NYU

Introduction to Data Science

Center for Data Science
Iddo Drori, Spring 2019

Agenda

- Expected value framework
- Feature engineering and data cleaning
- Combining models

- Example midterm questions

NYU

Expected Value Framework

Expected Value Framework: Marketing Example

Expected benefit of targeting:

$$P(R \mid x)\, VR(x) + (1 - P(R \mid x))\, VnotT(x) > 0$$

Response R

Consumer x

Value V

$$P(R \mid x)\, (dR(x) - c) + (1 - P(R \mid x))\, (-c) > 0$$

$dR(x)$ payment of responding customer x

Targeting cost c

$$P(R \mid x)\, dR(x) > c$$

class probability estimation — regression

Expected Value Framework: Churn Example

Expected benefit of targeting:

$$EBT(x) = P(S \mid x, T)\,(Us(x) - c) + (1 - P(S \mid x, T))\,(UnotS(x) - c)$$

Stay S
Customer x, Target T
Profit U

Expected benefit of not targeting:

$$EBnotT(x) = P(S \mid x, notT)\,Us(x) + (1 - P(S \mid x, notT))\,UnotS(x)$$

Maximize value of targeting with new offer
Greatest change in value as a result of targeting

$$EBT(x) - EBnotT(x) = (P(S \mid x, T) - P(S \mid x, notT))\,Us(x) - c$$

proxy data or 1 or target available data regression

NYU

Feature Engineering and Data Cleaning

Feature Engineering

- One-hot encoding: categorical to numerical, multiple binary
  A = [1,0,0] B = [0,1,0] C = [0,0,1]

- Binning: numerical to categorical
  Ages: 0-12 = child 13-18 = adult

- Normalization: convert to range to [0,1] (or [-1,1])
  xj = (xj - min(j)) / (max(j) - min(j))

- Z-score normalization: convert to 0 mean and 1 standard deviation
  xj = (xj - mu(j)) / std_dev(j)

Data Cleaning: Problems and Solutions

Tidy Data, Wickham 2013

(1) Column headers are values not variable names.
(2) Multiple variables are stored in one column.
(3) Column contains variable names, not values.
(4) Multiple types of observational units stored in same table.
(5) Variables of a single entity are spread across multiple tables
(6) Observations of a single entity spread across multiple tables.

Drori et al, 2018

(7) Missing values.
(8) Duplicates.
(9) Outliers.
10) Categorical values.
11) Non-standardized data.
12) Non-canonical data.
13) Missing target labels.
14) Missing target variable.
15) Invalid data defined by a domain expert.

NYU

Tidy Data

Problem 1

Problem 1: *column headers are values, not variable names.*

- Problem: for each row $i = 1, \ldots, m$, columns $(j_1, \ldots, j_k)$ contain values corresponding to mutually exclusive cases for a single category. For example, as shown in Table 3, the table has types A, B, and C as headers, with missing values.

Table 3: Data cleaning problem 1

Dataset	Type A	Type B	Type C
breast cancer	3366	556	460
hill valley	17951	N/A	8411
vehicle	N/A	N/A	5315

Solution 1

- Solution: a tidy way to reshape this table is to create a new column j, whose entries are the corresponding values in the column header of the original data table $c(A_{i,[j_1,\ldots,j_k]}) = A_{[i_1,\ldots,i_k],j}$. For example, as shown in Table 4, the Type A, Type B, and Type C columns are merged into a single Type column with multiple rows:
  Notice that missing values are discarded.

Table 4: Data cleaning solution 1

Dataset	Type	Value
breast cancer	A	3366
breast cancer	B	556
breast cancer	C	460
hill valley	A	17951
hill valley	C	8411
vehicle	C	5315

Problem 2

Problem 2: *multiple variables are stored in one column.*

- Problem: for each row $i = 1, \ldots, m$ column j contains multiple variables $A_{i,j} = A_{i,[j_1,\ldots,j_k]}$, where k is the number of variables in the cell $A_{i,j}$. For example, as shown in Table 5, the categorical resource field is a combination of two variables: type and format:

Table 5: Data cleaning problem 2

Dataset	Resource	Size
breast cancer	image-pgn	129MB
hill valley	table-csv	2.16GB
vehicle	audio-wav	1.33GB

Solution 2

- Solution: a tidy way is to split the column into two separate categorical features. Split the column into several columns $c(A_{i,j}) = A_{i,j_1}, \ldots, A_{i,j_k}$, for all i. For example, as shown in Table 6, the Resource column is split into two columns: Type and Format.

