

**DEVVOX**  
United Kingdom

[illegible]

@theNeomatrix369 @jeremiecharlet

What you upto?  
Busy!!! Training my model!

Tensorflow?  
Pytorch? Keras?

Kubeflow?

XGBoost? Random Forest?

Dask for hyperparameter  
tuning

# Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org

● Backfed Input Cell

XGB + SVM gives 99%  
accuracy

△ Spiking Hidden Cell

● Output Cell

○ Match Input Output Cell

● Recurrent Cell

○ Memory Cell

△ Different Memory Cell

● Kernel

○ Convolution or Pool

Hurray! 99.5% accuracy on  
training set!

Increase epoch!  
Re-run training!

Deep Feed Forward (DFF)

(P)

Feed Forward (FF)

Radial Basis Network (RBF)

Recurrent Neural Network (RNN)

Long / Short Term Memory (LSTM)

Gated Recurrent Unit (GRU)

Auto Encoder

Sparse AE (SAE)

Boltzmann Machine (BM)

Restricted BM (RBM)

0.5% for 100% accuracy



**Wrong results  
on test set!**

**CHICKEN**  
98% confidence

**Model does not  
make sense at all!**

**Model is not  
generalising!**

**Training & Test set  
contain corrupt data,  
missing values**



**I HAVE NO  
IDEA WHAT  
I'M DOING**



**But, do we really know our data?**

Do we know our data,  
as good as we know our tools?

...

8-11th May 2019 \* Devovx UK 2019 \* London, UK

# About us



Mani Sarkar  
[@theNeomatrix369](https://twitter.com/theNeomatrix369)

Freelance  
Software  
Developer

Java / JVM

Cloud/Infra/DevOps

Polyglot developer

LJC, Devoxx, dev.  
communities.

AI / ML / DL / DS

JCP member, F/OSS projects:  
[@adoptopenjdk](https://twitter.com/adoptopenjdk) [@graalvm](https://twitter.com/graalvm)

Java Champion, Software Crafter, Blogger, Speaker, Conferences,  
Events



# About us

Co founder & CTO  
Trackener  
(fitbit for horses)

Co-Host  
MaM ML Study group

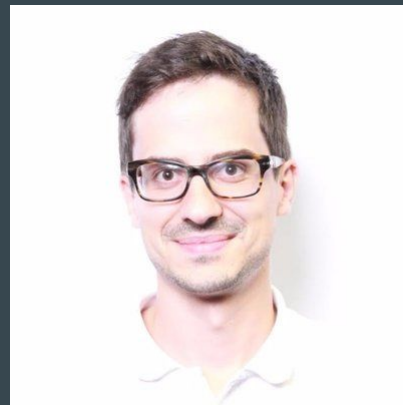
Polyglot developer

Personal Dev /  
Learning Addict

Datascience ML/DL

Mentor/Mentee  
MeetAMentor.co.uk

JAVA Spring Boot <3



Jeremie Charlet  
[@jeremiecharlet](https://twitter.com/jeremiecharlet)

Presentation: live slides

<http://bit.ly/do-we-know-our-data>



Look inside folder  
**02-devovx-uk-2019**

<https://github.com/neomatrix369/awesome-ai-ml-dl/tree/master/presentations/data>



Overwhelmed with mountains of information

It has been a humbling  
experience...

[Credits to original creator / author on Medium](#)

# Thank you

- All our data science mentors who helped
  - Ovidiu Serban (Imperial College)
  - Mark Bell (TNA Gov. UK)
  - Daniel Hulme (Satalia, UCL lecturer) and Joshua Cooper (Satalia)
  - Miguel Martinez (Nvidia, Deep Learning Solutions Architect),
  - Kerry O'Neill
  - Ole Moeller-Nilsson (Pivigo)
  - Antoine Paré (geophysicist at baker hughes)
  - And everyone else who has helped with our talk...

# About us



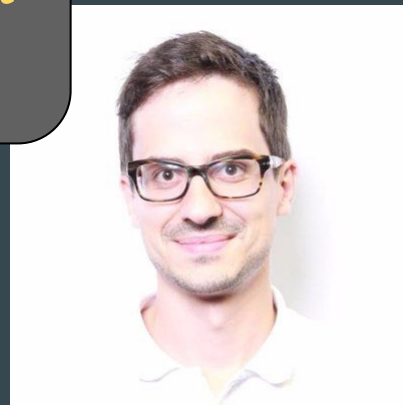
Mani Sarkar  
[@theNeomatrix369](#)

PhD in  
Everything

I wish I knew  
everything

Chief fool stack  
DevOps scientist  
(self inflicted)

I know, I know  
nutthing!



Jeremie Charlet  
[@jeremiecharlet](#)



