

REPORT- ML CRASH COURSE

1. FRAMING:

In this section, we learnt the common Machine Learning terminology that will be frequently used henceforth.

Suppose you want to develop a supervised machine learning model to predict whether a given email is "spam" or "not spam." Which of the following statements are true?

We'll use unlabeled examples to train the model. ✓

The labels applied to some examples might be unreliable. ✓

Definitely. It's important to check how reliable your data is. The labels for this dataset probably come from email users who mark particular email messages as spam. Since most users do not mark every suspicious email message as spam, we may have trouble knowing whether an email is spam. Furthermore, spammers could intentionally poison our model by providing faulty labels.

2 of 2 correct answers.

Emails not marked as "spam" or "not spam" are unlabeled examples. ✓

Because our label consists of the values "spam" and "not spam", any email not yet marked as spam or not spam is an unlabeled example.

1 of 2 correct answers.

Suppose you're working on a recommender system. You've trained a supervised ML model that will provide personalized shoe recommendations to users. That is, the model will recommend certain pairs of shoes to Marty and different pairs of shoes to Janet. The system will use past user behavior data to generate training data. Which of the following statements are true?

"Shoe beauty" is a useful feature. ✓

"The user clicked on the shoe's description" is a useful label. ✓

Users probably only want to read more about those shoes that they like. Clicks by users is, therefore, an observable, quantifiable metric that could serve as a good training label. Since our training data derives from past user behavior, our labels need to derive from objective behaviors like clicks that strongly correlate with user preferences.

2 of 2 correct answers.

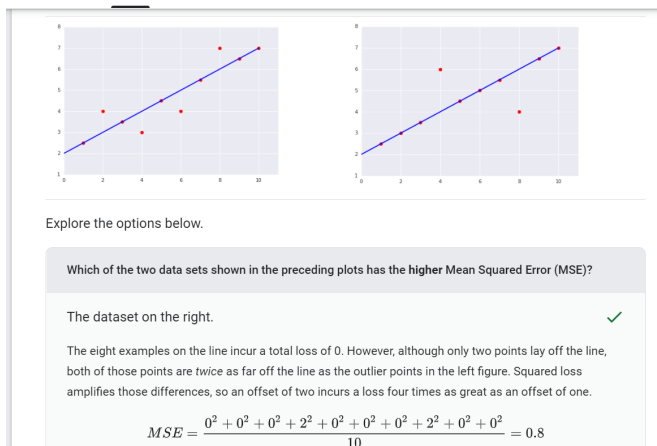
"Shoe size" is a useful feature. ✓

"Shoe size" is a quantifiable signal that likely has a strong impact on whether the user will like the recommended shoes. For example, if Marty wears size 9, the model shouldn't recommend size 7 shoes.

1 of 2 correct answers.

2.DESCENDING INTO ML:

Here, we brush upon the basic concept of Linear regression. We also learn about a few loss functions like L2(squared) loss and mean squared error(MSE). Minimising loss functions makes the model more accurate while training.



3.REDUCING LOSS:

We randomly guess the bias and weights of the linear equation and compute loss function. Then we accordingly change them(using gradient descent) until we find the optimised parameters having minimum loss value. Then the model is said to be converged. We use learning rate as a hyperparameter to go to the next point. Finding a suitable learning rate is very important. There are two types of gradient descent- Stochastic and batch. stochastic gradient deals with only one example. But a batch considers a set of examples.

Playground Exercise

- Check Your Understanding
- First Steps with TF (65 min)
- Generalization (15 min)
- Training and Test Sets (25 min)
- Validation Set (35 min)
- Representation (35 min)
- Feature Crosses (70 min)
- Regularization: Simplicity (40 min)
- Logistic Regression (20 min)
- Classification (90 min)
- Regularization: Sparsity (20 min)
- Neural Networks (65 min)
- Training Neural Nets (10 min)
- Multi-Class Neural Nets (45 min)
- Embeddings (50 min)

ML Engineering

- Production ML Systems (3 min)
- Static vs. Dynamic Training (7 min)
- Static vs. Dynamic Inference (7 min)

Check Your Understanding: Batch Size

Explore the options below.

When performing gradient descent on a large data set, which of the following batch sizes will likely be more efficient?

A small batch or even a batch of one example (SGD). ✓

Amazingly enough, performing gradient descent on a small batch or even a batch of one example is usually more efficient than the full batch. After all, finding the gradient of one example is far cheaper than finding the gradient of millions of examples. To ensure a good representative sample, the algorithm scoops up another random small batch (or batch of one) on every iteration.

Correct answer.

