Hotel Reservation Classification

Machine Learning B - DIS Spring 2024

Elise Hedlund, Ethan Ernst, & Leonard Mayorga

Introduction

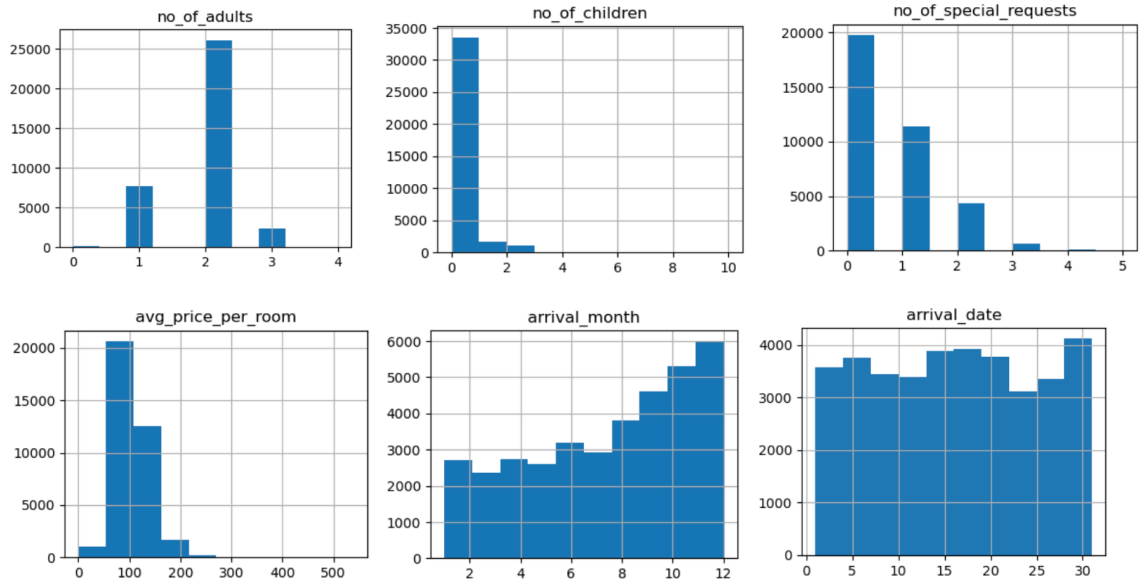
As current study abroad students who have now been to many hotels in multiple different countries, we thought hotel reservations would be the perfect dataset to look at. We all like to see a hotel that has free of charge cancellations which allow for increased flexibility when traveling, but this can often come at a cost to the hotel company because they could lose revenue from not having a room booked. A way to minimize this cost would be for hotels to be able to predict which reservations are most likely to cancel or not show up. This is exactly what we will be investigating in this project. We have decided to do a binary classification in order to attempt to predict which reservations are likely to cancel and which are not likely to cancel.

Data

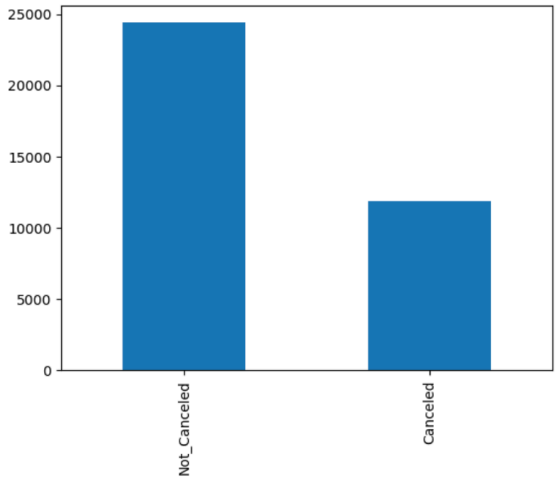
We located our dataset on Kaggle; it can be found at the following link: https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset

The dataset contains roughly 36,300 instances, each representing an individual hotel reservation between 2017 and 2018. There are 19 features, which are the attributes of customers' hotel reservations. The features are as follows: Booking_ID, no_of_adults, no_of_children, no_of_weekend_nights, no_of_week_nights, type_of_meal_plan, required_car_parking_space, room_type_reserved, lead_time, arrival_year, arrival_month, arrival_date, market_segment_type, repeated_guest, no_of_previous_cancellations, no_of_previous_bookings_not_canceled. avg_price_per_room, no_of_special_requests, and booking_status.

