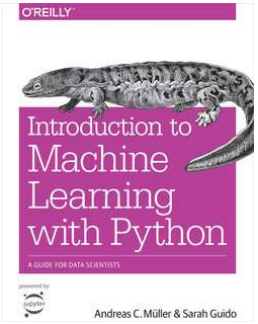
Applied Machine Learning
Multiple Features

BSc course Informatiekunde 2026
https://staff.fnwi.uva.nl/a.visser/education/AML

Arnoud Visser
Intelligent Robotics Lab & Computer Vision Lab
Informatics Institute
Universiteit van Amsterdam
A.Visser@uva.nl

Illustrations courtesy of Maarten Marx, Sarah Guido, Yolanda Hagar, and many others.



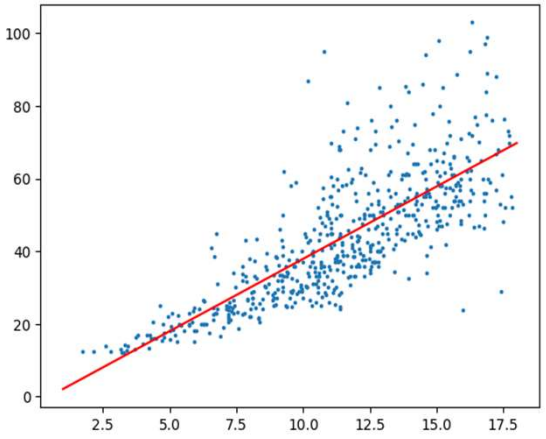
Section 4.7-4.8

Regression of a single feature

- Modelling 775 datapoints

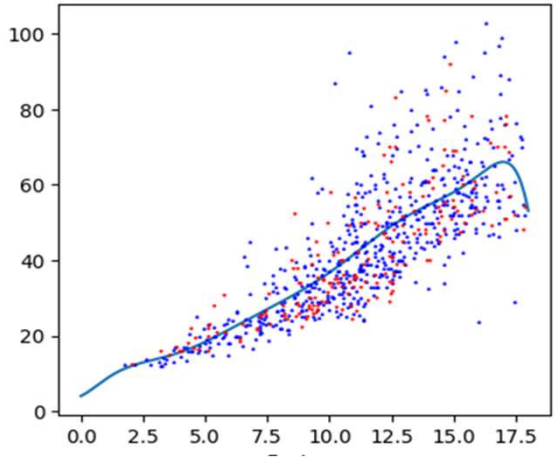
regensburg_pediatric_appendicitis

Linear model



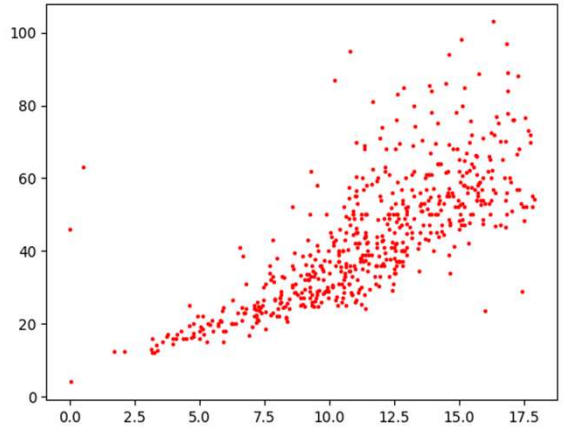
2 values ($y = \alpha + \beta x$)

Polynomial model



10 values ($y = \alpha + \ldots + \kappa x^9$)

