Evaluations

Every Model is Wrong, but Some Are Useful

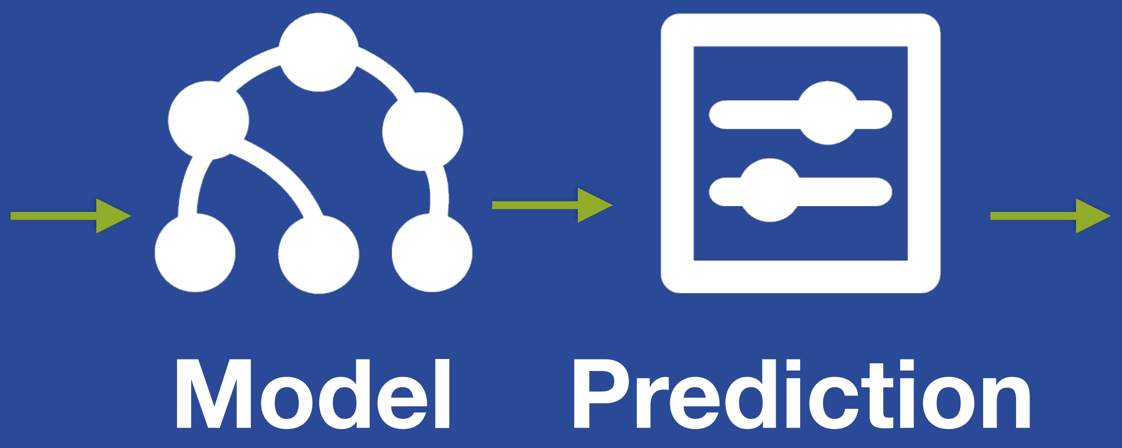
Poul Petersen
CIO, BigML, Inc

Why Evaluations

- FACT: No model is perfect - they **all** make mistakes
  - Your data has mistakes
  - Models are "approximations"
- Today you will/have seen models that predict:
  - Churn:
  - Diabetes:
  - Home Prices:
- You have also seen several different kinds of models
  - Decision Trees / Ensembles / Logistic Regression / Deepnets
  - Which one works the best for **your** data

Easy Right?

INTL MIN	INTL CALLS	INTL CHARGE	CUST SERV CALLS	CHURN
8.7	4	2.35	1	False
11.2	5	3.02	0	False
12.7	6	3.43	4	True
9.1	5	2.46	0	False
11.2	2	3.02	1	False
12.3	5	3.32	3	False
13.1	6	3.54	4	False
5.4	9	1.46	4	True
13.8	4	3.73	1	False



PREDICT CHURN
False
True
True
False
False
False
False
False
False

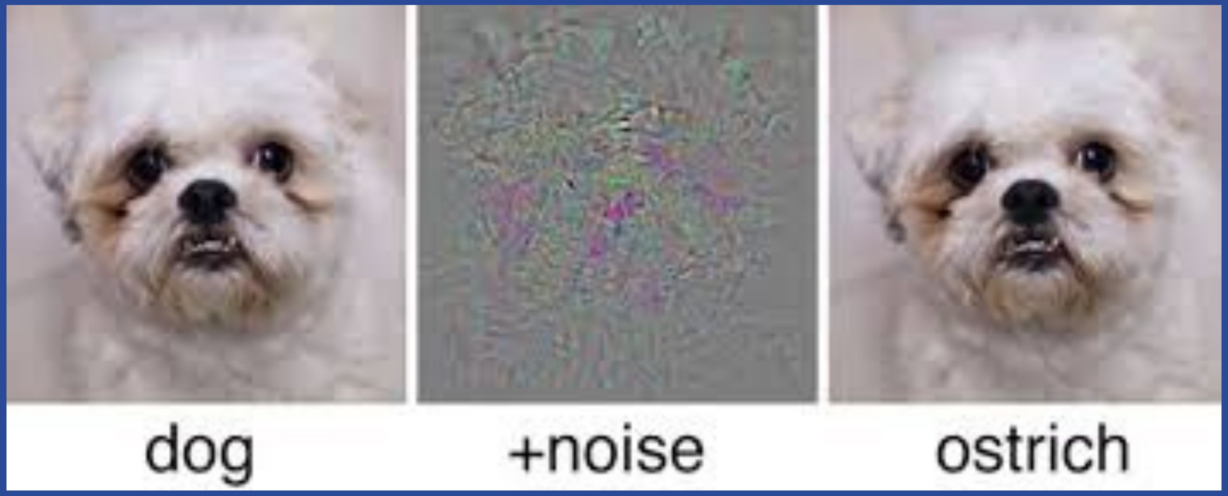
Count up mistakes!

Evaluations Demo #1

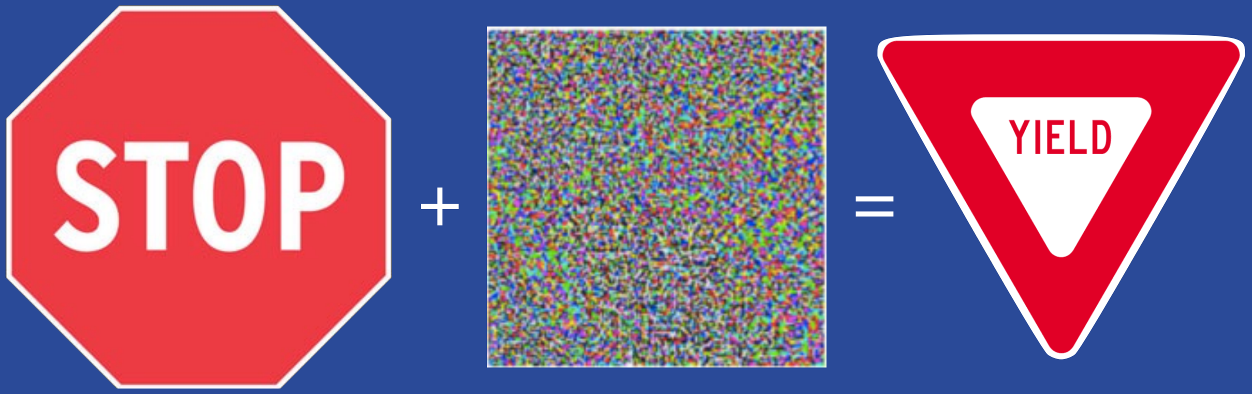
What Just Happened?

