**Lecture 1 Introduction**

图示

AI 生成的内容可能不正确。

**Cross Industry Standard Process for Data Mining**

Step 1: Business Understandi Step 2: Data Understanding

Step 3: Data Preparation Step 4: Model Building

Step 5: Testing and Evaluation Step 6: Deployment

import datetime studentID='58609426'

print('My StudentID:%s , currentTime:%s' % (studentID, datetime.datetime.now()))

import pandas as pd

myInfoTable=pd.DataFrame(columns=['StudenId' ,'CurrentTime'])

myInfoTable.loc[0]=[studentID, datetime.datetime.now()]

import time time.sleep(3)

myInfoTable = pd.DataFrame(columns=['id', 'name', 'time', 'expected\_scores'])

myInfoTable.loc[0] = [studentID,studentName,datetime.datetime.now(), None]

myInfoTable.loc[1] = [studentID,studentName,datetime.datetime.now(),95]

import numpy as np

matrix = df[["score", "grade"]].to\_numpy()

data = {

"id": ["111", "222", "333", "444"], "name": ["L", "Z", "W", "C"],

"grade": [4.1, 2.8, 3.1, 3.6], "score": [95, 70, 80, 88],

} df = pd.DataFrame(data) df.describe() #min,max,median...

matrix = df[["score", "grade"]].to\_numpy() 2x4

transpose = matrix.T 4x2

sum\_scores = np.sum(matrix,axis=0) print("\nSum of all scores:") [333. 13.6]

df["score"].apply(grade\_category) #apply function, elif ...: return “xxx”

**Lecture 2 Linear Regression**

Dependent variable: Y Independent variable: X Y=𝛽0 + 𝛽1 X+ 𝜀 𝛽0：intercept 𝛽1 slope 𝜀 random error

Linear regression **assumes**: **1**. The relationship between X and Y is linear **2.** Y is distributed normally at each value of X **3**. The variance of Y at every value of X is the same (homogeneity of variances) **4**. The observations are independent

图示

AI 生成的内容可能不正确。The estimated standard deviation of the slope: Roughly, there is a 95% chance that the true slope lies within 2 standard deviations 真实斜率位于 2 个标准差以内

**Extrapolation:** 如果你的样本数据中X的范围是[1, 10]，那么用X=1到X=10之间的值预测Y是相对可靠的。但如果用X=11(超出样本范围）预测Y，就是外推（Extrapolation），结果可能极不可靠。

**Outliers**: Drop or Predict

**Simple linear regression**: Using ONE independent variable to predict ONE dependent variable

**Multiple linear regression**: Using MORE THAN ONE independent var to predict a dependent var

图示

AI 生成的内容可能不正确。**Dummy Variable**: represent categorical variables, n cases can be represented with n-1 dummy variables. The value without a dummy variable is base level, which is indicated with all dummy variables are zero.

**Least Sum of Squares Curve Fitting**

**Conditional Likelihood p(y|x)**

文本

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。示意图

AI 生成的内容可能不正确。

文本, 信件

AI 生成的内容可能不正确。**文本

AI 生成的内容可能不正确。Minimizing the Squared Error**

**Optimazation**

图示

AI 生成的内容可能不正确。**文本, 信件

AI 生成的内容可能不正确。**

**Gradient Descent**

图表, 图示

AI 生成的内容可能不正确。

**Using Validation Set**: use for estimating generalization error, but Less data available for training

文本, 信件

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。**Regularization**

λ>=0

文本, 信件

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。**Ridge**  **LASSO**

**import numpy.random as rnd** # this is a library for random number generation

X = 2 \* rnd.rand(100, 1) y = 4 + 3 \* X + rnd.randn(100, 1)

**import matplotlib.pyplot as plt**  from matplotlib.pylab import rcParams rcParams['figure.figsize'] = 10, 8

plt.plot(X, y, "b.") plt.xlabel("$x\_1$", fontsize=18) plt.ylabel("$y$", rotation=0, fontsize=18) plt.axis([0, 2, 0, 15]) plt.savefig("generated\_data\_plot.png") plt.show()

**from sklearn.linear\_model import LinearRegression** #Ridge(alpha=alpha,normalize=True)

lin\_reg = LinearRegression() lin\_reg.fit(X, y)   lin\_reg.coef\_, lin\_reg.intercept\_

**from sklearn.metrics import r2\_score** evaluation\_result=r2\_score(Y\_new, Y\_new\_predict)

**import pandas as pd** ResearchData=pd.read\_csv('Airbnb.csv') #load data as pandas dataframe ResearchData.head() from sklearn.preprocessing import OneHotEncoder # OneHotEncoder(drop='first')

**from sklearn.model\_selection import train\_test\_split** x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y,

test\_size=0.30, random\_state=0) **from sklearn.metrics import mean\_absolute\_error**

mean\_absolute\_error(y\_eval, y\_pred) **from sklearn.preprocessing import OneHotEncoder**

**Lecture 3 Data Mining**

图示, 示意图

AI 生成的内容可能不正确。**Data**: Collection of data objects and their attributes

**Attribute values** are numbers or symbols assigned to an attribute for a particular object

**Types of Attributes**：**Nominal(象征,名义)**, distinguish one object from another; sex,zipcode,**Ordinal(有序型)**, provide enough information to order objects, educated degree, level

**Types of Attributes**: **Interval（间隔）**: the differences between values are meaningful.温度 (no ratios,可加减，不能乘除); **Ratio（比例）**, both differences and ratios are meaningful（加减乘除均可）长度