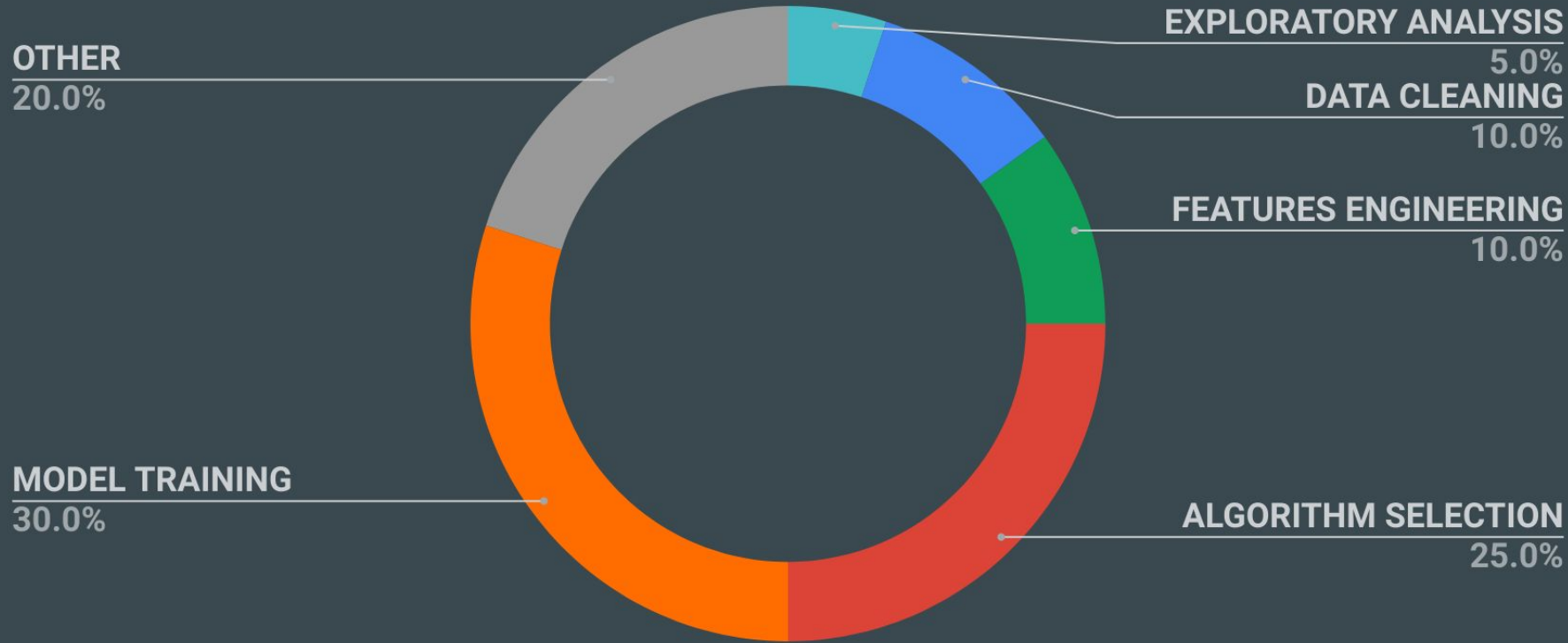


**Take it with a pinch of salt**

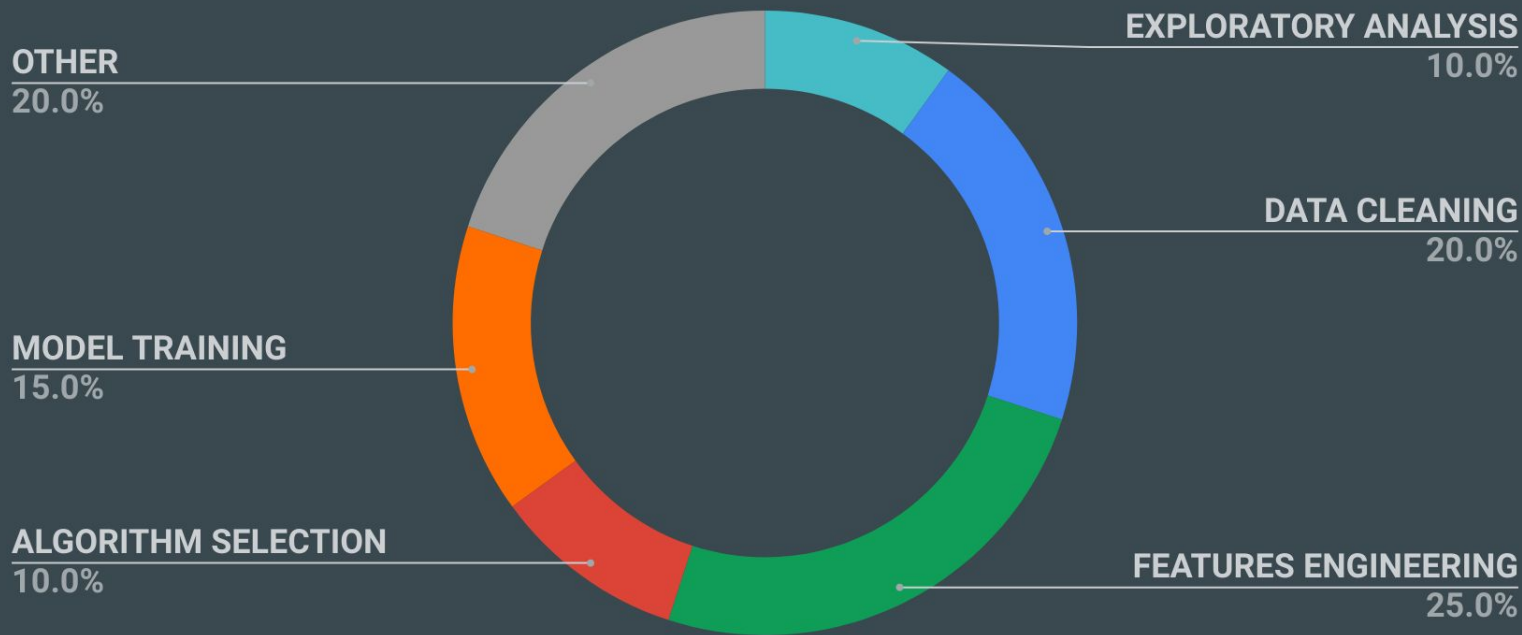


# Disclaimer

- *YMMV*
- our *first attempt* at a Devovx UK event
- might have rough edges and *inaccuracies*
- sharing our *learnings* over the past year
- gathered thoughts and ideas from *various sources*
- *sharing guidelines, not a silver-bullet*
- if it's not clear, *tell us!*



# Are these figures correct?



Estimated figures, a mere guideline, NOT a canonical source!

