LINEAR CLASSIFIERS IN PYTHON

Linear classifiers: prediction equations

Michael (Mike) Gelbart
Instructor
The University of British Columbia

Dot products

```
In [1]: x = np.arange(3)

In [2]: x
Out[2]: array([0, 1, 2])
```

```
In [3]: y = np.arange(3,6)

In [4]: y
Out[4]: array([3, 4, 5])
```

```
In [5]: x*y
Out[5]: array([0, 4, 10])
```

```
In [6]: np.sum(x*y)
Out[6]: 14
```

```
In [7]: x@y
Out[7]: 14
```

- `x@y` is called the dot product of `x` and `y`, and is written $x \cdot y$.

Linear classifier prediction

- $\text{raw model output} = \text{coefficients} \cdot \text{features} + \text{intercept}$
- Linear classifier prediction: compute raw model output, check the

  sign

  - if positive, predict one class
  - if negative, predict the other class
- This is the same for logistic regression and linear SVM
  - `fit` is different but `predict` is the same

How LogisticRegression makes predictions

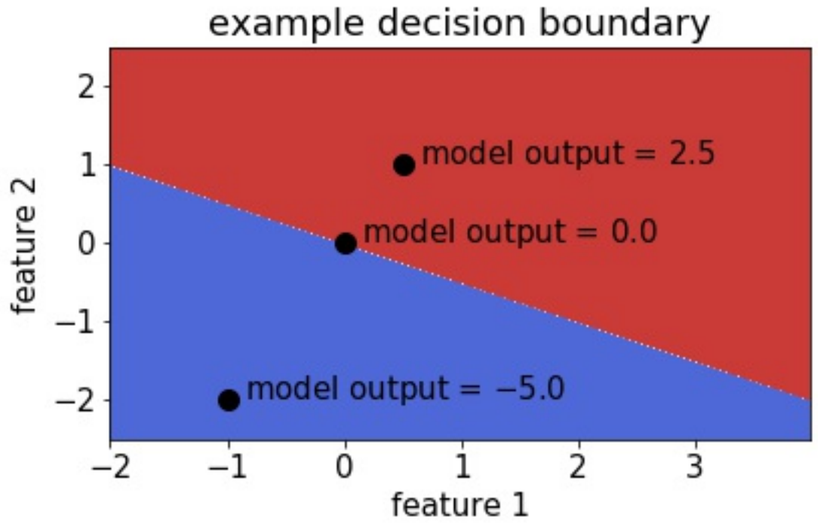
$$\text{raw model output} = \text{coefficients} \cdot \text{features} + \text{intercept}$$

```
In [1]: lr = LogisticRegression()

In [2]: lr.fit(X,y)

In [3]: lr.predict(X)[10]
Out[3]: 0

In [4]: lr.predict(X)[20]
Out[4]: 1
```

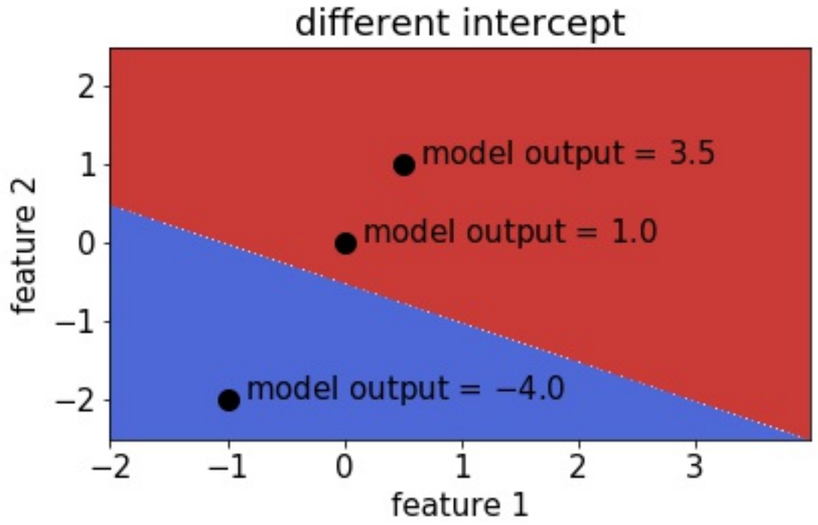
```
In [5]: lr.coef_ @ X[10] + lr.intercept_ # raw model output
Out[5]: array([-33.78572166])
```

```
In [6]: lr.coef_ @ X[20] + lr.intercept_ # raw model output
Out[6]: array([ 0.08050621])
```