**Properties of Attribute Values** :Distinctness: =, != Order: > < Addition: + -, Multiplication: \*/ Nominal attribute: distinctness ; Ordinal attribute: distinctness & order Interval attribute: distinctness, order & addition Ratio attribute: all 4 properties

**Discrete and Continuous Attributes**: Discrete Attribute: represented as integer variables; Continuous Attribute: real numbers as attribute values

**Asymmetric Nominal Attributes**: Only presence (a non-zero attribute value) is regarded as important

**Types of data sets**: Data Matrix, Graph Data...

Important Characteristics of Structured Data : **Data Quality**; **Dimensionality**: The number of attributes; Curse(诅咒) of Dimensionality **Sparsity**: Only presence counts; Less than 1% of the entries are non-zero. **Resolution**: Patterns depend on the scale; The surface of the Earth seems very uneven at a resolution of a few meters, but is relatively smooth at a resolution of tens of kilometers.

**Descriptive Analytics**: Numerical summary measures，A variety of graphs，

**Non-Numerical Variables**: for describing categorical variables, based on counting

1. the number of categories 2. the number of observations in each category 3. frequency of observations in nonoverlapping classes/cells(将成绩分为"A/B/C/D"四组，统计每组人数。) 4. Excel COUNTIF

**Numerical Variables**: **A.Single Variable:**，1. **Measures of Central Tendency**: Mean, Median, Mode(众数), Min, Max, Percentiles/Quartiles(四分位数，分成四等分) 2.**Measures of Dispersion/Variability:** Range(=max-min), **IQR: Interquartile Range(四分位差Q3-Q1)**, Variance, Standard Deviation 3. Measures of Distribution Shape: Skewness(偏度,衡量对称性), Kurtosis(描述尾部的极端值)

**B.Multiple variables**: Covariance(两个变量的线性关系方向); Correlation(相关系数，标准化后的协方差)

The second quartile (p = 50%) is equal to the median

图示

AI 生成的内容可能不正确。图示

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

Correlation is linear concept, it may not be able to recover nonlinear relationships.

Correlation ≠ Causality(因果，多开银行分行和储蓄没有因果关系)

**Association Rule Mining**: Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

Between values: Apple=> Coke Between categories of values: Food => Magazine Between values of attributes: Married:yes => OwnHouse:yes Over time period: year 1: Database => year 2: Data Mining

**Product Placement, Recommendations, Bundling**

**Basic Concepts:** ItemSet(k-item), Support count(σ)(σ({Milk, Bread,Diaper}) = 2) Support: s({Milk, Bread, Diaper}) = 2/5 Frequent Itemset: s>=threshold save.

文本, 信件

AI 生成的内容可能不正确。Association Rule**:** An implication expression of the form X -> Y, where X and Y are itemsets {Milk, Diaper} ->{Beer}

Support: Fraction of transactions that contain both X and Y

Confidence: Measures how often items in Y appear in transactions that contain X

**Valid Association Rules: Minimum Support(保证频率够高), Minimum Confidence(保证预测可靠)**

low support, simply occur by chance. High confidence represents strong reliability, but do not imply causality, like **Umbrellas → Raincoats**, rain is co-reason, not causality in their relation.

Lift: Lift = 0.75 / 0.9 = 0.8333 < 1 → this rule is worse than not having any rule (negatively associated) Ø < 1: substitution effect无效 =1无关 >1: dependence effect有效

**Data Processing**

**Aggregation**: Combining two or more attributes (or objects) into a single attribute (or object)(Data reduction, Change of scale, More “stable” data(Aggregated data tends to have less variability))

**Sampling**: **Simple Random Sampling**：Sampling without replacement(无放回), Sampling with replacement(有放回，独立) **Stratified(分层) Sampling:** Split the data into several partitions; then draw random samples from each partition(独立)

**Dim Reduction**：**Purpose**: Avoid curse of dim; Reduce amount of time and memory required; Allow data to be more easily visualized; May help to eliminate irrelevant features or reduce noise **Technique**：PCA; Singular Value Decomposition(SVD); supervised and non-linear techniques

**Feature Subset Selection**: **Redundant features**: Highly correlated with other features, Redundancy could be removed using correlation analysis (e.g., Pearson's coefficient) or PCA. **Irrelevant features**: Characteristics not significantly associated with the target variable, Screening by statistical tests (e.g., chi-square test, information gain) or importance of model features (e.g., random forest)

**Feature Creation**: Create new attributes that can capture the important information in a data set much more efficiently than the original attributes Feature extraction: e.g. : extracting edges from images Feature construction: e.g. : dividing mass by volume to get density Mapping data to new space:e.g. Fourier and wavelet analysis

**Histogram**, **Box Plot**s from high to low(outlier 90th percentile 75th percentile 50th percentile 25th percentile 10th percentile)框代表数据的四分位数范围,Q1-Q3,箱中间线是Q2，下线Q1 - 1.5 \* IQR，上线是Q3 + 1.5 \* IQR **Scatter plots**, **Parallel Coordinates**

Import pandas as pd data = pd.read\_csv(‘iris.txt’,header=None); data.columns=[‘xxx’,’xxx’...] data.head(5)

Col=’xxx’ data[col].mean() .std(). min() .max() .quantile(0.25) data[‘xxx’].value\_counts() data.describe()

