Logistic Regression.

Day-21.

Overview of Linear Regression →

there is depend^n & indep^n variable.

Linear Reg. → dependence technique.

correlation → interdependence technique.

there is no dependent & no independent var.

Statistical Variable.

(response var)

Depend^n — O/P var. — Ex. crop production

Indepn^n. — Ex. rainfall, fertilizer usage.

Purpose of LR.

↳ it is used to find causal effect relationship b/w independ^n var on depend^n var.

(How much one var. is going to affect another var. → Regression)

Indep^n affect depend^n going to

On regression,

deciding factor → r^2

(When both var are continous:
dep^n & indep^n → we go for Linear Regr.)

Regression

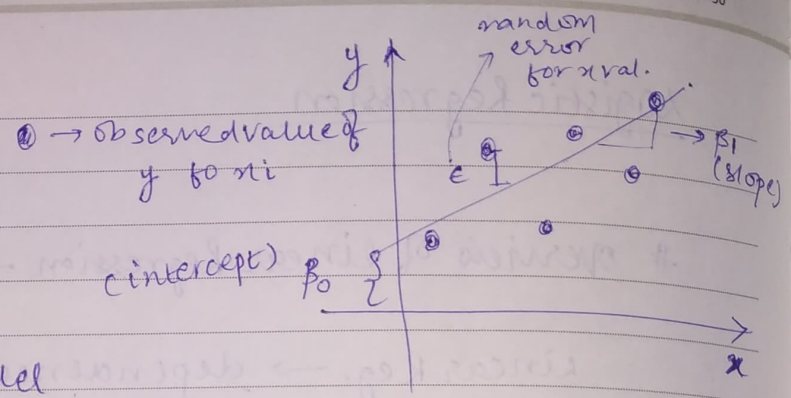
(one Indep^n) → Simple Regr. → Linear / Non-linear

(More than 1 indep^n var.) → Multiple Reg. → Linear / Non-linear.

15 Friday April 2011

Week 15 ▪ 105-260

(DV) $y = \beta_0 + \beta_1 x + \epsilon$ → random error.
intercept, coef, indepn var. (IDV)



Conducting Regr. Analysis.

1. Plot Scatter Diagram
2. Formulate General Model
3. Estimate Parameter
4. Determine strength & significance of Association.
5. Test the for Significance.

Ex.
- R^2 → (58.05%)
- infer p-value (p < 0.05 ie it is affecting y)
- frame eqn. (coeff + intercept)

(formed this eqn by stats).

Ex housepice = 98.2483 + 0.10977 (sq. feet)

const. ie no one can construct 0 square feet. min. amount you have to pay.

on an avg. value (of house increases ⅔ by 0.1097 (1000) = $109.7 you need to pay for each one sq. foot of size.

for an additional sq. feet, you have min 109.7 dollars.

$y = \beta_0 + \beta_1 x + \epsilon$

const, avg. (coef). (1/2/3) sq. feet.

Ex 95% confidence interval)

CI: upper 95% 0.18580 × 1000 = $185.80 max you have to pay this for house.

confidence interval
C.I
lower 95% 0.03374 × 1000 = $33.74 min you have pay this

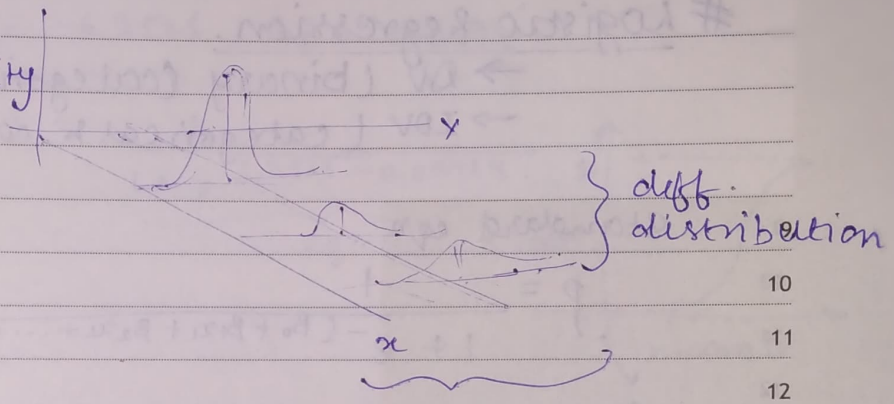
- This is for simple linear regression

o Multiple linear Regression.