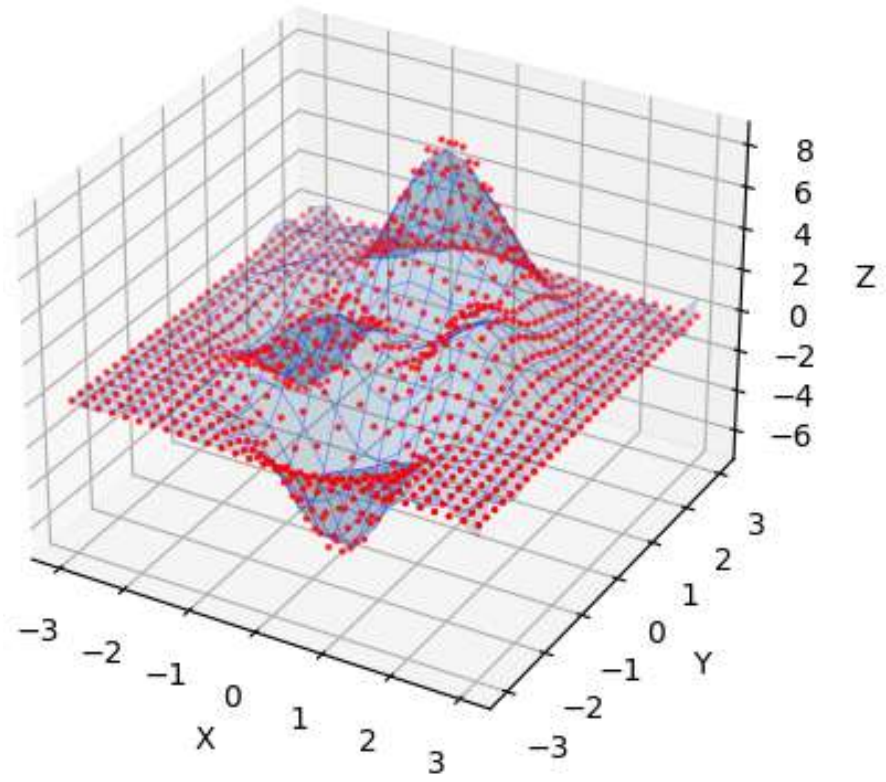
1−nn model



584 training values

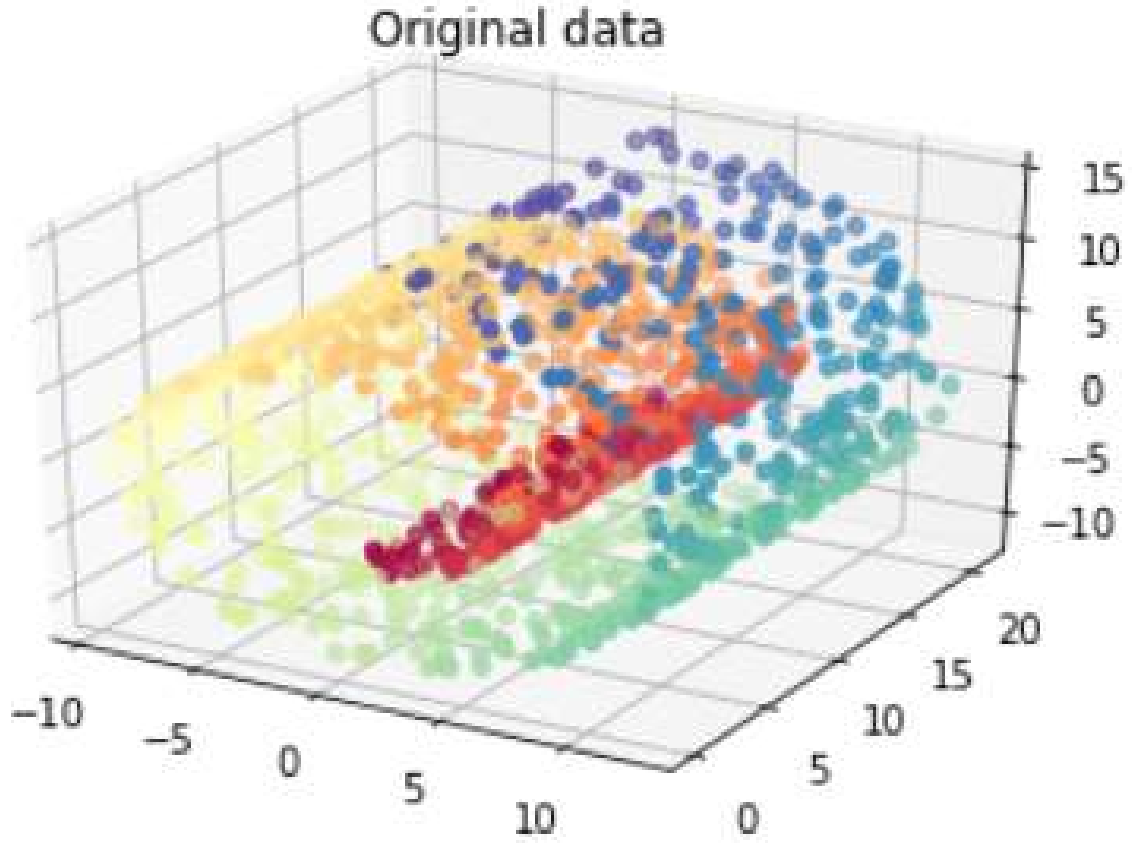
Regression for two features

- No longer fitting a line, but a plane



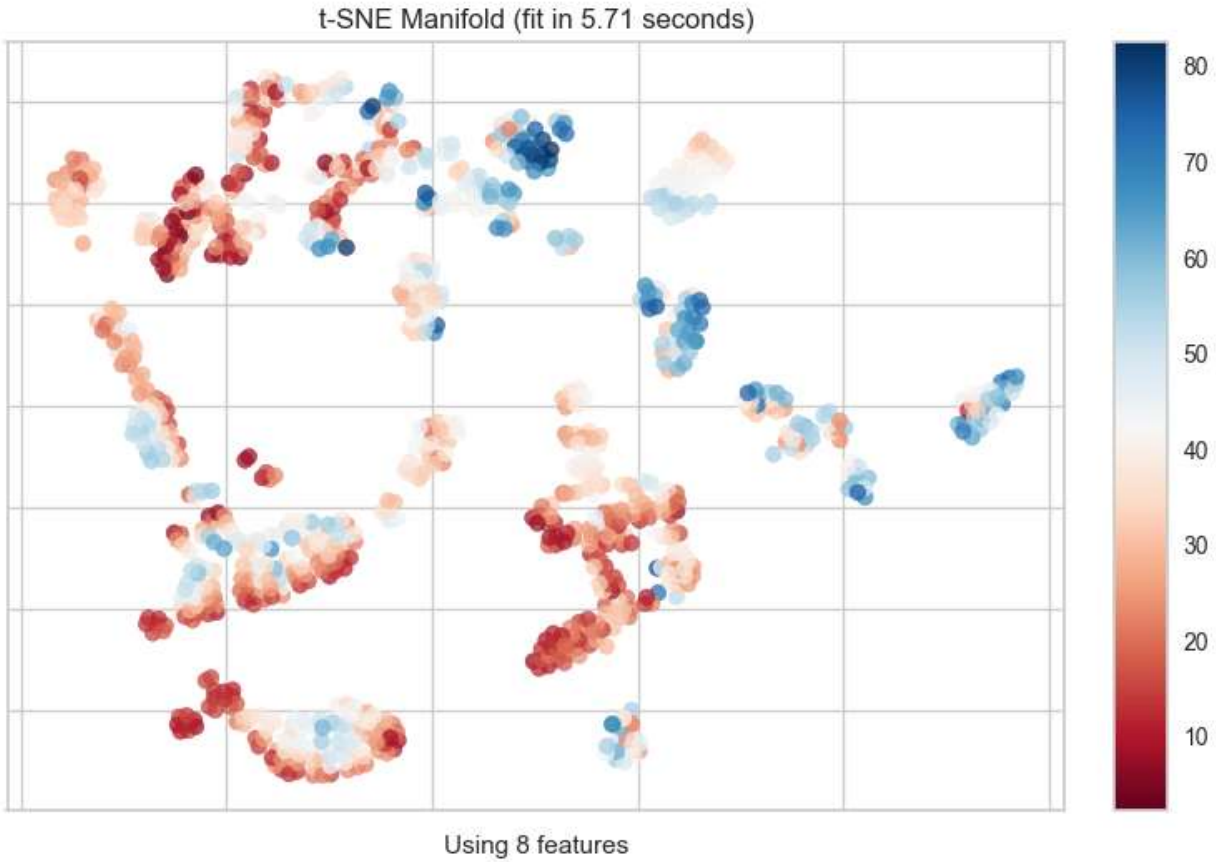
Regression for three features

- No longer fitting a plane, but a manifold



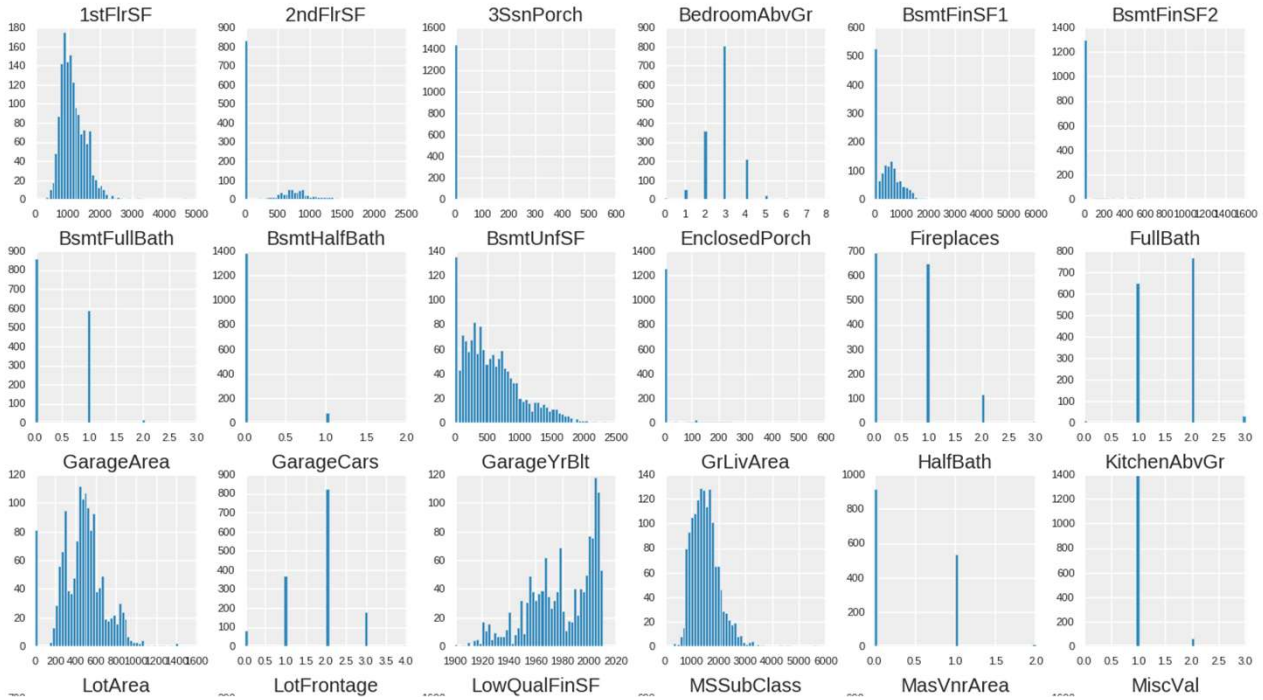
Regression for more features

- Still fitting a manifold



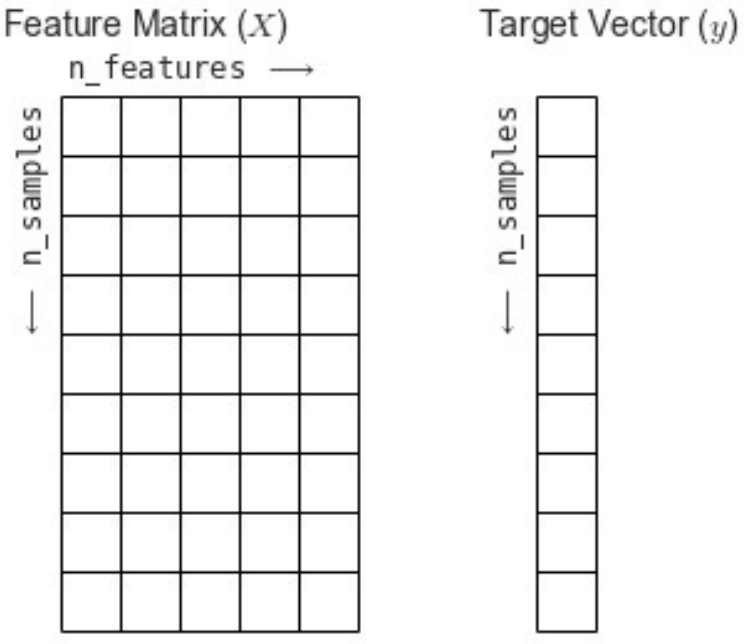
A dataset has typical more features

- Multiple features, one prediction y



Regression for multiple features

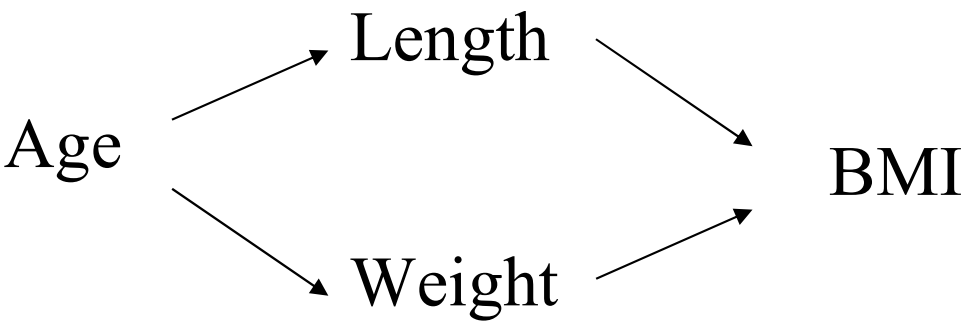
• Model.predict(X) $\longrightarrow$ y