From sklearn.preprocessing import StandardScaler features=[‘xxx’...] x=data[features]

scaler = StandardScaler() scaler.fit(x)# get mean & σ from raw data x\_scaled=scaler.transform(x) x\_transformed = pd.DataFrame(x\_scaled) a.k.a x\_scaled=scaler.fit\_transform(x) #the fit and transform can be integrated into a single function "fit\_transform"

data[‘xxx’].hhist(bins=20) data.boxplot() import matplotlib as plt groups = data.groupby("class")

**for** name, group **in** groups: plt.scatter(group['petal length'],group['petal width'], marker="o", label=name) plt.legend() plt.show() from pandas.plotting import parallel\_coordinates parallel\_coordinates(data, 'class') from sklearn.impute import SimpleImputer #this tool fill missing value with mean **import** **numpy** **as** **np** imp = SimpleImputer(missing\_values=np.nan, strategy='mean')

missing\_filled\_new=imp.fit\_transform(ResearchData[['review\_scores\_rating','bedrooms']]) ResearchData[['review\_score\_rating\_filled','bedrooms\_filled']]=missing\_filled\_new

**Lecture 4 Feature Engineering**

Feature: A set of attributes to describe a complex object

Feature Engineering: select/create the most representative features.

Feature Engineering Challenges: 1.Create/Choose Features from Raw Data 2. Duplicate/Redundant Feature 3.Curse of dimensionality(Cause Sparsity)

Trade-off between information, data space, and computation time

**Evaluation Measures for Ranking and Selecting Features**: 1.Information measures 2.Distance measures 3.Dependence measures 4.Consistency measures 5.Accuracy measures

**Classification of evaluation methods**：1.Filter: Without learning algorithm involvement 2.Wrapper: With learning algorithm involvement

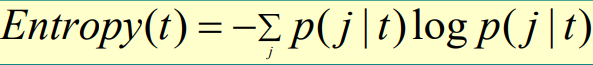
Filter model: class purity, The purer, the better 图示

AI 生成的内容可能不正确。

Measures of Node Impurity

绿色的钟表

AI 生成的内容可能不正确。



Univariate(单变量) Score: −𝐿𝑜𝑔(P𝑣𝑎𝑙𝑢𝑒)

图片包含 文本

AI 生成的内容可能不正确。1. Compute impurity measure (P) before splitting 2. Compute impurity measure (M) after splitting 2.1Compute impurity measure of each child node 2.2Compute the average impurity of the children (M) 3. Choose the attribute test condition that produces the highest gain (Gain=P-M) or equivalently, lowest impurity measure after splitting (M)

图示

AI 生成的内容可能不正确。

图示

AI 生成的内容可能不正确。

**文本

AI 生成的内容可能不正确。**表格, 日历

AI 生成的内容可能不正确。**Entropy**

文本

AI 生成的内容可能不正确。

maximizes GAIN

useInID3 and C4.5 decision tree algorithms

文本, 信件

AI 生成的内容可能不正确。图示

AI 生成的内容可能不正确。

Step 1: use a pre-determined learning model to input the feature

Step 2: use the model performance as feature importance score

钟表的特写

AI 生成的内容可能不正确。**Set 1**：Setosa: 50, Non-Setosa: 10 **Set 2**：Setosa: 0, Non-Setosa: 90

Set1: p(Setosa)=50/60=5/6 p(Non-Setosa)=10/60=1/6

Set2: GINI 0 1 1-1=0 Set W: 2/5 Set2 W: 3/5 GINI =0.1112

**Feature Selection and Dimension Reduction**

**Feature Selection**: chooses an optimal subset Objectives: reduce dim & remove noise, improve res

**Feature reduction** refers to the mapping of the original high-diml data onto a lower-dim space

Unsupervised setting: minimize the information loss Supervised setting: maximize the class discrimination

图示

AI 生成的内容可能不正确。**Optimal Subset**: Five Boolean features C = F1∨F2 F3 = ┐F2 , F5 = ┐F4 Optimal subset: {F1, F2} or {F1, F3} A **Subset Search Problem**: Forward，从空特征集（Empty Feature Set）开始，逐步添加特征 Backward：从完整特征集（Full Feature Set）开始，逐步移除特征

**Feature Reduction: Principal Component Analysis**

**Principal component analysis**: Reduce the dim, Retains most of the sample's information

图示

AI 生成的内容可能不正确。The new variables, called principal components (PCs), are uncorrelated,

图示, 文本

AI 生成的内容可能不正确。

PCA is to find a transformation of data satisfies the following properties: 1.Each pair of new attributes has 0 covariance (for distinct attributes). 2.The attributes are ordered with respect to how much of the variance of the data each attribute captures. 3.The first attribute captures as much of the variance of the data as possible. 4.Each successive attribute captures as much of the remaining variance as possible.

**PCA-Key** **concepts** 1. **Covariance Matrix S**: Given an m by n data matrix D, whose m rows are data objects and whose n columns are attributes, If the data matrix D is preprocessed so that the mean of each attribute is 0, then S = DTD. 2. **Eigenvalues of S**: Let λ1, . . . , λn be the eigenvalues of S. The eigenvalues are all nonnegative and can be ordered such that λ1 ≥ λ2 ≥ . . . λm−1 ≥ λm. 3. **Eigenvectors of S**: Let U = [u(1)...u(n)] be the matrix of eigenvectors of S. These eigenvectors are ordered so that the ith eigenvector corresponds to the ith largest eigenvalue.