- We started with the churn **Datasource**
- Created a **Dataset**
- Built a **Model** to predict churn
- We used the **Model** to predict churn for each customer in the **Dataset** using a **Batch Prediction**
- Downloaded the **Batch Prediction** as a CSV and looked for errors. That is, when the **Prediction** did not match the known true value for churn
- The comparison was tedious!
    - Examining one line at a time
    - Hard to understand - need some metrics!!!

BUT Mistakes can be Costly



FUN!



DANGER!

Insight: Labeling a Yield as a stop is not as bad as labelling a stop as a yield…
We REALLY need better metrics!

Evaluation Metrics

- Imagine we have a model that can predict a person's dominant hand, that is for any individual it predicts `left / right`
- Define the **positive** class
  - This selection is arbitrary
  - It is the class you are interested in!
  - The **negative** class is the "other" class (or others)
- For this example, we choose : `left`

Evaluation Metrics

- We choose the positive class: `left`
- True Positive `(TP)`
  - We predicted `left` and the correct answer was `left`
- True Negative `(TN)`
  - We predicted `right` and the correct answer was `right`
- False Positive `(FP)`
  - Predicted `left` but the correct answer was `right`
- False Negative `(FN)`
  - Predict `right` but the correct answer was `left`

Evaluation Metrics

Remember…

True *Positive*: Correctly predicted the *positive* class

True *Negative*: Correctly predicted the *negative* class

False *Positive*: Incorrectly predicted the *positive* class

False *Negative*: Incorrectly predicted the *negative* class

Accuracy

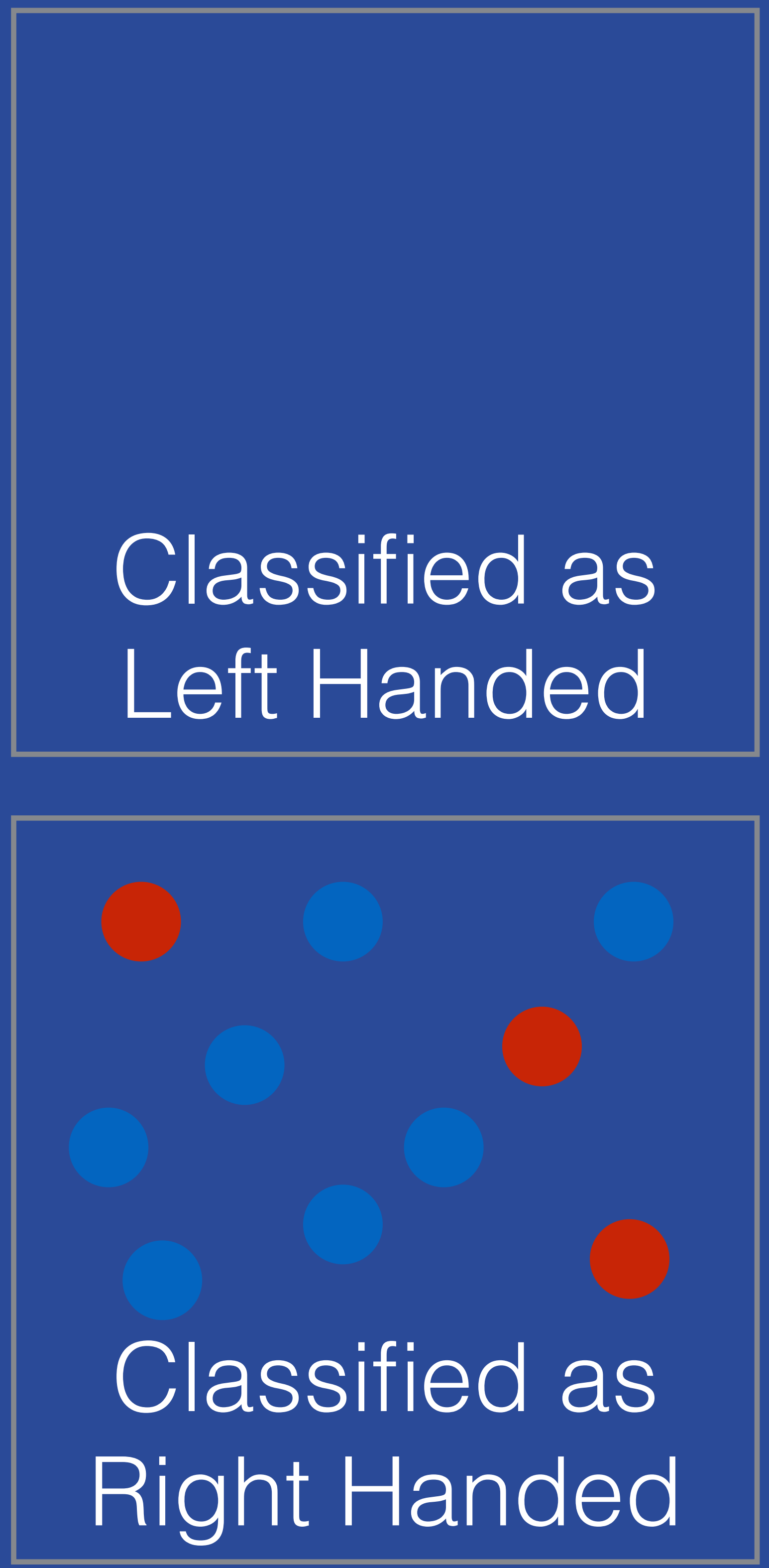
$$\frac{TP + TN}{Total}$$

- "Percentage correct" - like an exam
- If `Accuracy = 1` then no mistakes
- If `Accuracy = 0` then all mistakes
- Intuitive but not always useful
- Watch out for unbalanced classes!
  - Ex: 90% of people are right-handed and 10% are left
  - A silly model which *always* predicts right handed is **90% accurate**