Table 6: Data cleaning solution 2

Dataset	Type	Format	Size
breast cancer	image	pgn	129MB
hill valley	table	csv	2.16GB
vehicle	audio	wav	1.33GB

Problem 3

Problem 3: *column contains variable names, not values.*

- Problem: a column contains variable names instead of values. For example, in Table 7, the column Event indicates the name of the variable in the Date column.

Table 7: Data cleaning problem 3

Dataset	Event	Date
criteo-display-ad-challenge	Competition	September 2014
criteo-display-ad-challenge	Solution	August 2015

Solution 3

- Solution: merge all the rows corresponding to the same observation $A_{i_1,j}, \ldots, A_{i_k,j}$ into one row i, and for each variable name create a separate column, $c(A_{[i_1,\ldots,i_k],j}) = A_{i,j_1}, \ldots, A_{i,j_k}$. This solution is illustrated in Table 8. Note that this case is the transpose of problem 1.

Table 8: Data cleaning solution 3

Dataset	Competition Date	Solution Date
criteo-display-ad-challenge	September 2014	August 2015

Problem 4

Problem 4: *multiple types of observational units stored in same table.*

- There exist $A_{i_1,j_1,\ldots,j_k} = A_{i_2,j_1,\ldots,j_k}$ for $i_1 \neq i_2$ where A_{*,j_l} represents the same observational unit. For example, the data and task meta-data are repeated for each solution, as shown in Table 9.

Table 9: Data cleaning problem 4

Dataset	Data	Task Type	Estimator	Performance
titanic	table	classification	Decision Tree	0.8134
titanic	table	classification	Random Forrest	0.8676

Solution 4

- A solution is to split the data into two datasets: a data and task meta-data and a solution meta-data, and use an index to join them, as shown in Table 10. While this solution avoids repeating values it requires a later join which may be cumbersome, and maintaining multiple tables and indices may be error prone. Therefore, we find this optional, and for simplicity and since for 58% of the datasets we have a single solution, decided to keep a single table.

Table 10: Data cleaning solution 4

Id	Dataset	Data Type	Task Type
5	titanic	table	classification

Id	Estimator	Performance
5	Decision Tree	0.8134
5	Random Forrest	0.8676

Problem 5

Problem 5: *variables of a single entity are spread across multiple tables.*

- Problem: originally data and task were spread across different tables, as illustrated in Table 11

Table 11: Data cleaning problem 5

Dataset	Data Type
titanic	table
higgs-boson	table
nyc-taxi-trip-duration	table

Dataset	Task Type
titanic	classification
higgs-boson	classification
nyc-taxi-trip-duration	regression

Solution 5

- Solution: a tidy way is merging these tables into a single table by corresponding datasets, as shown in Table 12.

Table 12: Data cleaning solution 5

Dataset	Data Type	Task Type
titanic	table	classification
higgs-boson	table	classification
nyc-taxi-trip-duration	table	regression

Problem and Solution 6

Problem 6: *observations of a single entity spread across multiple tables.*

- Problem: For example, originally separate tables containing rows of meta-data about a data, task, and solution were separately prepared (each week by each graduate student).
- Solution: combine all tables rows into a single table with multiple rows.

Problem and Solution 7

Problem 7: *missing values.*

- Problem: variable $A_{i,j}$ is missing.
- Solutions [20]:

(1) Add a boolean feature indicating missingness.
(2) Add default value, $c(A_{i,j}) = a'_j$
(3) Discard data record i, $c(A_{i,*}) = \emptyset$.
(4) Discard feature j, $c(A_{*,j}) = \emptyset$ if feature has at least probability p of missing values.
(5) Set $c(A_{i,j})$ to the mean, median, or most frequent value.
(6) Regression models: define the target variable $y = A_{*,j}$ and partition the vector $y = (y_o, y_m)$ into observed and missing values. Partition the matrix of predictors $X = A_{*,*\setminus\{j\}}$ into rows X_o where y is observed, and rows X_m where y is missing. Train a regression model for $y = f(X, \beta)$ and use it to impute missing values. Regression models include:
   – Linear regression:

$$\hat{\beta} = \arg\min_{\beta} \|y_o - X_o\beta\|^2 \quad (1)$$

The missing value is then imputed as:

$$\hat{y}_m = X_m\hat{\beta} \quad (2)$$

Drori et al, 2018

NYU

Box's Loop

Solution 7

– Lasso and Ridge regression:

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2}\|y_o - X_o\beta\|^2 + \lambda\|\beta\| \tag{4}$$

– Decision tree, random forest.