The full batch. ▼

4. FIRST STEPS WITH TF:

This section deals with the introduction to TensorFlow- open source code library to develop ML models. The term epoch means the no. of times the model runs through the dataset. There are 3 hyperparameters we need to tune to create a model that converges efficiently. They are learning rate, epochs, batch size. There are no hard rules to tune these parameters. It purely depends on the dataset. We make use of correlation matrix to understand which feature is best for prediction of the label.

5. GENERALISATION:

We learnt about overfitting models. An overfitting model works very well with the training data with very little loss, but fails to adapt properly to the new set of data given for prediction. The key is to make the model quite simple and as generalised as possible without hardcoding around the peculiarities in the training set. A way is to divide the dataset into 2 subsets- training set and test set. But we need to make sure the subsets have randomly chosen examples and that the test set is large.

6. TRAINING AND TEST DATASETS:

We need to slice the dataset into training and test dataset. We need to make sure that the test data is representative of the whole set. Also we must confirm that we do not train our model on the test dataset itself.

7. VALIDATION SET:

Repetitive training, testing and tweaking the model on the dataset will lead to overfitting of exceptions in the dataset. To avoid that, we can introduce another subset- Validation set. Testing the model after making any changes to the hyperparameters can be done on validation set, leaving the test data untouched. Once the validation loss almost equals testing loss, we can say that we have avoided overfitting.

Crash Course Problem Framing Data Prep Clustering Recommendation Testing and Debugging GANs

Training and Test Sets (25 min)
 Validation Set (35 min)
 Check Your Intuition
 Video Lecture
 Another Partition
 Programming Exercise
 Representation (35 min)
 Feature Crosses (70 min)
 Regularization: Simplicity (40 min)
 Logistic Regression (20 min)
 Classification (90 min)
 Regularization: Sparsity (20 min)
 Neural Networks (65 min)
 Training Neural Nets (10 min)
 Multi-Class Neural Nets (45 min)
 Embeddings (50 min)

ML Engineering
 Production ML Systems (3 min)

Explore the options below.

We looked at a process of using a test set and a training set to drive iterations of model development. On each iteration, we'd train on the training data and evaluate on the test data, using the evaluation results on test data to guide choices of and changes to various model hyperparameters like learning rate and features. Is there anything wrong with this approach? (Pick only one answer.)

This is computationally inefficient. We should just pick a default set of hyperparameters and live with them to save resources. ✓

Doing many rounds of this procedure might cause us to implicitly fit to the peculiarities of our specific test set. ✓

Yes indeed! The more often we evaluate on a given test set, the more we are at risk for implicitly overfitting to that one test set. We'll look at a better protocol next.
 Correct answer.

Totally fine, we're training on training data and evaluating on separate, held-out test data. ✓

8. REPRESENTATION:

This section deals with Feature engineering. Extracting useful features from the dataset and modifying them into a form which we can use for training the model is very important. It takes up most of the time. Mapping numerical features is simple. We can simply scale it down or up according to our convenience. But categorical variables need some encoding to be done. Common practices are One Hot Encoding and Multi Hot Encoding. It creates binary feature vectors. If there are many categories then sparse representation is used. We can use binning for features that don't linearly depend on the label.

9. FEATURES CROSSES:

When we make new synthetic features by crossing 2 features, it is known as feature crosses. They help in bringing non linearity in linear models. Often we cross one hot encoded features.

Crash Course Problem Framing Data Prep Clustering Recommendation Testing and Debugging GANs

Training and Test Sets (25 min)
 Validation Set (35 min)
 Representation (35 min)
 Feature Crosses (70 min)
 Video Lecture
 Encoding Nonlinearity
 Crossing One-Hot Vectors
 Playground Exercises
 Programming Exercise
 Check Your Understanding
 Regularization: Simplicity (40 min)
 Logistic Regression (20 min)
 Classification (90 min)
 Regularization: Sparsity (20 min)
 Neural Networks (65 min)
 Training Neural Nets (10 min)
 Multi-Class Neural Nets (45 min)
 Embeddings (50 min)

predict housing prices. Which of the following sets of features or feature crosses could learn city-specific relationships between roomsPerPerson and housing price?

Three separate binned features: [binned latitude], [binned longitude], [binned roomsPerPerson] ✓

One feature cross: [binned latitude X binned longitude X binned roomsPerPerson] ✓

Crossing binned latitude with binned longitude enables the model to learn city-specific effects of roomsPerPerson. Binning prevents a change in latitude producing the same result as a change in longitude. Depending on the granularity of the bins, this feature cross could learn city-specific or neighborhood-specific or even block-specific effects.
 Correct answer.