Accuracy

Positive Class

Negative Class



= Left

= Right

$$TP = 0$$
$$FP = 0$$
$$TN = 7$$
$$FN = 3$$

$$\frac{TP + TN}{Total} = 70\%$$

Precision

$$\frac{TP}{TP + FP}$$

- “accuracy” or “purity” of <u>positive</u> class
- How well you did separating the positive class from the negative class
- If `Precision = 1` then no `FP`.
  - You may have missed some left handers, but of the ones you identified, <u>**all**</u> are left handed. No mistakes.
- If `Precision = 0` then no `TP`
  - None of the left handers you identified are actually left handed. All mistakes.

Precision

= Left

= Right

$$TP = 2$$
$$FP = 2$$
$$TN = 5$$
$$FN = 1$$

$$\frac{TP}{TP + FP} = 50\%$$

Recall

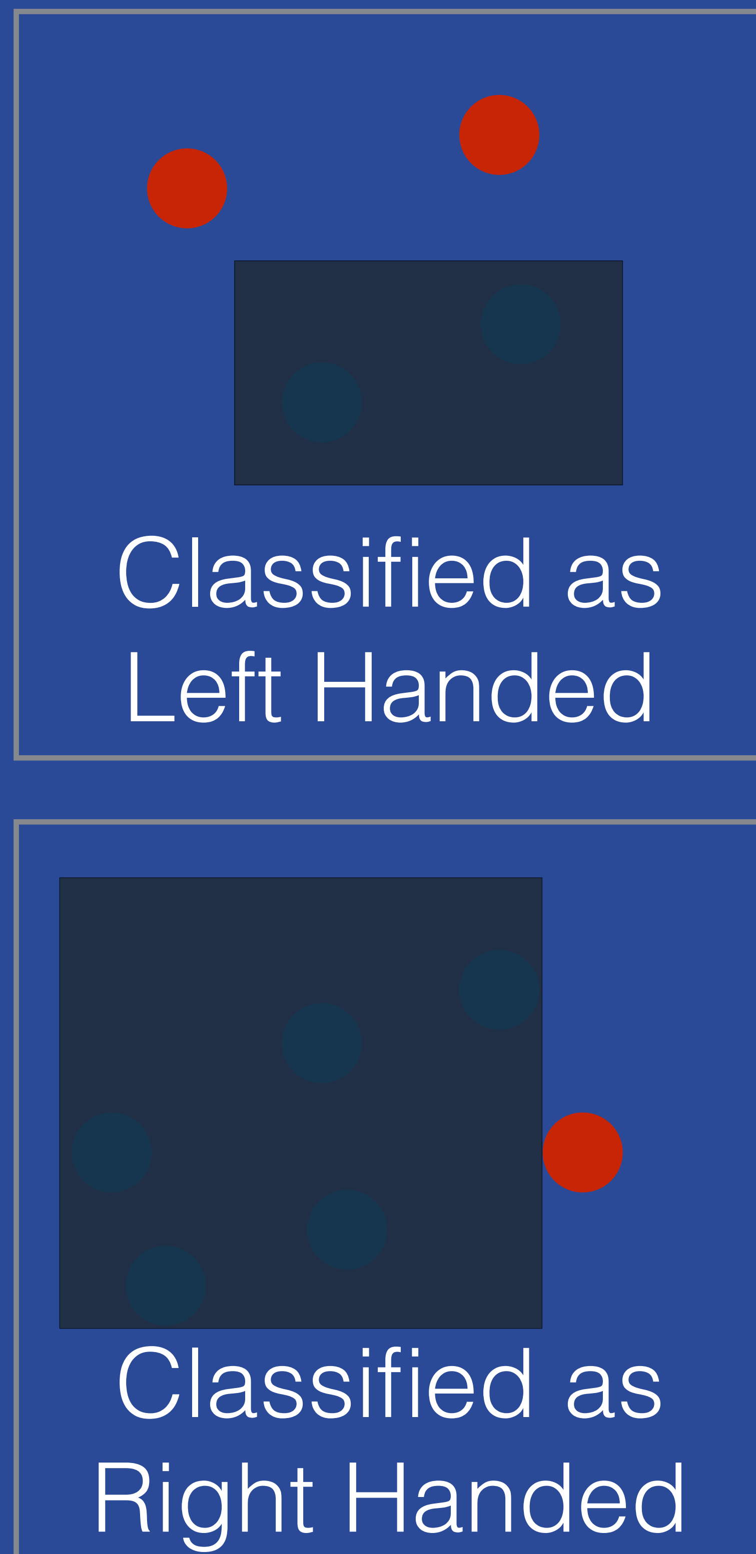
$$\frac{TP}{TP + FN}$$

- percentage of positive class correctly identified
- A measure of how well you identified all of the positive class examples
- If `Recall = 1` then no FN → **All** left handers identified
  - There may be FP, so precision could be <1
- If `Recall = 0` then no TP → **No** left handers identified

Recall

Positive Class

Negative Class



= Left

= Right

$$TP = 2$$
$$FP = 2$$
$$TN = 5$$
$$FN = 1$$

$$\frac{TP}{TP + FN} = 66\%$$

f-Measure

$$\frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

- harmonic mean of `Recall` & `Precision`
- If `f-measure = 1` then `Recall == Precision == 1`
- If `Precision` OR `Recall` is small then the `f-measure` is small

Phi Coefficient

```
__________TP*TN_-_FP*FN__________
SQRT[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]
```

- Returns a value between `-1` and `1`
- If `-1` then predictions are opposite reality
- =`0` no correlation between predictions and reality
- =`1` then predictions are always correct

Evaluations Demo #2

What Just Happened?