(7) Donor imputation: replace the missing value in a record with an observed value copied from other records. For example, using KNN and predictive mean matching.

Drori et al, 2018

Problem and Solution 8

Problem 8: *duplicates.*

- Problem: duplicate data records.
- Solutions: handle duplicates to reduce bias.

(1) Discard duplicates.

(2) Add feature counting number of duplicates.

Drori et al, 2018

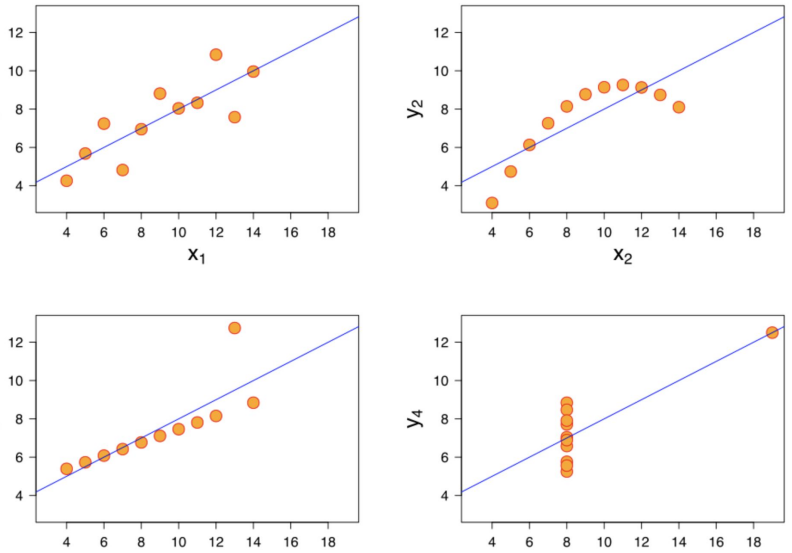
Problem and Solution 9

Problem 9: *outliers.*

- Problem: a data entry $A_{i,j}$ which is distant from other values, or statistically inconsistent with respect to other values. Methods for detecting outliers include:
  (1) Modified Thomson τ method: compute the mean μ_j and standard deviation σ_j of feature j, and compute the absolute value of the deviation from the mean $\delta_i = |A_{i,j} - \mu_j|$ for each $i = 1, \ldots, m$. Use the student t distribution to detect an outlier if: $\delta_i > \sigma_j \frac{t_{\alpha/2}(m-1)}{\sqrt{m}\sqrt{m-2+t_{\alpha/2}{}^2}}$.
  (2) Clustering.
  (3) Visual inspection [1, 24].
  (4) Histogram conditional on other features.
  (5) Distribution of raw unaggregated values.
  (6) Unsupervised density estimation, low probability point for that model.
- Solutions:
  (1) Discard data record.
  (2) Replace with nearest non-outlier, or using any of the solutions for handling missing values (problem 7).

Drori et al, 2018

Anscombe's Quartet



Property	Value	Accuracy
Mean of *x*	9	exact
Sample variance of *x*	11	exact
Mean of *y*	7.50	to 2 decimal places
Sample variance of *y*	4.125	±0.003
Correlation between *x* and *y*	0.816	to 3 decimal places
Linear regression line	$y = 3.00 + 0.500x$	to 2 and 3 decimal places, respectively
Coefficient of determination of the linear regression	0.67	to 2 decimal places

Problem and Solution 10

Problem 10: *categorical values.*

- Problem: a feature is not given as a number but as a discrete category.
- Solutions:

(1) One-hot encoding.
(2) Embedding.

Drori et al, 2018

Problem and Solution 11

Problem 11: *non standardized data.*

- Problem: the data format of a feature changes for different data points.
- Examples: date, time, location, address, currency, phone.
- Solutions:

(1) Make uppercase or lowercase.
(2) Remove letters, numbers, or punctuation.
(3) Trim spaces.
(4) Stemming.

Drori et al, 2018

Problem 12

Problem 12: *non canonical data.*

- Problem: many variations or different representation of the same unique value or element.
- Examples: all available features as well as their unique values are usually not known in advance, and discovered during the collection process.

