Chapter X – Data Centric Learning

Instructor: Dr. Hongfu Liu
Email: hongfuliu@brandeis.edu

Data Visualization

- Tools
  - Bar, scatter plot, line, histograms, pie, box plots, bubble chart
  - 2D plot, not 3D

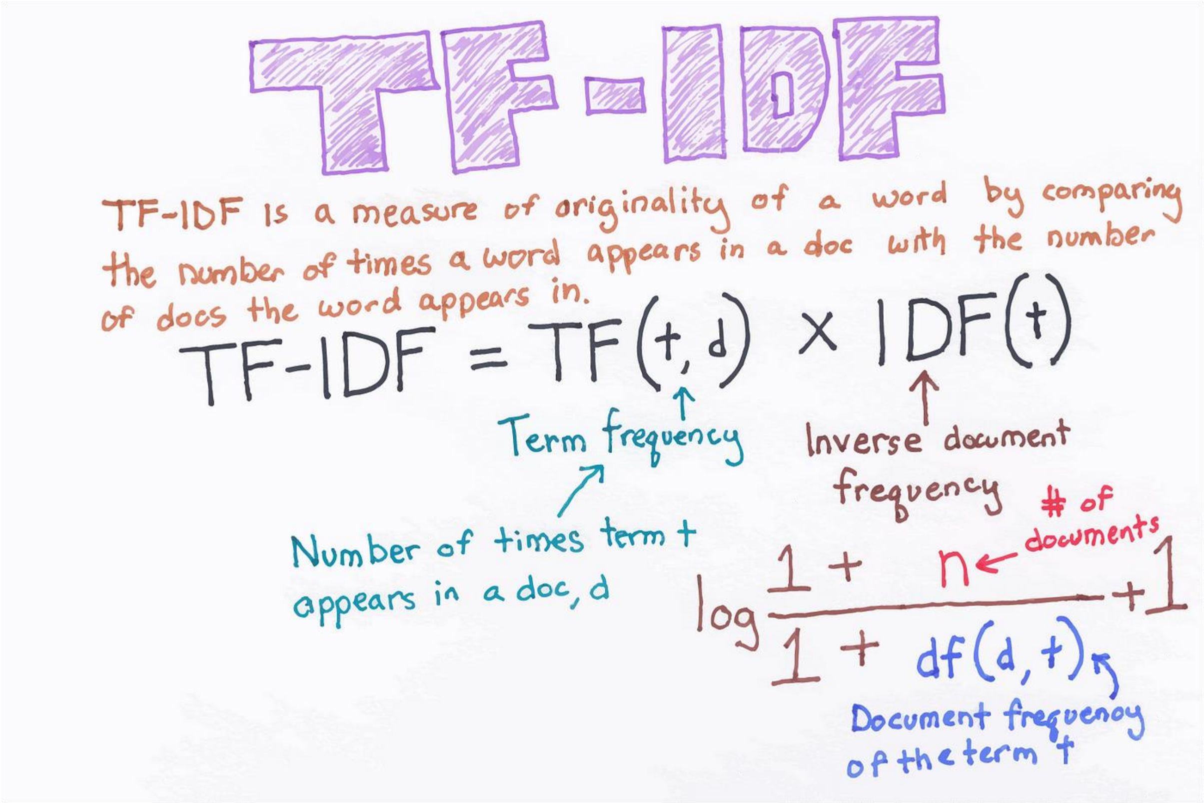
- Check samples or features

- The key is to understand the intrinsic structure or the predictive power (linear or non-linear correlation).

https://www.youtube.com/watch?v=csXmVBw8cdo

Normalization

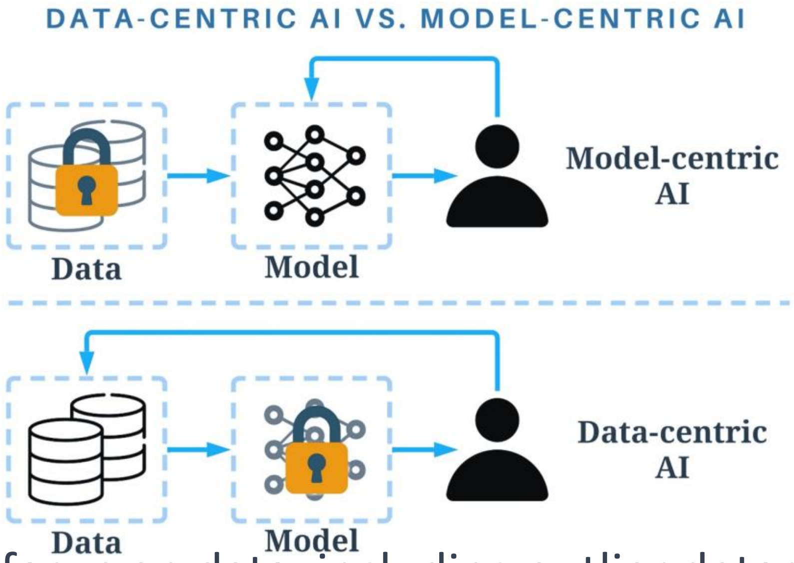
- Attribute Type
  - Zero-mean
  - Min-max
  - TF-IDF