Some feature visualization:



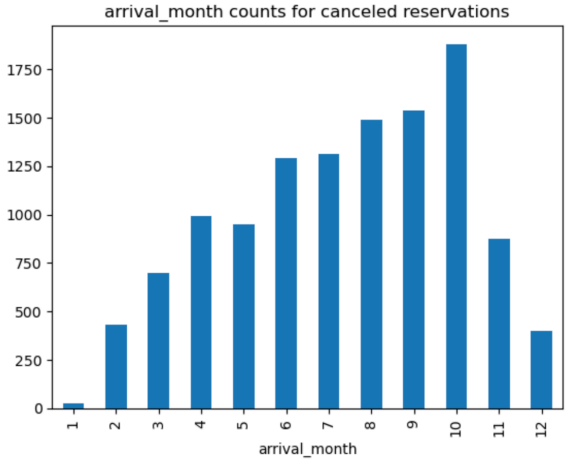
Our target variable is booking_status, which is defined in the dataset information as a "flag indicating if the booking was canceled or not." We chose this feature as our target variable because the aim of our project is to determine the likelihood of a booking being canceled and which features are the best predictors of this cancellation. As it stands in the dataset 67% of reservations are labeled as not canceled and 33% are labeled as canceled. Since there is a relatively large number on each side of the classification we decided not to do any overfitting in order for the model to classify better.

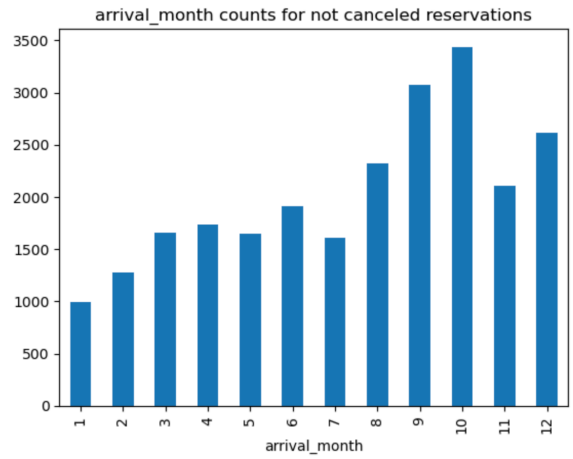


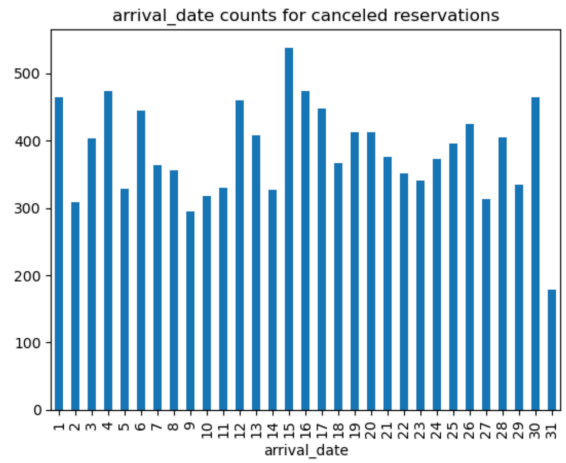
Data Preprocessing:

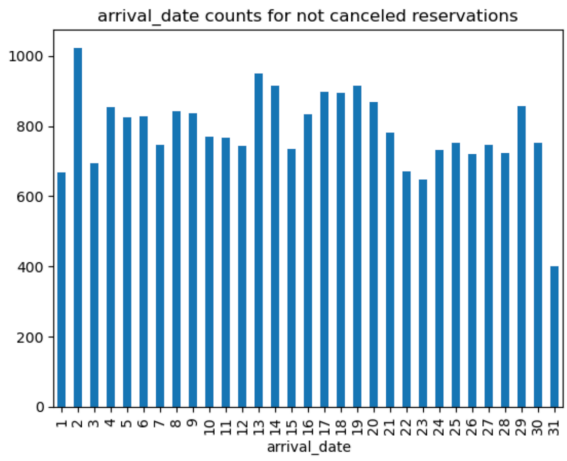
There was not an extensive preprocessing of the data needed for our project, but we did do some. We decided to do binary encoding for the booking_status feature in order to allow the models to work more efficiently. We also decided to use One Hot Encoding for the features; type_of_meal_plan, room_type_reserved and market_segment_type. This would allow us to look at these as individual features that do not interact with each other and are separate entities.