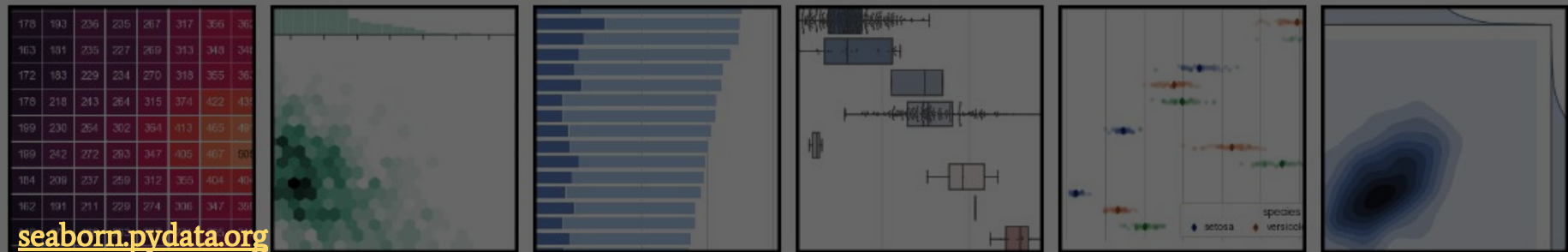
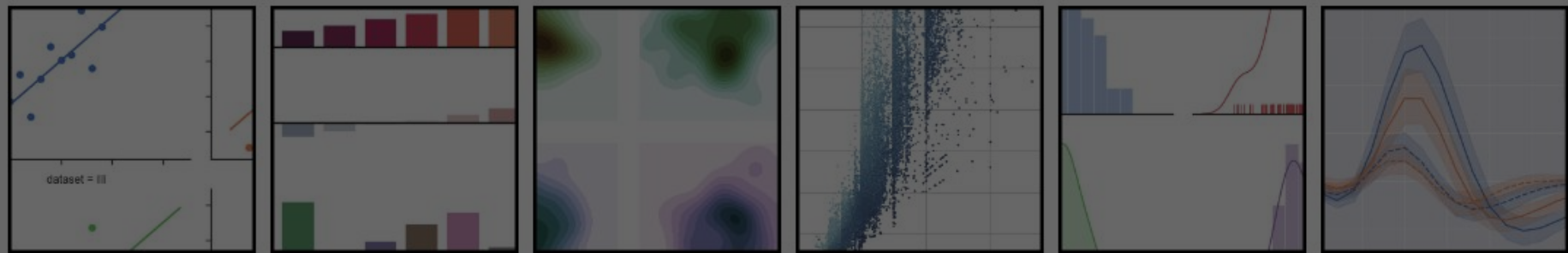
# What we won't cover

- Time series (resampling)
- Computer vision, Unstructured data (NLP)
- Generating synthetic data

For the sake of simplicity, we will focus on *tabular data* !

# Agenda

- Introduction
  - Data collection
  - Exploratory Data Analysis
  - Data Preparation
  - Feature Engineering
  - Periphery
- Conclusion: what others do?
- Resources
- Thank you, feedback, stay in touch!
- Appendix





# Data Collection

# Data Collection





# Data Collection



12  
percents

Flying stuff

Flying stuff

Flying stuff

Papa Roach

12l:kjd&^\*

dragonfly

My favourite  
dragonfly

The Beatles


dragonfly

dragonfly

dragonfly

# Data Collection - questions to ask

- Requirements?
- Good enough details
- How much data is enough?
- Reflect reality ? Bias ?



Iterative  
process

[See Appendix for further details](#)

# Exploratory Data Analysis

# Exploratory Data Analysis

- Know the domain knowledge
- Check basic characteristics of dataset
- Check descriptive statistics
- Plot distribution of features
- Check correlations between features, with target column



# Exploratory Data Analysis - why?

- Black box
- Feeling lost
- Be better prepared
- Prevent wasting time
- To achieve the goal

[See Appendix for further details](#)

# Exploratory Data Analysis

Go to the

[Exploratory Data Analysis - Jupyter Notebook](#)

[See Appendix for further details](#)

#DoWeKnowOurData

# Exploratory Data Analysis - questions to ask

- Domain knowledge
- Source of data
- Nature of data accumulation
- Bias
- Dirty data
- What to fix ?

[See Appendix for further details](#)

# Exploratory Data Analysis

Correlation not equal to causality (or does not mean they are related)



# Comedians really know their data!



55  
seconds  
of fun

[Link to the video](#)

#DoWeKnowOurData

# Comedians really know their data!

Ellen D.  
knew her  
time-series

Ellen D.  
knew her  
graphs

She made us  
laugh, so she  
must know  
something...



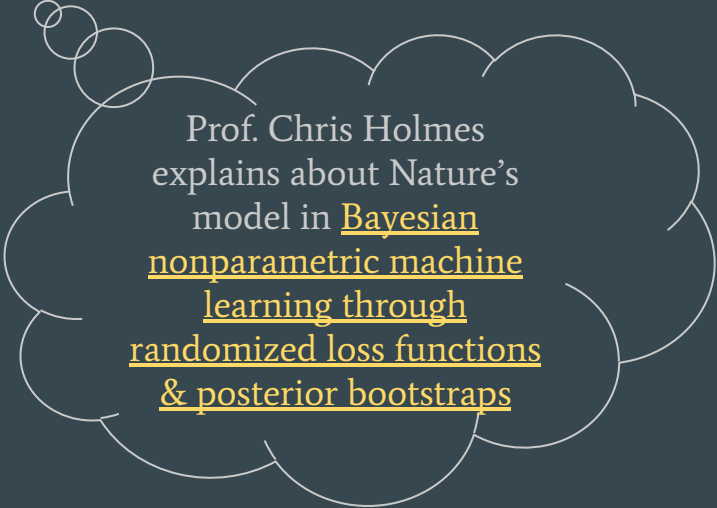
# Data Preparation

# Data Preparation

- Data cleaning
  - Deal with errors
  - Deal with duplicates
  - Deal with outliers
  - Deal with missing data
- Deal with too much data

# Data Preparation - why?

- Garbage in, garbage out
- Clean dataset
- Replicate nature's model



Prof. Chris Holmes  
explains about Nature's  
model in Bayesian  
nonparametric machine  
learning through  
randomized loss functions  
& posterior bootstraps

See Appendix for further details

# Data Preparation

Go to the

[Data Preparation - Jupyter Notebook](#)

[See Appendix for further details](#)

#DoWeKnowOurData

# Data Preparation - questions to ask

- Outliers
- Missing data
- Class overload
- Too many features
- Unbalanced dataset
- Bias
- More data?

[See Appendix for further details](#)

# Feature engineering

# Feature Engineering

- Find hidden information
  - Feature extraction
  - Applying math / statistical functions
  - Apply physics functions
- Deal with too many features / too much data
  - Dimensionality reduction
  - Feature selection
- Statistical Inference
- Improve training efficiency: accuracy, speed, save resources



Too many features:  
revisiting this  
topic to extract  
relevant data  
after cleaning  
and preparation



# Feature Engineering - why?

- Hidden information
- Extract the essence of the data
- Improve training

[See Appendix for further details](#)

