**ML Models**

Supervised Learning

* Regression models:
  + Linear regression
    - Assumes independence, sensible to outliers, quick and easy
    - Tells us how linear the data is
  + KNN
    - Take one observation $$x\_0$$ or x0 in test set. Find the K neighbors in training set.
    - Estimate $$f(x\_0)$$ by the average of the output of the neighbors.
  + Polynomial regression
    - This is the same as linear reg, but with transformed features.
  + Decision trees
    - Segments feature space into simple regions, where each region represents an outcome (mean/mode).
    - Start at the top and make splitting rules based on single features taking a top-down greedy approach.
    - At each split, always make the best split - highest drop of loss function
    - Stop when a region has less than $$n$$ observations
    - Very likely to overfit: solve with pruning
    - Better for nonlinear, complex relationships
    - Easily handle qualitative predictors
    - Not robust, not generalizable (high variance)
  + bagging
    - Train $$B$$ trees on $$B$$ bootstrapped datasets (from the original one)
    - Average all predictions (summing them decreases variance CLT)
    - Number of trees $$B$$ is not critical, it just needs to be large enough. Large $$B$$ does not lead to overfitting.
    - Less interpretable because we don’t know the importance of each variable anymore. But can check mean decrease of loss caused by each feature!
  + RF’s
    - Same as bagging, but at each split, it only considers $$m=\sqrt p$$ predictors out of $$p$$.
    - This is useful b/c if there is one very strong predictor, it will appear in all trees (correlations). Averaging correlated variables does not decrease variance
    - The more correlated variables there are, the smaller $$m$$ should be
  + Boosting
    - Same as bagging, but now trees are grown sequentially
    - No bootstrapping, instead use a modified version of the training set.
    - Parameters: number of trees (now can overfit), learning rate, number of splits in a tree (often 1)
  + MLPs
    - Easy
* Classification models:
  + Logistic regression
    - Only for binary outcomes
    - Assumes independence of features (no multicollinearity, no high corrs)
    - Needs large $$N$$.
    - Sensible to outliers
    - Quick and easy to implement
    - Best when dataset is linearly separable, and provides +/- relationship  btw each feature and outcome
    - Not so good for well-separated classes. Use LDA or SVM instead.
  + Naive Bayes
    - Assumes features are independent. (Naive). If true, performs better than sophisticated models. Typically they are not independent.
    - Easy to build, useful for big datasets
    - $$P(class|x)=P(x|class)P(class)/P(x)$$, $$x$$ is predictor
    - Better for categorical rather than numerical
    - Zero frequency issue. Solved by smoothing Laplace estimation.
    - Used a lot for NLP (freqs of words).
  + Linear Discriminant Analysis
    - Good when $$n$$ is small and distributions of each predictor is approx normal.
  + KNN
    - Take one observation $$x\_0$$ in test set. Find the K neighbors in training set.
    - Estimate the probability of class $$j$$ by the most frequent class in the neighbors.
  + Decision trees, bagging, RFs, boosting
    - See above.
  + Support Vector Machines
    - Originally intended for binary but multi class works too
    - Support vectors are the point closest to the boundary. Changes in them, change the boundary. Other points do not affect boundary.
    - SV Classifiers (linear): allow a few observations to be on the wrong side of the margins. Observations on the correct side do not affect the hyperplane. SV’s are now the observations lying on the incorrect side of the classifier.
    - SVM (nonlinear): Enlarge feature space of SVC using kernels
    - Linear SVC: $$f(x)=\beta\_0+\sum\_{i=1}^n\alpha\_i\langle x,x\_i\rangle$$, where $$\alpha\_i\neq 0$$ if $$i$$ is a SV.
    - For SVM, instead of inner product, use a kernel $$K(x,x\_i)$$. People use polynomial, radial(exponential) kernels. Radial kernels are local; only nearby training jobs play a role in the classification of a test obs.
    - Due to their similarity, SVM and logistic regression give similar results. Observations on the correct side of decision boundary have loss of zero in SVM. While observations far from decision boundary in LR give very small loss.
    - When classes are well separated, use SVM. For more overlap, use LR.
  + MLPs
    - Easy

Unsupervised Learning

* PCA
  + Eigenvalues of covariance matrix
  + Standardize beforehand
* K-means clustering
  + Want to minimize distance between observations in each cluster.
  + Randomly assign observations to one of 1—K clusters.
  + Compute the cluster centroid (vector of $$p$$ feature means for the observations in the cluster)
  + Assign each observation to the cluster whose centroid is closest.
  + May arrive to local minima, run multiple realizations.
  + Consider standardizing data

Hypothesis testing

1. Formulate a null and alternative $$H$$.
2. Assumptions: Independence, etc?
3. Pick a test and test statistic
4. Select significance level $$\alpha$$. Probability threshold below which the $$H\_0$$ will be rejected.
5. Compute observed test statistic.
6. Compute p-value. Probability of sampling a test statistic at least as extreme as observed one.
7. Reject $$H\_0$$ if p-value is less than $$\alpha$$.

Anova (Analysis of Variance)

* Consider two models $$M\_1$$, $$M\_2$$, where predictors of $$M\_1$$ are a subset of predictors of $$M\_2$$(more complex model).
* ANOVA tests: $$H\_0:$$$$M\_1$$ is a sufficient model to explain data. $$H\_A$$: $$M\_2$$ is required to explain data.

$$R^2 = 1-RSS/TSS$$, residual and total. Measures variances explained.

Sensitivity: % of true fraud identified.

Specificity: % of nonfraud identified.

