BREAK THROUGH TECH

Machine Learning Foundations

Lab 5

Week of June 23

Icebreaker: Student Feedback regarding Course

Icebreaker: Student Feedback

Objectives:

- Discuss what's going well in the Labs and course
- Discuss what can be improved in the Labs and course
- Propose solutions for addressing concerns/issues

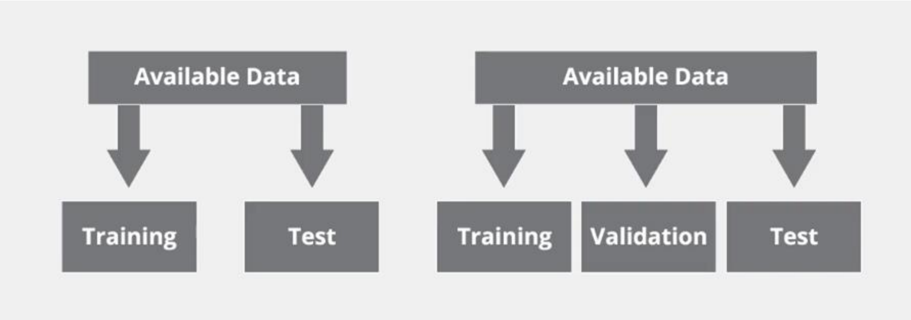
Week 5 Concept Overview + Q&A

Model Selection: Choosing an Optimal Model

- ❑ Out-of-Sample Validation
- ❑ Choose hyperparameters: GridSearchCV
- ❑ Feature Selection
- ❑ Evaluate using quantitative metrics

Out-of-Sample Validation

- Perform out-of-sample validation

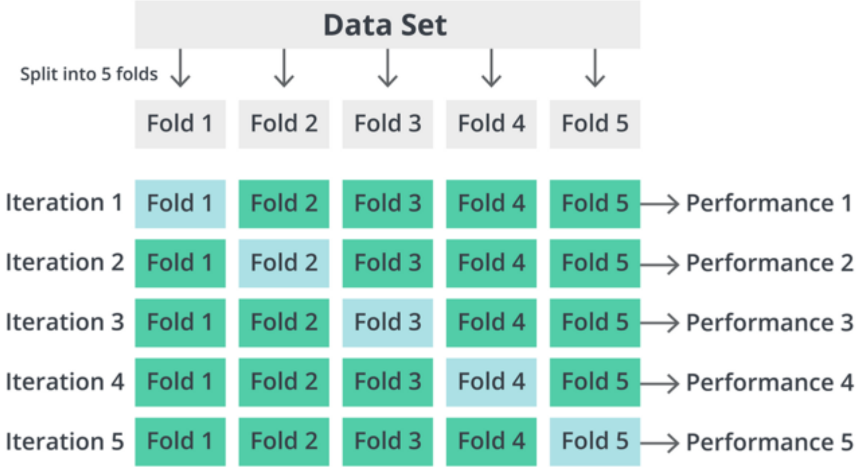


- **Training set:** used to actually fit the model to the data
- **Validation set:** used to evaluate model candidates for model selection
- **Test set:** used for estimating the generalization performance of the best selected model

Splitting for out-of-sample Validation

Typical

Not enough data: k-fold cross-validation



Choose Hyperparameters

- **Grid search:** goes through all combinations systematically
- **Random search:** goes through combinations randomly

Feature Selection

Why do feature selection?

- Less overfitting
- Better interpretability
- Better scalability
- Lower maintenance costs

Feature selection methods:

- **Heuristic selection:** filter out features using heuristic rules prior to modeling
- **Stepwise selection:** iteratively add/reduce features based on empirical model performance
- **Regularization:** include penalties for feature count in the algorithm’s loss function

Feature Selection:
Heuristic Feature Selection

Heuristic rules:

- Feature has a minimum level of correlation or mutual information with the label
- Feature has sufficient support (i.e., % examples where feature is not 0/NULL)
- Domain specific rules (i.e., feature is too expensive to operate, feature not allowed to regulations)

Feature Selection

Why do feature selection?