Drori et al, 2018

Solution 12

(1) Dictionary of canonical values for features defined and expanded during collection.
(2) Group values which are pronounced alike, spell checking [23].
(3) Group values with common characters and numbers [23].
(4) Use feature extractors $\phi(x)$.

Drori et al, 2018

NYU

Partial Task Schema

Problem and Solution 13

Problem 13: *missing target labels.*

- Problem: rows $i_1, \ldots, i_k$ are missing the target labels $y_{i_1,\ldots,i_k}$.
- Solutions:

(1) Discard data records $A_{[i_1,\ldots,i_k],*}$.

(2) Semi-supervised learning:
  - Generative model: extend supervised learning using a prior on the data or extend unsupervised learning, clustering, using partial labels.
  - Transductive support vector machine [12]: label the unlabeled data to obtain a maximum margin over all data records.
  - Graph based method [3]: connect data records by their similarity, and use the graph Laplacian as a regularization term enforcing smoothness.

Drori et al, 2018

Problem and Solution 14

Problem 14: *missing target variable.*

- Problem: the target variable of the regression problem is not specified.
- Solutions:

(1) Human in the loop [6]. For example, knowing if a target variable is age or gender requires a human defining the task.

(2) Datasheets for datasets [13]: using additional meta-data to infer the intended usage of the dataset.

(3) Learn a generative model for the joint distribution over all the columns $p(x_1, \ldots x_n)$. Whenever a prediction for target x_j is required, use Bayes rule: $p(x_j|x_{-j}) = \frac{p(x_1, \ldots x_n)}{p(x_{-j})}$ where $x_j = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$.

Drori et al, 2018

NYU

Data Validation

Problem 15

Problem 15:

- Problem: checking variable types, value ranges, consistency, and dependencies [25].
- Solutions:

(1) Human in the loop: data validation usually requires a domain expert.
(2) Discard data record.
(3) Repair: error correction rules can be formulated using mixed integer programming and are beyond the scope of this work.

Drori et al, 2018

NYU

Combining Models

Ensemble

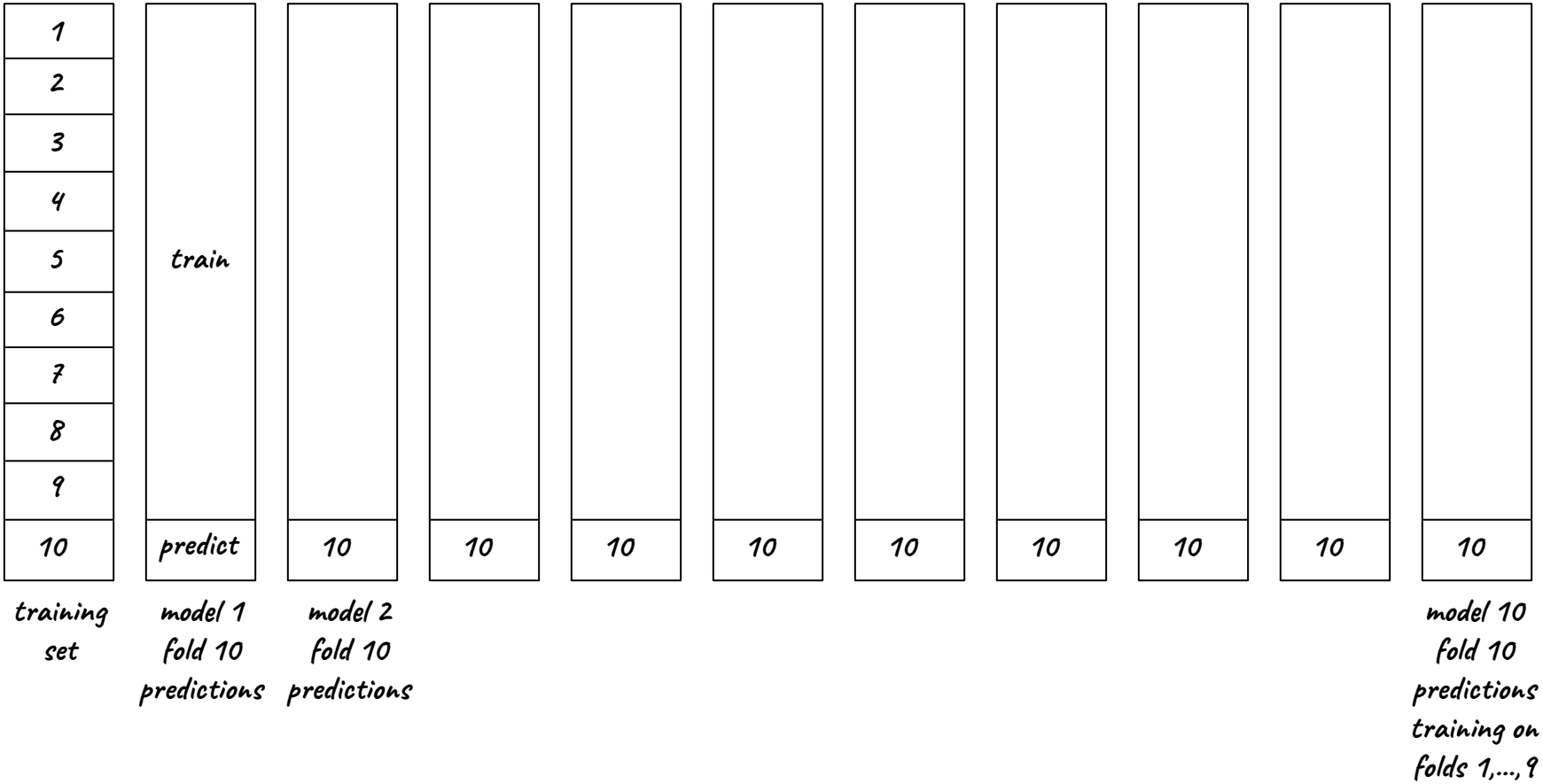
- Taking argmax over the average of probabilities over the models
- Models m
- Classes j

