nsform: kaggle_brazil_houses_rental_data.csv	TARGET COLUMN	TYPE	DATASET	DATE
	rent amount (R$)	Regression	kaggle_brazil_houses_rental_data.csv	April 29, 2024 at 9:58 AM PDT

SUMMARY

Dataset statistics

Key	Value	Feature type	Count
Number of features	13	numeric	8
Number of rows	10692	categorical	2
Missing	0%	text	0
Valid	100%	datetime	0
Duplicate rows	5.65%	binary	2
		unknown	0

High Priority Warnings

2 high severity warnings were detected. See the list below.

Duplicate rowsHigh

We found that 5.65% of the data are duplicate. Some data sources could include valid duplicates and in other cases these duplicates could point to problems in data collection. Duplicate samples resulting from faulty data collection, could derail machine learning processes that rely on splitting to independent training and validation folds. For example quick model scores, prediction power estimation and automatic hyper parameter tuning. Duplicate samples could be removed from the dataset using the Drop duplicates transform under Manage rows.

Target leakageHigh

The feature fire insurance (R$) predicts the target extremely well on it's own. A feature this predictive often indicates an error called target leakage. The cause is typically data that is not available at time of prediction. For example, a duplicate of the target column in the dataset can result in target leakage. Alternatively, if the machine learning task is "easy", then a single feature can have legitimately high prediction power. If you think that a single feature is very highly predictive, you don't need to do anything further. However, if you think there's target leakage, we recommended that you remove the highly predictive column from the dataset using the Drop column transform under Manage columns.

DUPLICATE ROWS

Data Wrangler detected that 5.65% of the data are duplicate. Some data sources could include valid duplicates. Other data sources could have duplicates that point to problems in data collection. Duplicate samples that result from faulty data collection could interfere with machine learning processes that rely on splitting the data into independent training and validation folds.

The following are examples of processes that can suffer from duplicated samples:

- Quick model analysis
- Prediction power estimation
- Hyper-parameters optimization

The most common duplicate rows are presented below. The number of occurrences of a row is given in left most column named Duplicate count.

You can remove duplicate samples from the dataset using the Drop duplicates transform under Manage rows.

Note: features of type vector are ignored in duplicate rows detection

Duplicate count	city	area	rooms	bathroom	parking spaces
22	Porto Alegre	47	1	1	1
14	São Paulo	20	1	1	0
9	São Paulo	45	1	1	1
7	São Paulo	35	1	1	0
7	São Paulo	20	1	1	0
7	São Paulo	40	1	1	0
6	São Paulo	50	1	1	0
6	São Paulo	35	1	1	0
5	Campinas	110	3	3	2
5	São Paulo	50	1	1	0

Duplicate rowsHigh

We found that 5.65% of the data are duplicate. Some data sources could include valid duplicates and in other cases these duplicates could point to problems in data collection. Duplicate samples resulting from faulty data collection, could derail machine learning processes that rely on splitting to independent training and validation folds. For example quick model scores, prediction power estimation and automatic hyper parameter tuning. Duplicate samples could be removed from the dataset using the Drop duplicates transform under Manage rows.

ANOMALOUS SAMPLES

Data Wrangler detects anomalous samples using the Isolation forest algorithm after basic preprocessing. The isolation forest associates an anomaly score to each sample (row) of the dataset.

- Low anomaly scores indicate anomalous samples.
- High scores are associated with non-anomalous samples.
- Samples with negative anomaly score are usually considered anomalous and samples with positive anomaly score are considered non-anomalous.

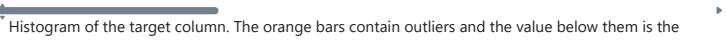
When you look at a sample that might be anomalous, we recommend that you pay attention to unusual values. For example, you might have anomalous values that result from errors in gathering and processing the data. The following is an example of the most anomalous samples according to the Data Wrangler's implementation of the isolation forest algorithm. We recommend using domain knowledge and business logic when you examine the anomalous samples.