- Starting with the Diabetes **Source**, we created a **Dataset** and then a **Model**.
- Using both the **Model** and the original **Dataset**, we created an **Evaluation.**
- We reviewed the metrics provided by the Evaluation:
    - Confusion Matrix
    - `Accuracy`, `Precision`, `Recall`, `f-measure` and `phi`
- This **Model** seemed to perform really, really well…

Question: Can we trust this model?

Evaluation Danger!

- **Never evaluate with the training data!**
  - Many models are able to “memorize” the training data
  - This will result in overly optimistic evaluations!

“Memorizing” Training Data

Training

plasma glucose	bmi	diabetes pedigree	age	diabetes
148	33.6	0.627	50	TRUE
85	26.6	0.351	31	FALSE
183	23.3	0.672	32	TRUE
89	28.1	0.167	21	FALSE
137	43.1	2.288	33	TRUE
116	25.6	0.201	30	FALSE
78	31	0.248	26	TRUE
115	35.3	0.134	29	FALSE
197	30.5	0.158	53	TRUE

Evaluating

plasma glucose	bmi	diabetes pedigree	age	diabetes
148	33.6	0.627	50	?
85	26.6	0.351	31	?

- Exactly the same values!
- Who needs a model?
- What we want to know is how the model performs with values never seen at training:

124	22	0.107	46	?

Evaluation Danger!

- **Never evaluate with the training data!**
  - Many models are able to “memorize” the training data
  - This will result in overly optimistic evaluations!
  - If you only have one Dataset, use a train/test split

Train / Test Split

Train

plasma glucose	bmi	diabetes pedigree	age	diabetes
148	33.6	0.627	50	TRUE
85	26.6	0.351	31	FALSE
183	23.3	0.672	32	TRUE
89	28.1	0.167	21	FALSE
137	43.1	2.288	33	TRUE
116	25.6	0.201	30	FALSE
78	31	0.248	26	TRUE
115	35.3	0.134	29	FALSE
197	30.5	0.158	53	TRUE

Test

- These instances were never seen at training time.
- Better evaluation of how the model will perform with “new” data

Evaluation Danger!

- **Never evaluate with the training data!**
  - Many models are able to “memorize” the training data
  - This will result in overly optimistic evaluations!
  - If you only have one Dataset, use a train/test split
- **Even a train/test split may not be enough!**
  - Might get a “lucky” split
  - Solution is to repeat several times (formally to cross validate)

Evaluation Demo #3

Your Turn!

- Start with the Diabetes source and create a dataset
- Split it 80/20 with the seed **bigml**
- Build a 1-click model on the 80%
- Evaluate with the 20%
    - What is the phi score?
    - **Bonus**: which class has the best recall?

Evaluation

DANGER ZONE

- **Never evaluate with the training data!**
  - Many models are able to “memorize” the training data
  - This will result in overly optimistic evaluations!
  - If you only have one **Dataset**, use a train/test split
- **Even a train/test split may not be enough!**
  - Might get a “lucky” split
  - Solution is to repeat several times (formally to cross validate)
- **Don’t forget that accuracy can be mis-leading!**
  - Mostly useless with unbalanced classes (left/right?)
  - Use weighting, operating points, other tricks…

Weighting

Instance	Rate	Payment	Outcome
1	23%	134	Paid
2	23%	134	Paid
3	23%	134	Paid
...	...	...	...
1000	23%	134	Paid
1001	23%	134	Default

Predict	Confidence
Paid	20%
Paid	25%
Paid	30%
...	
Paid	99.5%
Paid	99.4%

Problem: Default is "more important",but occurs less often than Paid
Solution: Weights tell the model to treat instances of a specific class (in this case Default) with more importance

Operating Points

- The default probability threshold is 50%
- Changing the threshold can change the outcome for a specific class

Rate	Payment	…	Actual Outcome
8.4%	$456	…	PAID
9.6%	$134	…	PAID
18%	$937	…	DEFAULT
21%	$35	…	PAID
17.5%	$1,044	…	DEFAULT

