Machine Learning from Idea to Production

Keynote, Big Data Conference Vilnius, November 2018

Oliver Zeigermann / @DJCordhose

Objective

Imagine we are in the car insurance business and want a system that predicts the risk of accidents for prospective customers.
ML Car Insurance Risk Check

You can check the risk group for a prospective customer simply by providing three inputs:

- **Speed in MPH**
  
  100

- **Age**
  
  47

- **Miles per Year (in thousand)**
  
  10

- **Calculate Risk Group**

- **Low Risk**

Customer Data - Risk of Accidents

Max Speed vs Age

How would you rank me (47) for a car having 100 mph top speed, driving 10k miles per year?
Our Process

1. Data Preparation
2. Training
3. Evaluation
4. Production
5. Post-Production
Part I

Data Preparation

- Collection
- Cleaning
- Analysis
Data is King

collecting data might be the hardest part of the job

- but also the most important
- no data, no supervised machine learning
- if you have a simulator, reinforcement learning might be an option
Basic Check on Collected Data

```
[ ] df.describe()
```

<table>
<thead>
<tr>
<th></th>
<th>speed</th>
<th>age</th>
<th>miles</th>
<th>group</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>1500.00000</td>
<td>1500.00000</td>
<td>1500.00000</td>
<td>1500.00000</td>
</tr>
<tr>
<td>mean</td>
<td>122.492667</td>
<td>44.980667</td>
<td>30.434000</td>
<td>0.998667</td>
</tr>
<tr>
<td>std</td>
<td>17.604333</td>
<td>17.130400</td>
<td>15.250815</td>
<td>0.816768</td>
</tr>
<tr>
<td>min</td>
<td>68.000000</td>
<td>16.000000</td>
<td>1.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>25%</td>
<td>108.000000</td>
<td>32.000000</td>
<td>18.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>50%</td>
<td>120.000000</td>
<td>42.000000</td>
<td>29.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>75%</td>
<td>137.000000</td>
<td>55.000000</td>
<td>42.000000</td>
<td>2.000000</td>
</tr>
<tr>
<td>max</td>
<td>166.000000</td>
<td>100.000000</td>
<td>84.000000</td>
<td>2.000000</td>
</tr>
</tbody>
</table>
## Clean Data and Select Features

<table>
<thead>
<tr>
<th>speed</th>
<th>age</th>
<th>miles</th>
<th>state</th>
<th>group</th>
</tr>
</thead>
<tbody>
<tr>
<td>98</td>
<td>44</td>
<td>25</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>118</td>
<td>54</td>
<td>24</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>26</td>
<td>34 Colorado</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>26</td>
<td>34 Colorado</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>114</td>
<td>38</td>
<td>22 California</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>130</td>
<td>55</td>
<td>34</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>118</td>
<td>40</td>
<td>51</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>143</td>
<td>42</td>
<td>34 New York</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>120</td>
<td>41</td>
<td>42</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>148</td>
<td>33</td>
<td>53 California</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>118</td>
<td>48</td>
<td>25</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>105</td>
<td>51</td>
<td>21 Oklahoma</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>speed</th>
<th>age</th>
<th>miles</th>
<th>state</th>
<th>group</th>
</tr>
</thead>
<tbody>
<tr>
<td>115</td>
<td>51</td>
<td>4</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>30</td>
<td>28 Texas</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>107</td>
<td>41</td>
<td>Texas</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>123</td>
<td>77</td>
<td>26</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>115</td>
<td>34</td>
<td>22</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>144</td>
<td>48</td>
<td>35</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>93</td>
<td>20</td>
<td>10 California</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>110</td>
<td>32</td>
<td>34</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>127</td>
<td>28</td>
<td>25 Florida</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1040</td>
<td>48</td>
<td>28</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>112</td>
<td>37</td>
<td>28 California</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>151</td>
<td>34</td>
<td>50</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Results: Data Cleaning und Feature Selection

Data Cleaning

• Typos: Califorina
• Outliers: Delete line or replace with decent value
• Doubles: Delete
• Missing Value: Delete line or replace with imputed value

Feature Selection

• Make sure which value to predict
• Row missing more than 50% of values: do not use
• Explore dependencies to decide what use for training input
Exploration
Part II
Training
Baselines

- we must be careful not to fool ourselves
- we can never be quite sure what is the highest possible score
- but we can provide a baseline of a prediction with very little effort
- every model must be matched to this baseline

https://blog.insightdatascience.com/always-start-with-a-stupid-model-no-exceptions-3a22314b9aaa
https://blog.insightdatascience.com/how-to-deliver-on-machine-learning-projects-c8d82ce642b0
Baseline 1: Random