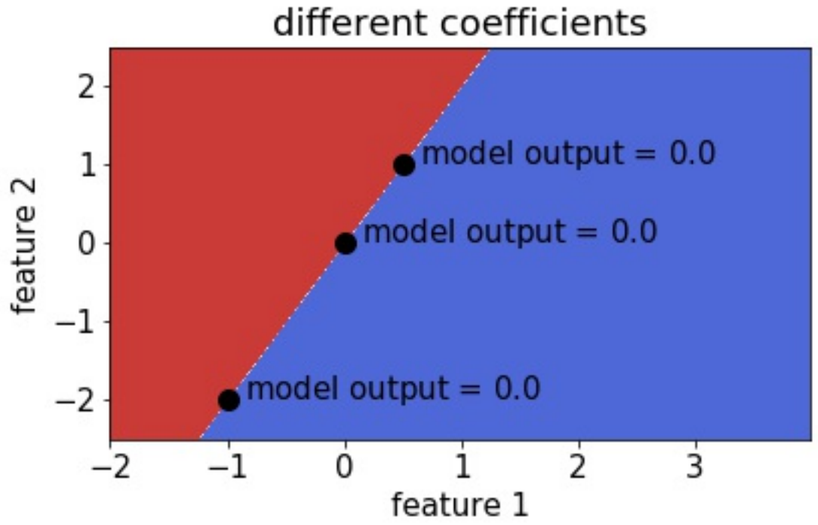
The raw model output



The raw model output



The raw model output



LINEAR CLASSIFIERS IN PYTHON

Let's practice!

LINEAR CLASSIFIERS IN PYTHON

What is a loss function?

Michael Gelbart
Instructor
The University of British Columbia

Least squares: the squared loss

- scikit-learn's LinearRegression minimizes a loss:

$$\sum_{i=1}^{n}(\text{true } i\text{th target value} - \text{predicted } i\text{th target value})^2$$

- Minimization is with respect to coefficients or parameters of the model.
- Note that in scikit-learn model.score() isn't necessarily the loss function.

Classification errors: the 0-1 loss

- Squared loss not appropriate for classification problems (more on this later).
- A natural loss for classification problem is the number of errors.
- This is the **0-1 loss**: it's 0 for a correct prediction and 1 for an incorrect prediction.
- But this loss is hard to minimize!

Minimizing a loss

```
In [1]: from scipy.optimize import minimize
```

```
In [2]: minimize(np.square, 0).x
Out[2]: array([0.])
```

```
In [3]: minimize(np.square, 2).x
array([-1.88846401e-08])
```

LINEAR CLASSIFIERS IN PYTHON

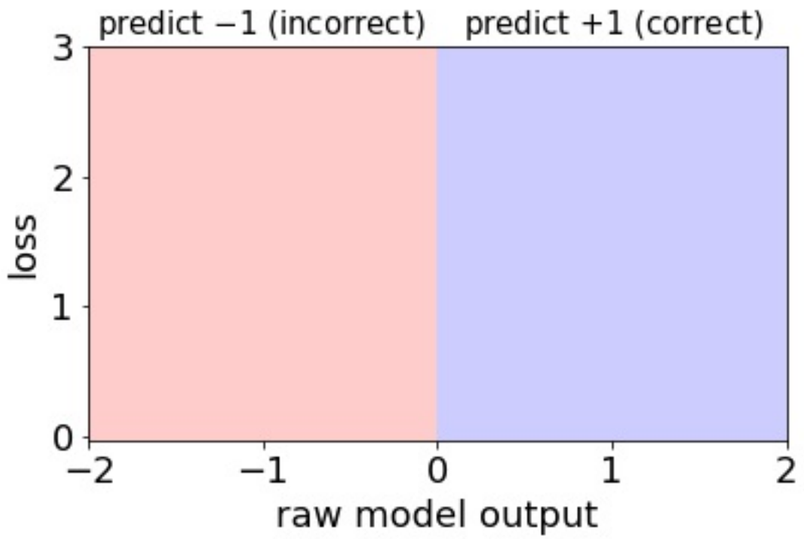
Let's practice!