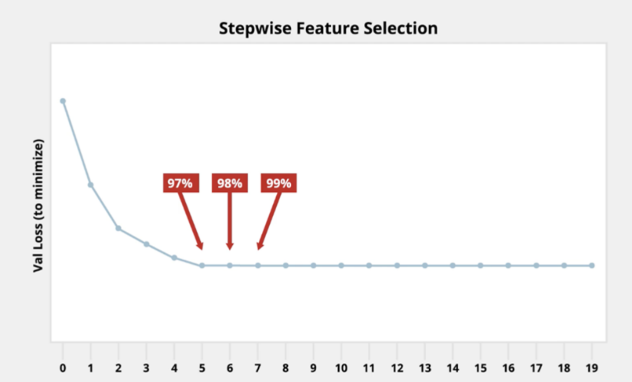
- Less overfitting
- Better interpretability
- Better scalability
- Lower maintenance costs

Feature selection methods:

- **Heuristic selection:** filter out features using heuristic rules prior to modeling
- **Stepwise selection:** iteratively add/reduce features based on empirical model performance
- **Regularization:** include penalties for feature count in the algorithm’s loss function

Feature Selection:
Stepwise Feature Selection Algorithm

1. Initialize: best_subset = {}
2. Initialize: candiate_features = all features
3. For each feature in candidate_features:
    a. Get: (cross) validation score with model built with: best_subset + feature
    b. Add to list: (feature, (cross) validated score)
4. Choose: best_feature = feature from step (3) with best performance
5. Update: best_subset = best_subset + best_feature
6. Remove: best_feature from candidate_features
7. Repeat: steps 3 - 6 until stopping criteria is met



Feature Selection: Regularization

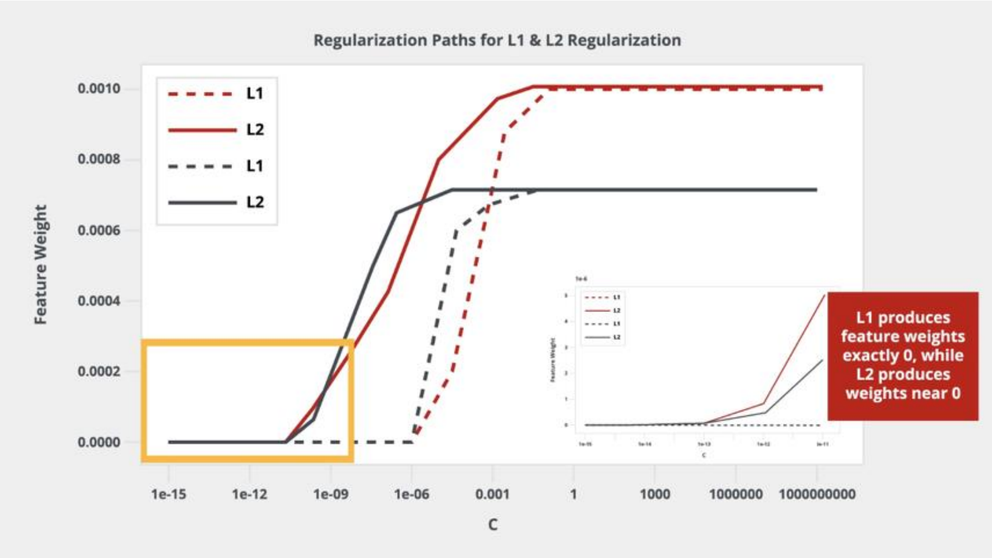
Implicit Feature Selection:
reducing feature count as a byproduct of the model training procedure

Regularization Loss = **Loss** + (1/**C**)*Penalty

- Loss can be any common loss function, such as log-loss or MSE
- C controls the weight of the penalty. Higher C = less regularization

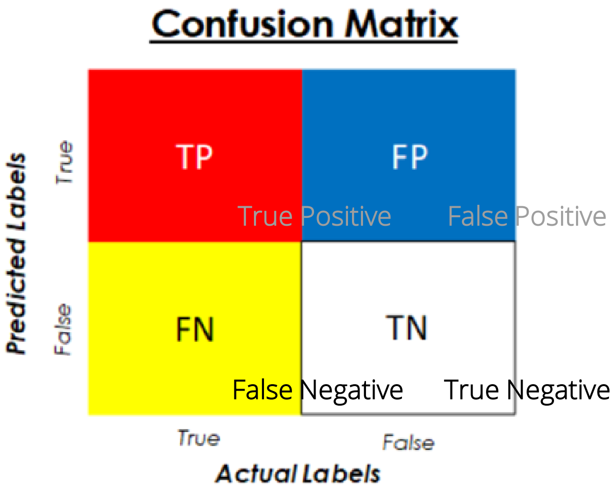
$$L1 - Penalty = \frac{1}{C}\sum_{j=0}^{m}|w_j|$$

a.k.a., the "Lasso"