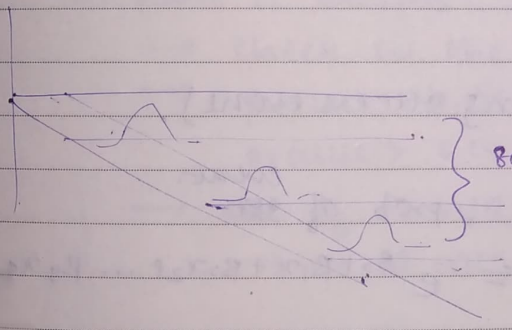
Regression Assumption.

① Heteroskedasticity. → This test should pass.
(different spread.)

all the IDV should be distributed the different way.



homoskedasticity.
same spread (equal)



This test should be passed.
(skewness, kurtosis etc.)

10 11 12 1 2 3 4 5 6

Sunday 17
107-258

② Multicollinearity.
IDV
whatever you are taking & all the IDV are correlated with each other is called Multicolinearity.

When you are going for Reg. ~~Mod~~, Multicollinearity test should fail.
(The IDV we are taking should not correlated with each other)

- Heteroskedaaity → Pass
- Autocorrelⁿ → Pass
- Multicollinearity → Fail

③ Autocorrelation

↳ Whatever DV we are taking in LR, The DV & IDV are correlated with each other.

Logistic Regression.

→ DV (binary (categorical)) — only 2 (0 or 1, Yes/No...)

→ IDV (categorical & continous.)

Standard eqn:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_q x_q)}}$$

$0 \le p \le 1$

p → prob. of belonging to class 1.

p → prob. of getting outcome.

(p prob. of getting success event).

Odds ratio:

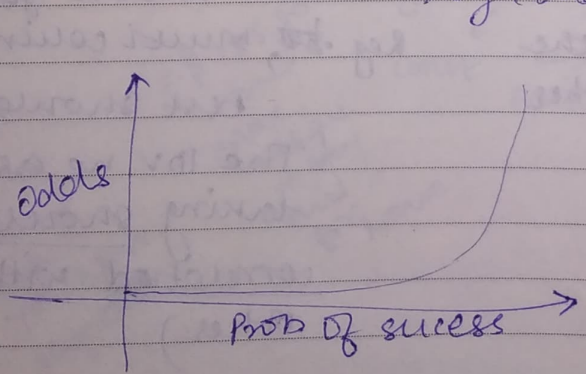
$$\text{odds} = \frac{p}{1-p} = e^{\beta_0 + \beta_1 x + \beta_2 x_2 + \dots \beta_q x_q}$$

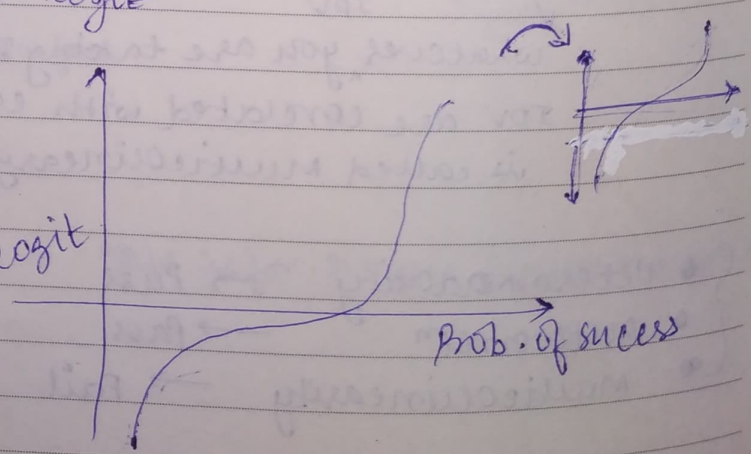
p → prob of event.

log on both sides.