- When to normalize?

https://medium.com/analytics-vidhya/demonstrating-calculation-of-tf-idf-from-sklearn-4f9526e7e78b

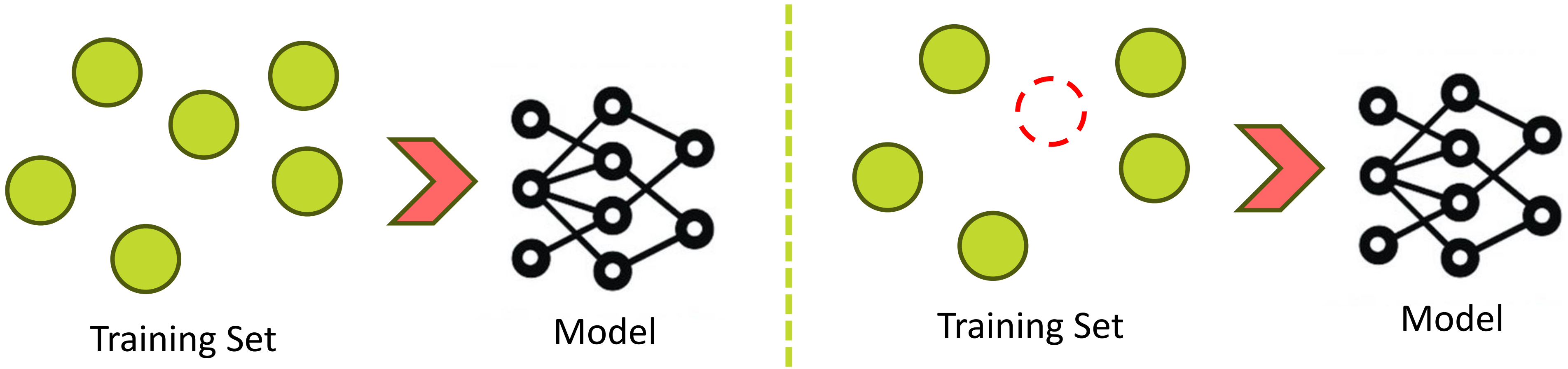
- One man's trash is another man's treasure
- The assessment of data valuation cannot be **isolated** from the overarching goal.



- Some techniques that also focus on data, including outlier detection, data augmentation and so on, do not belong to data-centric learning.

Data-Centric Learning

- The fundamental question in data-centric learning is how to assess the data valuation.
- One naïve way is the famous *leave-one-out* influence.