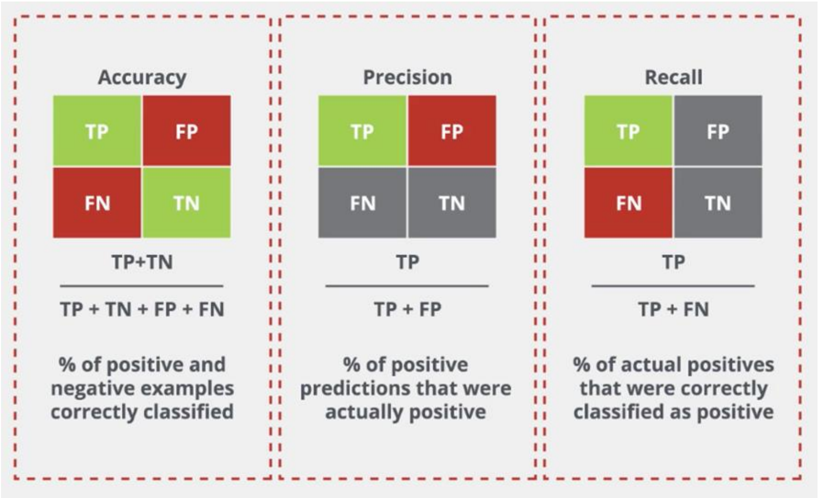
Run hyperparameter selection to identify best C, then remove features with weight close to 0.

Evaluate Using Quantitative Metrics



Practical Tips: When to use each

- **Accuracy:** most commonly used in multi-class problems
- **Precision:** favored in binary classification when false positives are much worse than false negatives
- **Recall:** favored in binary classification when false negatives are much worse than false positives



from stackoverflow.com

Evaluate Using Quantitative Metrics

Confusion matrix: can be used for multi-class classification

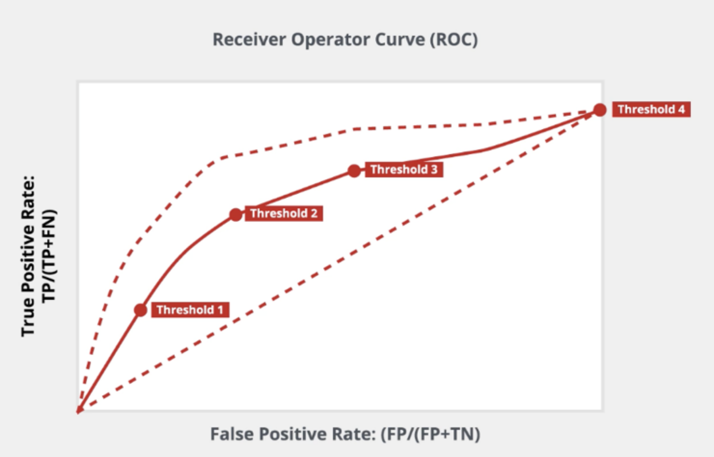
Example: Iris data set

		Predicted Values		
		Setosa	Versicolor	Virginica
Actual Values	Setosa	16 (cell 1)	0 (cell 2)	0 (cell 3)
	Versicolor	0 (cell 4)	17 (cell 5)	1 (cell 6)
	Virginica	0 (cell 7)	0 (cell 8)	11 (cell 9)

Bharathi, "Latest Guide on Confusion Matrix for Multi-Class Classification" *Analytics Vidhya*, 2013. https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/

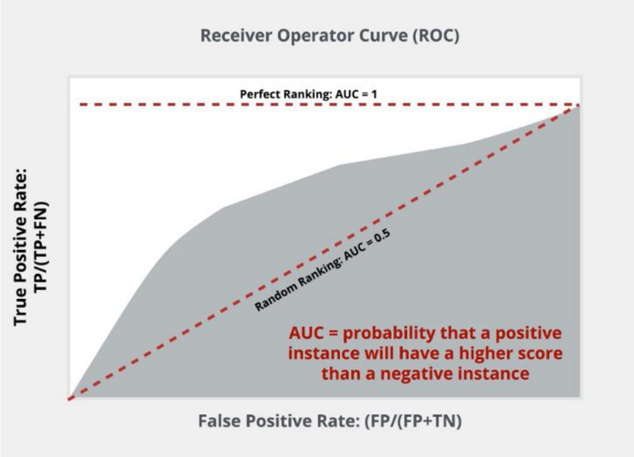
Use Recall, Precision to Choose Thresholds

ROC (receiver operating characteristic) Curve:
visualizes performance of binary classifier



