Lecture 1

* A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. (Tom Mitchell, 1997)
* A hyperparameter is a parameter of a learning algorithm (not the model) which is set prior to training and remains constant during training
* Business Problem 🡪 Goal Definition 🡪 Data Collection and Preparation 🡪 Feature Engineering 🡪 Model Training 🡪 Model Evaluation 🡪 Model Deployment 🡪 Model Serving 🡪 Model Monitoring 🡪 Model Maintenance
* Business Analyst and data analyst 🡪 Data Engineer and Data Labeler 🡪 Data Analyst 🡪DevOps 🡪 Data Analyst and Data LabelerThree factors influence the cost: Difficulty of the problem, cost of the data, and need for accuracy. When to use ML: Too complex for coding, constantly changing, perceptive, unstudied phenomenon, simple objective, cost effective. When NOT to use ML: Need for explainability, errors / failures are too costly, hard to get data, simpler methods work well, can manually create a lookup table.
* Most time is spent cleaning and organizing data
* Precision = TP / TP + FP, Recall / Sensitivity / True Positive Rate = TP / TP + FN, FP is a Type 1 error, FN is a Type 2 error, Increasing precision reduces recall and vice versa, Specificity = TN / TN + FP, False Alarm Rate = FN / FP + TN = 1 – Specificity, Accuracy = TP + TN / TP + TN + FP +FN, F-Measure = (2 x recall x precision) / (recall + precision) = TP / (TP+((FN+FP)/2)), ROC – TPR (Recall) vs FPR (FP / (FP + TN)), AUC > .5 is good.
* Precision recall tradeoff
* ROC – Receiver Operating Characteristic
* Accuracy is not the preferred method for analyzing a classifier. A confusion matrix is better.
* A perfect classifier will have a ROC AUC equal to 1, whereas a purely random classifier will have a ROC AUC equal to 0.5.

Lecture 2

* Linear regression, RMSE, MSE. Training a model means searching for a combination of model parameters that minimizes a cost function (over the training set). Scikit-Learn’s GridSearchCV Use cross-validation to get an estimate of a model’s generalization performance. / If a model performs well on the training data but generalizes poorly according to the cross-validation metrics, then your model is overfitting./ If it performs poorly on both, then it is underfitting. This is one way to tell when a model is too simple. Ridge Regression (also called Tikhonov regularization) is a regularized version of Linear Regression: (L2 regularization) a regularization term equal to half of the square sum of the coefficients is added to the cost function. Note that the regularization term should only be added to the cost function during training. Once the model is trained, you want to use the unregularized performance measure to evaluate the model’s performance. 
* It is recommended to scale the features before using Ridge Regression. Least Absolute Shrinkage and Selection Operator Regression (usually simply called Lasso Regression) is another regularized version of Linear Regression - uses the L1 norm 
* tends to eliminate the weights of the least important features (i.e., set them to zero). automatically performs feature selection and outputs a sparse model (i.e., with few nonzero feature weights).
* With early stopping, you just stop training as soon as the validation error reaches the minimum.
* The Softmax Regression classifier predicts the class with the highest estimated probability The Softmax Regression classifier predicts only one class at a time (i.e., it is multiclass, not multioutput), so it should be used only with mutually exclusive classes
* Bias – how much the model fits the underlying function Variance – how general is the model class
* Underfitting – when the model performs poorly on training data Overfitting – when the model performs well on training data but performs poorly on test data it hasn’t seen before
* There is a trade off between three factors, 1 – Complexity of Hypothesis, 2 – Training Set Size N, 3 – Generalization Error e on new data. As N increase, e decreases. In practice increasing N has an outsize impact. Training set – learn the model parameters, Validation set – select the algorithm, select the hyper parameters, Test Set – Publication of results
* Normalization – scale to 0-1. Could scale for -1 to +1. Standardization – fit to a normal distribution. No bounds.
* Decision trees Don’t require feature scaling or centering, eager learner, batch learning, make axis aligned splits, approximate diagonals with many splits 🡪 overfitting
* For decision trees: Im=-plog2(p)- (1-p)log2(1-p) where p = P(X=x), for K greater than 2, sum each up with negative sign in front, Gini index = 2p(1-p), misclassification error = 1 – max(p, 1-p)
* Decision Trees are prone to overfitting since they make split decisions that are focused on local optima, rather than global optima. Pruning – use Chi Square and if p-value is higher than say 5%, prune the leaf nodes, Decision Trees are sensitive to training set rotation

Lecture 3

K Nearest Neighbor (KNN). Leave One Out, Correct count over total is A, incorrect over total is error, k is too small, high variance, sensitive to each data point changing (overfitting), k is too large, high bias, smooth average of the outcomes of many neighbors (underfitting), When k is small, single instances matter; bias is small, variance is large: High complexity. As k increases, we average over more instances and variance decreases but bias increases: Low complexity. Cross-validation is used to fine tune k. **Manhattan Distance** = Sum from 1 to m of the absolute value of the difference. **Euclidian distance** between (x1,y1) and (x2, y2) is the square root of (x2 – x1)2 + (y2-y1)2.Set the value of k by cross validation. Detect outliers in training data by finding labeled members far away from similarly labeled or surrounded by differently labeled, k-nn summary: **Advantages**: Fast Training, Learns complex target functions easily, Retains information (e.g. training set) **Disadvantages**: Slow at testing and inference time, Needs a lot of storage, Easily fooled by irrelevant attributes & noise

Fisher’s Linear Discriminant: Fisher’s linear discriminant is optimal if the classes are normally distributed. / Fisher’s linear discriminant can be used even when the classes are not normal. / We have projected the samples from d dimensions to 1.