Anomaly scores	city	area	rooms	bathroom	parking spaces
-0.208	São Paulo	884	5	7	6
-0.207	São Paulo	700	4	7	8

-0.204	Belo Horizonte	758	5	4	5
-0.201	São Paulo	890	5	6	8
-0.199	São Paulo	600	6	7	4
-0.199	São Paulo	856	5	7	6
-0.198	São Paulo	998	7	10	4
-0.197	São Paulo	850	6	7	4
-0.194	São Paulo	900	4	9	8
-0.192	Rio de Janeiro	35	1	1	0

TARGET COLUMN

key	value
Valid	100%
Missing	0%
Outliers	0.103%
Min	450
Max	4.5e+04
Mean	3.9e+03
Median	2.66e+03
Skew	1.84
Kurtosis	4.62
Number of unique	1195


Histogram of the target column. The orange bars contain outliers and the value below them is the outliers average.

See below several samples with outlier target values.

city	area	rooms	bathroom	parking spaces	floor
São Paulo	700	4	7	8	-
São Paulo	350	3	3	3	-
São Paulo	486	8	4	6	-
São Paulo	80	2	1	1	1
São Paulo	900	3	4	8	-

Outliers in targetMedium

The target column contains a few outliers. They are probably harmless, however, they might point to bugs in data collection or processing. Because the outliers induce high errors during model training the machine learning algorithms tend to focus on them. Thus, you might get poor prediction quality for the non-outlier samples. In case you are interested in predicting extreme values well or plan to use a machine learning algorithm that has the ability to handle outlier values there is no need for further action. However, if extreme values are not the point of interest consider removing or clipping them using the Robust standard deviation numeric outliers transform under Handle outliers.

QUICK MODEL

Quick model provides an estimate of the expected predicted quality of a model that you train on your data.
The data is split into training and validation folds where Data Wrangler uses 80% of the samples for training and 20% of the values for validation. For classification the sample is stratified split. For a stratified split, each data partition has the same ratio of labels. For classification problems, it's important to have the same ratio of labels between the training and classification folds. Data Wrangler trains the XGBoost model with the default hyper-parameters. It applies early stopping on the validation data and performs minimal feature pre-processing.

Metric	Validation scores	Train scores
R2	0.988	1
MSE	1.54e+05	1.82e+03
RMSE	392	42.6
MAE	60	27
Max error	1.67e+04	558
Median absolute error	24.2	18.7

FEATURE SUMMARY

See a summary of the features ordered by the prediction power. Prediction power is measured by stratified splitting the data into 80%/20% training and validation folds.
We fit a model for each feature separately on the training fold after applying minimal feature pre-processing and measure prediction performance on the validation data.

- The scores are normalized to the range [0,1].
- Higher prediction power scores, toward 1, indicate columns that are more useful for predicting the target on their own.
- Lower scores, toward 0 point to columns that contain little useful information for predicting the target on their own. Although it can happen that a column is uninformative on its own but is useful in predicting the target when used in tandem with other features, a low score usually indicates the feature is redundant.

- A score of 1 implies perfect predictive abilities, which often indicates an error called target leakage. The cause is typically a column that will not be available at prediction time such as a duplicate of the target.

Feature	Prediction power	Type	Valid	Missing	High severity wa
fire insurance (R$)	0.979	numeric	100%	0%	1
total (R$)	0.863	numeric	100%	0%	0
area	0.514	numeric	100%	0%	0
property tax (R$)	0.488	numeric	100%	0%	0
bathroom	0.46	numeric	100%	0%	0
parking spaces	0.379	numeric	100%	0%	0
hoa (R$)	0.35	numeric	100%	0%	0
rooms	0.314	numeric	100%	0%	0
city	0.0712	categorical	100%	0%	0
floor	0.0597	categorical	100%	0%	0

Prediction power of the features. Higher prediction power scores, toward 1, indicate columns that are more useful for predicting the target on their own. Lower scores, toward 0 point to columns that contain little useful information for predicting the target on their own.