$$y = \arg\max_j \frac{1}{m}\left(\sum_{i=1}^{m} p_i^{(j)}\right)$$

Stacking: Train and Predict Same Folds for Each Model

Stacking: Train and Predict Same Folds for Each Model

Stacking: Train and Predict Same Folds for Each Model



Stacking: Super-Model using Predictions as Meta Features

1	model 1 fold 1 predictions	model 2 fold 1 predictions	model 3 fold 1 predictions	model 4 fold 1 predictions	model 5 fold 1 predictions	model 6 fold 1 predictions	model 7 fold 1 predictions	model 8 fold 1 predictions	model 9 fold 1 predictions	model 10 fold 1 predictions
2	model 1 fold 2 predictions	model 2 fold 2 predictions	model 3 fold 2 predictions	model 4 fold 2 predictions	model 5 fold 2 predictions	model 6 fold 2 predictions	model 7 fold 2 predictions	model 8 fold 2 predictions	model 9 fold 2 predictions	model 10 fold 2 predictions
3 4 5 6 7 8 9										
10	model 1 fold 10 predictions	model 2 fold 10 predictions	model 3 fold 10 predictions	model 4 fold 10 predictions	model 5 fold 10 predictions	model 6 fold 10 predictions	model 7 fold 10 predictions	model 8 fold 10 predictions	model 9 fold 10 predictions	model 10 fold 10 predictions

training

validation

test

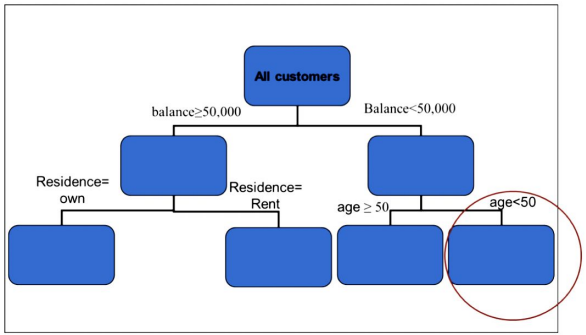
use level-1 model predictions as meta features for super stacked model

NYU

Example Midterm

Example Question

- You are constructing an information gain based decision tree to predict a binary (yes/no) dependent variable Loan Payment. Your features include: Balance, Residence, Age, Gender, Employment, Marital Status.
- After beginning to build the tree your reach the tree structure:



- Which variables would you may consider useful for splitting the marked node?

Example Question

- When would you use a more complex model, such as with polynomial features rather than a simple model such as linear regression?

Answer

- **More complex models** should be used when **more data is available**.
- If only few samples are available such models may overfit; whereas simpler model, with fewer degrees of freedom, may be prefered.