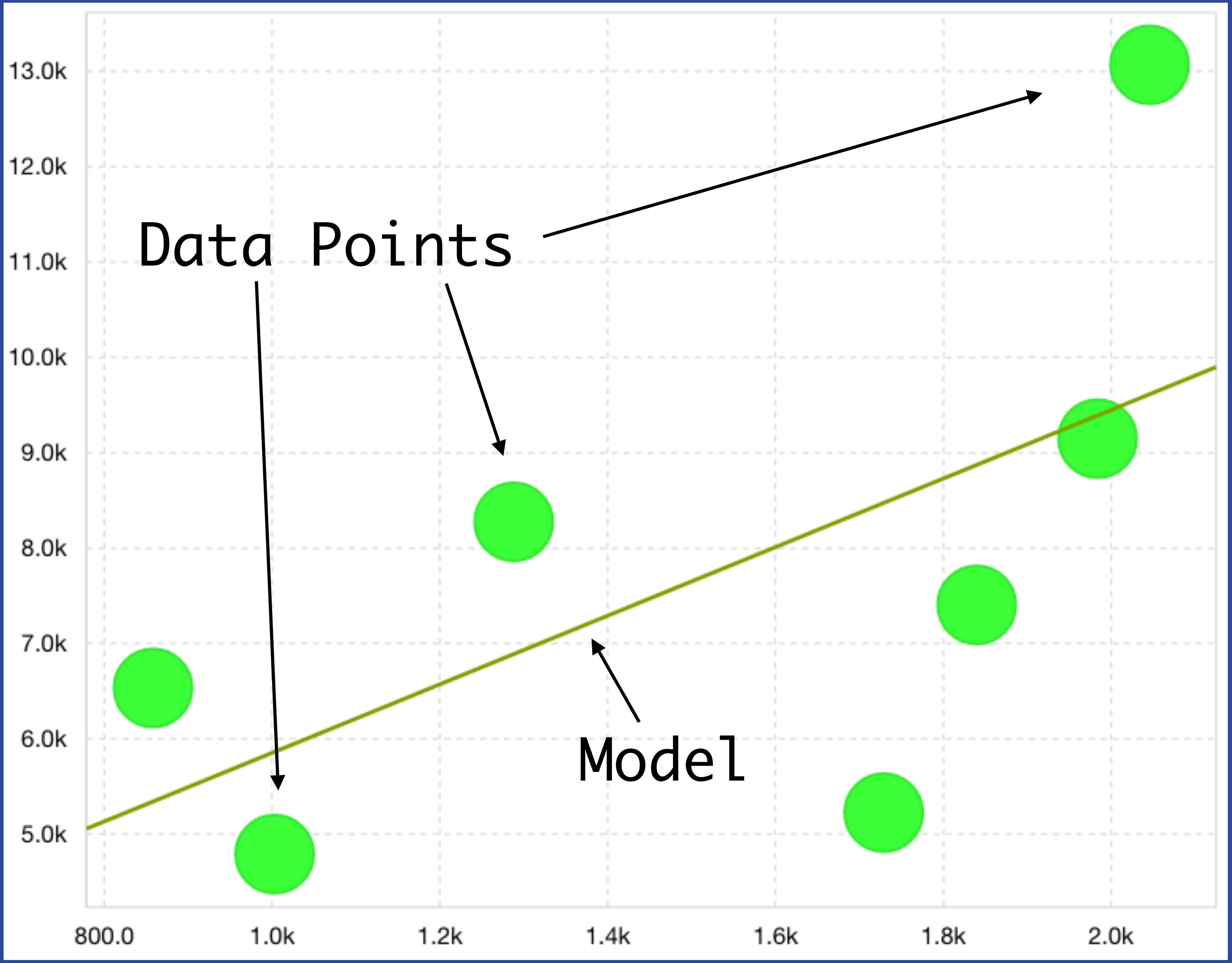
Probability PAID	Threshold @ 50%	Threshold @ 60%	Threshold @ 90%
95%	PAID	PAID	PAID
87%	PAID	PAID	DEFAULT
36%	DEFAULT	DEFAULT	DEFAULT
88%	PAID	PAID	DEFAULT
55%	PAID	DEFAULT	DEFAULT

Evaluations Demo #4

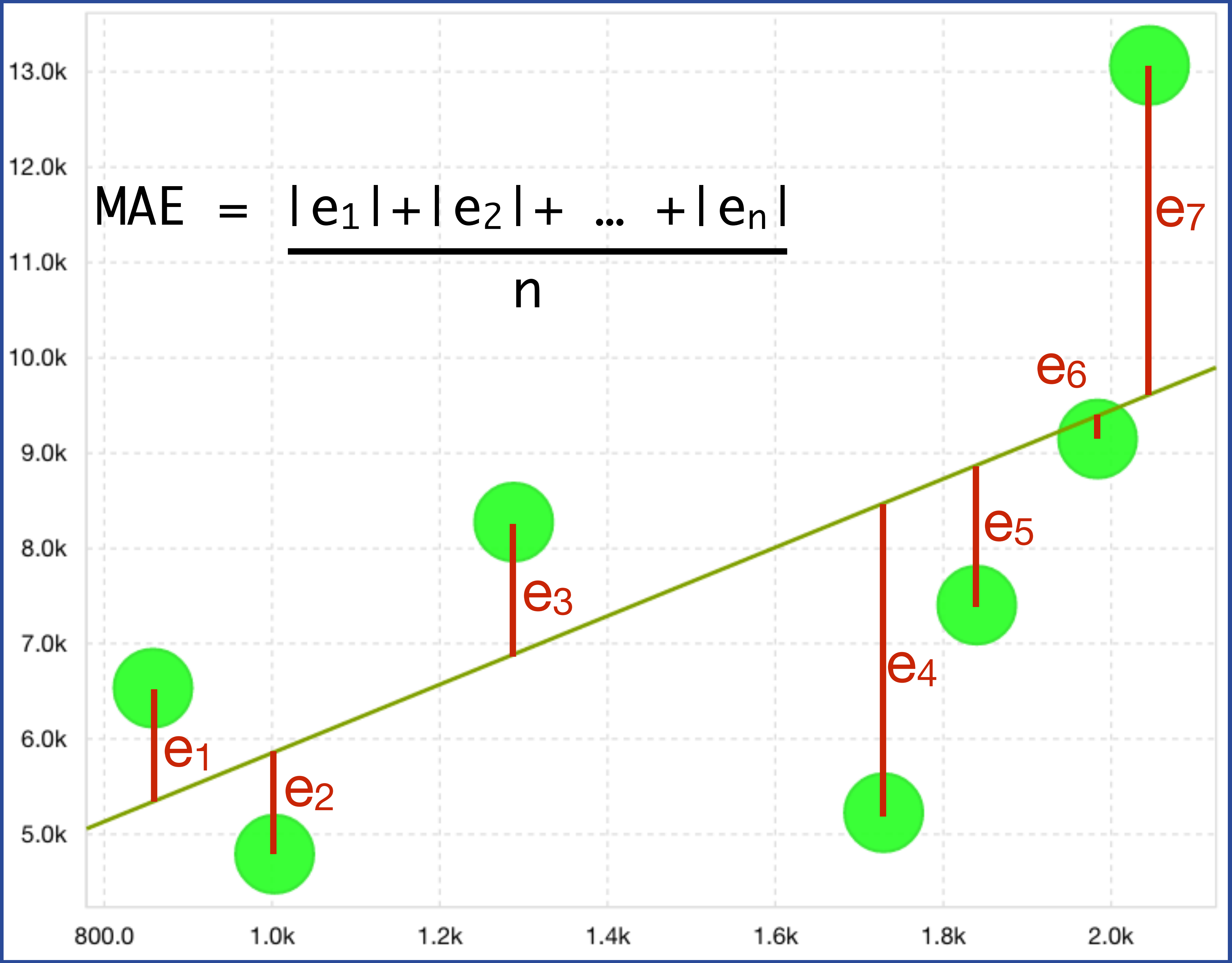
Your Turn!