Bad features

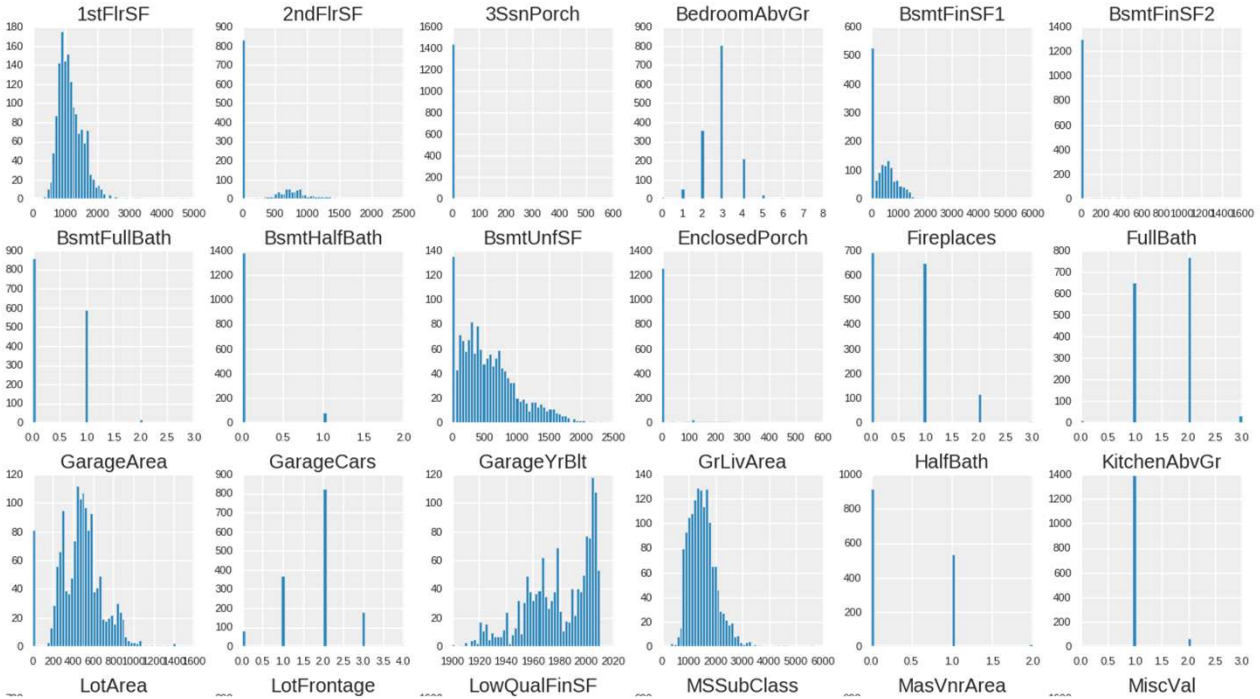
- Sparse features, nearly always zero
- Features not related to y – random noise
- Groups of features – all related to y
- Features out of scale to other features

regensburg_pediatric_appendicitis



What is a good feature?

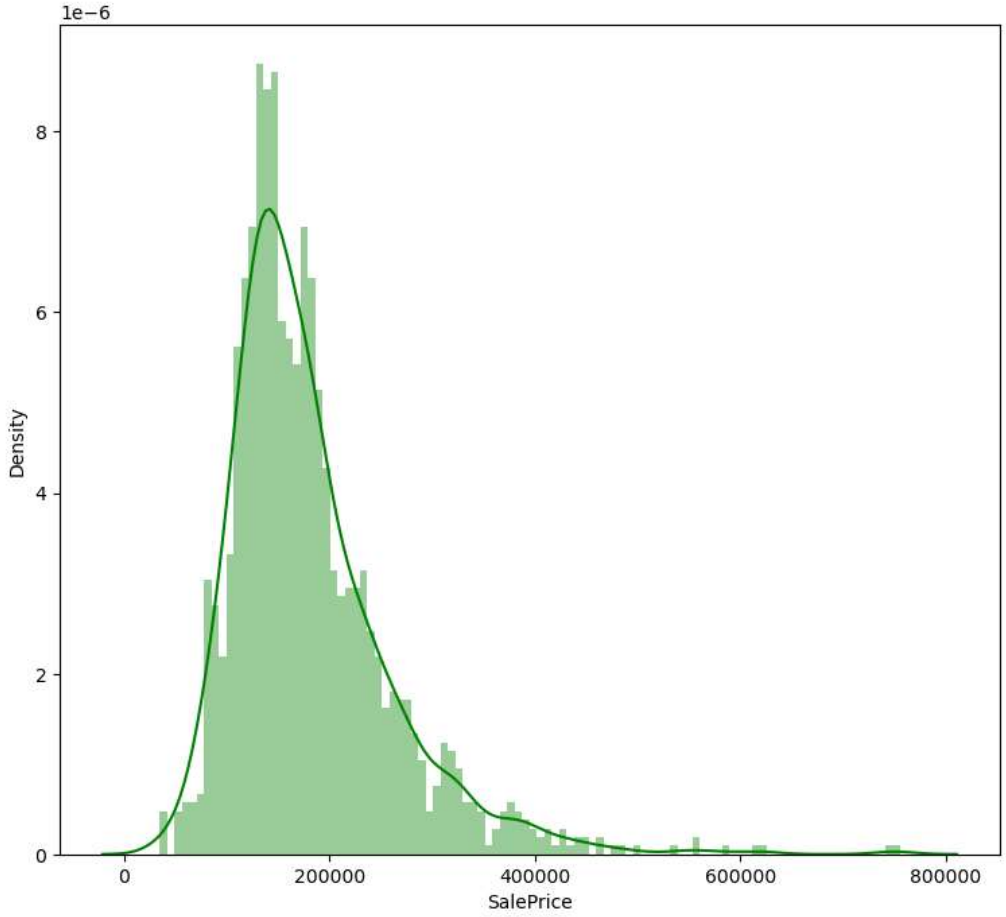
- Predict SalePrice based on 79 features



House Prices - Advanced Regression Techniques – [ongoing Kaggle competition]

Data Exploration

- First look at the SalePrice



± Normal distribution, yet with a positive skewness and long tail

Lognormal distribution with a small σ

House Prices in $ for individual residential property in Ames, Iowa from 2006 to 2010

Dean de Cock, Journal of Statistics Education, Volume 19, 2011 - Issue 3

Data Exploration

Remove sparse features

If a non-null value for only 30% of the observations, remove

```
df2 = df[[column for column in df if df[column].count() / len(df) >= 0.3]]
```

```
df.info
```

```
Alley          91 non-null object
PoolQC          7 non-null object
Fence         281 non-null object
MiscFeature    54 non-null object
```

4 features removed – 75 features remaining

Data Exploration

- Remove textual features

```
df_num = df.select_dtypes(include = ['float64', 'int64'])
```