**PCA-Data Transformation**:1.The data matrix D′ = DU is the set of transformed data that satisfies the conditions posed above. 2. U matrix is also an [n x n] matrix, turns out the columns of U are the u vectors we want! -So to reduce a system from n-dimensions to k-dimensions, just take the first k-vectors from U (first k columns)

1. Origin dataset [77 x 13] 2. Correlation matrix (it is self-standardized)

3. Get principle components

• Dimension is reduced from 13 to 4

• Only four of them are having Eigenvalue greater than 1.

From sklearn.feature\_selection import SelectKBest from sklearn.feature\_selection import f\_classif,chi2,f\_regression f\_classif分类任务的标签/特征之间的方差分析 F 值, chi2：分类任务的非负特征的卡方统计, f\_regression 回归任务的标签/特征之间的 F 值。

Import seaborn as sns sns.pairplot(iris.hue=”class”) display(X) X\_new=SelectKBest(chi2, k=2).fit\_transform(X,y) #we use chi2 as the criteria and select k=2 best features X\_new.shape # you get a new feature set of dimension k=2 特征选择使用Wrapper**from** **sklearn.ensemble** **import** ExtraTreesClassifier #此类实现了适合多个随机决策树的元估计器 #（又名额外树）在数据集的各个子样本上，以及#使用平均值来提高预测准确性并控制过度拟合。 **from** **sklearn.feature\_selection** **import** SelectFromModel y=y.values.ravel() #please note y is a one-dimensional data, we need to use ravel() to reshape it to (n,) clf = ExtraTreesClassifier （ n\_estimators = 50 ） *#从库中获取模型*clf = clf.fit ( X , y ) # 拟合*数据* clf.feature\_importances\_ #现在我们可以从模型中*获取特征重要性得分*

**Recursive Feature Elimination递归特征消除 from sklearn.feature\_selection import RFE** clf = ExtraTreesClassifier(n\_estimators=50) selection = RFE(estimator=clf, n\_features\_to\_select=2, step=1)

selection.fit(X,y) X\_new=selection.transform(X) display(X\_new[0:5,:]) PCA **from** **sklearn.decomposition** **import** PCA **from** **sklearn.discriminant\_analysis** **import** LinearDiscriminantAnalysis pca = PCA(n\_components=2) z = pca.fit(X).transform(X) from sklearn.feature\_selection import SequentialFeatureSelector

**Lecture 6 Clustering**

Types of Clusterings: **partitional(规则的形状), hierarchical(嵌套), and density-based(密度)**

**proximity or density measure**: Central to clustering; Depends on data and application

**Data characteristics that affect**: Dim(Sparseness); Attributes type; special relation; distribution

**Noise & Outliers**: Often interfere with the operation of the clustering algorithm

**Similarity Metric**: d(x,y)=square√(Σi=1 n(xi-yi)2)



**Attribute Normalization**:

Basic Clustering Algorithms:1.K-means & its variants 2.Hierarchical clustering 3.Density-based clustering

**K-means Clustering**: Partitional clustering approach; Number of clusters, K, must be specified; Each cluster is associated with a centroid (center point); Each point is assigned to the cluster with the closest centroid(until it not change); Simple.

**Step**: Select K points as the initial centroids. **repeat** Form K clusters by assigning all points to the closest centroid. Recompute the centroid of each cluster **until** The centroids don't change

**K-means++**(smartly picking initial centroids): **Core idea**: starts with one random centroid, then chooses others based on their distance from existing ones, spreading them out. **improve**s both the speed and the accuracy of k-means. Step 1. choose first center c1 Step 2. Choose the next center c2 from all possible points with probability D(x’)2/(ΣD(x)2)D(x) denote the shortest distance from a data point x to the closest center we have

**Limitations of K-means**: Sub-optimal(次优) results; 对不同Sizes,Densities, Non-globular(球形) shapes处理不佳; It is sensitive to Outliers

**Bisecting K-means**: Variant of K-means that can produce a partitional or a hierarchical clustering

**Hierarchical Clustering**：Produces nested clusters as a hierarchical tree(dendrogram树状图), Do not have to assume any particular number of clusters(‘cutting’ the dendrogram);有意义taxonomies(分类)

**Types of hierarchical clustering**: **Agglomerative**(凝聚式，自底向上): -Start with the points as **individual** clusters - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left **Divisive**(分列式，自顶向下): -Start with one, all-inclusive cluster -At each step, split a cluster until each cluster contains an individual point (or there are k clusters)

Traditional hierarchical algorithms use a similarity or distance matrix-Merge or split one cluster at a time

**Agglomerative Clustering Algorithm**: **常用**；**Step**: 1. Compute the proximity matrix 2. Let each data point be a cluster 3. Repeat 4. Merge the two closest clusters 5. Update the proximity matrix 6. Until only a single cluster remains; **Key operation** is the computation of the proximity of two clusters -Different approaches to defining the distance between clusters distinguish the different algorithms

**How to Define Inter-Cluster Distance**? MIN;MAX;Group Average;Distance Between Centroids;Other methods driven by an objective function – Ward’s Method uses squared error

日历

AI 生成的内容可能不正确。表格

AI 生成的内容可能不正确。**Single Link(MIN)**:

**Density-based Clustering**: DBSCAN is a density-based algorithm.

Density = number of points within a specified radius (半径是Eps) 落在这个圆内的点的数量就是该点的密度 **Core point:** 一个点被认定为**核心点**，如果在它的Eps邻域内至少有MinPts个点(包括点自身)