- Use the same 80/20 split of Diabetes from before
- Build a model with balance objective: true
- Evaluate the model
- How does it compare to the previous model?
- Which class, if any, is performing differently? Why?
- Can you detect more diabetes by changing the operating point? If so, at what cost?

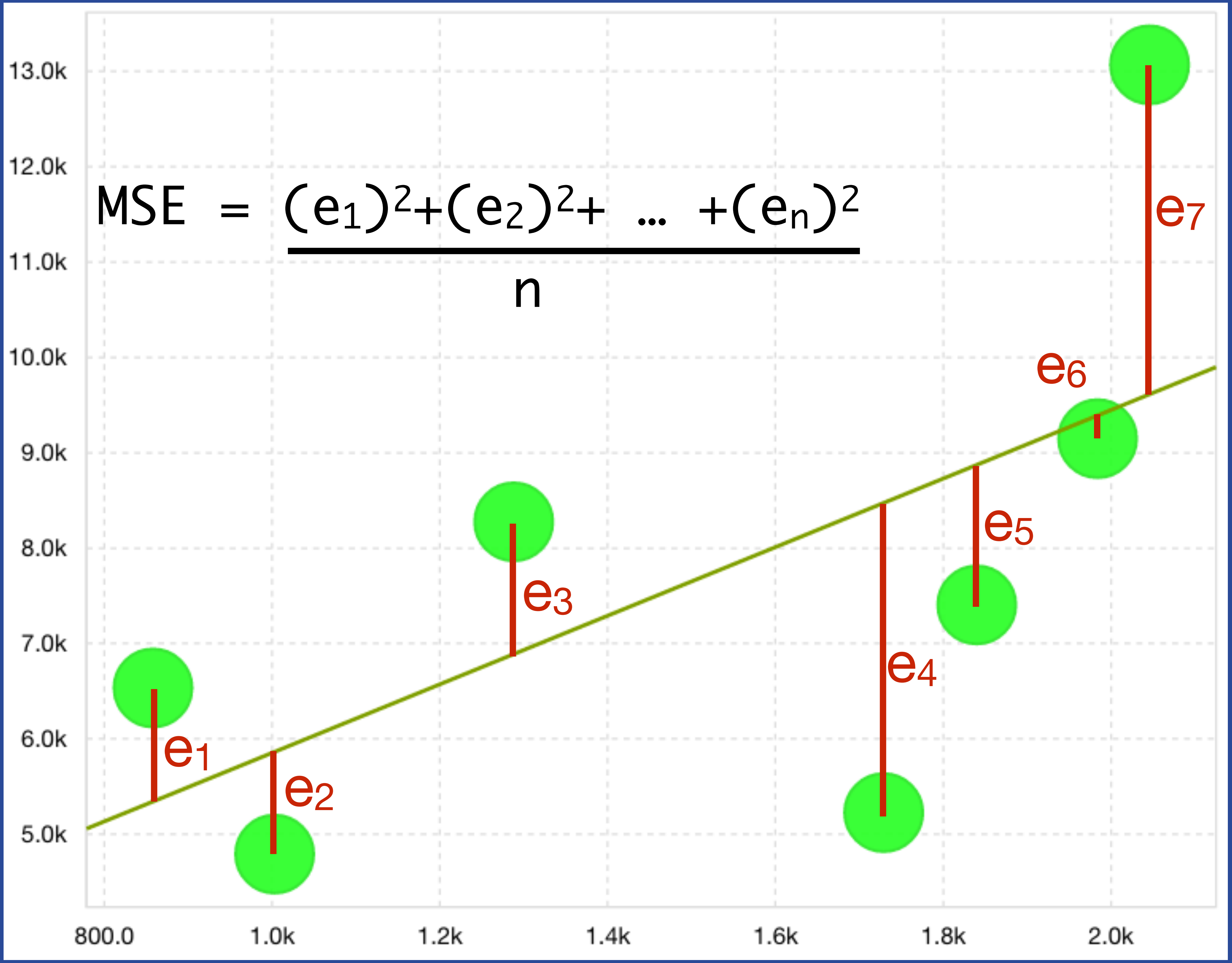
Regression - Fitting a Line



Mean Absolute Error



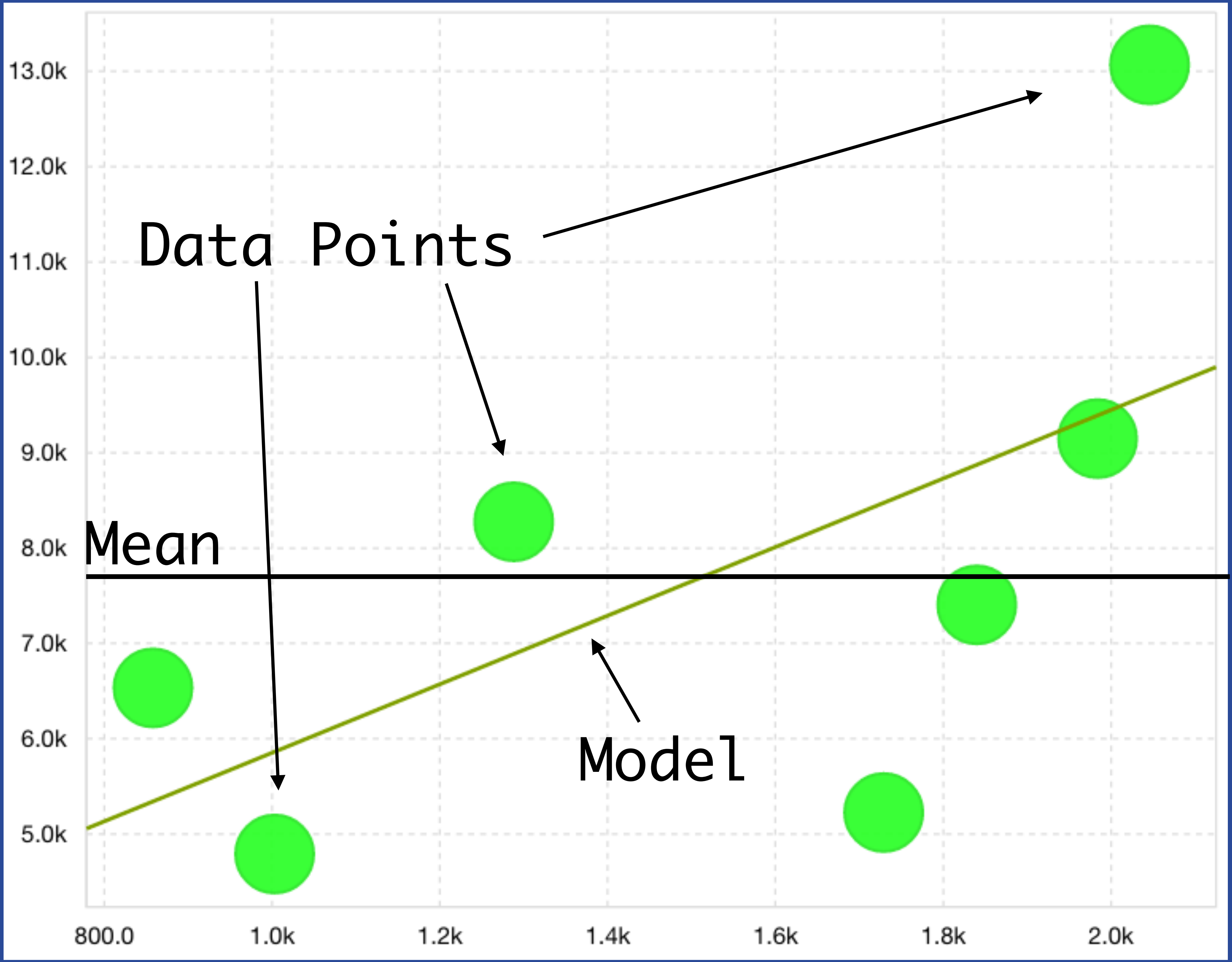
Mean Squared Error



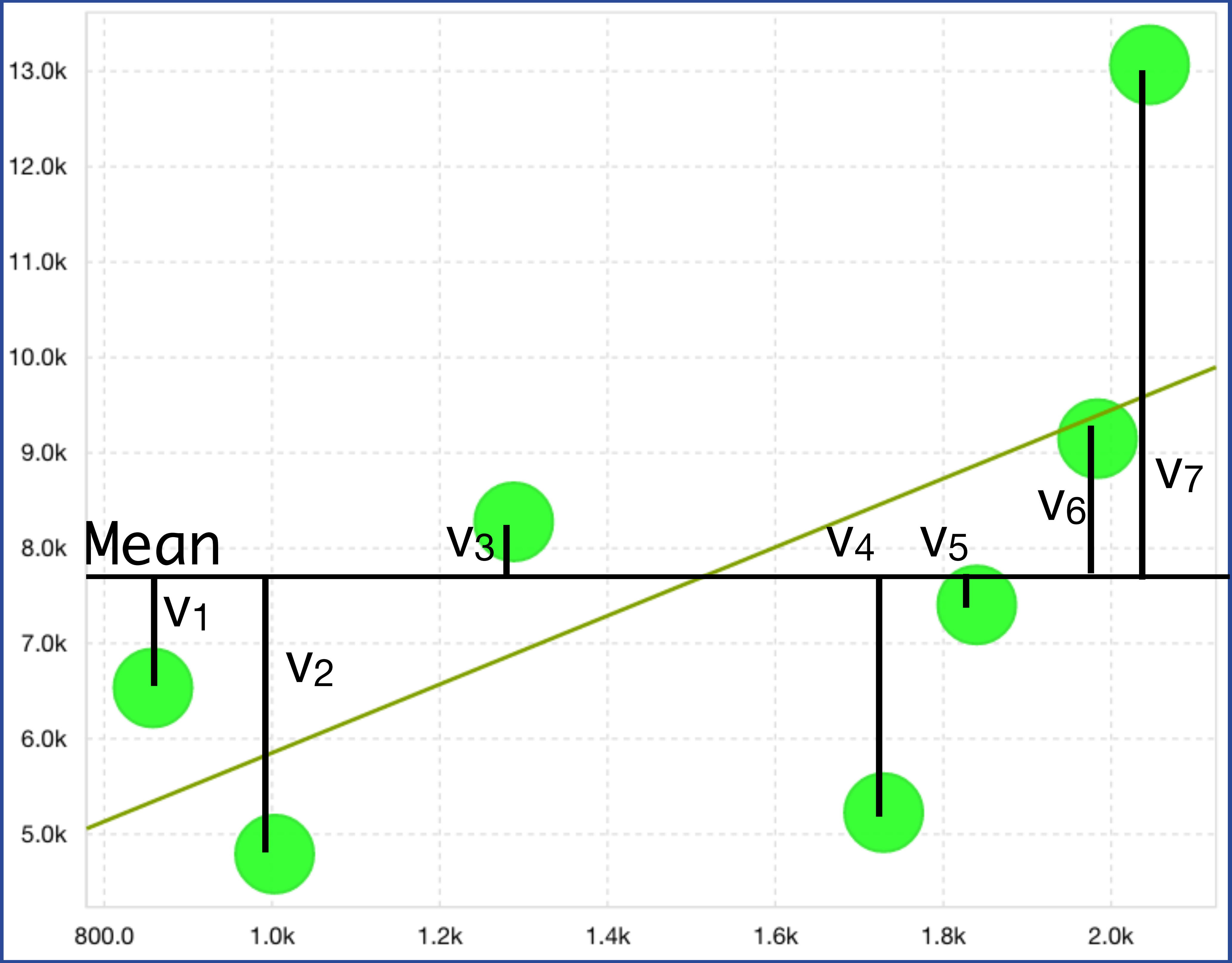
MSE versus MAE