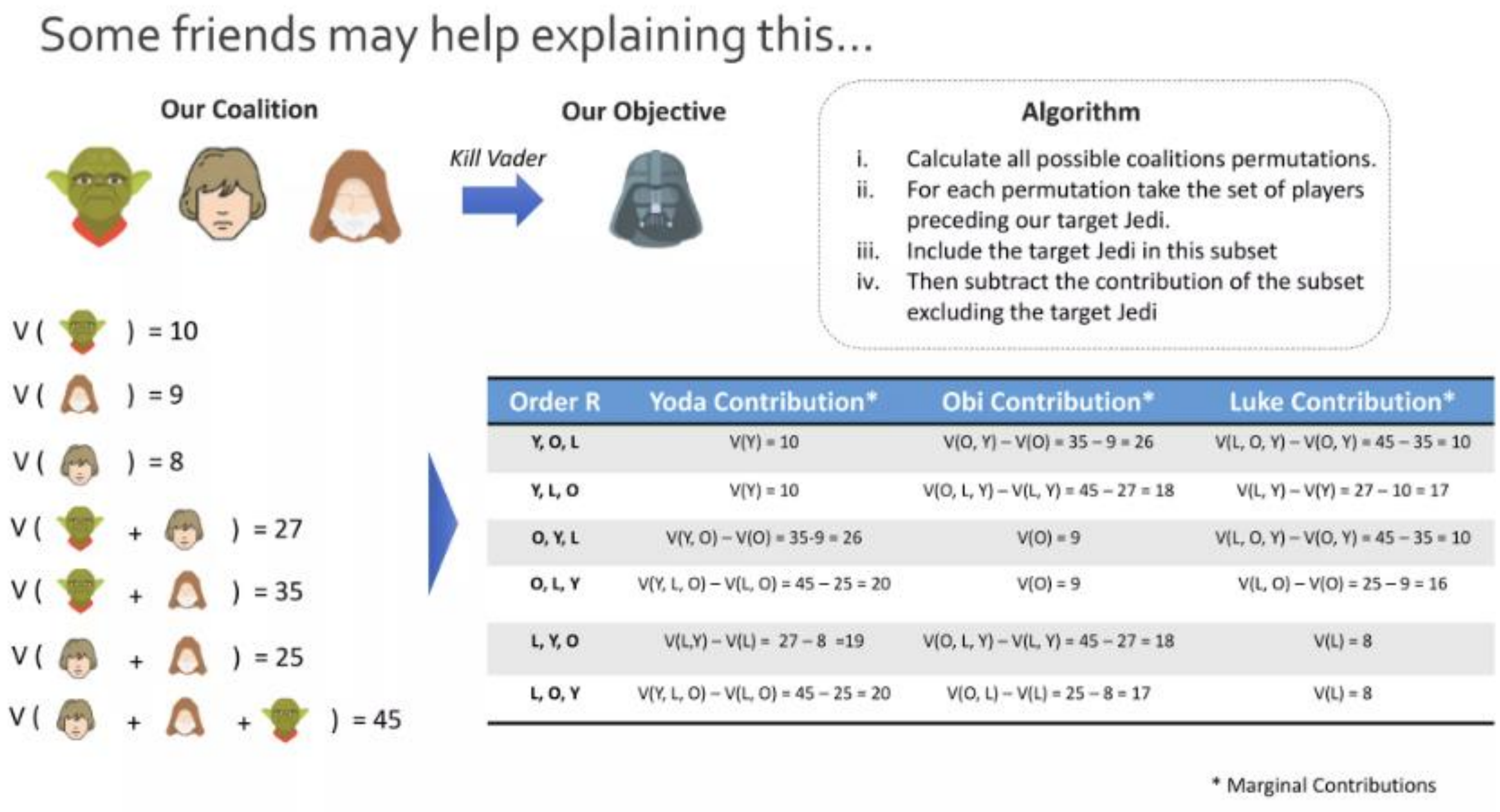
- We measure the *performance change on the validation set* with one data sample removed from the training set.

Shapley - Sample

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\;(n - |S| - 1)!}{n!}(v(S \cup \{i\}) - v(S))$$

$$= \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S))$$

Shapley - Sample



Algorithm

i. Calculate all possible coalitions permutations.
ii. For each permutation take the set of players preceding our target Jedi.
iii. Include the target Jedi in this subset
iv. Then subtract the contribution of the subset excluding the target Jedi

Order R	Yoda Contribution*	Obi Contribution*	Luke Contribution*
Y, O, L	V(Y) = 10	V(O, Y) – V(O) = 35 – 9 = 26	V(L, O, Y) – V(O, Y) = 45 – 35 = 10
Y, L, O	V(Y) = 10	V(O, L, Y) – V(L, Y) = 45 – 27 = 18	V(L, Y) – V(Y) = 27 – 10 = 17
O, Y, L	V(Y, O) – V(O) = 35-9 = 26	V(O) = 9	V(L, O, Y) – V(O, Y) = 45 – 35 = 10
O, L, Y	V(Y, L, O) – V(L, O) = 45 – 25 = 20	V(O) = 9	V(L, O) – V(O) = 25 – 9 = 16
L, Y, O	V(L,Y) – V(L) = 27 – 8 =19	V(O, L, Y) – V(L, Y) = 45 – 27 = 18	V(L) = 8
L, O, Y	V(Y, L, O) – V(L, O) = 45 – 25 = 20	V(O, L) – V(L) = 25 – 8 = 17	V(L) = 8

* Marginal Contributions

https://www.slideshare.net/slideshow/shap-144149270/144149270#19

Shapley - Sample

Now we can calculate the payout for each Jedi

	Initial Value	Payout (SHAP Value)
	10	10 + 10 + 26 + 20 + 19 + 20 = **17,5**
	9	26 + 18 + 9 + 9 + 18 + 17 = **16,2**
	8	10 + 17 + 10 + 16 + 8 + 8 = **11.5**

So what? ...After calculating each player marginal contributions* we realize that although Luke is 20% weaker the **contributed** 34% less than Yoda. Obi in terms of contribution is much closer to Yoda!