**border point**: 不是核心点，但它位于某个核心点的Eps邻域内。边界点位于聚类的边缘区域

**noise point**: 既不是核心点，也不是边界点。这些点被认为是异常值或噪声，不属于任何聚类

原理：随机选择一个未访问的点 确定该点的Eps邻域内的点数：如果≥MinPts，标记为核心点并创建一个新聚类 否则，暂时标记为噪声点 对于每个核心点，递归地找出所有密度可达的点，将它们加入同一聚类 边界点被分配给与之关联的核心点的聚类 重复上述过程直到所有点都被访问

from sklearn.cluster import KMeans Kmeans(n\_clusters=2,max\_iter=50,random\_state=1) k\_means.cluster\_centers\_ from sklearn.cluster import kmeans\_plusplus kmeans\_plusplus(data.values,n\_clusteers=2) from scipy.cluster import hierarchy hierarchy.linkage .dendrogram from sklearn.cluster import DBSCAN DBSCAN(eps=15.5,min\_samples..)

from sklearn.cluster import SpectralClustering **from** sklearn.cluster **import** BisectingKMeans

**Lecture 7 Classification**

Learn a model that maps each attribute set x into one of the predefined class labels y

**Classification Techniques**: **Base Classifiers**:-Decision Tree based Methods -Rule-based Methods -Nearest-neighbor -Neural Networks -Naïve Bayes and Bayesian Belief Networks -Support Vector Machines **Ensemble Classifiers** -Boosting, Bagging, Random Forests

Decision Tree: Train Set induction model deduction test set

Decision Tree Induction: Hunt’s Algorithm (one of the earliest) -CART -ID3, C4.5 -SLIQ,SPRINT

The idea is same. Link the features to the right prediction label

Hunt’s Algorithm: 如果某属性的值一面可以得出，则无需再分，若不可以，则加入其他属性继续分，当然在选择的过程当中我们会采用GINI 作为一个选择的指标，尽可能地保证purity

**Decision Tree** Based Classification： **Adv**: -Inexpensive to construct -Extremely fast at classifying unknown records -Easy to interpret for small-sized trees -Robust to noise (especially when methods to avoid overfitting are employed) -Can easily handle redundant or irrelevant attributes (unless the attributes are interacting) **Disadvantages**: -Space of possible decision trees is exponentially large. Greedy approaches are often unable to find the best tree. -Does not take into account interactions between attributes -Each decision boundary involves only a single attribute

**Build a Regression Tree**: Divide the predictor space into J distinct not overlapping regions R1 ,R2 ,R3 ,…,RJ, make the same prediction for all observations in the same region; use the mean of responses for all training observations that are in the region

**Ensemble Methods**: -Construct a set of classifiers from the training data -Predict class label of test records by combining the predictions made by multiple classifiers

Original Training data -> Multiple Data set -> Multiple Classifiers -> Combine Classifiers

**Types of Ensemble Methods**❖Bayesian ensemble -Example: Mixture of Gaussian ❖ Manipulate data distribution -Example: Resampling method ❖ Manipulate input features -Example: Feature subset selection ❖ Manipulate class labels -Example: error-correcting output coding ❖ Introduce randomness into learning algorithm -Example: Random forests

Bagging: - Sampling with replacement(替换) - Build classifier on each bootstrap(自助) sample

多次生成有放回采样子集，给出判断，投票求和

**Boosting**：An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records. -Initially, all N records are assigned equal weights -Unlike bagging, weights may change at the end of each boosting round,错误的会分到更高的权重，正确的权重降低

卡通人物

AI 生成的内容可能不正确。**AdaBoost Algorithm**: Base classifiers: C1 , C2 , …, CT Error rate:

卡通人物

AI 生成的内容可能不正确。Importance of a classifier:

Base error rate >0.5 will not be used

文本, 信件

AI 生成的内容可能不正确。AdaBoost:

1.用当前样本训练基分类器,然后计算错误率

2.计算分类器的权重αi 错误率越低，分类器权重越大

3.更新样本权重

4.当达到预设的迭代次数t时，或者错误率足够低，或者错误率超过50%时，我们进行条件的终止

最终我们通过加权投票来进行最终的预测

**Support Vector Machine**

文本, 信件

AI 生成的内容可能不正确。Find a linear hyperplane (decision boundary) that will separate the data, Find hyperplane maximizes the margin

Support vector: data point located along the margin boundary. • Support vectors shape the boundaries and decide the SVM model • More non-support vectors will not change the model parameters (the boundary)

Decision boundary depends only on support vectors

文本

AI 生成的内容可能不正确。

Nonlinear SVM

Introduce quadratic mapping (|)(xi)

文本, 信件

AI 生成的内容可能不正确。Kernel function mapping

-可以转化为凸优化问题找到全局最优 -过拟合通过最大化间隔来解决 -Difficult to handle missing values - Robust to noise - High computational complexity for building the model

**Imbalanced classification**：Lots of classification problems where the classes are skewed (more records from one class than another) -Credit card fraud -Intrusion detection

表格

AI 生成的内容可能不正确。

TP True Positive FN: False negative

图片包含 文本

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。FP False Positive TN True Negative

表格

AI 生成的内容可能不正确。



C(i,j): Cost of misclassifying class i example as class j

**Handling Class Imbalanced Problem**: Class-based ordering(RIPPER 稀有类有更高优先级) Cost-sensitive classification(少分多的代高于多分少) Sampling-based approaches

文本, 信件

AI 生成的内容可能不正确。图形用户界面, 文本

AI 生成的内容可能不正确。

**Sampling-based Approaches**:

**Under-sampling** Prototype generation: generate a smaller new set (not selected from the original set) Prototype selection: select samples from the original set Instance Hardness Threshold: measure how difficult it is to classify an instance correctly, remove these hard instances from the dataset.