LINEAR CLASSIFIERS IN PYTHON

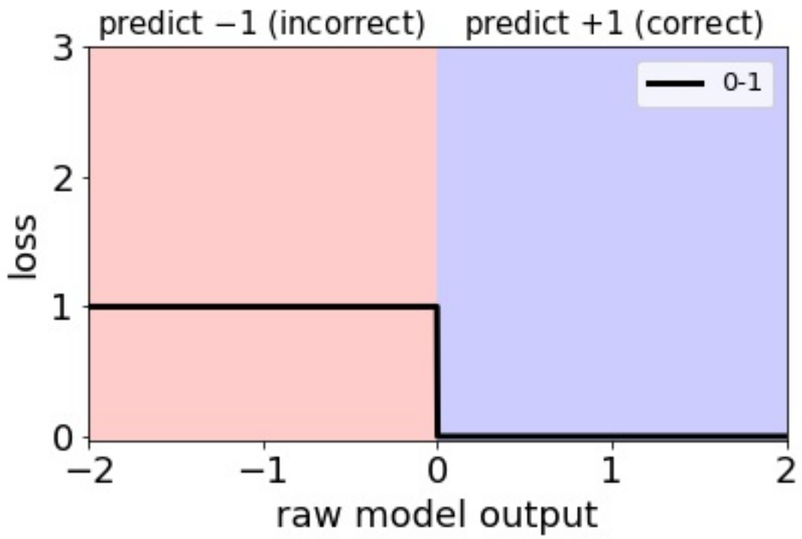
Loss function diagrams

Michael (Mike) Gelbart
Instructor
The University of British Columbia

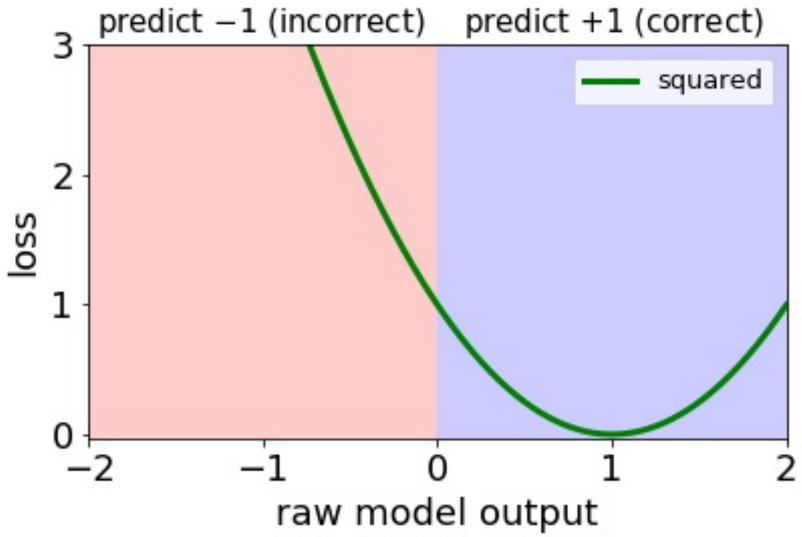
The raw model output



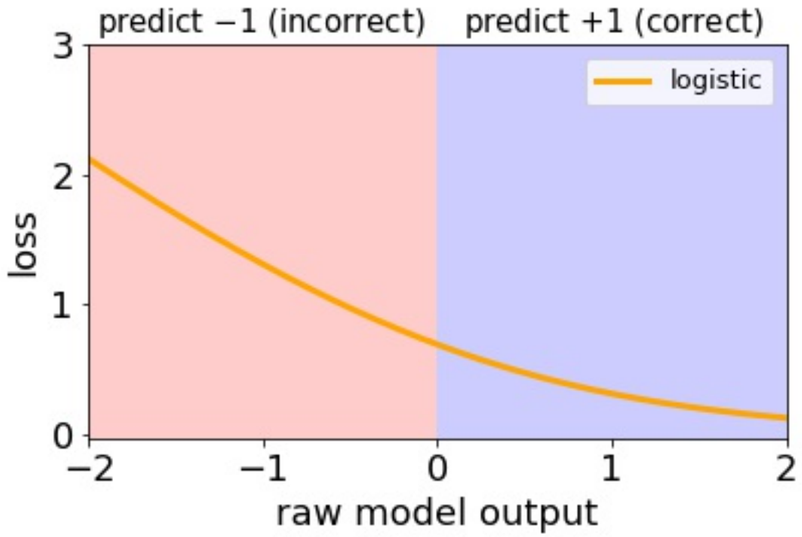