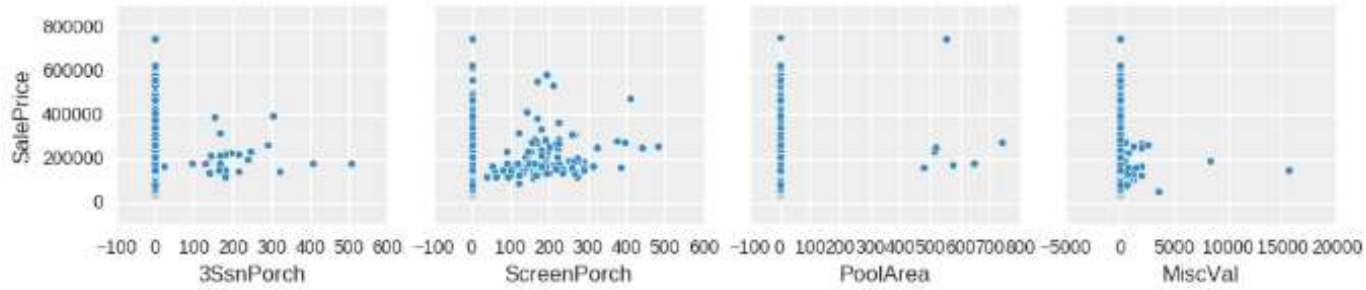
	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2
0	60	65.0	8450	7	5	2003	2003	196.0	706	0
1	20	80.0	9600	6	8	1976	1976	0.0	978	0
2	60	68.0	11250	7	5	2001	2002	162.0	486	0
3	70	60.0	9550	7	5	1915	1970	0.0	216	0
4	60	84.0	14260	8	5	2000	2000	350.0	655	0

38 features removed – 37 features remaining

Data Exploration

- Many features have 0 instead of null

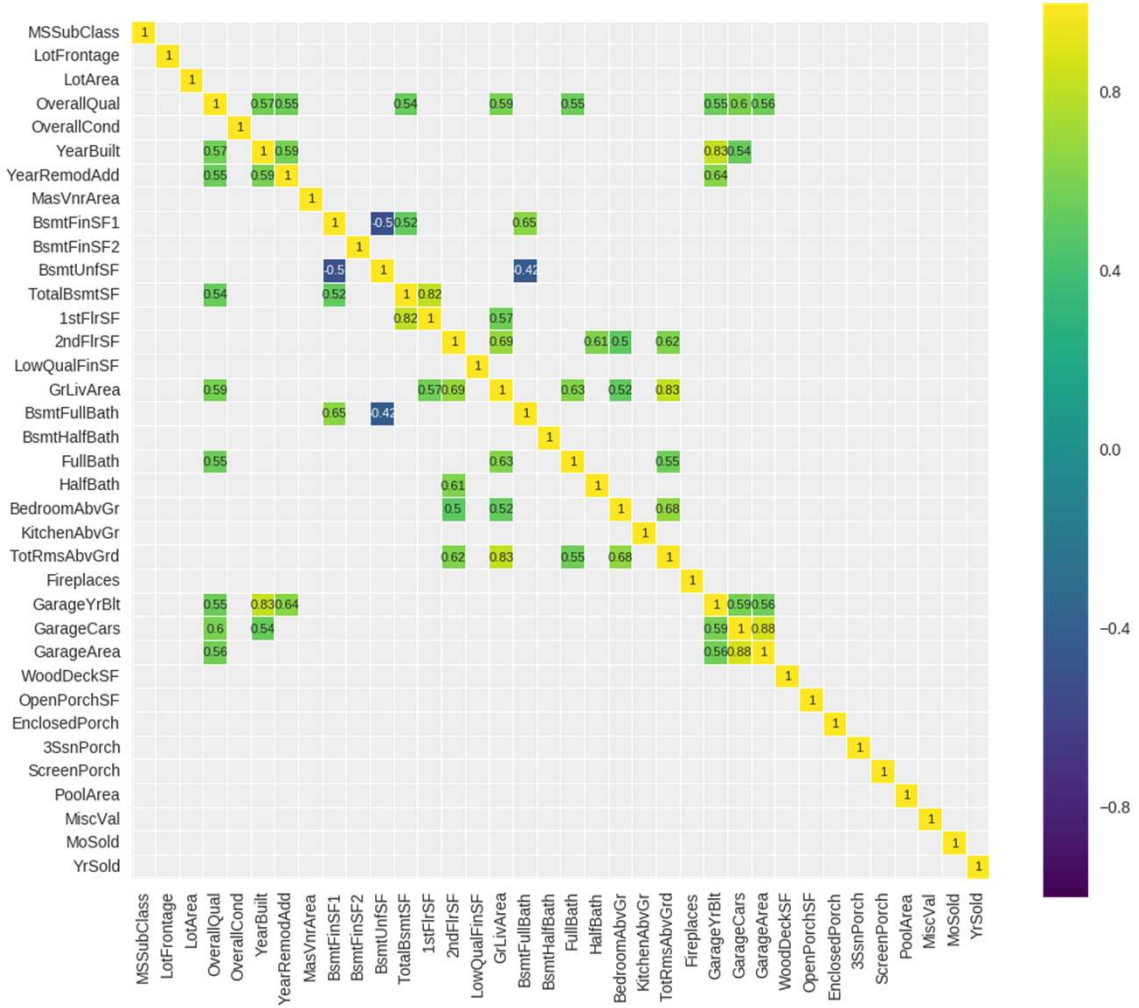
```
import operator

individual_features_df = []
for i in range(0, len(df_num.columns) - 1):
    tmpDf = df_num[[df_num.columns[i], 'SalePrice']]
    tmpDf = tmpDf[tmpDf[df_num.columns[i]] != 0]
    individual_features_df.append(tmpDf)

all_correlations = {feature.columns[0]: feature.corr()['SalePrice'][0] for feature in individual_features_df}
all_correlations = sorted(all_correlations.items(), key=operator.itemgetter(1))
```



no features removed – only observations removed

Data Exploration

- Look for correlations between features



Data Exploration

- separate the categorical from quantitative features

	LotFrontage	LotArea	MasVnrArea	BsmtFinSF1	BsmtFinSF2	TotalBsmtSF	1stFlrSF	2ndFlrSF	LowQualFinSF	GrLivArea	...
0	65.0	8450	196.0	706	0	856	856	854	0	1710	...
1	80.0	9600	0.0	978	0	1262	1262	0	0	1262	...
2	68.0	11250	162.0	486	0	920	920	866	0	1786	...
3	60.0	9550	0.0	216	0	756	961	756	0	1717	...
4	84.0	14260	350.0	655	0	1145	1145	1053	0	2198	...