There are two additional features that we looked at in the preprocessing and that was arrival_month and arrival_date. We wanted to figure out if these would be beneficial to keep with cyclical encoding or if we should drop them from the data set as they may not have any noticeable effect on the outcome. We decided to use a bar graph to visualize the arrival_month and arrival_date of canceled vs not canceled bookings to see if there is any noticeable effect.









From looking at these graphs and comparing the number of bookings kept per month and the number of bookings canceled per month, it looks like the only two months that might have any significance would be January and December. However, all of the other months seem to have no significance, so they do not make sense to use or make cyclical. Looking at the number of bookings for arrival date kept vs number of bookings canceled for arrival date, it does not look like there is any significance between days of the month, so it is unnecessary and not helpful to the model to keep these variables. With this analysis, we dropped the arrival_month and arrival_date features along with Booking_ID and arrival_year because none of them had any impact on the target variable. By removing them, we allow the features that do matter to have a more significant effect.

Methods

Based on our project's goal, binary classification was the most fitting approach. We chose to test out three different types of models in order to find the best performing one for our dataset. We chose to look specifically at logistic regression, support vector machines (SVM), and decision trees. Because of the large number of features (which also increased after one-hot encoding), we also performed dimensionality reduction using principal component analysis (PCA) and applied our findings to the decision tree, as that was our best performing model.

Logistic Regression:

Logistic regressions are very valuable models to use for binary classification because they are very simple and efficient algorithms. They allow for us to train a model using many instances and features, and they evaluate each feature independently to the other features in their predictive power towards the target variable and then create a coefficient for each feature. That coefficient displays how large of a predictive role the feature plays in predicting the target variable, booking_status. Logistic regression was good for this dataset because its dimensionality is not too high and the classes are relatively evenly distributed; one side is slightly lopsided to the other but logistic regression would still work well. It is a very interpretable model as well which makes it very easy to use.

SVM:

Support vector machines are powerful tools for classification problems, so it seemed fitting to include them in our analysis. Additionally, SVMs can perform both linear and nonlinear

classification, so this provides further flexibility (and, by looking at the results of each, we can gain more insight into the shape of our data). SVMs perform best on small to medium datasets and may not scale well to large datasets: our dataset has roughly 36,300 instances, which indicates that it may be too large for SVMs to be effective, but it is not so large for it to not be worth testing the model.

Decision Tree:

Decision trees can provide powerful insight when doing classification: namely, the feature importances, which are accessible after training, in which the decision tree highlights which features are most impactful when creating the decision tree. Access to feature importance with decision trees also gives us an alternate report to the feature importance given by PCA. Apart from feature importance, we decided to implement a decision tree because it is also easy to visualize the logic behind how it predicts new instances given the training data.

Results

Correlation Analysis:

We began our analysis by examining which features were most correlated with our target variable, booking_status. We found lead_time (the number of days between the booking and arrival dates) was the most correlated feature with a value of -0.438538. Keeping in mind that 1 signifies a non-cancelled booking, we determine that, as lead time increases, likelihood of booking cancellation also increases. We found that no_of_special_requests was the next most correlated, with a value of 0.253070, indicating that a higher number of special requests is associated with lower risk of booking cancellations.

PCA:

Since the hotel reservations dataset contains a large number of features, we chose to begin with PCA (principal component analysis) for dimensionality reduction before testing any specific models. For our initial implementation, we start by reducing it to 2 dimensions. With this, the explained variance ratio is [0.85694104 0.14212837].

The explained variance ratio indicates the proportion of a dataset's variance explained by each principal component. The results show that the first principal component in the hotel reservations dataset accounts for the majority of the variance, 86%, and the second accounts for 14%.

Next, we implement PCA with the optimal number of dimensions. In this case, we reduce down to the number of dimensions accounting for 95% of the variance. As expected based on the explained variance ratio, the optimal number of dimensions is two, indicating that the vast majority of the variance in the hotel reservations dataset can be explained by two principal components. Next, it is useful to know what these principal components are, so we will examine their feature contributions.