Target leakageHigh

The feature fire insurance (R$) predicts the target extremely well on it's own. A feature this predictive often indicates an error called target leakage. The cause is typically data that is not available at time of prediction. For example, a duplicate of the target column in the dataset can result in target leakage. Alternatively, if the machine learning task is "easy", then a single feature can have legitimately high prediction power. If you think that a single feature is very highly predictive, you don't need to do anything further. However, if you think there's target leakage, we recommended that you remove the highly predictive column from the dataset using the Drop column transform under Manage columns.

FEATURE DETAILS

fire insurance (R$)

numeric

key	value
Feature name	fire insurance (R
Type	numeric
Prediction power	0.979
Valid	100%
Missing	0%
Outliers	0.327%
Min	3
Max	677
Mean	53.3
Median	36

Histogram of fire insurance (R$) with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

Target leakageHigh

The feature fire insurance (R$) predicts the target extremely well on it's own. A feature this predictive often indicates an error called target leakage. The cause is typically data that is not available at time of prediction. For example, a duplicate of the target column in the dataset can result in target leakage. Alternatively, if the machine learning task is "easy", then a single feature can have legitimately high prediction power. If you think that a single feature is very highly predictive, you don't need to do anything further. However, if you think there's target leakage, we recommended that you remove the highly predictive column from the dataset using the Drop column transform under Manage columns.

total (R$)

numeric

key	value
Feature name	total (R$)
Type	numeric
Prediction power	0.863
Valid	100%
Missing	0%
Outliers	0.402%
Min	499
Max	1.12e+06
Mean	5.49e+03
Median	3.58e+03

Histogram of total (R$) with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

area

numeric

key	value
Feature name	area
Type	numeric

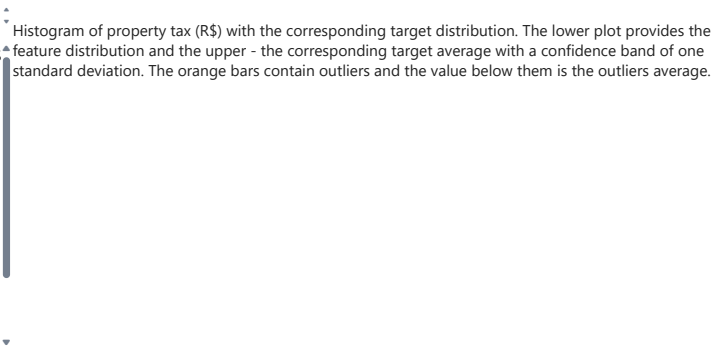
Histogram of area with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

Valid	100%
Missing	0%
Outliers	1.23%
Min	11
Max	4.63e+04
Mean	149
Median	90

property tax (R$)

numeric

key	value
Feature name	property tax (R$)
Type	numeric
Prediction power	0.488
Valid	100%
Missing	0%
Outliers	2.34%
Min	0
Max	3.14e+05
Mean	367
Median	125

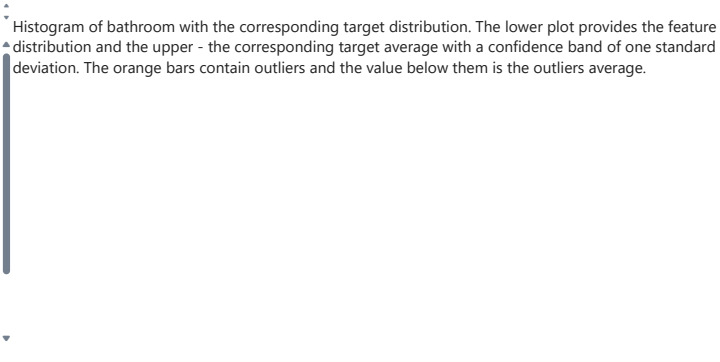


Histogram of property tax (R$) with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

bathroom

numeric

key	value
Feature name	bathroom
Type	numeric
Prediction power	0.46
Valid	100%
Missing	0%
Outliers	0.0655%
Min	1
Max	10
Mean	2.24
Median	2