5 rows × 28 columns

```
features_to_analyse = [x for x in quantitative_features_list if x in golden_features_list]
features_to_analyse.append('SalePrice')
features_to_analyse
```

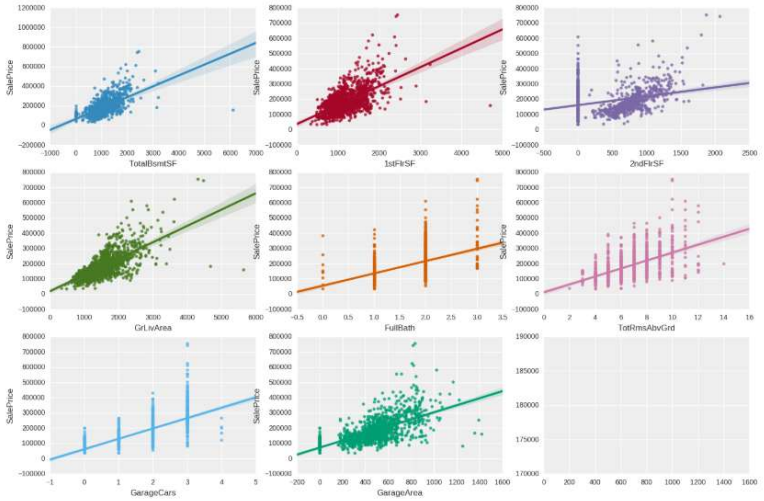
'YearRemodAdd', 'YearBuilt', 'OverallQual'

3 categorial features removed – 9 features remaining

Data Exploration

Visual inspect results

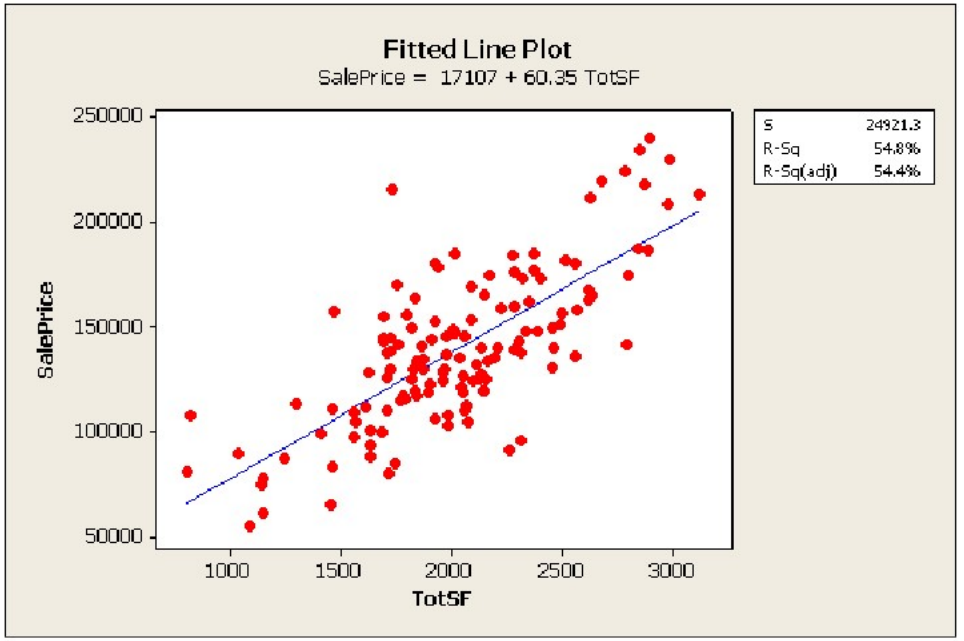
```
fig, ax = plt.subplots(round(len(features_to_analyse) / 3), 3, figsize = (18, 12))

for i, ax in enumerate(fig.axes):
    if i < len(features_to_analyse) - 1:
        sns.regplot(x=features_to_analyse[i],y='SalePrice', data=df[features_to_analyse], ax=ax)
```



5 area-related features dominant, but there is a lot variance

Linear Model

- Combine area features

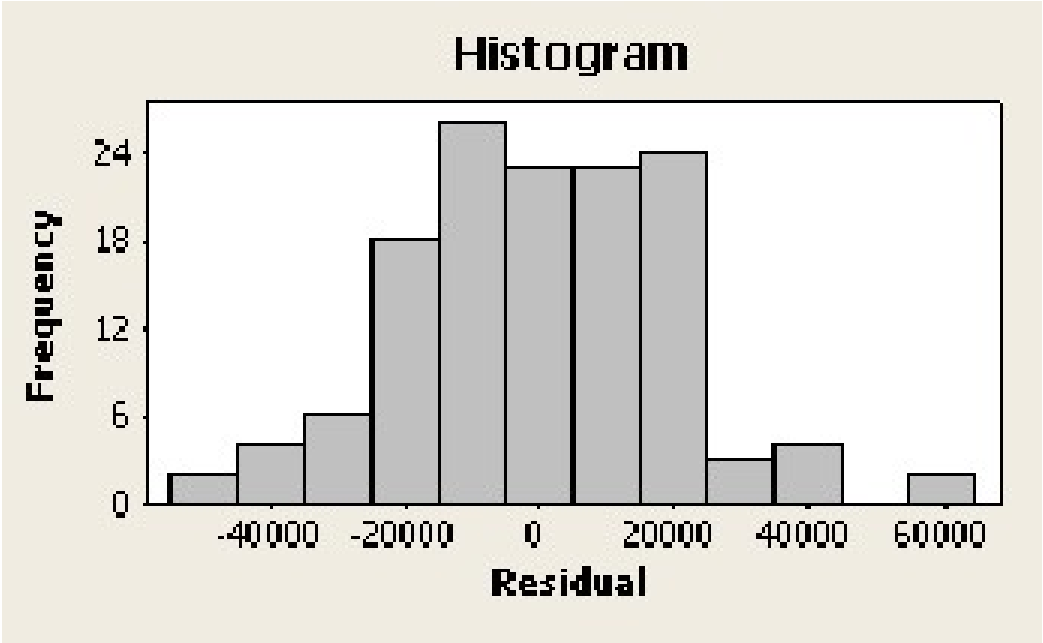


Total square footage = 'TotalBsmtSF' + 'GrLivArea'

House Prices in $ for individual residential property in Ames, Iowa from 2006 to 2010
Dean de Cock, Journal of Statistics Education, Volume 19, 2011 - Issue 3