- Optional: using auxiliary data, ie transfer learning or embeddings

Example Question

- A linear regression problem of the form:

  $$\text{minimize}_x \; \| y - Ax \|^2$$

  is not always solved analytically, and optimization techniques used instead.

- Why? Describe a solution.

Answer

- The model may have a very large number of parameters. Linear system may be too large and require memory which quadratic in number of parameters.

- **Stochastic gradient descent** requires memory linear in number of parameters and may be a practical solution.

Problem 1

Problem 1. Visualizing Model Performance (22 points):
(a) Define the ROC curve, explain the meaning of each point on the curve. (4 points)
(b) Write pseudo code for generating the ROC curve. (5 points)
(c) Define the AUC, describe its usage and range of values. (5 points)
(d) Describe a case in which AUC is a better performance measure than accuracy. (3 points)
(e) Define a profit graph and explain its limitation. (3 point)
(f) Define learning curves. (2 points)

Solution 1

(a) In a ROC curve the true positive rate (Sensitivity) is plotted as a function of the false positive rate (100-Specificity). It is plotted for different cut-off points. Each different point in ROC space corresponds to a specific confusion matrix. The important features in ROC curve are as follows:

- The lower left point(0, 0) represents the strategy of never issuing a positive classification. Such a classifier commits no false positive errors but also gains no true positives.
- The opposite strategy of unconditionally issuing positive classifications is represented by the upper right point (1, 1).
- The point (0, 1) represents perfect classification, represented by a star.
- The diagonal connecting (0, 0) to (1, 1) represents the policy of guessing a class.
- One point in ROC space is superior to another if it is northwest of the other.

Solution 1

(b)

- Assign a score to each instance of your sample.
- Order them decreasing from bottom to top
- Start at the bottom with an initial confusion matrix where everything is classified as N
- Moving upward, every instance moves a count of 1 from N row to the Y row, resulting in a new confusion matrix

Solution 1

Each confusion matrix maps to a (fp rate, tp rate) pair in ROC space
(c) An important summary statistic is the area under the ROC curve(AUC). As the name implies, this is simply the area under a classifiers curve expressed as a fraction of the unit square. Its value ranges from zero to one. Though a ROC curve provides more information than its area, the AUC is useful when a single number is needed to summarize performance, or when nothing is known about the operating conditions.
(d) AUC is a better metric than accuracy for datasets which are skewed as accuracy is a misleading metric in such cases.

Solution 1

(e) Using a ranking classifier, we can produce a list of instances and their predicted scores, ranked by decreasing score, and then measure the expected profit that would result from choosing each successive cut-point in the list. Conceptually, this amounts to ranking the list of instances by score from highest to lowest and sweeping down through it, recording the expected profit after each instance. At each cut-point we record the percentage of the list predicted as positive and the corresponding estimated profit. Graphing these values gives us a profit curve.
The disadvantage of a profit graph is that it requires that operating conditions be known and specified exactly.

Solution 1

(f) A learning curve shows model performance on testing data plotted against the amount of training data used. Usually model performance increases with the amount of data, but the rate of increase and the final asymptotic performance can be quite different between models.

