Lab 3: Training a Model

Dataset

In this lab, we will use the New York City Airbnb Open Data:

https://raw.githubusercontent.com/fenago/datasets/main/AirBnB_NYC_2019.csv

We'll be working with the 'price' variable, and we'll transform it to a classification task.

Features

For the rest of the lab, you'll need to use the features from below. So the whole feature set will be set as follows:

'neighbourhood_group',
'room_type',
'latitude',
'longitude',
'price',
'minimum_nights',
'number_of_reviews',
'reviews_per_month',
'calculated_host_listings_count',
'availability_365'
Select only them and fill in the missing values with 0.

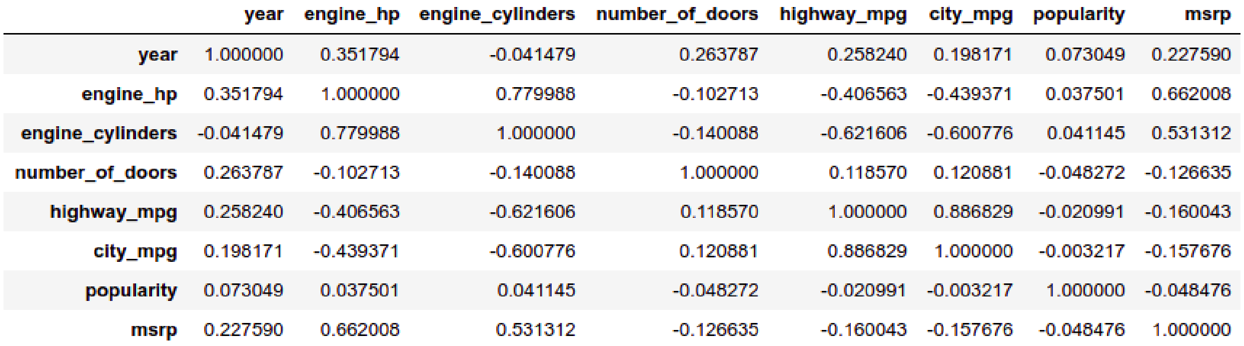
Question 1

What is the most frequent observation (mode) for the column 'neighbourhood_group'?

1. Split the data
2. Split your data in train/val/test sets, with 60%/20%/20% (or 80/20) distribution.
3. Use Scikit-Learn for that (the train_test_split function) and set the seed to 42.
4. Make sure that the target value ('price') is not in your dataframe.

Question 2

Create the correlation matrix for the numerical features of your train dataset.
In a correlation matrix, you compute the correlation coefficient between every pair of features in the dataset.
What are the two features that have the biggest correlation in this dataset?
Example of a correlation matrix for the car price dataset (I know this is not your dataset):



	year	engine_hp	engine_cylinders	number_of_doors	highway_mpg	city_mpg	popularity	msrp
year	1.000000	0.351794	-0.041479	0.263787	0.258240	0.198171	0.073049	0.227590
engine_hp	0.351794	1.000000	0.779988	-0.102713	-0.406563	-0.439371	0.037501	0.662008
engine_cylinders	-0.041479	0.779988	1.000000	-0.140088	-0.621606	-0.600776	0.041145	0.531312
number_of_doors	0.263787	-0.102713	-0.140088	1.000000	0.118570	0.120881	-0.048272	-0.126635
highway_mpg	0.258240	-0.406563	-0.621606	0.118570	1.000000	0.886829	-0.020991	-0.160043
city_mpg	0.198171	-0.439371	-0.600776	0.120881	0.886829	1.000000	-0.003217	-0.157676
popularity	0.073049	0.037501	0.041145	-0.048272	-0.020991	-0.003217	1.000000	-0.048476
msrp	0.227590	0.662008	0.531312	-0.126635	-0.160043	-0.157676	-0.048476	1.000000

Make price binary

We need to turn the price variable from numeric into binary.
Let's create a variable above_average which is 1 if the price is above (or equal to) 152.

Question 3

Calculate the mutual information score with the (binarized) price for the two categorical variables that we have. Use the training set only.
Which of these two variables has bigger score?
Round it to 2 decimal digits using round(score, 2)

Question 4

Now let's train a logistic regression
Remember that we have two categorical variables in the data. Include them using one-hot encoding.
Fit the model on the training dataset.
To make sure the results are reproducible across different versions of Scikit-Learn, fit the model with these parameters:
model = LogisticRegression(solver='lbfgs', C=1.0, random_state=42)
Calculate the accuracy on the validation dataset and round it to 2 decimal digits.

Question 5

We have 9 features: 7 numerical features and 2 categorical.
Let's find the least useful one using the feature elimination technique.

Train a model with all these features (using the same parameters as in Q4).
Now exclude each feature from this set and train a model without it. Record the accuracy for each model.
For each feature, calculate the difference between the original accuracy and the accuracy without the feature.
Which of following feature has the smallest difference?
neighbourhood_group
room_type
number_of_reviews
reviews_per_month
note: the difference doesn't have to be positive

Question 6

For this question, use the Classification template and identify the top 3 models based on ROC_AUC score.

Question 7 (Optional)

For this question, we'll see how to use a linear regression model from Scikit-Learn
We'll need to use the original column 'price'. Apply the logarithmic transformation to this column.
Fit the Ridge regression model on the training data.
This model has a parameter alpha. Let's try the following values: [0, 0.01, 0.1, 1, 10]
Which of these alphas leads to the best RMSE on the validation set? Round your RMSE scores to 3 decimal digits.
If there are multiple options, select the smallest alpha.

Bonus:

Try to exclude least useful features

Find the best regularization parameter for Ridge
There are other ways to implement one-hot encoding. E.g. using the OneHotEncoding class.
Sometimes numerical features requeire scaling, especially for iterative solves like "lbfgs". Other projects:

Lead scoring - https://www.kaggle.com/ashydv/leads-dataset
Default prediction - https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients