COMP47350: Data Analytics (Conv)

Dr. Georgiana Ifrim
georgiana.ifrim@ucd.ie

Insight Centre for Data Analytics
School of Computer Science
University College Dublin

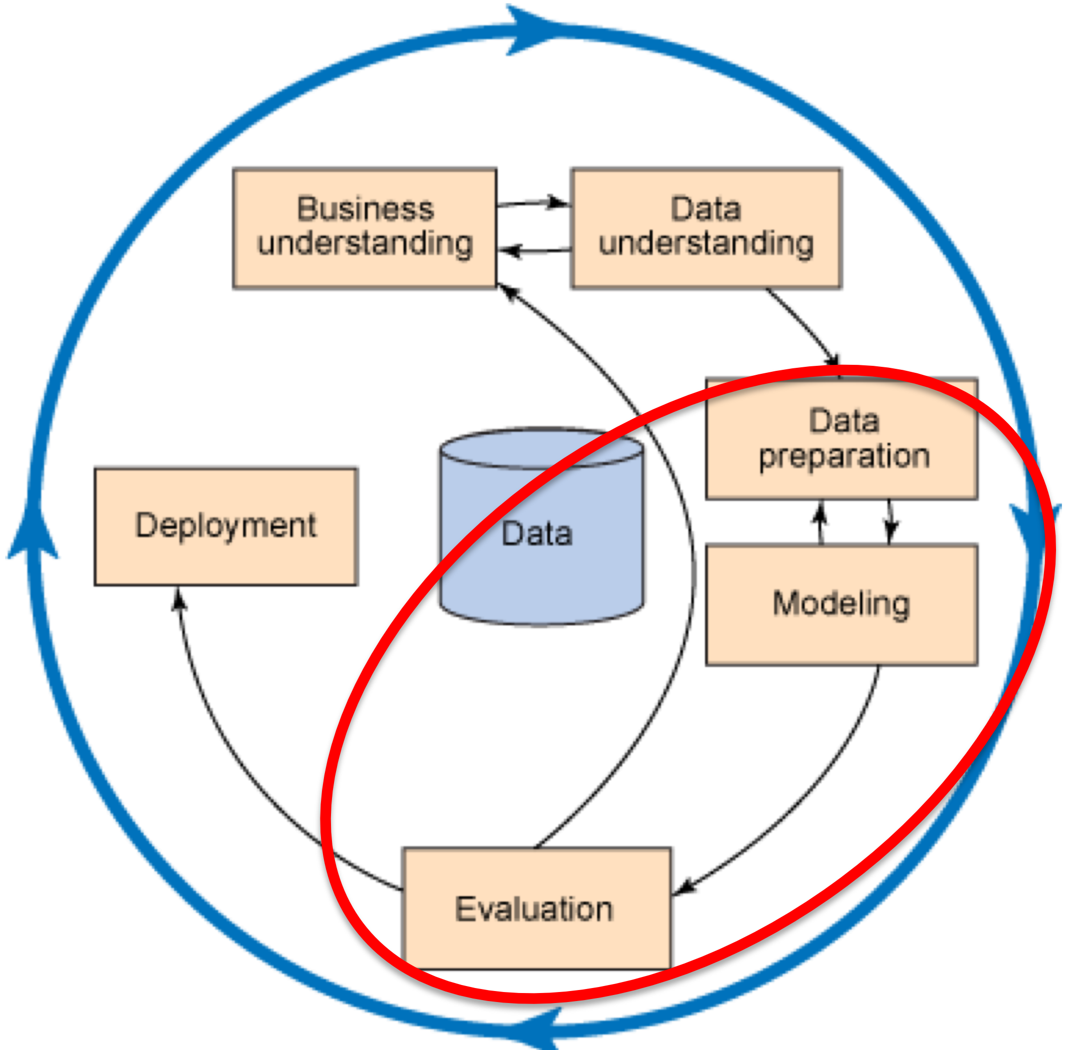
2018/19

Module Topics

- **Python Environment** (Anaconda, Jupyter Notebook)
- **Getting Data** (Web scrapping, APIs, DBs)
- **Understanding Data** (slicing, visualisation)
- **Preparing Data** (cleaning, transformation)
- **Modeling & Evaluation** (machine learning)

Data Analytics Project Lifecycle: CRISP-DM Methodology

CRISP-DM: CRoss-Industry Standard Process for Data Mining



Modeling Data

- Modeling:
  - How to build prediction models
  - **How to evaluate prediction models**

Model Evaluation

Experiment Design

- Underfitting/Overfitting
- Out-of-sample Testing
- K-fold Cross-Validation

Evaluation Metrics

- **Regression** (evaluation for models that predict a numeric target feature)
- **Classification** (evaluation for models that predict a categorical target feature)

Model Evaluation

Very important for the design of an evaluation experiment

- To avoid overfitting:
  - Make sure the data used to evaluate the model is not the same as the data used to train the model **(rote-learning vs generalized knowledge)!**
  - Always evaluate the trained model on a **realistic hold-out test set**: the test set should be similar to the real setting (e.g., balanced vs unbalanced, available features at prediction time, etc.).

Model Evaluation

The purpose of evaluation is threefold:

1. To determine which model/algo is the most suitable for a task (problem modeling)

2. To estimate how the model will perform when deployed (generalization on unseen test set)

3. To convince users that the model will meet their needs (e.g., accuracy, train/test efficiency, interpretability)

Model Evaluation

Model accuracy (usually the main focus)

- The best model depends on the application and the evaluation metric used.
- What is meant by <u>good predictive performance</u>? What is the right evaluation metric to optimize? The Accuracy metric is not always appropriate.

Other important properties: Model efficiency, interpretability, concept drift

- <u>Efficiency:</u> the model is too slow (runtime for train/test takes hours or weeks!) or takes too much memory (RandomForest trained model is 10Gb in size! Need to load model in memory to make predictions)
- <u>Interpretation:</u> the model is a black-box, hard to understand the model or the prediction
- <u>Concept drift</u>: the model goes stale and needs to be re-trained often

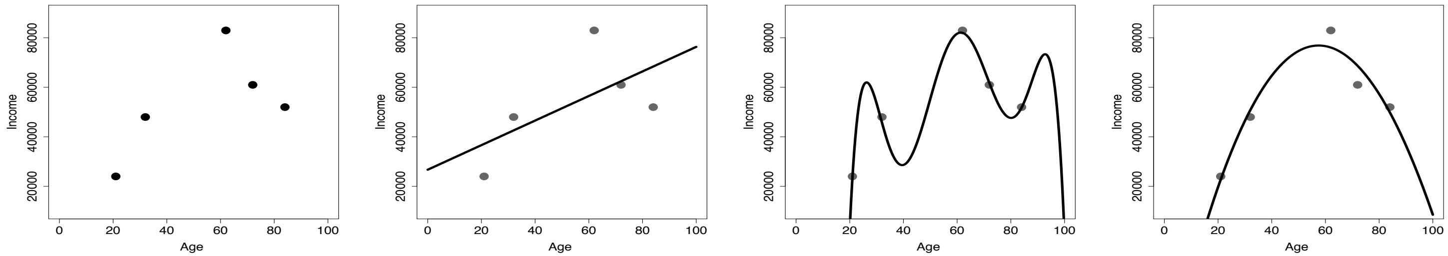
Overfitting

- How low can we push the training error?
  - We can make the model arbitrarily complex (effectively “memorizing” the entire training set)
  - Example: Rote learning - memorizing a set of questions and their answers on training set
  - We can push the training error down to zero! This problem is called **overfitting.**

- Training error is not a good estimate of accuracy beyond training data. It is important to measure error on new test data to check if the algo can generalize to new data.
  - **Generalizing Ability** – Can you apply the learned concept to a new set of (similar) questions? (a good exam should test generalization ability not rote learning)
  - Can the model predict well on a new set of data from the same population?

Underfitting/Overfitting

- **Underfitting:** Simple models may not capture the underlying relationship between descriptive and target feature.
- **Overfitting:** Complex models may pick up unimportant details in the training data.
- Using hold-out test sets (where we know the true labels) we try to estimate the out-of-sample (OOS) error (i.e., how well will the model do in the real world, on new examples).



(a) Dataset (b) Underfitting (c) Overfitting (d) Just right

Figure: Striking a balance between overfitting and underfitting when trying to predict age from income.

Model Fitting/Evaluation

- **Evaluation using only one split of the dataset**
- Typically used when lots of data available.

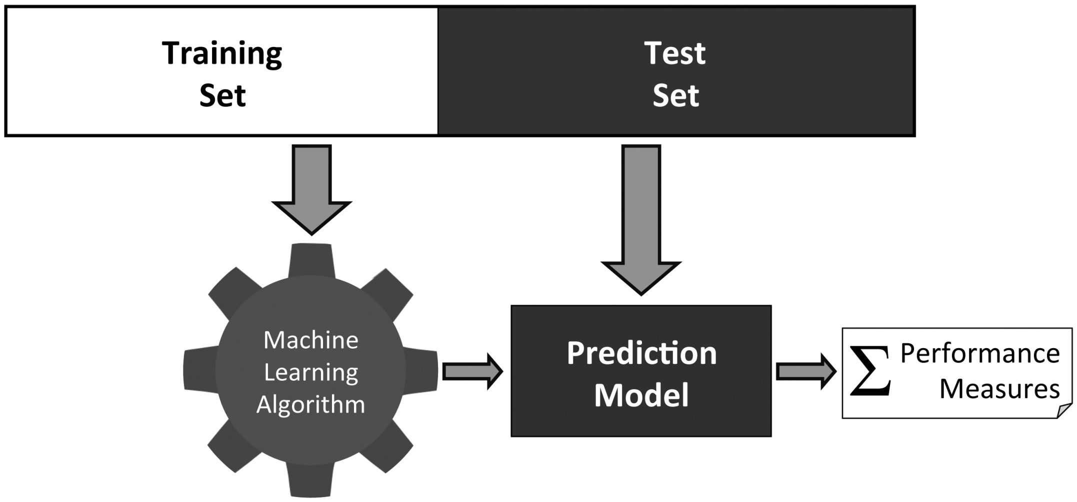


Figure: The process of building and evaluating a model using a **hold-out test set**.

Model Fitting/Evaluation

- Evaluation using only one split of the dataset (train/validation/test)
- Typically used when lots of data available.

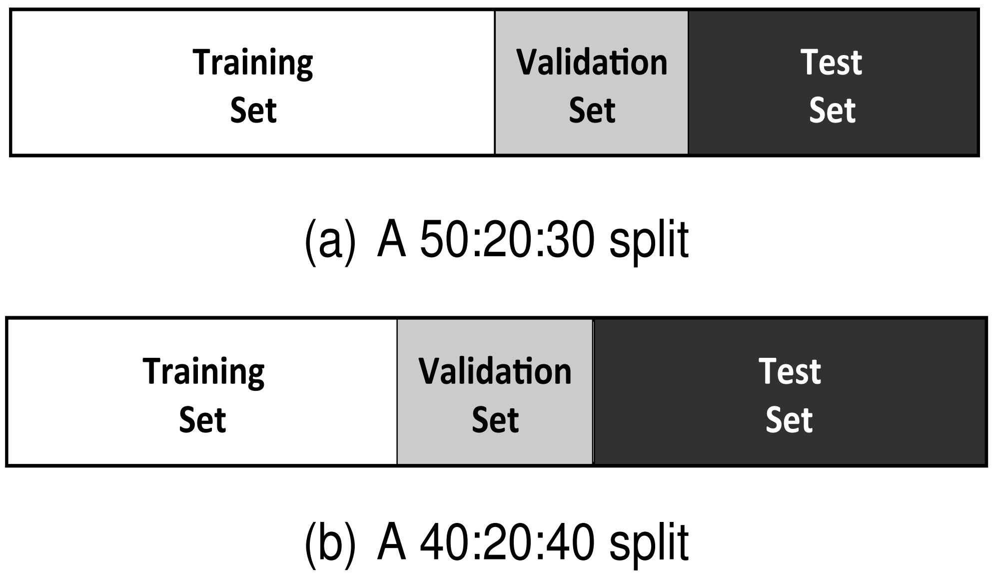


Figure: Hold-out sampling can divide the full data into training, validation, and test sets.

Model Fitting/Evaluation

Evaluation using only one split of the dataset

1. Given a labeled dataset, randomly shuffle the rows of the dataset.
2. Split dataset into: training/validation/test datasets.
3. For each model and set of parameters, repeat: train model, check error on validation set (aka parameter tuning, model selection).
4. Select the model + parameters with the lowest error on validation set.
5. Retrain best model on full training + validation data.
6. Evaluate final model on test data.

Model Fitting/Evaluation

Example: Evaluate and compare 2 linear regression models: LR with 10 features (LR10) or LR with 5 features (LR5).

Evaluation using only one split of the dataset

1. Given a labeled dataset, randomly shuffle the rows of the dataset.
2. Split dataset into: training/validation/test datasets.
3. For

   3.1 LR10: train model, check error on validation set. Accuracy is 0.80.

   3.2 LR5: train model, check error on validation set. Accuracy is 0.95.
4. Select LR5 as better model.
5. Retrain LR5 on full training + validation data.
6. Evaluate LR5 (retrained at point 5) on test data.

Model Fitting/Evaluation

Evaluation using multiple splits of the dataset

Example: 10-fold Cross-validation (90% train, 10% test)

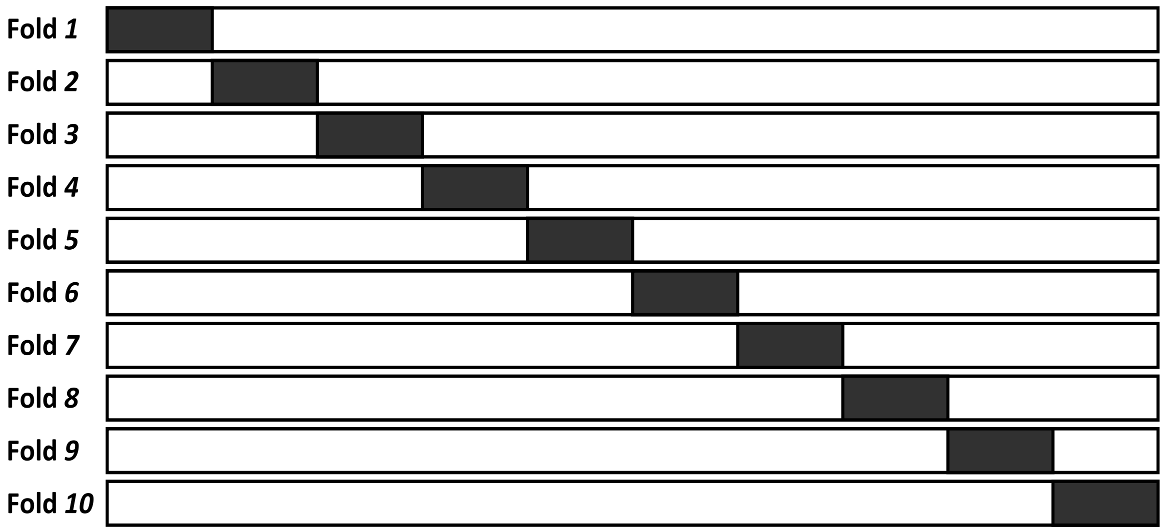


Figure: The division of data during the ***k*-fold cross validation** process. Black rectangles indicate test data, and white spaces indicate training data.

Model Fitting/Evaluation

Cross-validation

- A single train/test split may be misleading (one time lucky!)
- Do repeated splits and average the error on the test datasets

Steps for **K-fold cross-validation**:

1. Randomly split the dataset into K equal partitions.
2. Use partition 1 as test set & union of other partitions as training set.
3. Calculate error on test set.
4. Repeat steps 2-3 using a different partition as the test set at each iteration.
5. Take the average test set error as the estimate of OOS accuracy.

Model Fitting/Evaluation

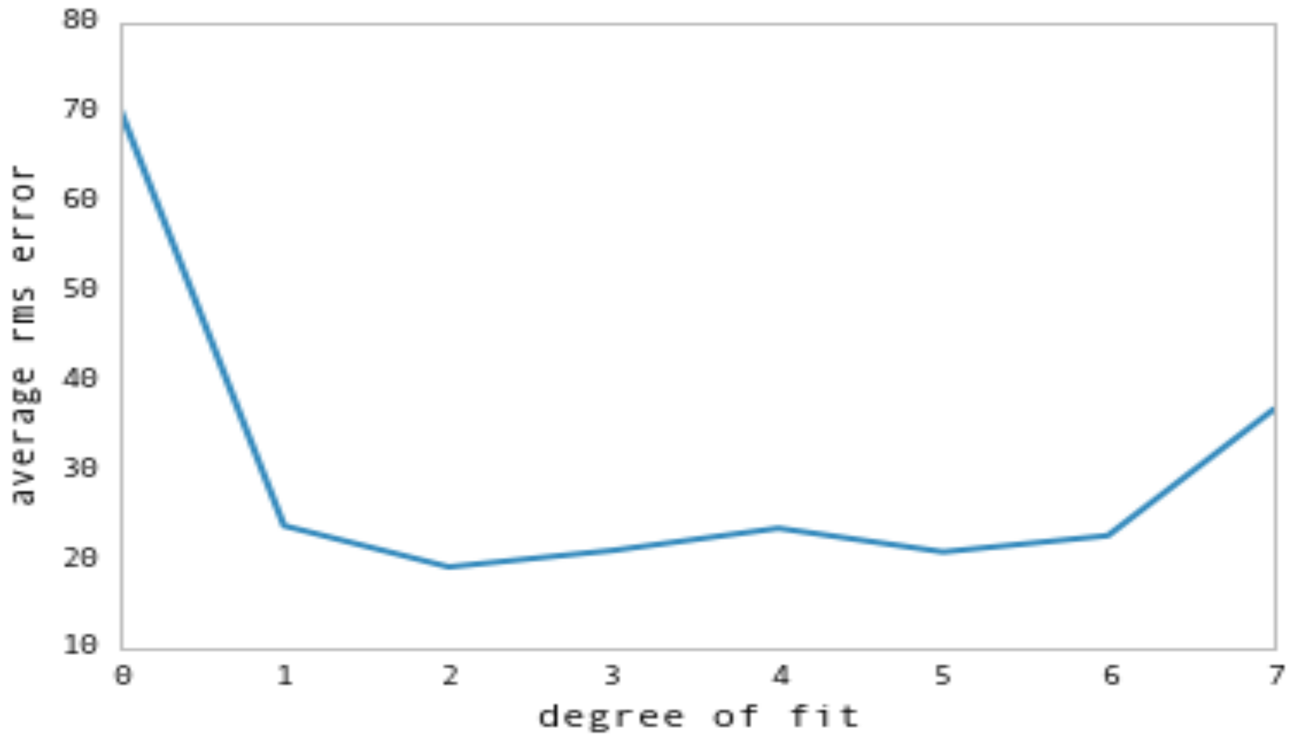
Features of K-fold cross-validation:

1. More accurate estimate of out-of-sample prediction error (i.e., error on unseen data).
2. More efficient use of data than single train/test split.
   - Each record in our dataset is used for both training and testing.
3. Presents tradeoff between accuracy estimate and efficiency
   - 10-fold CV is 10x more expensive than a single train/test split
   - 5-fold CV also popular
   - If data is small, LOO (leave-one-out) CV is recommended.

Model Fitting/Evaluation

Example model selection using cross-validation:

- Checking average test error using CV, for different polynomial models (linear, quadratic, etc)
- Lowest avg test error is at polynomial of degree = 2, so we select this as best model (e.g., quadratic features work best)



Cross-Validation

- If data has a special structure, random shuffling and split is not a good idea
- For example, in time series data, we need to train on past data and test on future data (we shouldn’t mix past and future during CV, need to avoid shuffling)

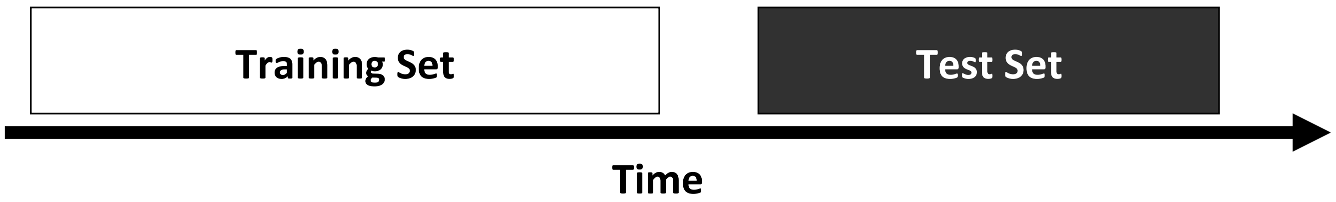


Figure: The **out-of-time sampling** process.

Model Evaluation

Experiment Design

- Underfitting/Overfitting
- Out-of-sample Testing
- Cross-Validation

Evaluation Metrics

- **Regression** (numeric target)
- Classification (class target)

Model Evaluation

Regression evaluation measures:

- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- R^2

Regression: Evaluation Metrics

Root Mean Squared Error (RMSE)

n = number of examples

y_j = true value of the target feature

ŷ_j = predicted value of the target feature

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Regression: Evaluation Metrics

Root Mean Squared Error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

- Root of average squared error
- Used for regression problems
- Easily interpretable (in the "y" units)
- "Punishes" larger errors (high penalty of outlier predictions)

Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

- Average of absolute errors
- Easily interpretable (in the "y" units)
- Less penalty for big outliers in the predictions
- RMSE sometimes recommended over MAE as it is more pessimistic (RMSE slightly overestimates the prediction error, so it is useful when large errors are undesirable)

Linear Regression: RMSE vs MAE

Table: Calculating the sum of squared errors for the candidate model (with $\mathbf{w}[0] = 6.47$ and $\mathbf{w}[1] = 0.62$) making predictions for the the office rentals dataset.

ID	RENTAL PRICE	Model Prediction	Error	Squared Error
1	320	316.79	3.21	10.32
2	380	347.82	32.18	1,035.62
3	400	391.26	8.74	76.32
4	390	397.47	-7.47	55.80
5	385	419.19	-34.19	1,169.13
6	410	440.91	-30.91	955.73
7	480	484.36	-4.36	19.01
8	600	552.63	47.37	2,243.90
9	570	577.46	-7.46	55.59
10	620	627.11	-7.11	50.51
			Sum	**5,671.64**
	Sum of squared errors (Sum/2)			**2,835.82**

MSE = average over the SquaredError column

RMSE = root of MSE

MAE = take absolute values for the Error column, then average them

MSE: 567.19, RMSE: 23.81 (about 24 euro off), MAE: 18.30 (about 18 euro off)

Regression: Evaluation Metrics

- Many other metrics: Mean Absolute Percentage Error (MAPE, WeightedMAPE), AIC, BIC, etc.
- RMSE, MAE and R^2 most popular, depending on the community
- RMSE, MAE are domain dependent (need to have an understanding of what the units mean); lower is better
- R^2 is domain-independent, higher is better

Regression: Evaluation Metrics

- R^2 is domain-independent (generally in [0,1] range)

$$R^2 = 1 - \frac{\text{sum of squared errors}}{\text{total sum of squares}}$$

Predictions of our model

Predictions of average model

$$\text{total sum of squares} = \frac{1}{2}\sum_{i=1}^{n}(t_i - \bar{t})^2$$

Average model: always predict the average of target feature, computed over training examples

Linear Regression: RMSE vs MAE vs R^2

Table: Calculating the sum of squared errors for the candidate model (with $\mathbf{w}[0] = 6.47$ and $\mathbf{w}[1] = 0.62$) making predictions for the the office rentals dataset.

ID	RENTAL PRICE	Model Prediction	Error	Squared Error
1	320	316.79	3.21	10.32
2	380	347.82	32.18	1,035.62
3	400	391.26	8.74	76.32
4	390	397.47	-7.47	55.80
5	385	419.19	-34.19	1,169.13
6	410	440.91	-30.91	955.73
7	480	484.36	-4.36	19.01
8	600	552.63	47.37	2,243.90
9	570	577.46	-7.46	55.59
10	620	627.11	-7.11	50.51
			Sum	**5,671.64**
			Sum of squared errors (Sum/2)	**2,835.82**

- R^2 does not tell us anything about the original units, but it gives an indication of how much better is our model vs simply predicting the average of the past targets
- Average price over training set: 455.5
- Sum of squared errors: 5671.64 ; Total sum of errors: 100122.5
- **R^2:** 1 – (5671.64 / 100122.5) = **0.94**
- **RMSE: 23.81 (about 24 euro off), MAE: 18.30 (about 18 euro off)**

Regression: Evaluation Metrics R^2

From Lab7-LinearRegression-updated

```
 Sum of squared errors:
 5671.9405389423

AverageModelPredictions:
 [455.5 455.5 455.5 455.5 455.5 455.5 455.5 455.5 455.5 455.5]
Actual - AvgPredictions:
 0   -135.5
1    -75.5
2    -55.5
3    -65.5
4    -70.5
5    -45.5
6     24.5
7    144.5
8    114.5
9    164.5
Name: RentalPrice, dtype: float64

(Actual - AvgPredictions) squared:
 0    18360.25
1     5700.25
2     3080.25
3     4290.25
4     4970.25
5     2070.25
6      600.25
7    20880.25
8    13110.25
9    27060.25
Name: RentalPrice, dtype: float64

 Total sum of squared errors:
 100122.5

 R2:
 0.9433499908717591
```

Regression: Evaluation Metrics

Some issues with R^2

- R^2 is domain-independent, so no link back to original units
- If our model is better than predicting the Average, than R^2 is in [0,1] range
- **If our model is worse than predicting the Average, R^2 is negative and unbounded**
- Can make R^2 arbitrarily good by adding more features (it will continue to improve); there are adjusted- R^2 variants used in practice to account for the number of predictors
- All metrics being equal, the simpler model is always better (less likely to overfit)

Regression: Evaluation Metrics RMSE vs MAE vs R^2

		Linear Regression		k-NN	
ID	Target	Prediction	Error	Prediction	Error
1	10.502	10.730	0.228	12.240	1.738
2	18.990	17.578	-1.412	21.000	2.010
3	20.000	21.760	1.760	16.973	-3.027
4	6.883	7.001	0.118	7.543	0.660
5	5.351	5.244	-0.107	8.383	3.032
6	11.120	10.842	-0.278	10.228	-0.892
7	11.420	10.913	-0.507	12.921	1.500
8	4.836	7.401	2.565	7.588	2.752
9	8.177	8.227	0.050	9.277	1.100
10	19.009	16.667	-2.341	21.000	1.991
11	13.282	14.424	1.142	15.496	2.214
12	8.689	9.874	1.185	5.724	-2.965
13	18.050	19.503	1.453	16.449	-1.601
14	5.388	7.020	1.632	6.640	1.252
15	10.646	10.358	-0.288	5.840	-4.805
16	19.612	16.219	-3.393	18.965	-0.646
17	10.576	10.680	0.104	8.941	-1.634
18	12.934	14.337	1.403	12.484	-0.451
19	10.492	10.366	-0.126	13.021	2.529
20	13.439	14.035	0.596	10.920	-2.519
21	9.849	9.821	-0.029	9.920	0.071
22	18.045	16.639	-1.406	18.526	0.482
23	6.413	7.225	0.813	7.719	1.307
24	9.522	9.565	0.043	8.934	-0.588
25	12.083	13.048	0.965	11.241	-0.842
26	10.104	10.085	-0.020	10.010	-0.095
27	8.924	9.048	0.124	8.157	-0.767
28	10.636	10.876	0.239	13.409	2.773
29	5.457	4.080	-1.376	9.684	4.228
30	3.538	7.090	3.551	5.553	2.014
	MSE		**1.905**		**4.394**
	RMSE		**1.380**		**2.096**
	MAE		**0.975**		**1.750**
	R^2		**0.889**		**0.776**

References

- Chapter5 from **An Introduction to Statistical Learning**, by G. James, D. Witten, T. Hastie, R. Tibshirani, Springer, 2016 (free book: http://www-bcf.usc.edu/~gareth/ISL/)
- Chapter8 from **FMLPDA Book: Fundamentals of Machine Learning for Predictive Data Analytics**, by J. Kelleher, B. Mac Namee and A. D'Arcy, MIT Press, 2015 (machinelearningbook.com)

- http://www.stat.cmu.edu/~ryantibs/datamining/lectures/19-val2.pdf
- https://sebastianraschka.com/blog/2016/model-evaluation-selection-part3.html
- https://github.com/justmarkham/DAT4/blob/master/slides/07_model_evaluation_procedures.pdf