One feature cross: [latitude X longitude X roomsPerPerson] ✓

Two feature crosses: [binned latitude X binned roomsPerPerson] and [binned longitude X binned roomsPerPerson] ✓

10. REGULARISATION FOR SIMPLICITY:

We do regularization to deal with overfitting. Instead of minimising only the loss, we minimize loss and complexity of the model. This is known as structural risk minimization. Regularization term measures the model complexity. In L2 regularization approach, we measure complexity as sum of squares of the weights. It is added with cost function. We use a hyperparameter lambda to tune the complexity of the model. Lambda must not be higher else it will lead to underfitting and improper predictions. Note that when we add regularization terms, training loss will increase but test loss will drop because we added new term. And also the weights decrease and come closer to zero. It tries to decrease the large weights.

- Representation (35 min)
- Feature Crosses (70 min)
- Regularization: Simplicity (40 min)
 - Playground Exercise: Overcrossing?
 - Video Lecture
 - L2 Regularization
 - Lambda
 - Playground Exercise: L2 Regularization
 - Check Your Understanding**
- Logistic Regression (20 min)
- Classification (90 min)
- Regularization: Sparsity (20 min)
- Neural Networks (65 min)
- Training Neural Nets (10 min)
- Multi-Class Neural Nets (45 min)
- Embeddings (50 min)

ML Engineering

- Production ML Systems (3 min)
- Static vs. Dynamic Training (7 min)

Imagine a linear model with 100 input features:

- 10 are highly informative.
- 90 are non-informative.

Assume that all features have values between -1 and 1. Which of the following statements are true?

L₂ regularization will encourage many of the non-informative weights to be nearly (but not exactly) 0.0. ✓

Yes, L₂ regularization encourages weights to be near 0.0, but not exactly 0.0.

1 of 2 correct answers.

L₂ regularization may cause the model to learn a moderate weight for some **non-informative** features. ✓

Surprisingly, this can happen when a non-informative feature happens to be correlated with the label. In this case, the model incorrectly gives such non-informative features some of the "credit" that should have gone to informative features.

2 of 2 correct answers.

L₂ regularization will encourage most of the non-informative weights to be exactly 0.0. ✓

- Representation (35 min)
- Feature Crosses (70 min)
- Regularization: Simplicity (40 min)
 - Playground Exercise: Overcrossing?
 - Video Lecture
 - L2 Regularization
 - Lambda
 - Playground Exercise: L2 Regularization
 - Check Your Understanding**
- Logistic Regression (20 min)
- Classification (90 min)
- Regularization: Sparsity (20 min)
- Neural Networks (65 min)
- Training Neural Nets (10 min)
- Multi-Class Neural Nets (45 min)
- Embeddings (50 min)

ML Engineering

- Production ML Systems (3 min)
- Static vs. Dynamic Training (7 min)

L₂ Regularization and Correlated Features

Explore the options below.

Imagine a linear model with two strongly correlated features; that is, these two features are nearly identical copies of one another but one feature contains a small amount of random noise. If we train this model with L₂ regularization, what will happen to the weights for these two features?

One feature will have a large weight; the other will have a weight of **exactly** 0.0. ✓

Both features will have roughly equal, moderate weights. ✓

L₂ regularization will force the features towards roughly equivalent weights that are approximately half of what they would have been had only one of the two features been in the model.

Correct answer.

One feature will have a large weight; the other will have a weight of **almost** 0.0. ✓

11. LOGISTIC REGRESSION:

Logistic regression is used for estimating probabilities. To ensure that our output is always between 0 and 1 we use sigmoid function. The linear equation having the learned weights and bias is also known as log-odds. The cost function here is called log loss. Regularization becomes very important when it comes to logistic. If no regularisation, then it'll become fully overfit. The model will try to reduce the loss zero (which is not possible in sigmoid), shooting the weights to infinity.

12. CLASSIFICATION:

Since the output is probabilistic, we need to have a threshold value to classify the data. In a confusion matrix with all possibilities, A **true positive** is an outcome where the model *correctly* predicts the positive class. Similarly, a **true negative** is an outcome where the model *correctly* predicts the negative class. A **false positive** is an outcome where the model *incorrectly* predicts the positive class. And a **false negative** is an outcome where the model *incorrectly* predicts the negative class. Accuracy is a metric to evaluate our classification model. But it does a poor job when it comes to class imbalance dataset. We have, in addition, Precision and Recall. They should be high for a good ML model.

Machine Learning Crash Course

Feature Courses (Primary)

Regularization: Simplicity (40 min)

Logistic Regression (20 min)

Classification (90 min)

Video Lecture

Thresholding

True vs. False; Positive vs. Negative

Accuracy

Precision and Recall

Check Your Understanding: Accuracy, Precision, Recall

ROC Curve and AUC

Check Your Understanding: ROC and AUC

Prediction Bias

Programming Exercise

Regularization: Sparsity (20 min)

Neural Networks (65 min)

Training Neural Nets (10 min)

Multi-Class Neural Nets (45 min)

Embeddings (50 min)

Courses

Practica

Guides

Glossary

Search

English

Precision

Explore the options below.

