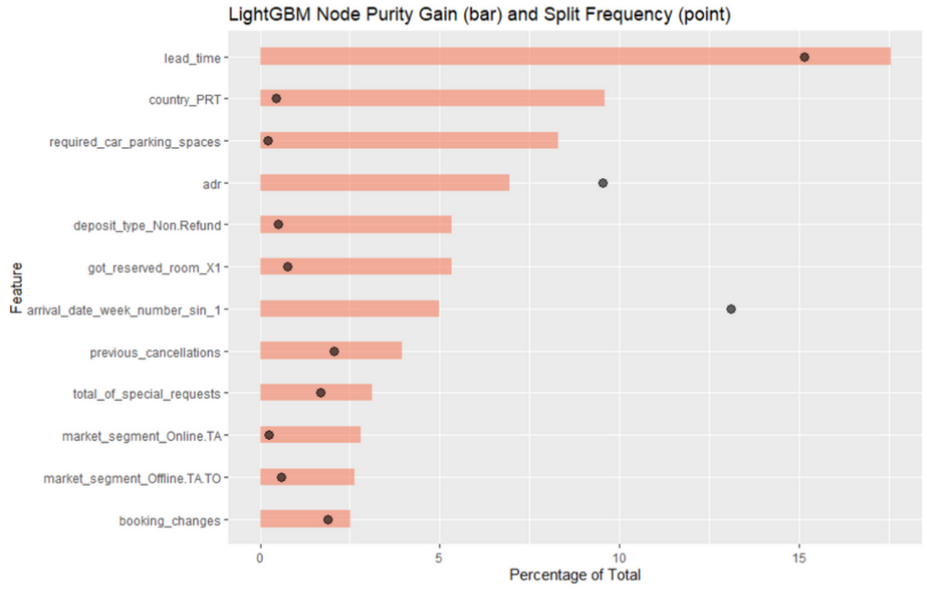
Advantages

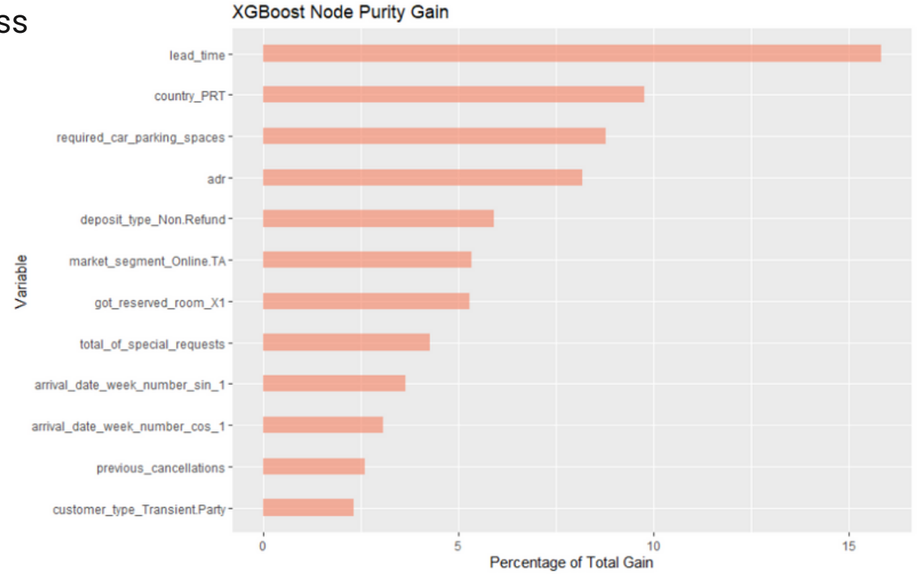
- Higher predictive accuracy than almost any other algorithm when tuned well
- Better at handling imbalanced data by iteratively fitting trees on the hard-to-predict class
- Better feature importance estimates
- Generally need fewer data to achieve good performance

Two main differences between XGBoost and LightGBM

- Using a histogram for determining split points
- Leaf-wise versus level-wise tree building

On this dataset: XGBoost slightly outperforms the Random Forest on F1-score, but not on accuracy. LightGBM takes the crown with almost a full percentage point increase in F1-score as well as better accuracy and area under the ROC-curve than the Random Forest.