# Feature Engineering

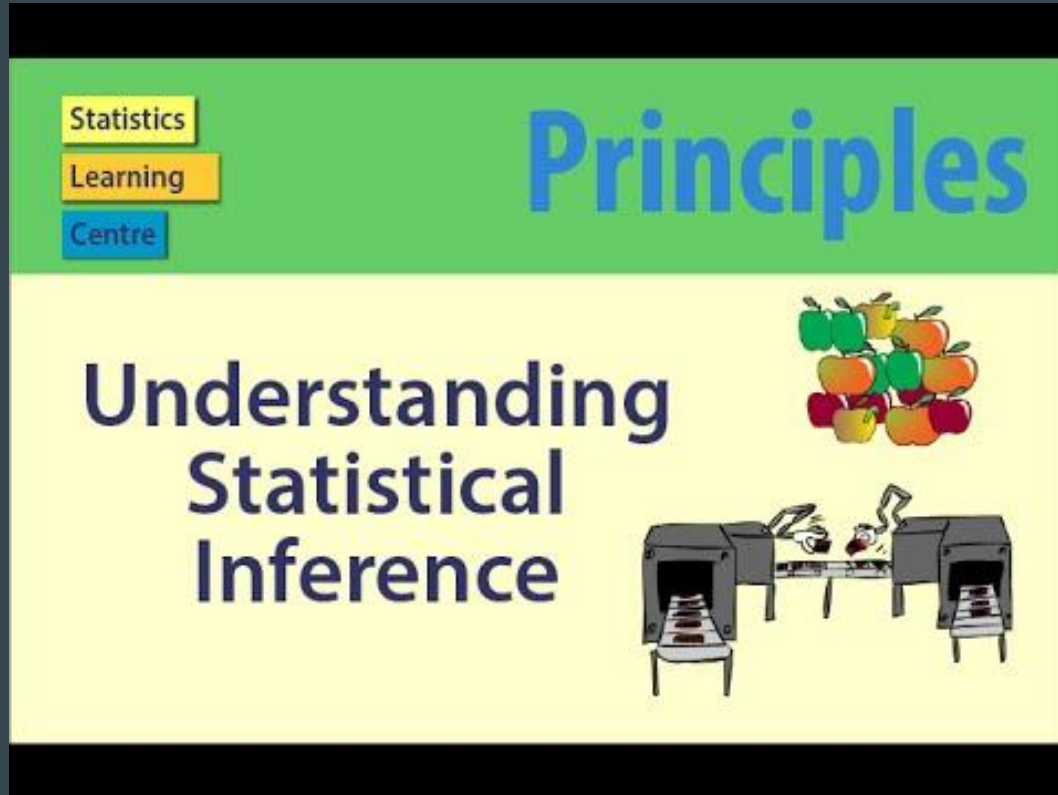
Go to the

[Feature Engineering - Jupyter Notebook](#)

[See Appendix for further details](#)

#DoWeKnowOurData

# Feature Engineering - statistical inference



Want to  
know  
more? See  
[Appendix.](#)

# Feature Engineering - questions to ask

- Achievable
- Rinse-and-repeat
- Iterate and retrospect
- Viability of model
- Simplify
- Essence

[See Appendix for further details](#)

# Conclusion



# What others do? Why?

- Consistency
- Do not reinvent the wheel
- Learn from others

[See Appendix for further details](#)

# What others do?

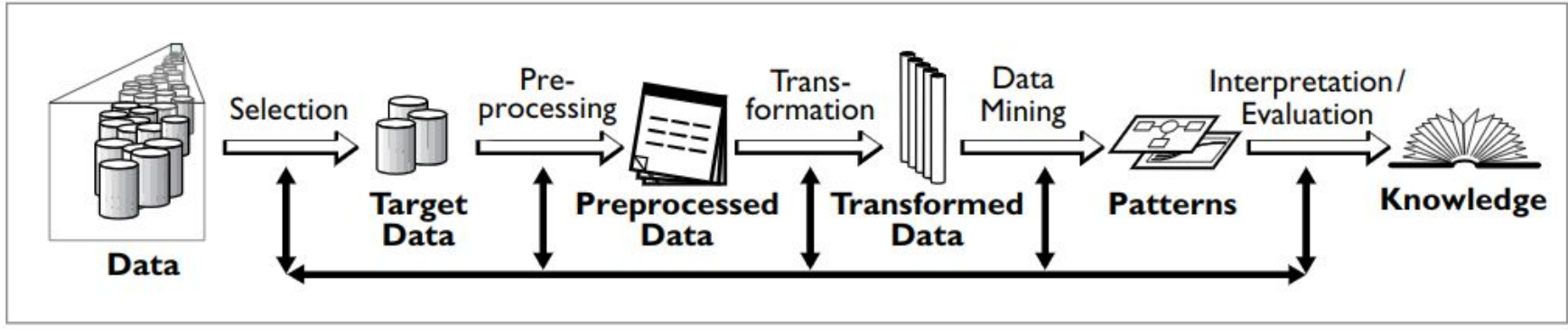
Our presentation so far is an example of  
the methodologies out there

# What others do?

- Frameworks and resources you can learn from, see links: [[1](#)] [[2](#)]
  - Some call it methodology
  - Others call it rules or best practices
- KDD - knowledge discovery from data
- Data Mining book
  - It has been done for the last 20 years
  - We didn't reinvent the wheel

# What others do?

**Figure 1.** Overview of the steps constituting the KDD process



# What others do?

- Motto and purpose: knowing our data is the most important
  - Model is dependent on the quality of the data
  - Garbage in, garbage out!!!



# Overall - questions to ask

- Data Collection - questions to ask
- Exploratory data analysis - questions to ask
- Data Preparation - questions to ask
- Features engineering - questions to ask
- What others do? - questions to ask
  
- Ours is just a guideline
- Ask right questions (and come up with your own)

# Periphery

- Data generation
- Unsupervised learning - clustering
- Data Science Ethics Checklist
- Know your data at Company Level
- AI Transformation Playbook. from Andrew NG
- Making your Neural Network say “I don’t know”

# Additional resources

- Treasure trove of links and resources:  
<http://github.com/neomatrix369/awesome-ai-ml-dl>
- Everything you wanted to know about data:  
<https://github.com/neomatrix369/awesome-ai-ml-dl/tree/master/data>
- Notebooks:  
<https://github.com/neomatrix369/awesome-ai-ml-dl/blob/master/data/README.md#notebooks>
  - [Data Exploratory Analysis](#)
  - [Data Preparation](#)
    - [Data Cleaning](#)
    - [Data Preprocessing / wrangling](#)
  - [Data Generation](#)
  - [Feature engineering](#)
  - [Statistics](#)
  - [Common mistakes](#)

# Additional resources

- Everything you wanted to know about data (2 / 2):
  - <https://github.com/neomatrix369/awesome-ai-ml-dl/tree/master/data>
    - [Cheatsheets](#)
    - [Courses and books](#)
    - [Best practices](#)
    - [Frameworks](#)
    - [Notebooks](#)
- [Understanding Data Science Problems - template of questions to ask](#)

# Shout out!

Join the Meet-a-Mentor  
ML Study Group based in **London, UK**

*We meet weekly!*

[Meetammentor.co.uk](https://www.meetammentor.co.uk)

[@RWmeetammentor](https://www.meetup.com/MaM-Machine-Learning-Study-Group)



# Thank you, feedback, stay in touch!

Please share your feedback, to be applied to the live slides and other resources for everyone's benefit

[@theNeomatrix369](#)

[@jeremiecharlet](#)

# Citations

The images used in this presentation are owned by the respective authors, and many of them come from the <https://thenounproject.com>.