	Principal Component 1
lead_time	0.999523
avg_price_per_room	0.030632
no_of_week_nights	0.002450
no_of_previous_bookings_not_canceled	0.001565
market_segment_type_Offline	0.001492

Above are the top five features for the first principal component. Below is the same for the second component.

	Principal Component 2
avg_price_per_room	0.999454
lead_time	0.030629
no_of_previous_bookings_not_canceled	0.006017
room_type_reserved_Room_Type 1	0.004662
no_of_adults	0.004517

These results indicate that lead_time is the most influential feature for predicting whether a hotel reservation will be canceled, which confirms the results of our correlation analysis. The avg_price_per_room, no_of_week_nights, no_of_previous_bookings_not_canceled, no_of_week_nights, room_type_reserved_Room_Type 1, and market_segment_type_Offline are also influential. While having a model to predict whether a hotel reservation will be canceled is useful, it is also beneficial for hotels to be able to pinpoint certain conditions that may lead to cancellations, so understanding that lead_time has a large influence is beneficial.

Logistic Regression:

The logistic regression seemed to work very well and looks similar to the correlations we found in the original analysis. Below are the coefficients created by the logistic regression model.

```
Coefficients:
   no_of_adults  no_of_children  no_of_weekend_nights  no_of_week_nights  \
0     -0.056202       -0.058754             -0.112828          -0.040708

   required_car_parking_space  lead_time  repeated_guest  \
0                    0.293952  -1.342494        0.364251

   no_of_previous_cancellations  no_of_previous_bookings_not_canceled  \
0                     -0.099010                              0.110753

   avg_price_per_room  ...  room_type_reserved_Room_Type 3  \
0           -0.658787  ...                       -0.016005

   room_type_reserved_Room_Type 4  room_type_reserved_Room_Type 5  \
0                        0.009508                        0.036232

   room_type_reserved_Room_Type 6  room_type_reserved_Room_Type 7  \
0                        0.126637                        0.075422

   market_segment_type_Aviation  market_segment_type_Complementary  \
0                     -0.061481                           0.602212

   market_segment_type_Corporate  market_segment_type_Offline  \
0                      -0.027052                     0.386228

   market_segment_type_Online
0                   -0.474623
```

From looking at the coefficients in the logistic regression we can conclude that the coefficients that seem to have the greatest predictive power to whether a reservation is canceled or not are lead_time, avg_price_per_room, market_segment_type_Complementary, market_segment_type_Offline, and market_segment_type_Online. It seems as though lead_time has the greatest predictive power; based on the coefficient, that means that the further in advance someone books, the more likely they are to cancel the reservation. A greater average room price and booking the reservation online also increase the chances that someone will cancel their reservation. On the other hand, it looks like a complementary reservation and reserving a room offline decrease the chance that someone will cancel the reservation.

Now we will look at the classification report of this model:

```
classification report:
              precision    recall  f1-score   support

           0       0.76      0.61      0.67      3026
           1       0.82      0.90      0.86      6043

    accuracy                           0.80      9069
   macro avg       0.79      0.76      0.77      9069
weighted avg       0.80      0.80      0.80      9069

MSE: 0.19572168927114345
R2: 0.1196887715588405
MAE: 0.19572168927114345
```

Looking at the classification report we can see that there is a decrease of 6% in precision and a decrease of 29% in recall which also lead to a decrease in F1 score. This makes sense because logistic

regression is meant for classifying when the two groups are even in size, but since the number of cancellations is lower, it has a harder time classifying those values. Overall, this model worked relatively well looking at the averages, but the low precision and recall for predicting cancellations could be concerning because that is the thing hotels would most want to predict.. You will also see in the next section that the logistic regression had almost the same performance as the linear SVM, which makes sense because they work in similar ways to classify data.