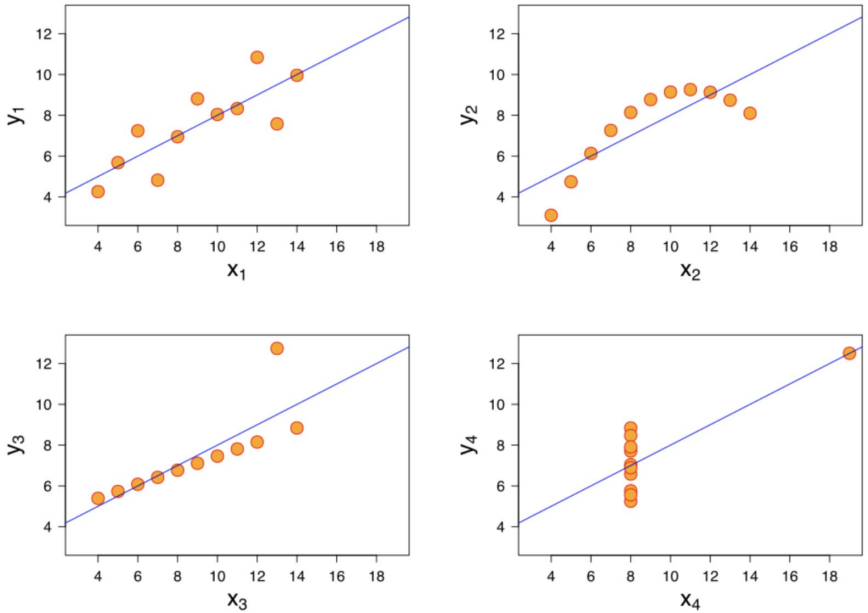


Figure 1: Anscombe's quartet

Problem 2

Problem 2. Similarity and Clustering (23 points):
(a) Define the following distance functions: Euclidean, Manhattan, Jaccard, Cosine, Levenshtein (5 points).
(b) Define a majority vote classifier (2 points).
(c) Define similarity moderated classification and regression (4 points).
(d) Write pseudo-code for K-means clustering. Is the algorithm guaranteed to converge? to the same clusters? (7 points).
(e) The four datasets shown in the figure, known as Anscombe's quartet, have the same mean, variance, correlation, and linear regression line, yet look very different. Describe a method for distinguishing between the top two and lower two datasets. (5 points)

Solution 2

(a) (i) Euclidean Distance

For $x, y \in R^n$: $d_2(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^2)^{\frac{1}{2}}$

(ii) Manhattan Distance

For $x, y \in R^n$: $d_1(x, y) = \sum_{i=1}^{n} |x_i - y_i|$.

(iii) Jaccard Distance: $1 - \frac{|X \cap Y|}{|X \cup Y|}$

(iv) Cosine Distance: $1 - \frac{A \cdot B}{\|A\| \|B\|}$

(v) Levenshtein Distance: Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

Solution 2

(b) Majority vote classifier is used in K nearest neighbors to decide the output of the unknown target variable. An instance belongs to a class where most of its neighbors belong to.
(c) Similarity moderated classification and regression uses the distance between points to determine the the cluster of that instance. The weight is computed by taking the reciprocal of the square of the distance. The nearer points get more weight. The score in case of classification and value in case of regression is calculated based on weighted scores or values of the neighbors

Solution 2

(d) Pseudo Code for K Means Clustering:
Step 1: Clusters the data into k groups where k is predefined.
Step 2: Select k points at random as cluster centers.
Step 3: Assign objects to their closest cluster center according to the Euclidean distance function.
Step 4: Calculate the centroid or mean of all objects in each cluster.
Step 5: Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.
Algorithm always converges may not end up in same clusters always.
(e) Clustering or outlier detection.

Problem 3

Problem 3. Overfitting (23 points):

(a) In your Kaggle competition, your position in the public leaderboard may have been different than your position in the private leaderboard. Explain why, and why Data Science competitions maintain two different leaderboards. (4 points)

(b) Describe how overfitting can be avoided in a decision tree model. (4 points)

(c) Define k-fold cross validation. (4 points)

(d) How does feature selection help prevent overfitting? Describe the forward stepwise algorithm for feature selection. (4 points)

(e) Define the Lasso and explain its regularization. (6 points)

(f) Can the Lasso be used as method for feature selection? Explain. (1 point)

Solution 3

Answer:

(a) Overfitting is the tendency of data mining procedures to tailor models to the training data, at the expense of generalization to previously unseen data points.As a model gets more complex,it is able to pick up harmful spurious correlations that are idiosyncracies of the specific training set used and do not represent characteristics of the population in general. Performance declines with overfitting as these spurious correlations produce incorrect generalizations in the model. In a Kaggle competition, the public leaderboard is calculated on a section of the test data and the private leaderboard is calculated on the entirety of the test data ,mimicking the test set and validation set approach.This is done to prevent to overfitting of the model on the basis of its performance on the test data.

Solution 3

(b) If no constraints are imposed the tree induction algorithm will keep growing the tree to fit the training data until it creates pure leaf nodes. To avoid overfitting we can(i) stop growing the tree before it gets too complex, and (ii) grow the tree until it is too large,then prune it back, reducing its size (and thereby its complexity). Commonly used techniques are limiting the maximum number of nodes a tree can have, specifying the minimum number of instances that must be present in a leaf and pruning the leaves/branches of a tree until any removal or replacement reduces accuracy

Solution 3

(c) K-fold Cross-validation involves splitting a labeled dataset into k partitions called folds. In each iteration of the cross-validation procedure, a different fold is chosen as the test data while the other k1 folds are combined to form the training data. When cross-validation is finished, every example will have been used only once for testing but k1 times for training.Typically performance estimates of the models from each iteration are averaged to produce an estimate of the model's out-of-sample performance.