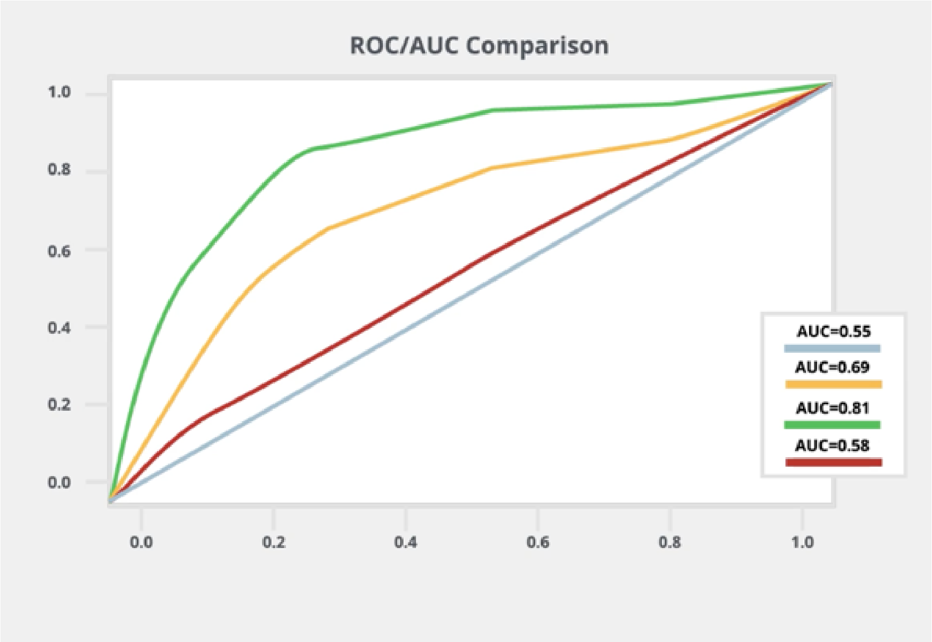
AUC (area under curve):

classifier performance for different thresholds



Use Recall, Precision to Choose Thresholds

Using ROC and AUC to compare different models



Some Sklearn evaluation metrics online resources

More details on the confusion matrix:

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

ROC curve:

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html

AUC curve:

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html

REMINDER: you use actual predictions in the confusion matrix, and you use prediction probabilities for auc and roc (where you plot with true positive vs false positive rates

Big Picture Questions

Ponder the following questions within your breakout groups:

- Why do we place significant importance on addressing overfitting and achieving generalization in machine learning?
- Can you identify specific applications where minimizing false negatives is more crucial than minimizing false positives? Additionally, what evaluation metric would be appropriate in such situations?
- What are some reasons why we might consider performing feature selection in machine learning? Furthermore, how can we determine the optimal number of features to include in a model?
- What is cross-validation and what is the benefit of performing cross-validation?

Breakout Groups: Lab Assignment

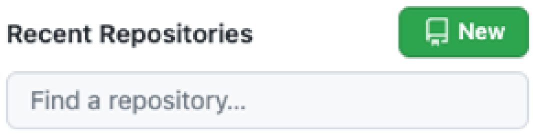
Set Up Your Repository on GitHub

Before you start this week's lab, let's create a new repository on GitHub.

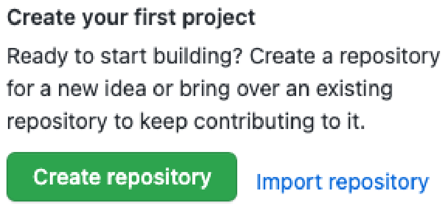
1. If you already have a personal GitHub account, sign in to github.com.

   a. If needed, you can sign up for a free personal GitHub account at github.com. Click Sign Up