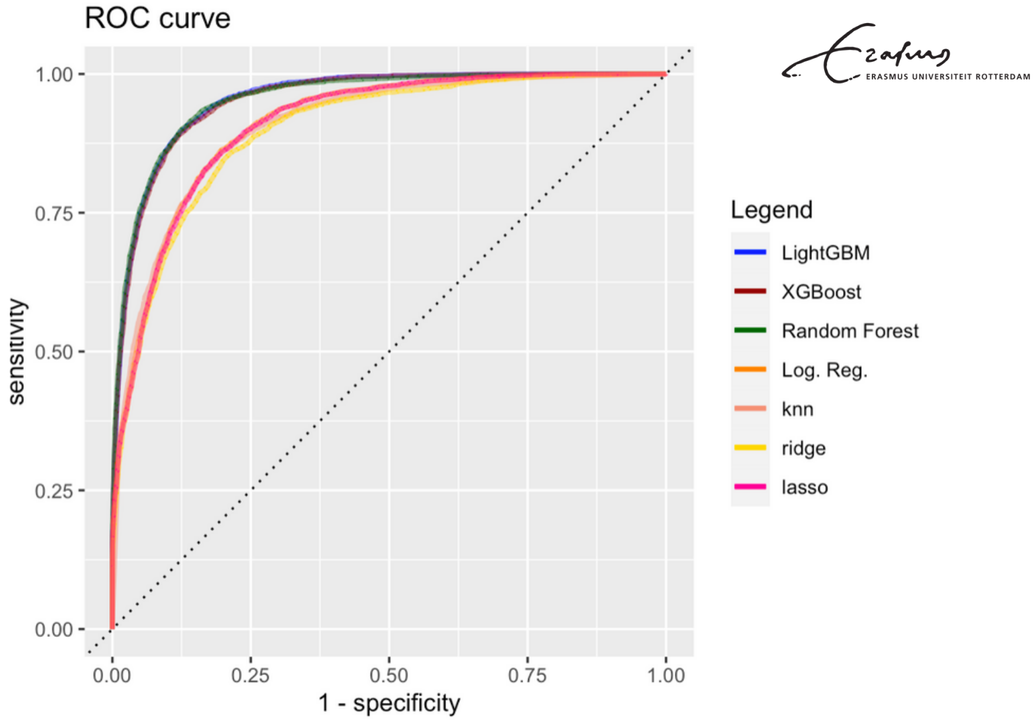
Model assessment

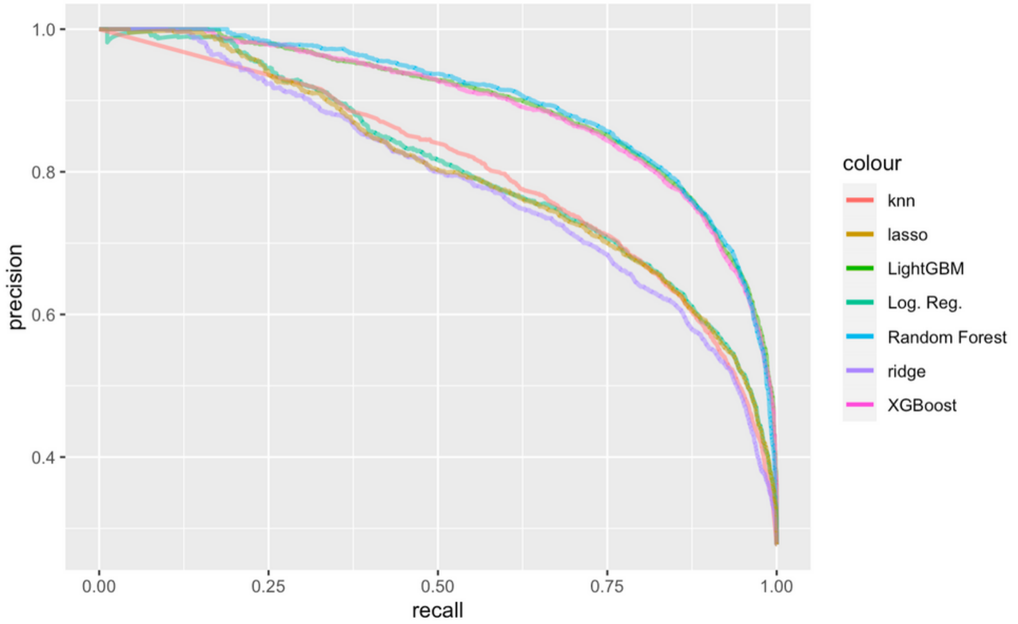
Preprocessing steps are performed on both the train and test data.