**Over-sampling** Naïve random over-sampling: generate new samples by randomly sampling with replacement the current available ones SMOTE and ADASYN: duplicating some of the original samples of the minority class ❖ADASYN: focuses more on samples next to wrongly classified samples next to the original ones ❖SMOTE: not make distinction between easy and hard samples to be classified

**Combination of Over- and Under-sampling** Over-sampling may bring noisy samples when interpolating new points Add cleaning methods to clean the space resulting from over-sampling E.g. SMOTETomek and SMOTEENN

**from** **sklearn.model\_selection** **import** train\_test\_split # this function provides a single "Hold-Out" Validation. **from** **sklearn.metrics** **import** accuracy\_score #similar to MAE, we use accuracy\_score evaluation metric. **from** **sklearn.tree** **import** DecisionTreeClassifier **from** **sklearn.ensemble** **import** RandomForestClassifier np.random.seed(1) **from** **sklearn.ensemble** **import** BaggingClassifier **from** **sklearn.ensemble** **import** AdaBoostClassifier **from** **sklearn.metrics** **import** plot\_confusion\_matrix

**from** **sklearn.datasets** **import** make\_classification **from** **numpy** **import** where *# Generate and plot a synthetic imbalanced classification dataset* **from** **collections** **import** Counter *# define dataset*

**from** imblearn.over\_sampling **import** SMOTE **from** sklearn.metrics **import** confusion\_matrix, accuracy\_score, precision\_score, recall\_score, f1\_score **from** sklearn.ensemble **import** AdaBoostClassifier, RandomForestClassifier **from** sklearn.naive\_bayes **import** GaussianNB **from** sklearn.svm **import** SVC

**from** sklearn.neighbors **import** KNeighborsClassifier

**Lecture 8-9 Time Series**

Time Series is a set of observations of a variable at regular time intervals, such as yearly, monthly, weekly, daily, 15 minutes interval, To predict the value of one dependent variable on the basis of time info

Quantitative(定量的): -Extrapolation (or time series) methods -Econometric (or causal) methods

Qualitative(定性) and judgmental: Historical analogy, Delphi Method(专家小组评估), Market Research

**Quantitative Forecasting**: Using past to predict future

**Time Series Patterns**: **Horizontal Pattern**(数据在平均值附近波动) **Trend Pattern**(随时间呈现持续的上升或下降趋势) **Seasonal Pattern**(在固定周期内呈现规律性的波动), **Seasonal+Trend**(数据既呈现长期的上升/下降趋势，又在固定周期内有规律的季节性波动), **Cyclical Pattern**(在较长时间跨度内呈现的波动，周期长度不固定，且与经济、商业或社会因素相关)

**Statistical Methods**: Time Series Component: The different components reflect the impact of different factors. The overall time series is the combination of different components.

Additive Model:Yt=Tt+St+Ct+It Multiplicative model (linear in log form): Yt=TtStCtIt

Tt = Trend value at period t St = Seasonality value for period t Ct = Cyclical value at time t It = Irregular (random) value for period t

**Simple Moving Average**: **Average random fluctuations in a time series** to infer short-term changes in direction **Assumption**: future observations will be similar to recent past **Forecast** = average of most recent k observations Ft+1=(Yt+...Yt-m+1)/m

**Weighted Moving Average**: A natural extension to moving average. -Weight the most recent k observations, with weights that add to 1.0 -**Higher weights on more recent observations generally provide more responsive forecasts to rapidly changing time series** -Rational: recent observation are more similar as what will happen next

墙上的钟表

AI 生成的内容可能不正确。

**Exponential Smoothing** -Weights decay exponentially -Sum of weights approach 1

**Exponential smoothing model**:



last prediction + adjustment on error

- Ft+1 is the forecast for time period t+1, - Lt is the level for period t, (L1=Y1 ) - Yt is the observed value in period t, and α is a constant between 0 and 1, called the smoothing constant.

文本

AI 生成的内容可能不正确。文本, 信件

AI 生成的内容可能不正确。Higher smoothing constant (higher weights on more recent observations) provide forecasts that respond more rapidly to changes in series.

**Holt-Winter** **Simple exponential smoothing + trend + seasonality** • Current level (𝑙𝑡 ), similar to the Holt-Linear • Current trend (𝑏𝑡 ), similar to the Holt-Linear • Current seasonality (𝑠𝑡 )

图示, 文本, 信件

AI 生成的内容可能不正确。

**ETS: error, trend, seasonality**

Time series data

Trend Seasonality Random

**Regression on Linear Trend**

Yt​=a+bt+et​

Yt时间t的观测值 a截距，表示时间 t=0 t = 0 t=0 时的期望值 b 斜率，表示每个时间周期的期望变化量 et随机误差项，反映模型未解释的波动。

**Regression on Exponential Trend**

适用于时间序列以**固定百分比**（而非固定量）变化的情况，表现为增长或衰减速度随时间加速或减缓。 c：常数（类似截距）。 b：增长率参数。 ut​：乘法误差项（随机波动） ebt：指数增长/衰减项

可以化为线性形式

**Regression on Seasonal Effect**

A regression approach to forecasting seasonal data uses dummy variables for the seasons.