It still gets 33% percents right
Baseline 2: KNN, $n=39$

Already 70% accuracy on test data, found using grid search with cross validation
Generalization

- The most important property of a model is if it generalizes well to unknown data.
- A machine learning model is of no use if it only works well on the data it has been trained on:
  - If it was, the easiest way to achieve this would be a dictionary translating from a set of inputs to the known output.
- Conceptually it is a little bit hard to optimize for something you do not know.
- So, we introduce a little trick here.
Split known data into training and test

Testing for generalization

80% Train model

20% Final test

Training

Test
Use some training data for validation

**Testing for Generalization**

- **80%** Train Model
- **20%** Constant Check
- **20%** Final Test

**Training**  **Validation**  **Test**
The Issue: Overfitting

Training Score 80% ↓ ↑ 70% Test Score

Training and test scores clearly divert
Regularization

Process to counter overfitting

Each ML strategy has its own means of Regularization, e.g.

- KNN: more neighbors
- Decision Trees: reduce depth, use ensembles
- SVM: gamma and cost
- NN: Dropout, Batch Normalization, Reduced Capacity, Reduced Training Time
Great results

Up to 80% accuracy on test data using a regularized neural networks
Part III

Evaluation
Great results

BUT ARE THEY?
Predictions outside of the trained range

A neural network deployed in the wild may be asked to make predictions for inputs that were drawn from a different distribution than that of the training data.

A plethora of work has demonstrated that it is easy to find or synthesize inputs for which a neural network is highly confident yet wrong.

https://arxiv.org/abs/1810.09136
Looking at areas not covered by training data

https://colab.research.google.com/github/djcordhose/ai/blob/master/notebooks/sklearn/overview.ipynb#sc...
How to deal with those blank spots?

First: Be aware of the reasonable range of values

- For requests in extended area
  - add a warning
  - reject request
- Identify suspicious scores
  - in extended area scores often are very biased towards one category (1.0 score)
- Add likelihoods to scores
  - Bayesian approaches are the new hype
- Try to get more data, even incrementally
  - Use user feedback, give it lower training score
Check for bias / unfairness

your model contains all biases and truths of the training data

- Your model might perform well for a few users, but badly for others
- Accuracy and loss only show the average
- Some areas might not have properly covered with data samples (including test)
- Curse of High Dimensions

https://developers.google.com/machine-learning/fairness-overview/
Dealing with Bias / Unfairness

- log real requests
- log matching user behavior
- evaluate those requests (Elastic)
- if you recommend, do users follow your advice?
- if there is probability, how good is it?
- do you see bias/unfairness?
- put more emphasis on data that suffers from bias

Always check with GDPR
Explaining your models

LIME:
https://github.com/marcotcr/lime/blob/master/README.md

Anchor:
https://github.com/marcotcr/anchor/blob/master/README.md
Part IV:
Production
ML Car Insurance Risk Calculator

ML Car Insurance Risk Check

You can check the risk group for a prospective customer simply by providing three inputs.

Speed in MPH
100

Age
47

Miles per Year (in thousand)
10

Calculate Risk Group
Low Risk

Serving Models

• Be very sure you have the same pipeline of steps when serving as you had for training and evaluation

• Log real world predictions (if ok with GDPR)


Version together

1. Code for Cleaning and Trained Model
2. Trained Model
3. Serving Code
4. Training data might not be feasible because of size or GDPR
Possible Deployment Environments

- Google Cloud ML
  - Can serve TensorFlow and Sklearn
  - your model and calls are in Google’s control
  - Takes away the scaling and installation burden
- Local Server
  - TensorFlow Serving
  - Flask Server
- Browser: https://js.tensorflow.org
Part V:

Post-Production
Compile-time and runtime are less relevant notions than before-shipping and after-shipping
Going to production changes everything

True for classic software as well as machine learning models

Every deployment of a newly trained model is like a major release in software development

- needs to be tested like this
- or you need to make sure it rather is like a minor or bugfix release
  - make incremental changes
  - combine models
Continuity in model predictions

- People who use your model regularly will expect most things to stay the way they know it.
- Even if it not totally accurate.
- That might mean you can not train a model from scratch even if you have better data or a better model architecture.
- Consider mixing more than one model.
The Turtle Effect

If the user expects a certain pattern to be in the model, they expect it to persist
Wrap Up

Training a model is the least part

• having reasonable data is the most important part
• evaluation is more than accuracy: blank spots, bias
• going to production changes everything
• people expect continuity

Machine Learning from Idea to Production