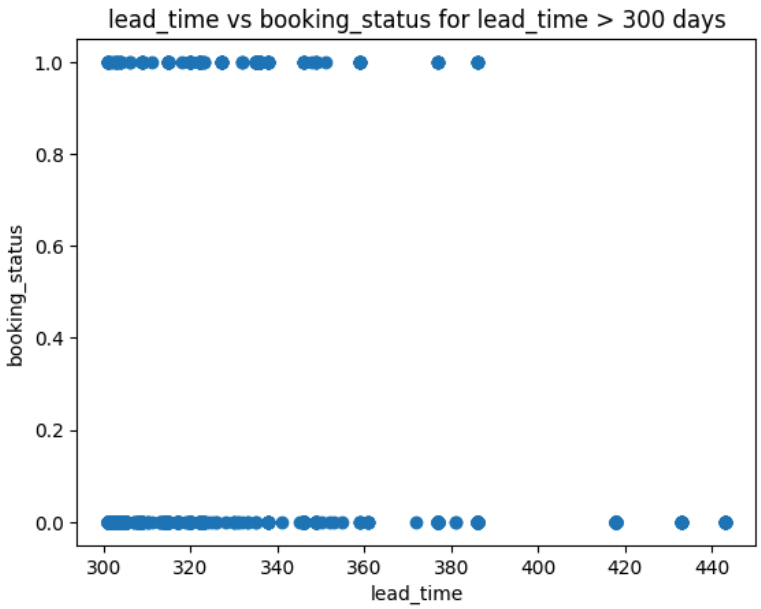
SVM:

We tested both linear and nonlinear SVMs using soft margin classification to account for outliers and add flexibility in our model. The linear SVM (implemented with LinearSVC from Scikit-Learn) performed with the following results:

```
 classification report:
              precision    recall  f1-score   support

           0       0.75      0.60      0.67      3026
           1       0.82      0.90      0.86      6043

    accuracy                           0.80      9069
   macro avg       0.79      0.75      0.76      9069
weighted avg       0.80      0.80      0.80      9069

MSE: 0.19781673833939795
R2: 0.11026572178961114
MAE: 0.19781673833939795
```

As mentioned above, we see the logical result that the linear SVM performs very similarly to the logistic regression. As with the logistic regression, the accuracy of 80% indicates that, while the model is working to some degree, there are changes that need to be made for it to be effective.

This implementation also gave a ConvergenceWarning, in which the model failed to converge even with increased iterations. Considering this and the graph of lead_time vs booking_status, the data does not appear to be linearly separable.



So, in addition to the linear SVM, we will also try a nonlinear SVM for comparison. For this, we use a polynomial kernel of degree 3.

```
 classification report:
              precision    recall  f1-score   support

           0       0.81      0.63      0.71      3026
           1       0.83      0.93      0.88      6043

    accuracy                           0.83      9069
   macro avg       0.82      0.78      0.79      9069
weighted avg       0.82      0.83      0.82      9069

MSE: 0.17433013562686073
R2: 0.2159030692025503
MAE: 0.17433013562686073
```

With a nonlinear SVM, we see improved results across our classification report, from accuracy to precision, recall, and f1-score. The mean square error and mean absolute error decrease while R-squared increases, both of which also indicate a better performing model. This indicates that nonlinear SVM classification is a better choice for this dataset.

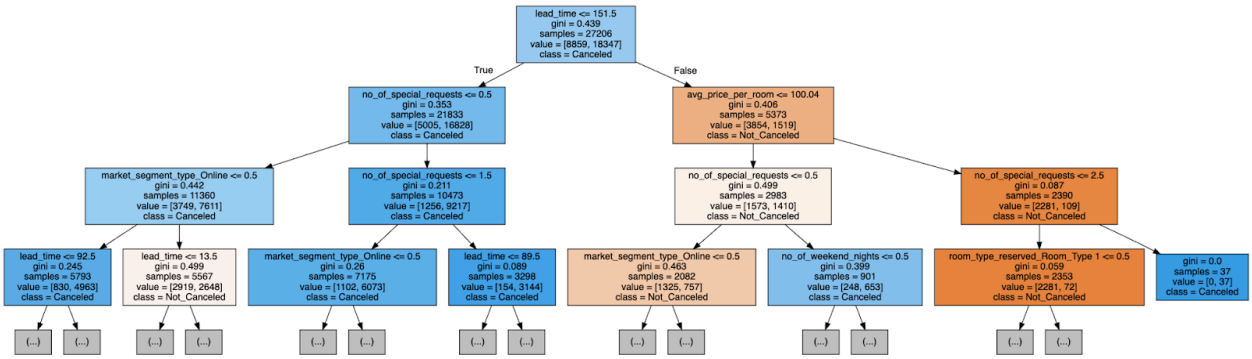
While the kernel trick does make the SVM less computationally expensive, using a polynomial kernel is still less efficient than a linear one, especially when the LinearSVC class is used. SVMs are generally best suited for small to medium-sized datasets, so the size of our dataset also indicates that there might be better models to use. The very low recall for canceled bookings in both