# Appendix

# Data Collection - questions to ask

- What are you going to need ?
- What level of details would be good enough ?
- How much data do you need to start ?
- Do your data will reflect reality ? What are the biases ?

Be ready to repeat the process

# Exploratory Data Analysis - why?

- Black box: if anything wrong, if we don't understand our data, we won't be able to correct it
- We want to solve a problem in the end, with ML, and have good results. If we don't understand our data, we might not be able to train a model because of incoherent or missing data, or have poor results because we have misleading (unbalanced, highly correlated, too many useless features) data. Thus we need to find out if there are incoherent/missing/misleading data
- Know the nature / boundaries / source of the data
- When I work with someone, I want to get to know that person to get the best out of your relationship
- If you don't have domain knowledge, you risk wasting time on finding the root causes of issues, and might never manage to solve the problem

# Exploratory Data Analysis

- Basic characteristics of dataset: get a feeling

```
[4]: df.columns
```

```
[4]: Index(['Movement', 'Gait', 'Rein', 'Comment', 'Date', 'ACCX', 'ACCY', 'ACCZ',  
         'GX', 'GY', 'GZ', 'PITCH', 'ROLL', 'YAW', 'HEIGHT', 'GAIT', 'name',  
         'height'],  
        dtype='object')
```

```
[5]: df.head(3)
```

```
[5]:
```

	Movement	Gait	Rein	Comment	Date	ACCX	ACCY	ACCZ	GX	GY	GZ	PITCH
0	NOT_ON_HORSE	NOT_ON_HORSE	NOT_ON_HORSE	None	2018-08-22 08:48:08.600	-0.00	-0.41	0.29	66.11	-78.55	-5.11	-20.11
1	NOT_ON_HORSE	NOT_ON_HORSE	NOT_ON_HORSE	None	2018-08-22 08:48:08.700	-0.01	-0.83	0.59	15.08	-18.35	0.00	-19.38
2	NOT_ON_HORSE	NOT_ON_HORSE	NOT_ON_HORSE	None	2018-08-22 08:48:08.800	-0.01	-0.83	0.59	0.00	0.00	0.00	-19.92

What fields have we collected for our dataset? What do they mean? Are they gonna help?

You can go a bit further, like check the shape of your dataset (enough data? Too much?), data types (only float in that column or more stuff?) class values (what's in Gait?)

Get a global idea and finer details of what exactly your data looks like

# Exploratory Data Analysis

- Check descriptive statistics: explore a bit more, take some notes, raise questions

Summary details of datasets always helps spot Garbage or incorrect data? Start raising questions about your data

Mmh a horse galloping towards the sky? On a rocket maybe? Some weird outliers we'll need to check / clean

Yaw always equals to 0. The data are useless. Is it gonna be a problem? Is there something wrong with the data collection? Can we simply ignore that column?

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

```
[8]:
```

	ACCX	ACCY	ACCZ	GX	GY	GZ	PITCH	ROLL	YAW	HEIGHT	GAIT	height
count	680859.000000	680859.000000	680859.000000	680859.000000	680859.000000	680859.000000	680859.000000	680859.000000	680859.0	680859.0	680859.000000	680859.000000
mean	0.017209	-0.019569	0.954397	-0.058867	-0.404106	-0.104984	-1.725721	-1.549733	0.0	0.0	7.823378	15.649734
std	0.200833	0.250091	0.195043	1.571396	1.461214	1.353676	11.981994	19.926747	0.0	0.0	22.704810	0.788629
min	-1.460000	-1.610000	-1.450000	-1.082390000	-5.0000000	-1.561130000	-89.429000	-179.981000	0.0	0.0	0.000000	14.100000
25%	-0.150000	-0.160000	0.960000	0.640000	-1.718000	0.540000	-9.790000	-8.550000	0.0	0.0	3.000000	15.000000
50%	0.050000	-0.050000	0.990000	0.000000	-0.540000	0.000000	-3.356000	-3.188000	0.0	0.0	3.000000	16.100000
75%	0.150000	0.100000	1.010000	0.520000	0.580000	0.000000	7.437500	5.641000	0.0	0.0	3.000000	16.200000
max	2.580000	1.390000	2.730000	669.980000	438.640000	358.260000	89.652000	179.904000	0.0	0.0	155.000000	16.200000

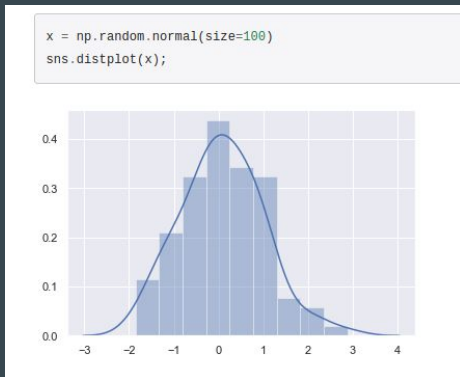
Most ACCZ data around 1. We need to remove gravity on the acceleration



# Exploratory Data Analysis

- Plot distribution of features: check overall distribution
  - Histogram for numeric data
  - Bar chart for categories

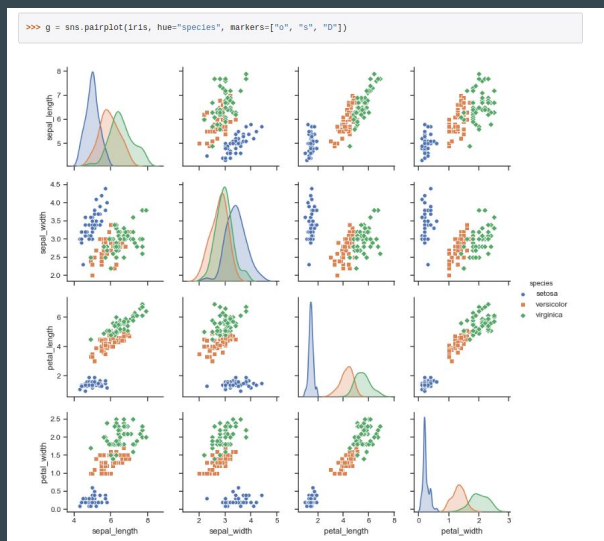
Are the category classes unbalanced? Are there sparse categories that we will need to group together?