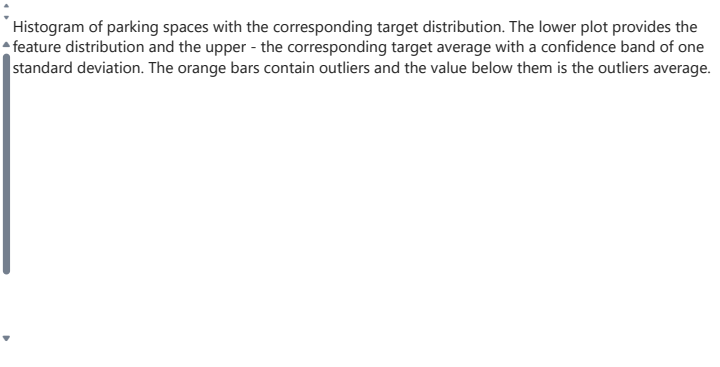


Histogram of bathroom with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

parking spaces

numeric

key	value
Feature name	parking spaces
Type	numeric
Prediction power	0.379
Valid	100%
Missing	0%
Outliers	0.0281%
Min	0
Max	12
Mean	1.61
Median	1

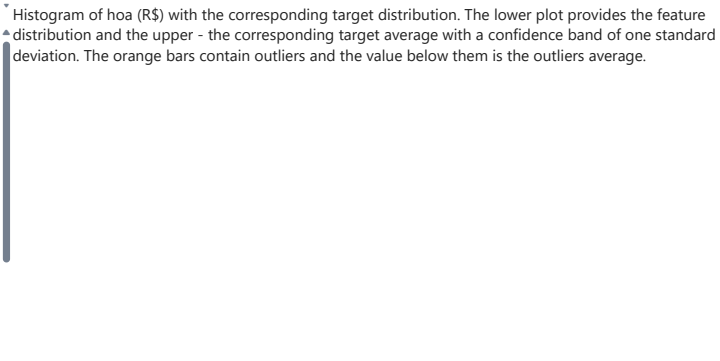


Histogram of parking spaces with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

hoa (R$)

numeric

key	value
Feature name	hoa (R$)
Type	numeric
Prediction power	0.35
Valid	100%
Missing	0%
Outliers	0.973%
Min	0
Max	1.12e+06



Histogram of hoa (R$) with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

Median	560

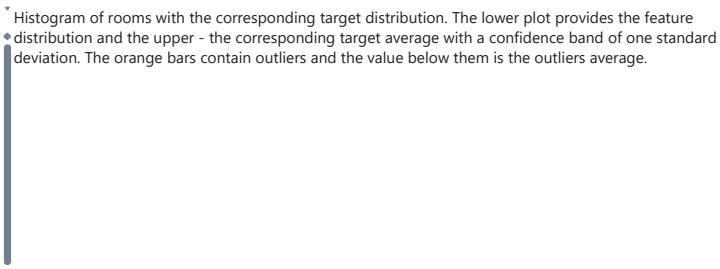
Disguised missing valueMedium

The frequency of the value "0" in the feature hoa (R$) is 22.2% which is uncommon for numeric features. This could point to bugs in data collection or processing. In some cases the frequent value is a default value or a placeholder to indicate a missing value. If that is the case, it is recommended to replace this value with NaNs using the Convert regex to missing transform under Search and edit.

rooms

numeric

key	value
Feature name	rooms
Type	numeric
Prediction power	0.314
Valid	100%
Missing	0%
Outliers	0.15%
Min	1
Max	13
Mean	2.51
Median	2

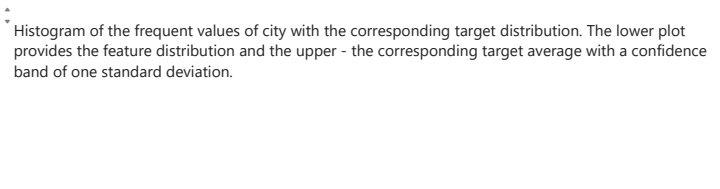


Histogram of rooms with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation. The orange bars contain outliers and the value below them is the outliers average.