**Testing for Randomness**

核心思想：在时间序列预测中，目标是构建一个模型，尽可能捕捉数据中的所有可预测成分（趋势、季节性、周期性等），使残差（Residuals）表现为**随机噪声（Random Noise）**。如果残差不是随机噪声，则说明模型遗漏了某些可建模的模式，需要进一步改进。

**Typical Forecasting Equation** Yt=Fitted Value +Residual et=Yt-Yhatt **观测值和预测值的差异**

文本

AI 生成的内容可能不正确。希望et是随机噪声，无趋势，季节性，自相关；均值为0，方差稳定

**Autocorrelations**衡量时间序列在不同时间滞后（Lags）之间的相关性,揭示了时间序列中相邻观测值之间的关系,创建滞后变量（Forming Lagged Variables）,计算相关系数(皮尔逊)

**The Random Walk Model**一种特殊的时间序列模型，时间序列本身非随机（可能有趋势或自相关），但其\*\*一阶差分（Differences）\*\*是随机的 Yt​=Yt−1​+m+et​ Yt Observed value at the time period of t Yt-1- Observed value at the time period of t-1 M - the average differences et - a random time series.

图形用户界面, 文本, 应用程序, 聊天或短信

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。ARIMA

文本

AI 生成的内容可能不正确。AR

文本, 信件

AI 生成的内容可能不正确。

文本, 信件

AI 生成的内容可能不正确。

文本, 信件

AI 生成的内容可能不正确。MA

图示

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。ARIMA

**Stationarity**: A process is stationary if it has ü No trends (i.e., constant mean) ü Constant variance ü No seasonality **Stationarity test** • Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test ü Null hypothesis: data is stationary ü If p-value < 0.05, then reject the null, i.e., the data is non-stationary, at a confidence level of 95%. Otherwise, we accept that the data is stationary at a confidence level of 95%.

图形用户界面, 文本, 应用程序, 聊天或短信

AI 生成的内容可能不正确。图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

图片包含 文本

AI 生成的内容可能不正确。电脑屏幕的照片上有字

AI 生成的内容可能不正确。

文本, 信件

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。文本, 信件

AI 生成的内容可能不正确。

**from** **statsmodels.tsa.exponential\_smoothing.ets** **import** ETSModel *#key library*

**from** statsmodels.tsa.ar\_model **import** AutoReg

**from** **scipy.stats** **import** boxcox

**from** **statsmodels.tsa.stattools** **import** kpss

**from** **statsmodels.tsa.arima.model** **import** ARIMA

**from** **statsmodels.graphics.tsaplots** **import** plot\_acf, plot\_pacf

**Lecture 10 Model Assessment**

图示

AI 生成的内容可能不正确。-Model selection - Parameter tuning - Avoid misleading evaluation result Under/Over fitting Imbalanced dataset - Adaptive learning - Pipeline evaluation…

Supervised Learning:

**Classification** • Classification matrix (also known as confusion matrix) • Precision • Recall • F-measure • Misclassification errors

图表

AI 生成的内容可能不正确。图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。**Prediction** • MAE or MAD (mean absolute error/deviation) • Average error • MAPE (mean absolute percentage error) • RMSE (root-mean-squar ed error) • Total SSE (total sum of squared error)

**Metrics for imbalanced data** Lots of classification problems where the classes are skewed (more records from one class than another) -Credit card fraud -Intrusion detection -Defective products in manufacturing assembly line

Validation Data

Direct measurement of training error using the training data that induces the learning model is misleading

Under-fitting

Large training error

Over-fitting

Small training error Large testing error

**Hold-out Method**使用训练集拟合模型。 使用验证集（独立于训练集）评估模型的泛化误差。 选择在验证集上表现最好的模型。Drawbacks: May not have enough data to afford setting one subset aside forgetting a sense of generalization abilities Validation error may be misleading (bad estimate of generalization error) if we get an “unfortunate" split Limitations of hold-out can be overcome by a family of random sub-sampling methods at the expense of more computation.

**Crossvalidation** : 1) CreateK-fold partition of the dataset. 2) Form K hold-out predictors, each time using one partition as validation and rest K-1 as training datasets. K predictors for each model class: fi...fk,然后看平均损失和损失的标准差

CV with random subsampling: 随机抽取一个固定比例 α的数据作为验证集,剩下的作为一个训练集训练一个预测器，然后最小化经验风险拟合，重复k次

**Rolling Forecasting**: Multi-step forecasting: we can put more records in the validation sets, e.g. next three days. • Fixed-sized training data: We can forget the oldest training records, e.g. keep a fixed length for training records

**Model Selection** – Early Termination Stop training as soon as the validation error reaches the min value

过拟合: If training data is under-representative, testing errors increase and training errors decrease on increasing number of nodes • Increasing the size of training data reduces the difference between training and testing errors at a given number of nodes

Hyperparameter Tuning: GridSearchCV & RandomizedSearchCV | Hyperparam Search + Cross-validation

Selection: **Nested-Cross Validation** How about just using a standard CV for

hyperparameter tuning? This can lead to overly optimistic estimates of the model’s generalization error

because the model has indirectly “seen” the test data during hyperparameter tuning, i.e., data

leakage. Nested cross-validation addresses this issue by separating the data used for hyperparameter tuning from the data used for performance evaluation.

图示

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

7. Finally, the mean and standard deviation of the model performance is computed by taking

all of the model scores calculated in step 5 for each of the K models.

8. Step 3 to Step 7 is repeated for different values of hyperparameters.

9. Finally, the hyperparameters which result in the most optimal mean and the standard

deviation of model scores get selected.

10. The model is then trained using the entire training data set (step 2) and the model

performance is computed on the test data set (step 1).