models is also very concerning, as being able to predict correctly that a canceled booking was in fact canceled would be very important to a hotel.

Since the accuracy and recall of the nonlinear SVM are still relatively low, we will continue exploring other classification models in order to find higher starting values, then we will seek to improve our model with techniques like hyperparameter tuning from there.

Decision Tree:

When training the decision tree, we wanted to be cautious of overfitting to ensure the model would perform well on unseen data. In order to help us achieve this, we used grid search to find the best hyperparameters for max_depth, min_samples_leaf, min_sample_split. Grid search found the best hyperparameters for max_depth, min_samples_leaf, min_sample_split to be 13, 1, and 2, respectively. Low values for min_samples_leaf and min_sample_split could be a bit concerning given the tens of thousands of instances in the training set. However, given the max_depth of 13, the tree would eventually be forced to generalize past a certain depth.




```
Top 5 features based on gini impurity:
lead_time 0.41825940588756305
avg_price_per_room 0.20487916192183514
market_segment_type_Online 0.1375665289201212
no_of_special_requests 0.09910823393573771
no_of_week_nights 0.03689049640385032
```

After training and visualizing the top layers of the decision tree, with a quick glance we can see that lead_time and avg_price_per_room are the two most important features based on gini impurity. This finding agrees with our feature importance findings earlier when using PCA.

```
confusion matrix:
[[2360  666]
[ 506 5537]]


classification report:
              precision    recall  f1-score   support

           0       0.82      0.78      0.80      3026
           1       0.89      0.92      0.90      6043

    accuracy                           0.87      9069
   macro avg       0.86      0.85      0.85      9069
weighted avg       0.87      0.87      0.87      9069

MSE: 0.1292314477891719
R2: 0.41874661423490767
MAE: 0.1292314477891719
```

Taking a look at the above metrics, although our model performed with 87% accuracy, we must take a closer look at other metrics to evaluate performance. As expected, the model performed better when predicting the more common not-canceled class. This comes through in the higher scores for precision and recall. However, taking a look at these same metrics for the canceled class, we see a 7% drop with regard to precision and a 14% drop with regard to recall when compared to the not-canceled class. Given the scope of the problem, it can be argued that recall with respect to a reservation being canceled is the most important metric as this is the percentage chance that a canceled reservation will correctly be identified by the model. The performance discrepancy between the two classes can be explained by the class size imbalances, and addressing this imbalance could be a good topic for future work.

Decision Tree using Principal Components:

```
confusion matrix:
[[1610 1416]
[ 439 5604]]


classification report:
              precision    recall  f1-score   support

           0       0.79      0.53      0.63      3026
           1       0.80      0.93      0.86      6043

    accuracy                           0.80      9069
   macro avg       0.79      0.73      0.75      9069
weighted avg       0.79      0.80      0.78      9069

MSE: 0.2045429485058992
R2: 0.08001277253050654
MAE: 0.2045429485058992
```

Using the two principal components from earlier, we wanted to see how a decision tree would perform compared to the non-reduced data. We employed the same methodology in training this decision tree and only changed the underlying data to use the two principal components. Although overall accuracy only dropped from 87% to 80%, the recall with regard to the canceled class dropped from 78% to 53%. We expected a decrease in the performance when using the reduced dataset, but not to this degree especially considering the 95% of variance captured by the two components. This model will only predict a cancellation when a cancellation will occur 53% of the time, which is not ideal for real world use. Going forward, we acknowledge PCA is a powerful pre-processing tool to simplify datasets with vast amounts of features, but it did not perform well for our use case. With our dataset, and even after one-hot encoding, we still had relatively few features, so we viewed it as an experimental step in our pre-processing..