Solution 3

(d) In general, fitting a model with numeric parameters involves finding the set of parameters that maximizes an objective function indicating how well it fits the data.

$$argmax_w fit(x, w)$$

Regularization avoids overfitting by adding to this objective function a penalty for complexity:

$$argmax_w[fit(x, w) - \lambda penalty(w)]$$

Ridge regression is when we incorporate the L2 norm as the penalty term in a standard least-squares regression

$$argmax_w[gIikelihood(x, w) - \lambda||w||_2]$$

Solution 3

(e)LASSO is when we incorporate the L1 norm as the penalty term in a standard least-squares regression

$$argmax_w[g_likelihood(x, w) - \lambda||w||_1]$$

Solution 1

(f) Yes. Usage of L1 norm means that the optimization algorithm can find a corner point optima and thus L1-regularization ends up zeroing out many coefficients.Since these coefficients are the multiplicative weights on the features, L1-regularization effectively performs an automatic form of feature selection.

Problem 4

Problem 4. Text Mining (22 points)

(a) Describe a bag of words representation, and its limitation. (3 points)

(b) Describe an n-gram representation, and its limitation. (3 points)

(c) What is the relationship between a bigram and a Markov model? (4 points)

(d) Define TF-IDF, and it's connection to entropy. (4 points)

(e) Explain the usage of cosine similarity with TF-IDF for document retrieval. (4 points)

(f) Define a RNN, and its advantage over a bag of words and Markov models. (4 points)

Solution 4

Answer:
(a) Multi-set. Does not preserve word order nor model long term dependencies.
(b) An n-gram is a contiguous sequence of n items from a given sample of text or speech. It does not model long term dependencies.
(c) They are the same.

Solution 4

(d) TF(t, d) term frequency of word in document.
IDF(t) = 1 + log (total # of documents)/(# of documents containing t). Inverse document frequency.
TFIDF(t,d) = TF(t, d) · IDF(t).
Entropy can be expressed as the expected value of IDF(t) and IDF(not t).
E(t) = -p log(p) - (1-p) log(1-p)
= p IDF(t) - (1-p)[-IDF(not t)]
= p IDF(t) + (1-p)[IDF(not t)]
(e) $d_{cosine}(x, y) = 1 - \frac{x \cdot y}{\|x\|_2 \|y\|_2}$.
Translate the query into its TFIDF representation and compute the TFIDF of each document. Use the cosine similarity to measure the similarity between the TFIDF representations of the query and documents, and find the nearest document.

Solution 4

(f) The advantage of an RNN is that it models long term dependencies while sharing weights across time.

Problem 5

Problem 5. Model Evaluation (10 points):
The confusion matrix for a binary classifier is as follows:

	Actual Class 1	Actual Class 0
Predicted Class 1	70	30
Predicted Class 0	140	470

(a) Define precision and recall, and compute them for the confusion matrix. (2 points)
(b) Define sensitivity and specificity, and compute them for the confusion matrix. (2 points)
(c) Define the F1-score, and compute it from the confusion matrix. (2 points).

For a given feature vector input your logistic regression is currently predicting 1 when

$$f_w(z) = g(w^T x) > 0.5$$

where g is a sigmoid function.
(d) If you change the threshold, predicting 1 for g(z) $>$ 0.9, how will precision and recall be affected? (2 points)
(e) If you change the threshold, predicting 1 for g(z) $>$ 0.3, how will precision and recall be affected? (2 points)

Solution 5

Answer:

(a) $Precision = TP/(TP + FP) = 0.7, Recall = TP/(TP + FN) = 0.3$

(b) $Specificity = TN/(TN + FP) = 0.94, Sensitivity = TP/(TP + FN) = 0.3$

(c) $F1 = 2 * (precision * recall)/(precision + recall) = 0.42$

In both the below examples the behaviour of recall can be easily determined withiout ambiguity. The behaviour of precision is little ambiguous. The behaviours given here are taken in general.

(d) When we increase the threshold, we are only predicting something as 1 when we are completely sure. So generally the precision increases. The recall decreases beacuse FN increases.

(e) When we decrease the threshold, we are predicting something as 1 when we are not completely sure. So generally precision decreases. Recall increases because FN decreases