city

categorical

key	value
Feature name	city
Type	categorical
Prediction power	0.0712
Valid	100%
Missing	0%

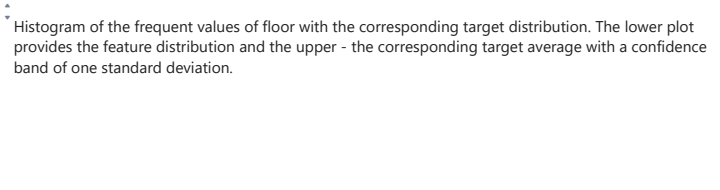


Histogram of the frequent values of city with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation.

floor

categorical

key	value
Feature name	floor
Type	categorical
Prediction power	0.0597
Valid	100%
Missing	0%

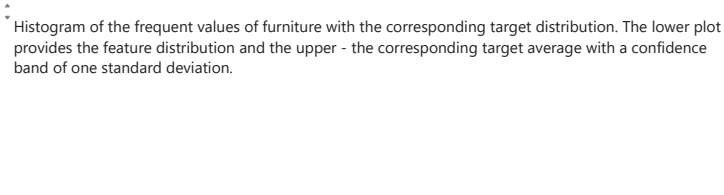


Histogram of the frequent values of floor with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation.

furniture

binary

key	value
Feature name	furniture
Type	binary
Prediction power	0.0328
Valid	100%
Missing	0%

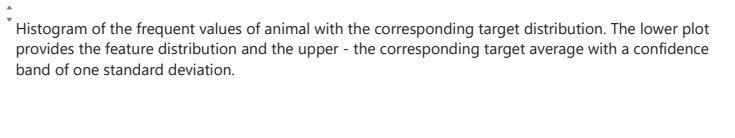


Histogram of the frequent values of furniture with the corresponding target distribution. The lower plot provides the feature distribution and the upper - the corresponding target average with a confidence band of one standard deviation.

animal

binary

key	value
Feature name	animal
Type	binary
Prediction power	0.0028
Valid	100%
Missing	0%



DEFINITIONS

Feature types

Numeric: Numeric values, either floats or integers. For example: age, income. The machine learning models assume that numeric values are ordered and a distance is defined over them, that is 3 is closer to 4 than to 10 and 3 < 4 < 10.
Categorical: The column entries belong to a set of unique values which is usually much smaller than the number of entries in the column. For example, a column of length 100 containing the unique values "Dog", "Cat" and "Mouse". The values could be numeric, text of combination of both e.g. "Horse", "House", 8, "Love" and 3.1 are all valid values and can be found in the same categorical column. As opposed to numeric features, the machine learning model does not assume order or distance on the values of a categorical feature (even when all the values are numbers).
Binary: A special case of categorical feature where the cardinality of the set of unique values is 2.
Text: A text column contains many non-numeric unique values. In extreme cases, all the elements of the column are unique i.e., no two entries are the same.
Datetime: This column contains date and/or time information.

Feature Statistics

PREDICTION POWER
Outliers (in numeric features and regression target): Outliers are detected using two statistics that are robust to outliers: median and robust standard deviation (RSTD). RSTD is derived by clipping the feature values to the range [5 percentile, 95 percentile] and calculating the standard deviation of the clipped vector. All values larger than median + 5 * RSTD or smaller than median - 5 * RSTD are considered to be outliers.
Skew (in numeric features and regression target): Skew measures the symmetry of the distribution and is defined as the third moment of the distribution divided by the third power of the standard deviation. The skewness of the normal distribution or any other symmetric distribution is zero. Positive values imply that the right tail of the distribution is longer than the left tail and negative values - vice versa. As a thumb rule, a distribution is considered skewed when the absolute value of the skew is larger than 3.
Kurtosis (in numeric features and regression target): Pearson's kurtosis measures the heaviness of the tail of the distribution and is defined as the fourth moment of the distribution divided by the square of the second moment. The kurtosis of the normal distribution is 3. Thus, kurtosis values lower than 3 imply that the distribution is concentrated around the mean and the tails are lighter than the tails of the normal distribution. On the other hand, Kurtosis values higher than 3 imply heavier tails and/or outliers.
VALUES
Numeric features / regression target: All values that could be casted to finite floats are valid. Missing values are not valid.
Categorical / binary / text features / classification target: All values that are not missing are valid.
Datetime features: All values that could be casted to datetime object are valid. Missing values are not valid.
Invalid values: Either missing or could not be casted to the desired type. See the definition of valid values for more information