Are there outliers ?  
Are the features following gaussian distribution ?

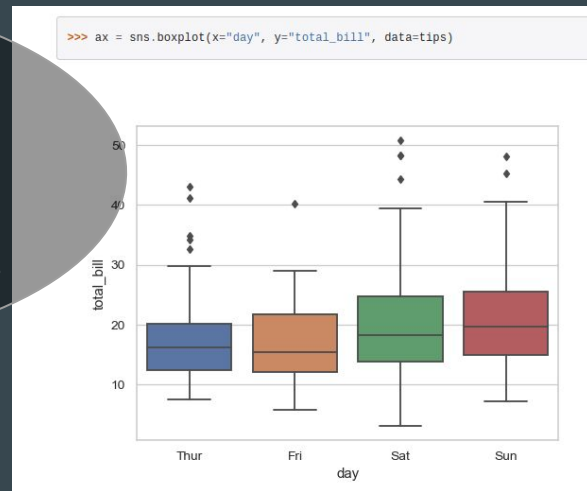
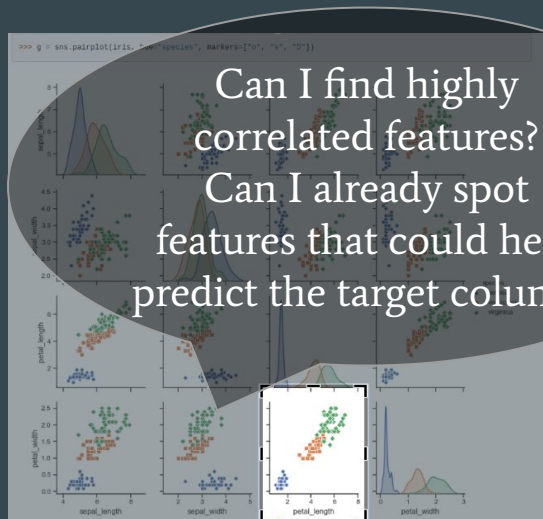
# Exploratory Data Analysis

- Find correlations between features, with target column
  - Scatter charts for numeric data
  - Box whisker / violin plots for category data vs numeric



```
>>> g = sns.pairplot(iris, hue="species", markers=["o", "s", "D"])
```

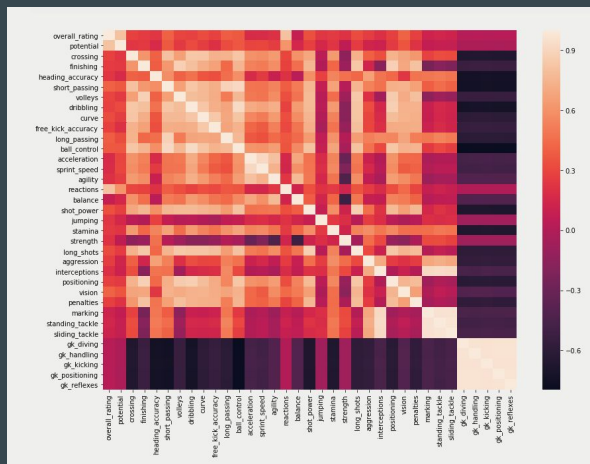
Can I find highly correlated features?  
Can I already spot features that could help predict the target column?



# Exploratory Data Analysis

- Study correlations between features
  - Correlation matrix

Change in the value of one field / variable impacting the other...



# Exploratory Data Analysis - questions to ask

Do you know:

- The domain knowledge needed to understand the data. Have you done your research ?
- Where it comes from
  - Is it manually/automatically generated data or mix of both?
  - Process used to create the data
- Do you know the nature of the bias in the data ? Does it need to be reduced and can it be ?
- Do we know the strengths and weaknesses of the data ? Is the data trustworthy, dirty -> do we need to fix it ?
  - Are there outliers ? should we keep them ?
  - Are there missing data ? what should we do about them ?
- Are the data representative of the domain ?

# Data Preparation - why?

- Garbage in, garbage out. If you work with dirty data, even the most sophisticated models won't be able to get satisfying results. Better data beats fancier algorithms
- To create a clean dataset (so that it has good enough accuracy and correctness)
- So that we can create models that are closer to nature's model

# Data Preparation - data cleaning

Deal with errors (structural)

Type of error (problem)	Technique to use
mislabeled	relabel data automatically or manually
dataset standardisation issue	uniformly replace them
sync issues between sources of data	standardise the data

# Data Preparation - data cleaning

Deal with duplicate data

Type of problem	Technique to use
duplicates	group values with frequencies

[Link to Pandas data wrangling cheat sheet  
from DataCamp](#)

#DoWeKnowOurData



# Data Preparation - data cleaning

Deal with duplicate data

```
>>> s3.unique()  
>>> df2.duplicated('Type')  
>>> df2.drop_duplicates('Type', keep='last')  
>>> df.index.duplicated()
```

Return unique values
Check duplicates
Drop duplicates
Check index duplicates

# Data Preparation - data cleaning

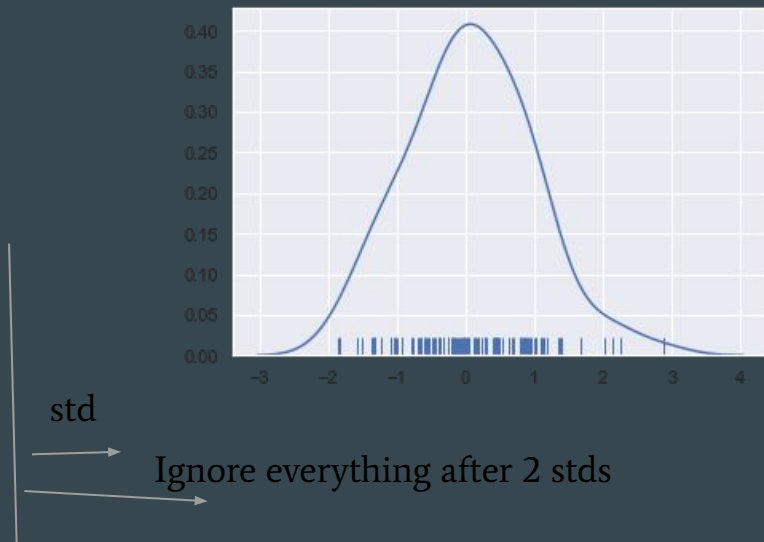
## Deal with outliers

We tend to just ignore them while they can be very important in some cases (like anomaly detection) - expand on it!