Learnings

One of the first challenges we came across in the process was not a problem with any of the coding of the project or machine learning models: it was actually with GitHub. We found GitHub to be a little challenging to use when it came to branching, as not all of us had experience working with GitHub in groups. Some of us had not used branching before, so we came across a few merge conflicts that were a little frustrating, since it was hard to understand what was going wrong.

However, after the initial merge conflicts and troubles with branching, we figured out a good way for all of us to work together using our own branches to modify the same main repository and learned a lot about how branching and merging works within a project. This was definitely the first important thing we learned from this project.

Additionally, we learned a lot about the experience of going through a full project cycle. While we were all familiar with the different models and techniques we employed in the project, using them to create a cohesive result from start to finish was more difficult. We gained more experience with the process, beginning with finding and pre-processing a dataset, then analyzing it using the tools learned in class, and finally interpreting our results and suggesting action steps for further analysis. Knowing that we were doing a binary classification narrowed down our options for models, but it was difficult to anticipate which out of our options would work best, which is why we tested multiple models. If we had known decision trees would be the most effective going in, we could have allotted more time to improving that model; however, the process of testing a variety of models did help us understand each of them better. We've learned about how they compare to each other and which situations they are best suited for. Being responsible for the entirety of the project cycle meant that sometimes we would try techniques that didn't work as expected and that we had to do additional analysis to really understand our variables and their relation to our data and results. Our results apply specifically to the management of hotels, which is interesting information but not necessarily the most relevant to our lives. What was of highest value to us was learning how to effectively execute a machine learning project from start to finish, which this project has equipped us to do.

Conclusion

When selecting this dataset, our aim was to examine data on hotel reservations in order to predict which reservations are likely to cancel and which are not likely to cancel. Through the implementation of three classification models - logistic regression, support vector machines, and decision trees - along with supplementary insight from correlation analysis and principal component analysis, we have gained more knowledge about how to predict cancellations and what features are most important in doing so.

While all our models performed fairly well in this analysis, in terms of accuracy and recall (particularly for canceled bookings), the decision tree performed the best. While this model could be

used as is to start predicting hotel reservation cancellations, further refining this model for improved accuracy and recall would be helpful. As suggested earlier, addressing the imbalance between the classes would be a good first step.

Additionally, we've also gained insight into what features are most indicative of booking cancellations. Lead time was consistently the most influential - the further in advance a room was booked, the more likely it was to be canceled. Based on this, hotels might want to target customers reserving rooms in the near future. Further analysis into this variable would also be beneficial in order for hotels to respond effectively. For example, if customers with a lot of lead time cancel shortly before their booked date, this would be problematic for the hotel, as they may not have time to fill the room. On the other hand, if they both book early and cancel early, this may not have as much of an impact as long as the hotel has enough notice to fill the empty room. More research into which of these is more prominent would further help hotels make the most sound business decisions.

It is also important to consider the ethical implications of any analysis like this one. For a hotel, knowledge of the factors most likely to cause cancellation could impact the ways they advertise to consumers and even the way they price rooms or accept reservations. While this could be beneficial for the hotels and their business model, it could also negatively impact customers that the model predicts as being more likely to cancel. Across our models, recall was consistently better for non-cancelled bookings, so the risk of a model predicting that a customer will cancel and the customer not canceling is lower than that of the model failing to predict a customer cancellation, which does benefit the customer. Additionally, because the greatest predictor is lead time, using these results is less ethically concerning because a customer can, for the most part, control how far in advance they are booking a hotel room. Further ethical considerations depend on what approach a hotel wants to take - something like targeted advertising seems less problematic, while something like changing room pricing would be a greater cause for concern.

In our analysis, we have implemented binary classification via three different model types in order to predict the likelihood of a hotel reservation cancellation. We created preliminary models that could be used to predict this and also pinpointed the most notable features. We recognized that there are still substantial improvements that could be made to our models, and we suggested further areas for improvement, like addressing the imbalance between class sizes and more research into the lead time variable. Finally, we considered what we have learned throughout this project and the ethical implications of our findings.

References

1. Raza, Ahsan. “Hotel Reservations Dataset.” *Kaggle*, 2023, https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset. Accessed 1 April 2024.

2. Géron, Aurélien. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Third ed., O'Reilly, 2022.