REGULARIZATION

Regularization:

- To prevent over fitting and improves generalization
- L1 or Lasso and L2 or Ridge regression or L1-L2 regularization
- Adds a penalty to the error term
- One penalizes the absolute term while the other penalizes in squared manner
- Used for the Bias-Variance trade off
- One makes the coefficients to zero while other makes them near zero
- Lasso can be used for feature selection as well as if there is multicollinearity in the input data

What is Bias Variance trade off

- Bias: fit the line predicted if zero bias
- Variance: more featutrres
- Needs to balance bias-variance for predicted unseen data
- Low bias – High variance → overfitting
- High bias – High variance → wrong model – low accuracy
- High bias – Low variance → Underfitting

L2 Regularization/Ridge Regression

- To avoid overfitting
- When trying to fit Ordinary Least Square linear regression we try to minimum sum of square (actual – predicted).. but the form add some penalty \lambda*Slopes^2
- \sigma (y_i- y_pred)^2 + \lambda*Slopes^2 and try to find minimum (Ridge)
- Less dependent on x than OLS

L1 /Lasso

- \sigma (y_i- y_pred)^2 + \lambda*|Slopes|
- Lasso reduces features to zero for multicollinearity → feature selections

Elasticnet Regularization

- Combine L1 and L2
- \sigma (y_i- y_pred)^2 + \lambda*|Slopes|+ \lambda*Slopes^2
- 1/2n * \sigma (y_i- y_pred)^2 + ([1-\alpha]/2 *\lambda*|Slopes|+\alpha* \lambda*Slopes^2)
  where n → Number of samples and \alpha → ridge/lasso parameter

BAGGING

- Various models are built in parallel
- All models vote to give the final prediction

BOOSTING

- Train the decision tree in a sequence
- Learn from the previous tree by focusing on incorrect observations
- Build new model with higher weight for incorrect observations from previous sequence
-

CROSS VALIDATION AND MODEL SELECTION

Cross Validation

- Get good model with better accuracy prediction
- Avoid Problem of overfitting – creating model not good for prediction
- Splitting – Train and test dataset with model never see data in test set

How CV works:

- Train – split sets
- K fold CV (special case: leave one out with K=N)
- Stratified CV

```
F1 = 2 * (precision * recall) / (precision + recall)
```

Predictive Model: Evaluation

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

predictive result / classification		actual result / classification: yes	actual result / classification: no
	yes	**tp** (true positive)	**fp** (false positive) ← Type 1 error
	no	**fn** (false negative)	**tn** (true negative)

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \qquad \text{True Negative Rate} = \frac{tn}{tn + fp}$$

Missing or corrupted data in a dataset:

- Deleting rows/columns
- Replacing with Mean/Median/Mode
- Create an unknow category (for Unknown – in profession column)
- Predicting the missing values
- Using algorithms support missing values: Random forest, KNN, carboost, xgboost, …

Data normalization

- Standardization: clustering, PCA, data with outliers
- Normalization: image, neural networks
- Yes – clustering, KNN, PCA, Gradient method
- No – tree models. Naïve Bayes, LDA

Tailed feature distribution

- Truncation
- Square root transformation
- Logarithmic transformation

SQL https://www.sqltutorial.org/sql-cheat-sheet/

Overfit https://elitedatascience.com/overfitting-in-machine-learning

- Cross validatiom
- Train with more data
- Remove features
- Early stopping
- Regularization
- Ensembling

Query embedding: I don't have time

Random forest: https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb

AdaBoost https://link.medium.com/15210kYXv7

Decision tree Notebook

Imbakanced https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18

1. Change the performance metric
2. Change the algorithm
3. Oversample minority class
4. Undersample majority class
5. Generate synthetic samples

NLP https://builtin.com/data-science/easy-introduction-natural-language-processing

Deep learning https://www.investopedia.com/terms/d/deep-learning.asp

C=CNN https://en.wikipedia.org/wiki/Convolutional_neural_network

Reinforelearning https://en.wikipedia.org/wiki/Reinforcement_learning