*"The Shapley value can be misinterpreted. The Shapley value of a feature value is not the difference of the predicted value after removing the feature from the model training. The interpretation of the Shapley value is: Given the current set of feature values, the contribution of a feature value to the difference between the actual prediction and the mean prediction is the estimated Shapley value" (https://christophm.github.io/interpretable-ml-book/shapley.html#general-idea)

Shapley - Sample

- KNN-Shapley

$$
\begin{aligned}
s_{\alpha_N} &= \frac{\mathbb{1}[y_{\alpha_N} = y_{test}]}{N} \\
s_{\alpha_i} &= s_{\alpha_{i+1}} + \frac{\mathbb{1}[y_{\alpha_i} = y_{test}] - \mathbb{1}[y_{\alpha_{i+1}} = y_{test}]}{K} \frac{\min\{K, i\}}{i}
\end{aligned}
$$

https://pydvl.org/stable/examples/shapley_knn_flowers/#building-a-dataset-and-a-utility

Shapley - Feature

Here we use a selection of 50 samples from the dataset to represent “typical” feature values, and then use 500 perterbation samples to estimate the SHAP values for a given prediction. Note that this requires 500 * 50 evaluations of the model.

[5]:
```
explainer = shap.KernelExplainer(f, X.iloc[:50, :])
shap_values = explainer.shap_values(X.iloc[299, :], nsamples=500)
shap.force_plot(explainer.expected_value, shap_values, X_display.iloc[299, :])
```

[5]:

Explain many predictions

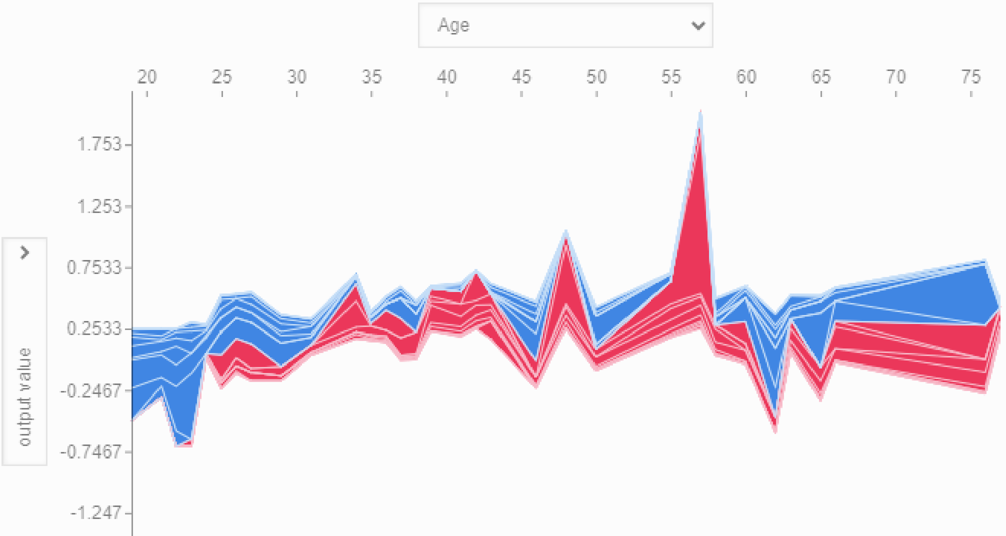
Here we repeat the above explanation process for 50 individuals. Since we are using a sampling based approximation each explanation can take a couple seconds depending on your machine setup.

[6]:
```
shap_values50 = explainer.shap_values(X.iloc[280:330, :], nsamples=500)
```

```
100%|██████████| 50/50 [00:53<00:00,  1.08s/it]
```

[7]:
```
shap.force_plot(explainer.expected_value, shap_values50, X_display.iloc[280:330, :])
```

[7]:



https://shap.readthedocs.io/en/latest/


Q
&
A