Steps that require some form of estimation (e.g.: scaling) are only estimated on the train set and applied to both the train and test set. (This way the mean of the train set is used for normalization on the test set)

The hyperparameters are picked according to cross-validation performance on the training set, after that the chosen model is evaluated on the test set just once to give a fair estimate of generalization error.

ROC- and PR-curves are used to evaluate models for different preferred levels of sensitivity, specificity, precision, and recall.





Analysis of statistics

KNN confusion matrix

	Truth	
Prediction	**yes**	**no**
yes	3270	1115
no	1179	10461

Ridge confusion matrix

	Truth	
Prediction	**yes**	**no**
yes	2565	718
no	1884	10858

Lasso confusion matrix

	Truth	
Prediction	**yes**	**no**
yes	2828	864
no	1621	10712

Logistic Regression confusion matrix

	Truth	
Prediction	**yes**	**no**
yes	2863	882
no	1586	10694

XGBoost confusion matrix

	Truth	
Prediction	**yes**	**no**
yes	3585	749
no	864	10827

LightGBM confusion matrix

	Truth	
Prediction	**yes**	**no**
yes	3629	734
no	820	10842

Random Forest confusion matrix

	Truth	
Prediction	**yes**	**no**
yes	3488	630
no	961	10946

Clearly LightGBM has the lowest error overall and also the best balance between false positives and false negatives

Final result and model selection

Model	F1	Accuracy	Precision	Recall	ROC AUC
Logistic (Base)	0.699	0.846	0.764	0.643	0.910
KNN	0.740	0.857	0.746	0.735	0.916
Lasso	0.695	0.845	0.766	0.636	0.910
Ridge	0.663	0.838	0.781	0.577	0.902
Random forest	0.814	0.901	**0.847**	0.784	0.957
LightGBM	**0.824**	**0.903**	0.832	**0.816**	**0.961**
XGBoost	0.816	0.899	0.827	0.806	0.958
Best performing model	**LightGBM 0.824**	**LightGBM 0.903**	**Random forest 0.847**	**LightGBM 0.816**	**LightGBM 0.961**

Final conclusion

Hoteliers want to increase the value of their hospitality services for their customers to have retention, and ultimately, increase revenue. For this case specifically, it is beneficial for staffing purposes to know how many bookings will be canceled in advance. The use of machine learning is beneficial in almost all industries as it optimizes processes and efficiency gains.

- **LightGBM** is the best-performing model across most metrics
- The data allows for a fairly accurate classification of cancellations and non-cancellations
- There is a good balance between over- and underprediction of cancellations

Further suggestions

- Performance should be monitored for model rot
- Machine learning projects can be rolled out for predictive maintenance and price optimization

	F1	Accuracy	Precision	Recall	ROC AUC
Best performing model	**LightGBM 0.824**	**LightGBM 0.903**	**LightGBM 0.832**	**LightGBM 0.816**	**LightGBM 0.961**