Linear Model

- Assumption verification

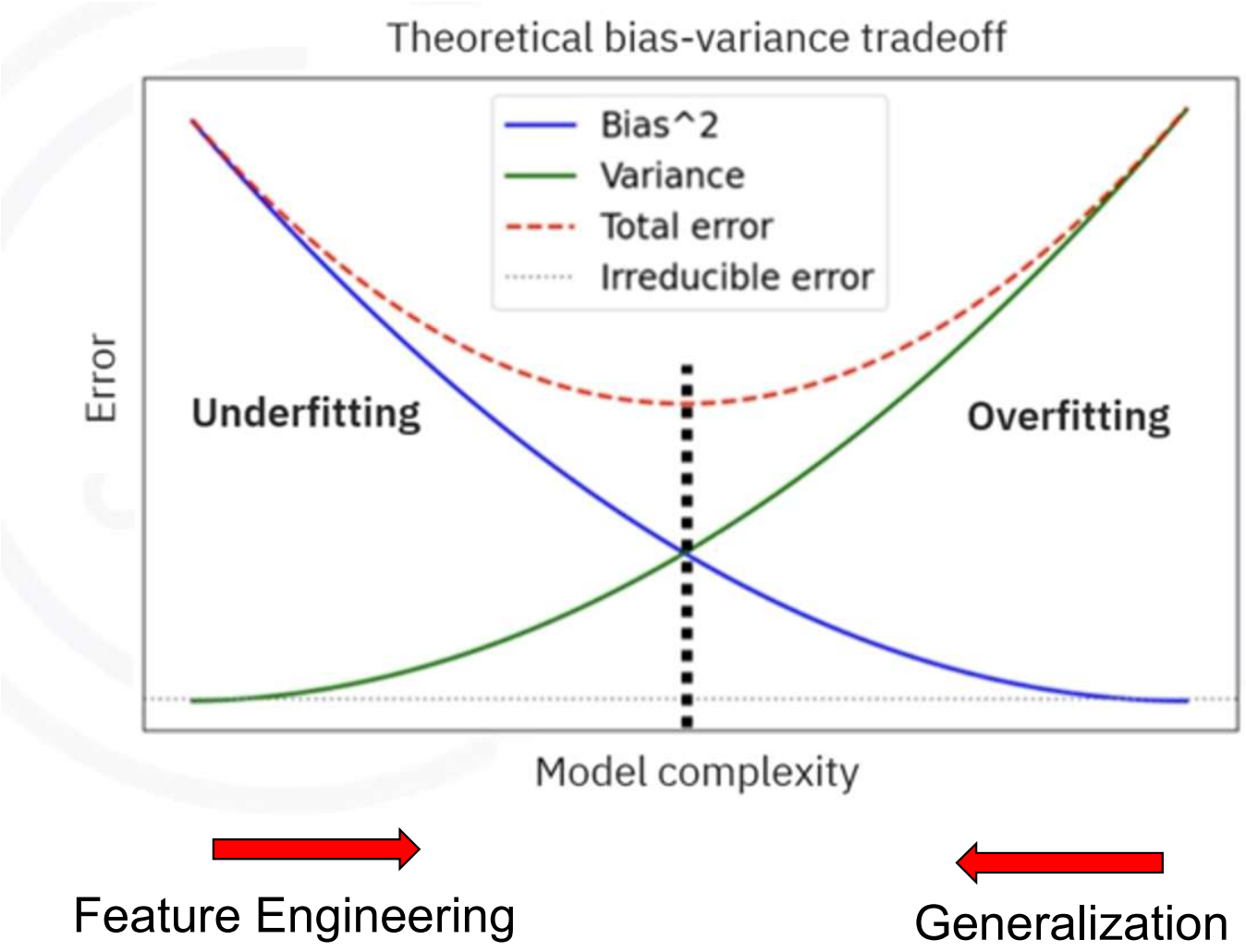


Total square footage = 'TotalBsmtSF' + 'GrLivArea'

House Prices in $ for individual residential property in Ames, Iowa from 2006 to 2010
Dean de Cock, Journal of Statistics Education, Volume 19, 2011 - Issue 3

What is good model?