ROC, we vary the threshold for classification for $$P(Y|X)$$

More topics

***Statistical Learning:***

* ISL
  + Quality of Fit
    - Measure the quality of fit using a metrics such as mean squared error, cross entropy, etc.
  + Bias-Variance Trade off
    - Refers to the level of complexity in a model. Low complexity typically gives a larger bias in its predictions with low variance. High complexity gives low bias (great fir to training), but high variance (poor fit to new data).
  + Classification setting
    - ?
* Linear Regression
  + Coefficient estimation
  + Accuracy of coefficient estimates
  + Accuracy of model
  + Multiple linear regression
  + Potential problems:
    - Nonlinearity of data
    - Correlation of error terms
    - Nonconstant variance of error terms
    - Outliers
    - High leverage points
    - Collinearity
  + Comparing Linear regression to KNN
* Classification
  + Overview
  + Logistic regression
    - Model
    - Coefficient estimation
    - Making predictions
    - Multiple logistic regression
    - Logistic reg for >2 classes
  + Linear discrimnant Analysis (LDA)
    - Bayes Thm for classification
    - LDA for p=1
    - LDA for p>1
    - Quadratic DA
  + Compare LDA, QDA, KNN
* Resampling methods
  + Cross-validation
    - Leave-one-out cross-validation
    - k-Fold CV
    - Bias-Variance Trade-off for k-Fold CV
    - CV on classification
  + Bootstrap
* Linear model selection and regularization
  + Subset selection
    - Best subset selection
    - Stepwise selection
    - Choosing optimal model
  + Shrinkage methods
    - Ridge regression
    - Lasso
    - Selecting tuning parameter
  + Dimension Reduction Models
    - PC regression
    - Partial Least Squares
  + Considerations in high-dim data
* Beyond linearity
  + Polynomial Regression
  + Step Functions
  + Basis functions
  + Regression Splines
  + Smoothing splines
  + Local regression
  + Generalized Additive Models
* Tree-based methods
  + Basics of decision trees
  + Bagging
  + Random Forests
  + Boosting
* Support Vector Machines
  + Maximal Margin Classifier
  + Support Vecotr Classifiers
  + SVM’s
  + SVM’s with more than 2 classes
  + Relationship to logistic regression
* Unsupervised learning
  + Overview
  + PCA
  + Clustering methods
    - K-means clustering
    - Hierarchical clustering
    - Issues in clustering

***Deep Learning:***

* Linear Algebra
  + Singular value decomposition — See the lectures for Pattern Recognition.
* Probability
  + Information theory
  + Structured probabilistic models
* Numerical computation
* Machine Learning basics
  + Learning algorithms
  + Capacity, overfitting, underfitting
  + Hyperparameters and validation sets
  + Estimators, bias and variance
  + MLE
  + Bayesian Stats
  + Supervised Learning algorithms
  + Unsupervised Learning algorithms
  + Stochastic gradient descent
  + Building an ML algorithm
* Deep Feedforward Networks
  + Learning XOR
  + Gradient-based learning
  + Hidden Units
  + Architecture Design
  + Backprop
* Regularization for DL
  + Parameter norm penalties
  + Norm penalties as constrained optimization
  + Regularization and under-constrained problems
  + Dataset Augmentation
  + Noise Robustness
  + Semi-supervised Learning
  + Multitask Learning
  + Early Stopping
  + Parameter tying and sharing
  + Sparse Representations
  + Bagging and Other ensemble methods
  + Dropout
  + Adversarial training
  + Tangent distance, tangent prop and manifold tangent classifier
* Optimization for training Deep Models
  + How learning differs from pure optimization
  + Challenges in NN optimization
  + Basic Algorithms
  + Parameter Initialization strategies
  + Algorithms with adaptive Learning rates
  + Approximate second order models
  + Optimization strategies and meta-algorithms
* Convolutional Networks
  + Conv op
  + Motivation
  + Pooling
  + Conv and pool as an infinitely strong prior
  + Variants of the basic conv function
  + Structrued outputs
  + Data types
  + Efficient Conv algorithms
  + Random or unsupervised features
  + Neuroscientific basis for conv nets
* Sequence Modeling
  + Unfolding computational graphs
  + RNN
  + Bidirectional RNNs
  + Encoder-decoder sequence-to-sequence architectures
  + Deep recurrent nets
  + Recursive NNs
  + Challenge of long-term dependencies
  + Echo State Nets
  + Leaky units and other strategies for multiple timescales
  + Long Short term Memory LSTM
  + Optimization for Long-term dependencies
  + Explicit memory
* Practical Methodology
  + Performance metrics
  + Default baseline models
  + Do we need more data?
  + Selecting hyperparameters
  + Debugging strategies
* Applications
  + Large-scale DL
  + Computer Vision
  + Speech Recognition
  + Natural Language Processing
  + Others
* Linear Factor Models
  + Probabilistic PCA and Factor Analysis
  + ICA
  + Slow Feature Analysis
  + Sparse Coding
  + Manifold interpretation of PCA
* Autoencoders (AE)
  + Undercomplete AEs
  + Regularized AEs
  + Representational power, layer size and depth
  + Stochastic encoders and decoders
  + Denoising AEs
  + Learning manifolds with AE
  + Contractive AEs
  + Predictive Sparse decomposition
  + Apps of AEs
* Representation Learning
* Structured Probabilistic Models for DL
* MonteCarlo Methods
  + Sampling and MCM
  + Importane sampling
  + Markov Chain MCM
  + Gibbs Sampling
* Confronting the partition function
* Approximate Inference
* Deep Generative Models
  + Boltzmann Machines
  + Restricted BMs
  + Deep Belief nets
  + Deep BMs
  + BMs for real-valued data
  + Convolutional BMs
  + BMs for structured or sequential outputs
  + Other BMs
  + Backprop through random operations
  + Directed Generative nets
  + Drawing samples from AEs
  + Generative Stochastic Nets