Quick model metrics

REGRESSION
R2 (coefficient of determination): R2 is the proportion of the variation in the target that is predicted by the model. R2 is in the range of [-infty, 1] where 1 is the score of the model that predicts the target perfectly and 0 is the score of the trivial model that always predicts the target mean.
MSE: Mean squared error. MSE is in the range [0, infty] where 0 is the score of the model that predicts the target perfectly
MAE: Mean absolute error. MAE is in the range [0, infty] where 0 is the score of the model that predicts the target perfectly
RMSE: Root mean square error. RMSE is in the range [0, infty] where 0 is the score of the model that predicts the target perfectly
Median absolute error: In the range [0, infty] where 0 is the score of the model that predicts the target perfectly
CLASSIFICATION
Accuracy: The ratio of samples that are predicted accurately. Accuracy is in the range [0, 1] where 0 is the score of the model that predicts all samples wrong and 1 is the score of the perfect model.
Balanced accuracy: The ratio of samples that are predicted accurately when class weights are adjusted to balance the data. That is, all classes are given the same importance, regardless of their frequency. Balanced accuracy is in the range [0, 1] where 0 is the score of the model that predicts all samples wrong and 1 is the score of the perfect model.
ROC-AUC: (Binary classification) Area under the receiver operating characteristic curve. ROC-AUC is in the range [0, 1] where a random model will yield a score of 0.5 and the perfect model 1.
ROC-AUC (OVR): (Multi-class classification) Area under the receiver operating characteristic curve calculated separately for each label using one vs rest and the average is reported. ROC-AUC is in the range [0, 1] where a random model will yield a score of 0.5 and the perfect model 1.
Precision: Precision is defined for a specific class. Precision is the fraction of true positives out of all the instances that the model classified as that class. Precision is in the range [0, 1] where 1 is the score of the model that have no false positives for the class. For binary classification, we report the precision of the positive class.
Recall: Recall is defined for a specific class. Recall is the fraction of the relevant class instances that are successfully retrieved. Recall is in the range [0, 1] where 1 is the score of the model that classifies correctly all the instances of the class. For binary classification, we report the recall of the positive class.
F1: F1 is defined for a specific class and it is the harmonic mean of the precision and recall. F1 is in the range [0, 1] where 1 is the score of the perfect model. For binary classification, we report F1 for the positive class.

Pattern Learning

TEXTUAL PATTERNS
Patterns describe the textual format of a string using an easy to read format. The following are examples of textual patterns:

- "{digits:4-7}" describes a sequence of digits of length between 4 and 7.
- "{alnum:5}" describes an alpha-numeric string of length exactly 5.

Data Wrangler infers the patterns by looking at samples of **non-empty** strings from your data. It can describe many of the commonly used patterns.
The **relevance** (%) indicates how much of the data is estimated to match the pattern. Using the textual pattern, you can see which rows in your data you need to correct or drop.
The following describes the patterns that Data Wrangler can recognize:

Pattern	Textual Format
{alnum}	alphanumeric strings
{any}	any string of word characters
{digits}	a sequence of digits
{lower}	a lowercase word
{mixed}	a mixed-case word
{name}	a word beginning with a capital letter
{upper}	an uppercase word
{whitespace}	whitespace characters

A word character is either an underscore or a character that may appear in a word in any language. For example, the strings ‘Hello_world’ and ‘écoute’ both consist of word characters. ‘H’ and ‘é’ are both examples of word characters.