Text, letter

Description automatically generated

SVMs are particularly well suited for classification of complex small- or medium- sized datasets. Find the widest street. The centerline is called a Hyperplane Lagrangian Is a strategy for finding the local maxima and minima of a function subject to the condition that one or more equations have to be satisfied exactly by the chosen values of the variables

Graphical user interface, text

Description automatically generated

The �!are the Lagrange Multipliers. The �! that are non-zero will lie in the “gutter” the rest of the samples have �!=0! Optimization only depends on the dot product of pairs of points.

Use scaling, use a kernel function to transform to a new space. Linear: K(x, xT) = x dot xT, Polynomial : K(x, xT) = (x dot xt)d, Gaussian: exp(-1/2||x-xT||/sigma) Lienar function: f(x) = sign(w × x + b), The kernel trick: instead of explicitly computing the lifting transformation φ(x), define a kernel function K such that K(xi,xj) = psi(xi) dot psi(xj), (to be valid, the kernel function must satisfy Mercer’s condition), • This gives a nonlinear decision boundary in the original feature space:, Mercer's condition ensures that the inner product in the feature space induced by the kernel function K(x,y) is always non-negative, which is a necessary condition for the associated kernel matrix to be positive semidefinite. Positive semi-definiteness of the kernel matrix is crucial for the stability and convergence of algorithms that rely on kernel methods, such as SVMs.

In Machine Learning, a kernel is a function capable of computing the dot product ϕ(a)⊺ ϕ(b), based only on the original vectors a and b, without having to compute (or even to know about) the transformation ϕ. By using kernel functions, we can compute the dot products without explicitly expanding the feature space. We can effectively capture complex relationships as if we had added numerous polynomial features, including high-degree polynomials, without the computational and storage costs associated with adding them explicitly. The kernel trick gets the same result as if you had added many polynomial features, even with very high-degree polynomials, without actually having to add them If the training set is not too large, you should also try the Gaussian Radial Basis Function (RBF) kernel; it works well in most cases. SVM Regression tries to fit as many instances as possible on the street while limiting margin violations (i.e., instances off the street). Introduce Slack variable. Minimize the distance include slack (w dot w C x sum of slack, soft margin, Large C, the focus is on having a low training error (smaller margin) - Overfitting / Small C, the focus is on having a low generalization error (larger margin) - Underfitting / Dual Problem: Primal problem - constrained optimization problem, Dual problem is a different but closely related problem, / The solution to the dual problem typically gives a lower bound to the solution of the primal problem, but under some conditions it can have the same solution as the primal problem. / The SVM problem happens to meet these conditions so you can choose to solve the primal problem or the dual problem; both will have the same solution. / The dual problem is faster to solve than the primal one when the number of training instances is smaller than the number of features. / More importantly, the dual problem makes the kernel trick possible, while the primal does not.

Lecture 4

No Free Lunch Theorem: There is no algorithm that is always the most accurate / Different Algorithms / different hyperparameters, Voting, Stacking, Bagging, Boosting, Randomization, can weight votes – gating. Stacking, variety of base learners combined by a usually simple learner. Combiner does not need to be a linear combiner as in voting. Output of first layer is used as features to the second layer. Can do multiple layers, No Free Lunch Theorem: There is no algorithm that is always the most accurate, Instead use scheme based on cross-validation, Cascading – one learner at a time. , linear methods are more reasonable when combining complex Gaussian kernels.

Bagging (bootstrap aggregating) - when sampling is performed with replacement (i.e. data instances can be sampled more than once to create a given training set) / Pasting - when sampling is performed without replacement (i.e. any particular data instance can only be sampled once) Unstable Algorithms improve with Bagging / Bagging parameters: Max samples, Max Features, Bootstrapping of Samples, Bootstrapping of Features / Bagging is suitable for high variance low bias models (for complex models). Out of Bag instances – sample not selected, about 37%, can be used for validation. but aggregation reduces both bias and variance / scales well because of parallelism. Random patches – sample both instances and features, Random Subspaces – all instances but random features / so bagging ends up with a slightly higher bias than pasting.

AdaBoost - The new classifier pays a bit more attention to the training instances that its predecessor underfitted (e.g. “got wrong”)

Text

Description automatically generated

It turns out that Adaboost is just scaling the weights for the answers we got correct by ½ and also scaling the weights For the answers we go wrong by ½ and then normalizing so that all the weights add to One to enforce a distribution! / weak classifiers are too complex 🡪 overfitting / week classifiers are too weak 🡺 underfitting 🡪 low margins 🡪 overfitting / susceptible to uniform noise / sequential / hard to parallelize

Gradiant boosting - tries to fit the new predictor to the residual errors made by the previous predictor.

Random Forests / **Advantages of Random Forests** / Handles both Classification and Regression / Handles missing values and maintains accuracy for missing data / Somewhat immune to overfitting / Handles large datasets with higher dimensionality / **Disadvantages of Random Forests** / Although good for classification, not as accurate on regression / “Black box” / User has little control over what model does (random seeds, parameters) / Relatively Slow during Testing so not good for real-time operation

Extremely Randomized Trees / This technique trades more bias for a lower variance.