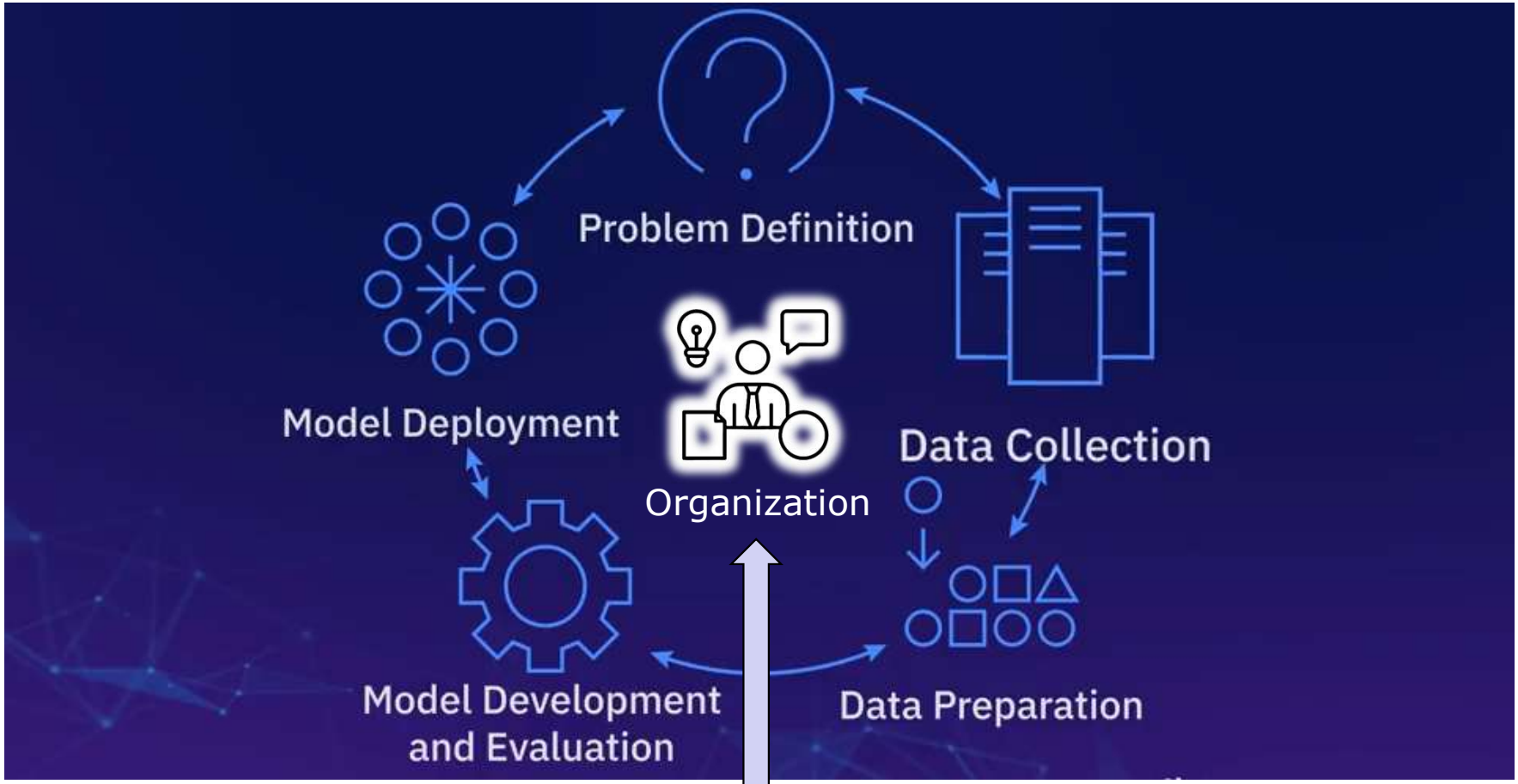
- Balance



Courtesy Joseph Santarcangelo & Jeff Grossman

Introduction to Machine Learning

Building a predictive model is a circular process.

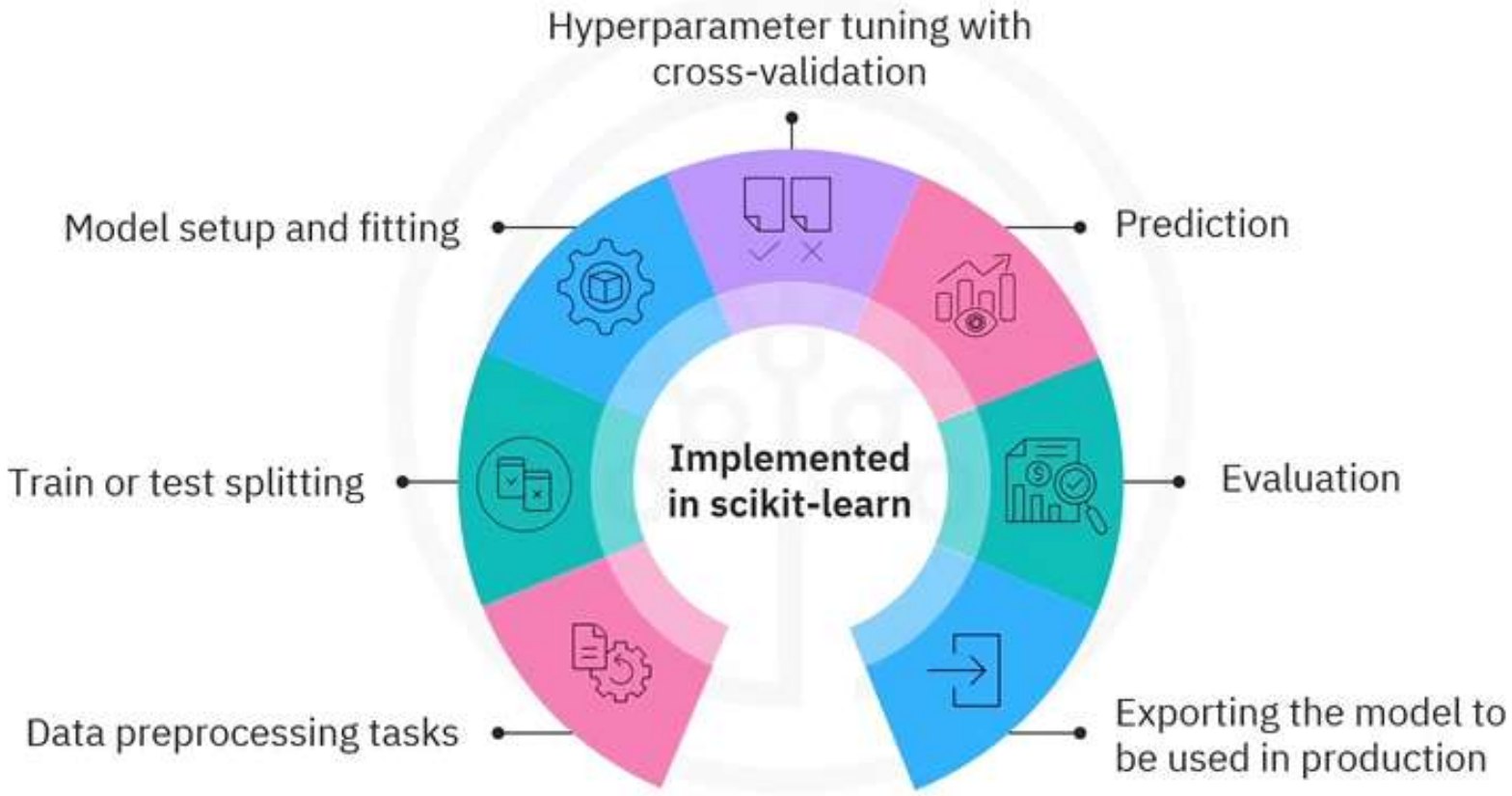


Expertise

- Depends on the organization
- Interpret the information in the right way

Introduction to Machine Learning

Building a predictive model is a circular process.

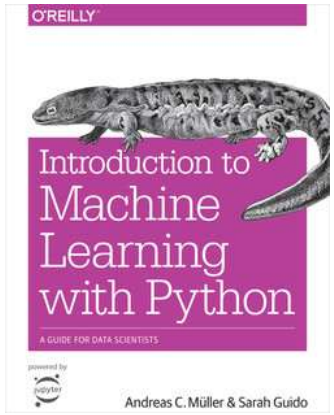


Joseph Santarcangelo & Joss Grossman, Machine Learning with Python, IBM Skills network, 2025

Introduction to Machine Learning

- 2.2 Generalization, overfitting & underfitting
- 2.3.3 Regularization

- Strengths:
  - Complexity of model can be controlled
- Weakness:
  - Additional hyperparameter.



Andreas C. Müller, Sarah Guido, Introduction to Machine Learning with Python, O'Reilly Media, October 2016

Conclusion

Learning outcomes of this course covered today

- Regularization depends on your data prepared (normalization & feature selection)
- You have to be aware of feature correlations & sparsity