- Welcome
- Introduction: Machine Learning concepts
- Module 1. The Predictive Modeling Pipeline
- Module 2. Selecting the best model
- Module 3. Hyperparameter tuning
- **Module 4. Linear Models**

Module overview

Intuitions on linear models
Quiz M4

Non-linear feature engineering for linear models
Quiz M4

Regularization in linear model
Quiz M4

Wrap-up quiz

Quiz M4.03

Note: For each question **make sure you select all of the correct options—** there may be more than one! Don't forget to use the sandbox notebook if you need.

Question 1 (1/1 point)

Which of the following estimators can solve linear regression problems?

- [x] a) `sklearn.linear_model.LinearRegression` ✔
- [ ] b) `sklearn.linear_model.LogisticRegression`
- [x] c) `sklearn.linear_model.Ridge` ✔

✔

Select all answers that apply

EXPLANATION

solution: a) c)

Logistic regression is a classification method even if it contains the words "regression" in its name. Logistic regression predicts the probability that a given sample belongs to a class among a finite set of possible classes.

Ridge regression is a regularized version of traditional linear regression.

You have used 1 of 2 submissions

- a) to create a model robust to outliers (samples that differ widely from other observations)
- b) to reduce overfitting by forcing the weights to stay close to zero ✔
- c) to reduce underfitting by making the problem linearly separable

EXPLANATION

solution: b)

For linear models, regularization refers to controlling to size the weights. This is especially important when the model is fit on a data with either a small number of training samples or a large number of features, especially if some of those features are not very related to the target variable.

Regularization makes the linear models more constraints and can augment underfitting rather than reduce it. If the problem is not linearly separable, one should instead try to engineer new predictive features as we will explain later in this module.

Fitting linear models that are robust to outliers is a complex issue that cannot be solved by simple regularization alone. Please refer to the scikit-learn documentation robust regression if you are interested.

You have used 1 of 1 submissions

Question 3 (1/1 point)

A ridge model is:

- [ ] b) the same as logistic regression with penalized weights
- [x] c) a linear model ✔
- [ ] d) a non linear model

✔

Select all answers that apply

EXPLANATION

Solution: a) c)

You have used 1 of 2 submissions

Question 4 (1 point possible)

Assume that a data scientist has prepared a train/test split and plans to use the test for the final evaluation of a `Ridge` model. The parameter `alpha` of the `Ridge` model:

- [ ] a) is internally tuned when calling `fit` on the train set
- [x] b) should be tuned by running cross-validation on a **train set** ✔
- [x] c) should be tuned by running cross-validation on a **test set**
- [x] d) must be a positive number ✔

✖

Select several answers

Solution: b) d)

a) is wrong: `alpha` is not changed during `fit`, only `coef_` and `intercept_` change when training the model.

c) is wrong: one should never choose any hyper-parameters based on the test set: this will overestimate the generalization performance of the model.

d) the data scientist can either specify a positive `alpha` or use the default value as mentioned in the notebook and in the documentation for `sklearn.linear_model.Ridge`.

You have used 2 of 2 submissions

Question 5

Scaling the data before fitting a model:

- [x] a) is often useful for regularized linear models ✔
- [ ] b) is always necessary for regularized linear models
- [x] c) may speed-up fitting ✔
- [ ] d) has no impact on the optimal choice of the value of a regularization parameter

✔

Select all answers that apply

EXPLANATION

Solution: a) c)

Here are some reasons for scaling features:

- When the original feature values have widely difference natural scales, fitting a linear model on the raw features can cause the

would have comparatively little impact on the features with the largest scales because those features would anyway be assigned comparatively smaller weights when training the model. If many such features are not predictive, the benefit of using regularization would therefore be reduced.
- Scaling is not always necessary, for instance if the features values naturally vary with similar ranges by default.
- Since scaling features has a impact on the relative magnitude of the weights of the trained model, deciding to scale the features or not can change the optimal value of the regularization parameter of a linear model when this parameter is tuned to improve a generalization metric estimated using cross-validation.
- Many models such as logistic regression use numerical solvers (based on gradient descent) to find their optimal parameters. These solvers often converge much faster when the features are scaled.

Note that the value of the regularization parameter can also impact the speed of convergence of gradient descent solvers.

You have used 1 of 2 submissions

Question 6 (1/1 point)

The effect of increasing the regularization strength in a ridge model is to:

- [x] a) shrink all weights towards zero ✔
- [ ] b) make all weights equal
- [ ] c) set a subset of the weights to exactly zero
- [ ] d) constrain all the weights to be positive

✔

Select all answers that apply

Solution: a)

You have used 1 of 2 submissions

Question 7 (1/1 point)

By default, a `LogisticRegression` in scikit-learn applies:

- a) no penalty
- b) a penalty that shrinks the magnitude of the weights towards zero (also called "l2 penalty") ✔
- c) a penalty that ensures all weights are equal

EXPLANATION

Solution: b)

The LogisticRegression documentation, in the "Parameters" description says:

```
penalty : {‘l1’, ‘l2’, ‘elasticnet’, ‘none’}, default=’l2’
```

The term "l2 penalty" refers to a type of regularization that also includes the type of regularization used in `Ridge` regression.

Also note that the default value of the parameter `C` of `LogisticRegression` is 1.0. Larger values of `C` would decrease the strength of the regularization.

You have used 1 of 1 submissions

Question 8 (1/1 point)

a) similar to the parameter `alpha` in a ridge regressor

b) similar to `1 / alpha` where `alpha` is the parameter of a ridge regressor ✔

c) not controlling the regularization

EXPLANATION

Solution: b)

The LogisticRegression documentation says:

```
C : Inverse of regularization strength; smaller values specify
stronger regularization.
```

The Ridge documentation says:

```
alpha : Regularization strength; Larger values specify
stronger regularization.
```

You have used 1 of 1 submissions

Question 9 (1/1 point)

In logistic regression, increasing the regularization strength (by decreasing the value of `C`) makes the model:

a) more likely to overfit to the training data

b) more confident: the values returned by $predict_proba$ are closer to 0 or 1

c) less complex, potentially underfitting the training data ✔

Solution: c)

Increasing the regularization strength adds a stronger penalty on the model's coefficients. This can restrict the model's complexity, leading it to potentially underfit the training data. On the flip side, too little regularization might make the model too flexible and risk overfitting to the training data. Thus, there's a tradeoff to be balanced when selecting the right regularization strength.

Furthermore, increasing regularization tends to make the probabilistic predictions more conservative (closer to the average proportions between classes in the training). We can interpret this as the model being less confident in its predictions.

You have used 1 of 1 submissions

YOUR EXPERIENCE

According to you, this whole 'Regularization in linear model' lesson was:

- ○ **Too easy, I got bored**
- ○ **Adapted to my skills**
- ○ **Difficult but I was able to follow**
- ○ **Too difficult**

Submit

To follow this lesson, I spent:

- ○ **less than 30 minutes**
- ○ **30 min to 1 hour**
- ○ **1 to 2 hours**
- ○ **2 to 4 hours**
- ○ **more than 4 hours**

Submit

FORUM (EXTERNAL RESOURCE)

Question 4

roy-aad 1m

Shouldn't the answer for question 4 be b,c and d?
How can we determine if the model is overfitting if we do not compare the train results to the test result? This is what is done in the ipynb file and also mentioned with the following statements:

- "As mentioned, the regularization parameter needs to be tuned on each dataset."
- "Model hyperparameter tuning should be done with care. Indeed, we want to find an optimal parameter that maximizes some metrics. Thus, it requires both a training set and testing set."