**Regularization**:

文本

AI 生成的内容可能不正确。We make some constraints when training our model Prevent an unnecessary complex model

文本

AI 生成的内容可能不正确。

图示

AI 生成的内容可能不正确。手机屏幕截图

AI 生成的内容可能不正确。Pessimistic Error

决策树 预剪枝 后剪枝

图形用户界面, 文本

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

图示

AI 生成的内容可能不正确。

**Clustering Evaluation**

Inter-cluster distances are maximized Intra-cluster distances are minimized

文本, 信件

AI 生成的内容可能不正确。

Kmeans

文本

AI 生成的内容可能不正确。

**Evaluate of BDA pipeline**

一、A/B 测试概述

定义：A/B 测试是将用户随机分配到两个版本（控制组 A 和实验组 B），通过比较关键性能指标（Key Performance Indicators, KPIs）来评估哪个版本更优。

目标：确定哪种设计或策略在目标指标（如点击率、转化率、收入等）上表现更好。

随机化：通过随机分配用户（随机评估者），确保实验结果不受偏差影响。

二、A/B 测试框架

A/B 测试通常遵循以下步骤，确保实验设计科学且结果可靠：

确定关键性能指标（KPI）及测试变体：

KPI：明确要跟踪的指标，例如网页点击率、购买转化率、用户停留时间等。

变体：定义 A 和 B 两个版本，例如网页按钮颜色（红色 vs. 蓝色）或定价策略（$10 vs. $12）。

例子：测试新网页布局是否提高转化率，KPI 是转化率，变体是旧布局（A） vs. 新布局（B）。

提出假设：

假设实验组（treatment）比控制组（control）表现更好，并说明原因。

例子：假设新网页布局（B）会提高转化率，因为它更简洁，用户体验更好。

假设基于理论、用户反馈或初步数据分析。

识别可能影响 KPI 的关键变体：

确定可能导致 KPI 差异的因素，例如：

设计的视觉元素（颜色、字体、布局）。

功能变化（添加新功能、简化流程）。

用户交互方式（提示信息、按钮位置）。

例子：新布局的按钮更大、更显眼，可能是转化率提升的关键。

运行随机化实验：

使用随机化方法（如随机分配用户到 A 或 B 组）进行实验。

常见统计方法：

DID（Difference-in-Differences）：比较实验前后两组的 KPI 变化。

T 检验：检验 A 和 B 组 KPI 的差异是否显著。

例子：将用户随机分为两组，分别体验旧布局和新布局，记录转化率。

评估 KPI 差异的显著性：

使用统计检验（如 T 检验、Z 检验）评估 A 和 B 组 KPI 差异是否具有统计显著性（通常 p 值 < 0.05）。

考虑置信区间和效应大小，确保结果可靠。

例子：如果新布局的转化率显著高于旧布局（p < 0.05），则认为差异显著。

得出结论：

根据实验结果，判断变体是否显著改善 KPI。

如果变体表现更好，考虑推广；否则，保留控制组设计或尝试其他变体。

例子：新布局显著提高转化率，决定全面采用新布局。

三、2 阶段分析 vs. 趋势分析

2 阶段分析（Two-Stage Analysis）：

第一阶段：运行 A/B 测试，收集初步数据，评估 KPI 差异。

第二阶段：深入分析结果，识别影响 KPI 的具体因素（如用户细分、时间趋势）。

优点：提供更细致的洞察，适合复杂场景。

趋势分析（Trending Analysis）：

持续监控 KPI 随时间的变化，观察变体的长期效果。

适合动态环境（如季节性变化）或需要长期验证的场景。

选择依据：

2 阶段分析适合快速决策和深入探索。

趋势分析适合长期优化和动态调整。

四、评估优秀设计（BDA 设计中的变体）

在大数据分析（BDA）或机器学习模型设计中，A/B 测试可以用于评估不同模型或处理流程的效果。设计中的变体包括：

属性选择/创建：选择不同的特征集或创建新特征。

预处理质量：比较不同数据清洗或标准化方法的表现。

混合模型：测试单一模型 vs. 集成模型（如随机森林 vs. 梯度提升）。

模型结构设计：测试新模型架构或超参数配置。

消融测试（Ablation Test）：逐一移除模型组件（如某层神经网络），评估其对性能的影响。

其他：如优化算法、损失函数等。

评估方法：

使用 A/B 测试框架，比较不同设计的 KPI（如预测准确率、F1 分数）。

结合统计检验（如 T 检验）评估差异显著性。

可视化结果（如 ROC 曲线、误差分布）辅助决策。

**from** matplotlib.pylab **import** rcParams

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.metrics **import** mean\_absolute\_error

**from** sklearn.preprocessing **import** MinMaxScaler,Normalizer

**from** sklearn.datasets **import** load\_breast\_cancer

**from** sklearn.tree **import** DecisionTreeClassifier

from sklearn.tree import plot\_tree

**from** sklearn.svm **import** SVC

**from** sklearn.model\_selection **import** GridSearchCV

**from** sklearn.metrics **import** classification\_report

**from** sklearn.datasets **import** make\_moons, make\_circles, make\_classification

**from** sklearn.neural\_network **import** MLPClassifier

**from** sklearn.neighbors **import** KNeighborsClassifier

**from** sklearn.gaussian\_process **import** GaussianProcessClassifier

**from** sklearn.gaussian\_process.kernels **import** RBF

**from** sklearn.naive\_bayes **import** GaussianNB

**from** sklearn.discriminant\_analysis **import** QuadraticDiscriminantAnalysis

**from** sklearn.metrics **import** classification\_report *# this library directly generates precision, recall, f-measure*