- For both MAE & MSE: Smaller is better, but values are unbounded
- MSE is always larger than or equal to MAE

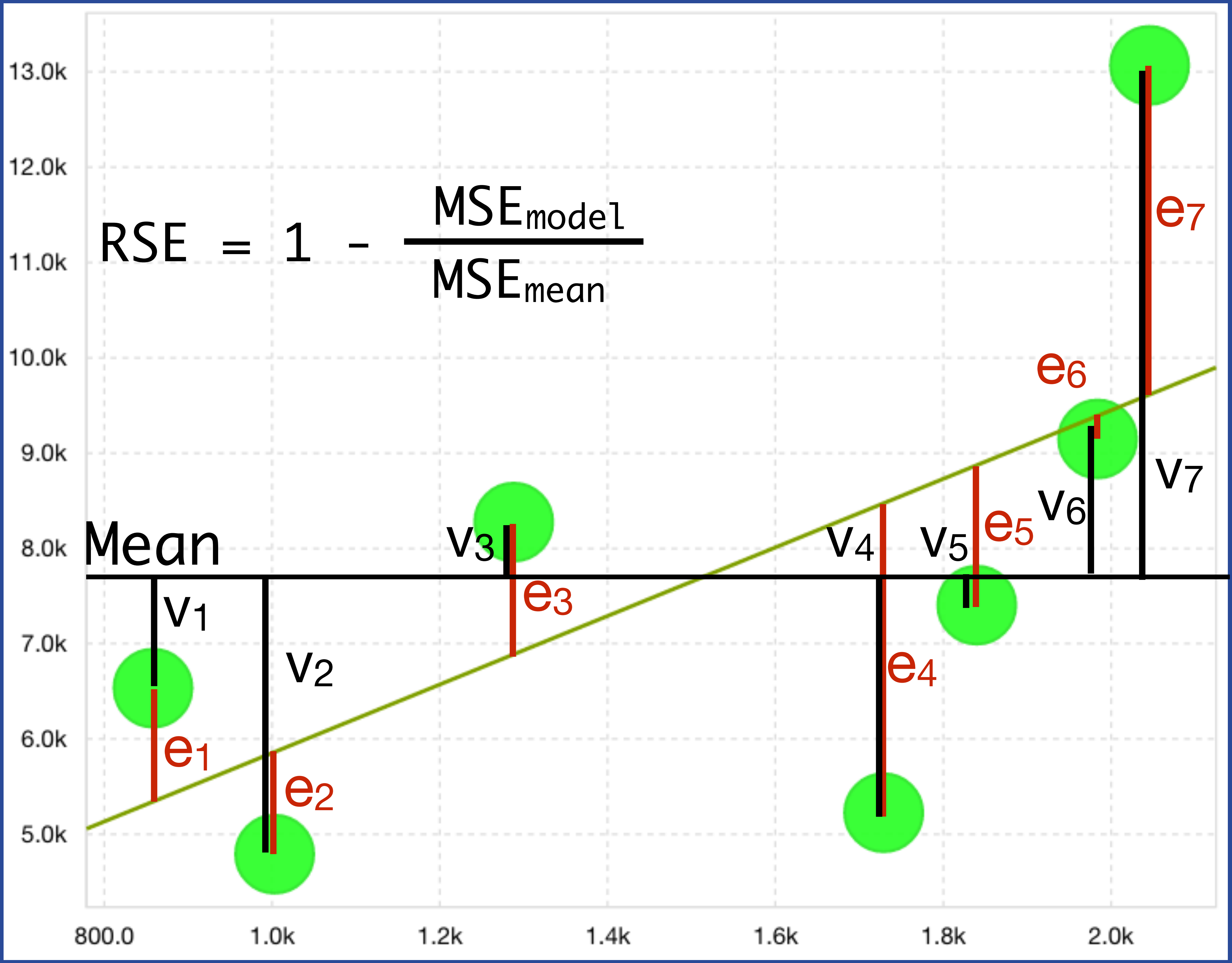
R-Squared Error



R-Squared Error



R-Squared Error



R-Squared Error

$$RSE = 1 - \frac{MSE_{model}}{MSE_{mean}}$$

- RSE: measure of how much better the model is than always predicting the mean
- < 0 model is worse then mean
  - $MSE_{model} > MSE_{mean}$
- $= 0$ model is no better than the mean
  - $MSE_{model} = MSE_{mean}$
- $\rightarrow 1$ model fits the data "perfectly"
  - $MSE_{model} = 0$ (or $MSE_{mean} >> MSE_{model}$)

Evaluations Demo #5

Multi-Class Evaluations

- All the evaluation metrics are built from TP/FP/TN/FN which requires that the class only have two states:
  - True/False
  - Left/Right
  - On/Off
- What happens with multiple classes?
  - Yes/No/Maybe
  - OK/Suspicious/Fraud
  - Brown/Orange/White/Yellow
- Basically, one-versus-all:
  - The positive class is still the one you are interested in
  - The negative class is “everything else”

Evaluation Demo #6

bigml®