Type of problem	Technique to use
outliers	<ul style="list-style-type: none"><li>- distribution graphs and/or pair plots</li><li>- use of mean and std dev</li><li>- filter out the outliers</li><li>- apply filter to smoothen out a curve</li><li>- to decide: to keep the outliers or try to remove them</li></ul>

# Data Preparation - data cleaning

Deal with outliers



# Data Preparation - data cleaning

Deal with missing data

Type of problem	Technique to use
missing data	the best way to handle missing data is to first flag them as missing and fill with default values: median, mean, weighted mean, predicted, zero
	<ul style="list-style-type: none"><li>- fill based on observations or correlations</li><li>- to generate synthetic data to fill missing values</li></ul> Using sklearn Imputer or KNN
	<ul style="list-style-type: none"><li>- decide whether to drop rows with missing values</li></ul>

# Data Preparation

Deal with too much data (information overload) [1 / 2]

Type of problem	Technique to use
<p>needle in a haystack problems</p> <p>(huge dataset with disproportionate class distribution: e.g. we try to detect a class which occurs in 0.5% of the data (horse rolling which is a rare event vs simply standing or lying))</p>	<p>Step1: group data + histogram - to identify the disproportion</p> <p>Step 2: Undersampling the classes to remove data</p> <p>Step 3: Oversampling by adding more data</p>
	<p>Step 1: Manage at the training stage (adjust hyperparameter) (check ML Mastery for more techniques in the google docs)</p>

# Data Preparation

Deal with too much data (information overload) [2 / 2]

Type of problem	Technique to use
<b>dataset with class overload problems</b>  (column with astronomical number of categories. e.g. city in house prices)	<ul style="list-style-type: none"><li>- Group together sparse categories</li><li>- Remove sparse categories</li><li>- Summarising categories into higher levels of abstractions</li></ul>

# Data Preparation - questions to ask

Do we have those problems to fix and have we ?

- Outliers
- Missing data
- Class overload
- Too many features
- Unbalanced dataset
- Have we removed or balanced any existing bias in the dataset?

# Feature Engineering - why?

- To find hidden information
- To extract the essence of the data which is representative of the rest of the data
- Improve training efficiency: accuracy, speed, good use of resources



# Feature Engineering

- Feature extraction

Type of problem	Technique to use
find hidden information	<ul style="list-style-type: none"><li>- group together sparse classes</li><li>- create new calculated columns, for e.g. extracting weekday from date</li><li>- generate relevant labels with the help of results from clustering</li></ul>

# Feature Engineering

- Applying math / statistical functions

Type of problem	Technique to use
<b>find hidden information</b>	<ul style="list-style-type: none"><li>- convert to absolute values</li><li>- apply root mean square</li><li>- use logarithmic functions</li><li>- applying rolling mean / stddev / min / max</li></ul>
<b>Improve distribution, remove skewness</b>	And manage precision of the data!

# Feature Engineering

- Applying physics related functions

Type of problem	Technique to use
find hidden information	<ul style="list-style-type: none"><li>- Energy</li><li>- Energy rate</li><li>- Short Term Average / Long term Avg</li><li>- Kurtosis</li><li>- FFT (Fast Fourier Transform)</li></ul>

# Feature Engineering

- Feature scaling

Type of problem	Technique to use
<b>Improve distribution, remove skewness.</b>  <b>Required for some models</b>	<ul style="list-style-type: none"><li>- Standardization</li><li>- Normalization</li><li>- Map to uniform or gaussian distribution</li></ul>

# Feature Engineering

- Dimensionality Reduction [1 / 2]

Type of problem	Technique to use
too many features	Factorisation (PCA)
	ICA Independent Component Analysis
	t-SNE t-Distributed Stochastic Neighbour Embedding
	UMAP Uniform Manifold Approximation and Projection

# Feature Engineering

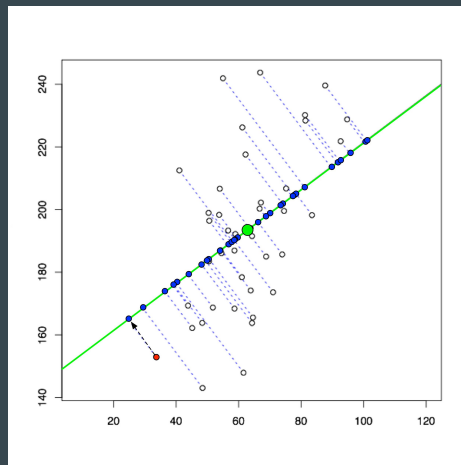
- Dimensionality Reduction [2 / 2]

Type of problem

too many features

Technique to use

Principal Component Analysis (PCA)



# Feature Engineering

- Feature selection [1 / 3]

Type of problem	Technique to use
too many features  (dataset on house prices has 50 features. where to start, what does really affect the prices? Should I train on everything, just a subset of the features, how to choose them, etc?)	Manual feature selection  Programmatically > Tree-based, during training  Genetic process of selecting potent features

# Feature Engineering

- Feature selection [ 2 / 3 ]

Type of problem	Technique to use
too many features	<p>(Manual)</p> <p>Filter out features which are highly correlated.</p> <p>Plot multi scatter chart</p> <p>Use correlation table (might need to remove features which were used for extraction)</p>



# Feature Engineering

- Feature selection [ 3 / 3 ]

Type of problem	Technique to use
too many features	<p>(Programmatically)</p> <ul style="list-style-type: none"><li>- Tree based feature selection</li></ul> <p>Use feature importance from XGBoost or RandomForest</p> <ul style="list-style-type: none"><li>- During training</li></ul> <p>Recursive feature elimination (select features by recursively considering smaller and smaller sets of features)</p>

# Feature Engineering

- Feature selection:

Type of problem	Technique to use
too many features	H2O driverless AI: Genetic algorithmic process of selecting potent features (automated)

# Statistical inference

- [Understanding statistical inference \[video\]](#)
- [Four ideas of Statistical Inference](#)
- [An Introduction to Statistical Learning \[book\]](#)
- [Statistical Inference \[course\]](#)



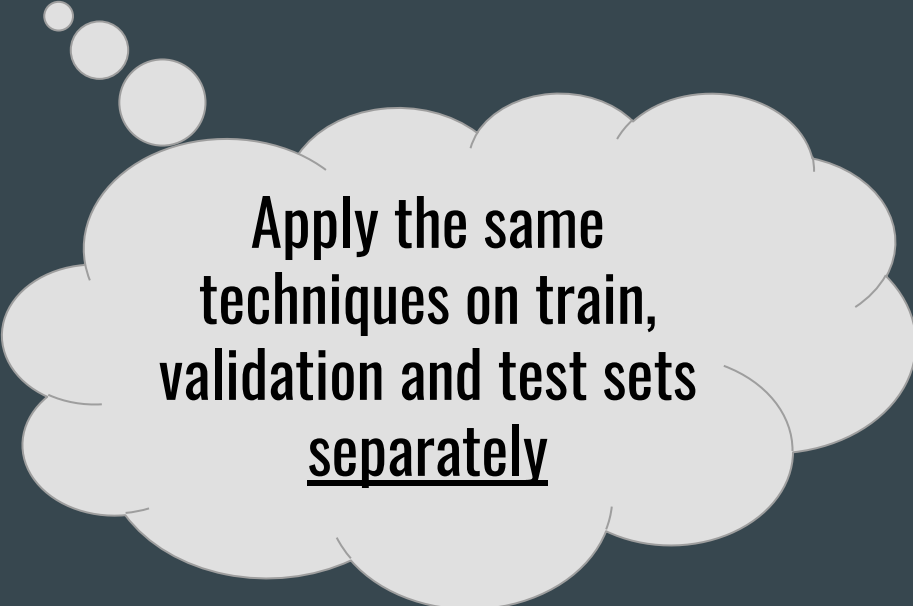
**Create a baseline first!**

**Feature engineering is an art!**



**Improve over training  
iterations!**

# Feature Engineering



**Apply the same  
techniques on train,  
validation and test sets  
separately**

# Feature Engineering

**Repeat steps in the  
Data Exploratory  
section, again**

**Check the  
distribution of your  
data (descriptive  
analytics)**

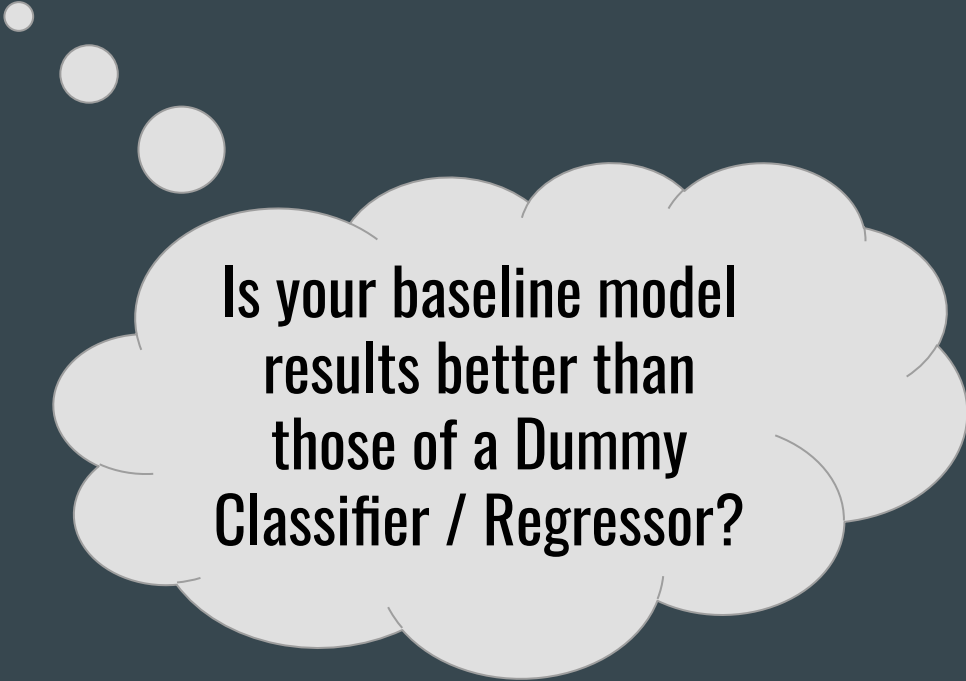
*Verify your data*

**Question your  
results !!!**

**Visualise feature  
engineered dataset**

# Feature Engineering

See Dummy Classifier or  
Dummy Regressor



Is your baseline model  
results better than  
those of a Dummy  
Classifier / Regressor?

# Feature Engineering - questions to ask

- Can I create a model from my dataset efficiently given my resources?
- How do we make best use of the trial-error process yet get the best out of the dataset / model?
- How can I make informed decisions and record them and reuse them in my next iteration?
- How can I make sure that the end model is useful to solve my business problem after going through the tedious process?
- From the trends we identified previously can we make it simpler for the model to pick and use such information from the dataset?
- How can I get the essence out of my original dataset ? Can I express the intuitions I got during the analysis in a more obvious way ?



# What others do? Why?

- Consistency
- Do not reinvent the wheel
- Learning from the good work of others from the past and current times
- Learning and applying from the lessons learnt from tried and tested empirical experiments from the past

# Tips from Mark Bell (TNA, Data Scientist / Research)

- Data is rarely clean
- Tidy your data
- Visualise your data
- Know your numbers
  - High values; Low values; Missing values
  - Quartiles
  - Mean; Medians
  - Correlations
- Create your own features
- Go to Kaggle!

# It's harder than it looks!

[Slides to talk](#)

- Keep your data & labels separate!
  - “How to Prevent Catastrophic Failure in Production ML Systems, Martin Goodson”, QCon London 2019
- Need a methodology - or can get overwhelmed by all the possibilities
- If it looks too good to be true...



Thanks [Dave](#)  
[Snowdon](#) &  
[Jeremie](#)  
[Charlet](#)

[callingbullshit.org](http://callingbullshit.org)

#DoWeKnowOurData