1. On the GitHub home page, click New.



1. Or click Create Repository if you're creating your first repository with a new GitHub account.



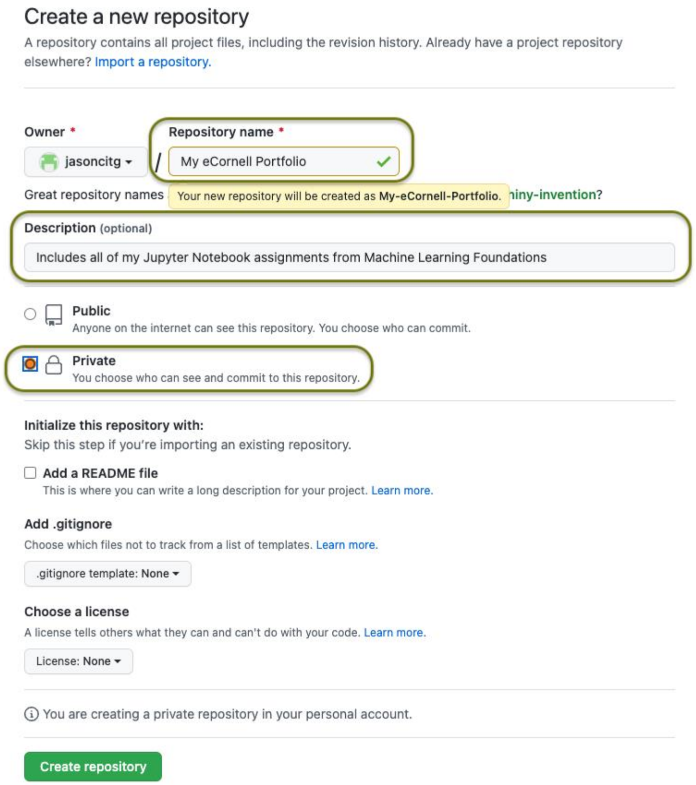
Set Up Your Repository on GitHub

4. Enter a name for the new repository in the Repository name field. (ex. My Cornell Portfolio)

   Optional:

   - Enter a description for the portfolio in the description field.
   - Check the Add a README file box if you wish to write a longer description or include file notes.

4. Click Create repository.



Practice Using Git

The following common commands will help you get started in Git:

git init	Turns a directory into an empty Git repository.
git add	Adds files into a staging area for Git.
git commit	Record the changes made to a file to a local repository.
git status	Returns the current state of the selected repository.
git config	Allows the user to assign settings and configurations.
git branch	Determine what branch the local repository is on, add a new branch, or delete a branch.
get checkout	Switch branches
git merge	Integrate branches
git remote	Connect a local repository with a remote repository.
git clone	Create a local working copy of an existing remote repository
git pull	Get the latest version of a repository.
git push	Sends local commits to the remote repository.
git stash	Save changes made when they’re not in a state to commit them to a repository.
git log	Show the chronological commit history for a repository.

Lab 5

In this lab, you will:

- Build your DataFrame and define your ML problem
- Create labeled examples from the data set, and split the data into training and test data sets
- Train, test and evaluate a logistic regression model using scikit-learn's default hyperparameter value for C.
- Find the optimal logistic regression model using GridSearchCV.
- Train, test and evaluate the optimal logistic regression model.
- Plot the precision-recall curve and the ROC, then compute the AUC for both models.
- Practice the SelectKBest feature selection method.
- Save your best performing model to a PKL file, and add the model and dataset to your GitHub repository.

Lab 5

Lab 5: ML Life Cycle: Evaluation and Deployment

In [ ]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, precision_recall_curve
```

In this lab, you will continue practicing the evaluation phase of the machine learning life cycle. You will perform model selection for logistic regression to solve a classification problem. You will complete the following tasks:

1. Build your DataFrame and define your ML problem:
    - Load the Airbnb "listings" data set
    - Define the label - what are you are predicting?
    - Identify the features
2. Create labeled examples from the data set
3. Split the data into training and test data sets
4. Train, test and evaluate a logistic regression (LR) model using the scikit-learn default value for hyperparameter C.
5. Perform a grid search to identify the optimal value of C for a logistic regression model.
6. Train, test and evaluate a logisitic regression model using the optimal value of C.
7. Plot a precision-recall curve for both models.
8. Plot the ROC and compute the AUC for both models.
9. Perform feature selection.
10. Make your model persistent for future use.

Note: Some of the code cells in this notebook may take a while to run.

Next week

In the following week, you will:

- Improve model performance with ensemble methods
- Understand the mechanics of three ensemble methods: stacking, random forests and gradient boosted decision trees
- Explore unsupervised learning
- Implement unsupervised clustering

And in the lab, you will:

- Build your DataFrame and define your ML problem
- Create labeled examples from the data set, and split the data into training and test data sets
- Train, test, and evaluate two individual regressors and three ensemble regressors to solve your ML problem
- Visualize and compare the performance of the individual models and the ensemble models

Content + Lab Feedback Survey

Content + Lab Feedback Survey

To complete your lab, please answer the following questions about BOTH your online modules and your lab experience. Your input will help pay it forward to the Break Through Tech student community by enabling us to continuously improve the learning experience that we provide to our community.

Thank you for your thoughtful feedback!

https://forms.gle/eUQQZgS6BPRpqgZ7A