Consider a classification model that separates email into two categories: "spam" or "not spam." If you raise the classification threshold, what will happen to precision?

Definitely increase.

Probably decrease.

Definitely decrease.

Probably increase.

In general, raising the classification threshold reduces false positives, thus raising precision.

Correct answer.

Crash Course

Problem Framing

Data Prep

Clustering

Recommendation

Testing and Debugging

GANs

Feature Courses (Primary)

Regularization: Simplicity (40 min)

Logistic Regression (20 min)

Classification (90 min)

Video Lecture

Thresholding

True vs. False; Positive vs. Negative

Accuracy

Precision and Recall

Check Your Understanding: Accuracy, Precision, Recall

ROC Curve and AUC

Check Your Understanding: ROC and AUC

Prediction Bias

Programming Exercise

Regularization: Sparsity (20 min)

Neural Networks (65 min)

Training Neural Nets (10 min)

Multi-Class Neural Nets (45 min)

Embeddings (50 min)

Accuracy

Explore the options below.

In which of the following scenarios would a high accuracy value suggest that the ML model is doing a good job?

In the game of roulette, a ball is dropped on a spinning wheel and eventually lands in one of 38 slots. Using visual features (the spin of the ball, the position of the wheel when the ball was dropped, the height of the ball over the wheel), an ML model can predict the slot that the ball will land in with an accuracy of 4%.

This ML model is making predictions far better than chance; a random guess would be correct 1/38 of the time—yielding an accuracy of 2.6%. Although the model's accuracy is "only" 4%, the benefits of success far outweigh the disadvantages of failure.

Correct answer.

An expensive robotic chicken crosses a very busy road a thousand times per day. An ML model evaluates traffic patterns and predicts when this chicken can safely cross the street with an accuracy of 99.99%.

A deadly but curable medical condition afflicts 01% of the population. An ML model uses symptoms as

Crash Course

Problem Framing

Data Prep

Clustering

Recommendation

Testing and Debugging

GANs

Feature Courses (Primary)

Regularization: Simplicity (40 min)

Logistic Regression (20 min)

Classification (90 min)

Video Lecture

Thresholding

True vs. False; Positive vs. Negative

Accuracy

Precision and Recall

Check Your Understanding: Accuracy, Precision, Recall

ROC Curve and AUC

Check Your Understanding: ROC and AUC

Prediction Bias

Programming Exercise

Regularization: Sparsity (20 min)

Neural Networks (65 min)

Training Neural Nets (10 min)

Multi-Class Neural Nets (45 min)

Embeddings (50 min)

Recall

Explore the options below.

Consider a classification model that separates email into two categories: "spam" or "not spam." If you raise the classification threshold, what will happen to recall?

Always stay constant.

Always decrease or stay the same.

Raising our classification threshold will cause the number of true positives to decrease or stay the same and will cause the number of false negatives to increase or stay the same. Thus, recall will either stay constant or decrease.

Correct answer.

Always increase.

Crash Course

Problem Framing

Data Prep

Clustering

Recommendation

Testing and Debugging

GANs

Feature Courses (Primary)

Regularization: Simplicity (40 min)

Logistic Regression (20 min)

Classification (90 min)

Video Lecture

Thresholding

True vs. False; Positive vs. Negative

Accuracy

Precision and Recall

Check Your Understanding: Accuracy, Precision, Recall

ROC Curve and AUC

Check Your Understanding: ROC and AUC

Prediction Bias

Programming Exercise

Regularization: Sparsity (20 min)

Neural Networks (65 min)

Training Neural Nets (10 min)

Multi-Class Neural Nets (45 min)

Embeddings (50 min)

Precision and Recall

Explore the options below.

Consider two models—A and B—that each evaluate the same dataset. Which one of the following statements is true?

If model A has better precision and better recall than model B, then model A is probably better.

In general, a model that outperforms another model on both precision and recall is likely the better model. Obviously, we'll need to make sure that comparison is being done at a precision / recall point that is useful in practice for this to be meaningful. For example, suppose our spam detection model needs to have at least 90% precision to be useful and avoid unnecessary false alarms. In this case, comparing one model at (20% precision, 99% recall) to another at (15% precision, 98% recall) is not particularly instructive, as neither model meets the 90% precision requirement. But with that caveat in mind, this is a good way to think about comparing models when using precision and recall.

Correct answer.

If model A has better recall than model B, then model A is better.