0-1 loss diagram



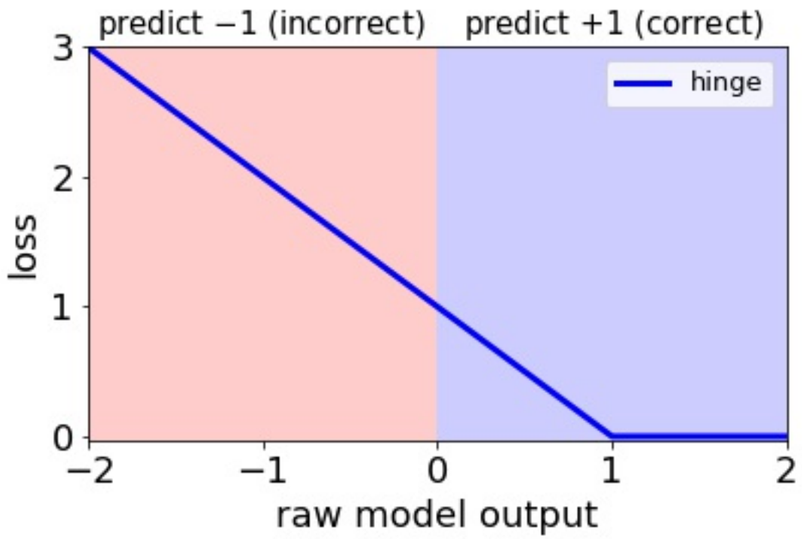
Linear regression loss diagram



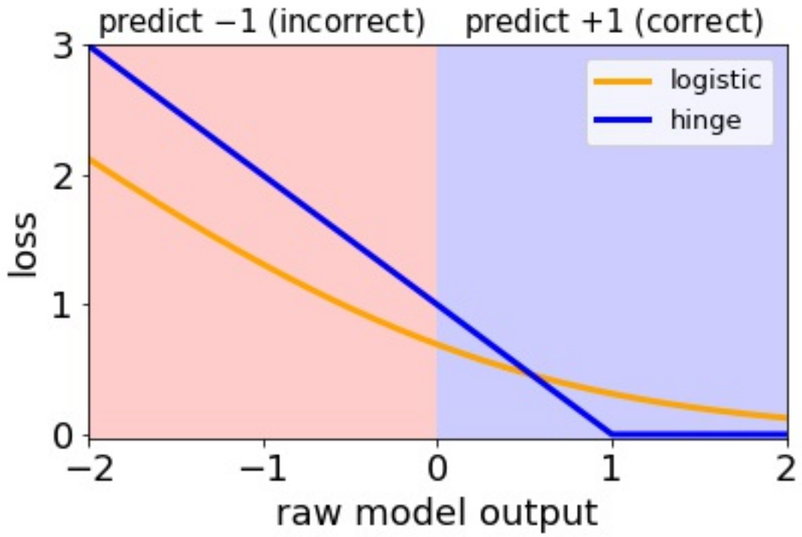
Logistic loss diagram



Hinge loss diagram



Hinge loss diagram



LINEAR CLASSIFIERS IN PYTHON

Let's practice!