$$\log(\text{odds}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_q x_q$$

$$\log(\text{odds}) = \text{logit}$$



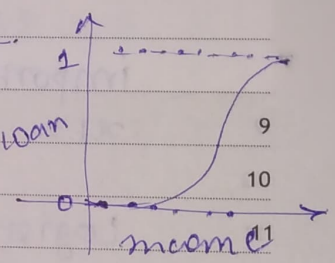


Single predictor Model → 1 DV, 1 IDV
Multiple predictor Model → 1 DV, more than

Ex. $P(\text{Loan} = \text{Yes} \mid \text{Income} = x) = \frac{1}{1 + e^{-(B_0 + B_1 x)}}$

after fitting, $b_0 = -6.3525$, $b_1 = 0.0392$

$P(\text{Loan} = \text{Yes} \mid \text{Income} = x) = \frac{1}{1 + e^{6.3525 - 0.0392x}}$ (eqn)



* Multiple predictor

Ex for prediction of loan -
we have 12 colums:
→ check all the pvalues of all coln.
→ pvalue < 0.05 (significant coln)
pvalue > 0.05 (insignificant coln)
→ out of 12, 10 coln have pvalue < 0.05
→ in 10 coln, pvalue = 0. (Most significant coln for predicting loan)

Conclusion

P < 0.5 ✗ (Prob of getting loan is less.)
P > 0.5 ✓ (Prob. of getting loan is high).

* When we go for logistic reg., multicolinearity should fail.

* Both linear & logistic regression, both are used to find casual relationship b/w variables.

• It can be used for explanatory task (profiling) or predictive tasks (classification).

Practical Implementation.

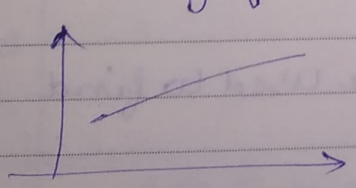
```
df = pd.read_excel('LogisticRegression.xlsx', sheetname=0)
df.head()

y = df.Admit
x = df[['GRE', 'GPA', 'Rank']]

import statsmodel.api as sm
x1 = sm.add_constant(x)

Logistic = sm.Logit(y, x1)
result = Logistic.fit()
result.summary()
```

→ takes value from $-\infty$ to $+\infty$

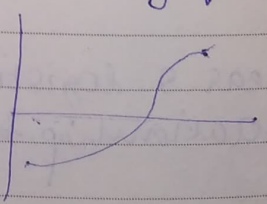
↳ check pvalue < 0.05 or not?
if $p < 0.05$ (var. are significantly imp.)

$p > 0.5$ (prob. of admission is high)
$p < 0.5$ (prob. of admission is less)

Linear Regression

1. DV, IDV → continous
2. deciding factor R^2
3. 
4. $y = \beta_0 + \beta_1 x_1 + \dots \beta_n x_n$

Logistic Regression.

1. DV → 0/1,
   IDV → categorical / cont.
2. deciding factor → pvalue.
3. 
4. $p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots \beta_n x_n)}}$

Assignment → BankLoanModeling, AttritionProject (Apply Logistic to both datasets.)

Thursday

April 2011

21

Week 16 ▪ 111-254

(5) condn:
- Heterosk
- Multicolinearity (x) fail
- Autocorelation

(6) Predictive algorithm.

(5) condn.
- Multicolinearity
- variable selection.

(6) Classification algorithm.

HW. Real Estate Analysis.

* LinearRegrn → Build 5 model (inside LR)
  - Price V/s sqft_living
  - Price V/s bedrooms
  - Price V/s bathrooms
  - Price V/s floors
  - Price V/s (all IDV). → Multiple regression.

9 10 11 12 1 2 3 4 5 6

Day-22.

Correlation → To find reln. b/w 2 var. only.
Regression → To find causal effect reln. b/w IDV & DV
1. Linear: Simple & Multiple.
2. Logistic: Singe predictor model & Multiple predictor model.

most important

Decision Tree & Random forest.

Classification Technique	Dependent Var.	IDV	Purpose of Algorithm
1. Decision Tree	categorical	categ. + continous	It is used to classify records in pictorial format with help of GINI index.

2. Random Forest. categorical cat^r. + continous

ensemble.
It is a sample DT. algo, also used to find imp. var. for DT.

Decision Tree Analysis:

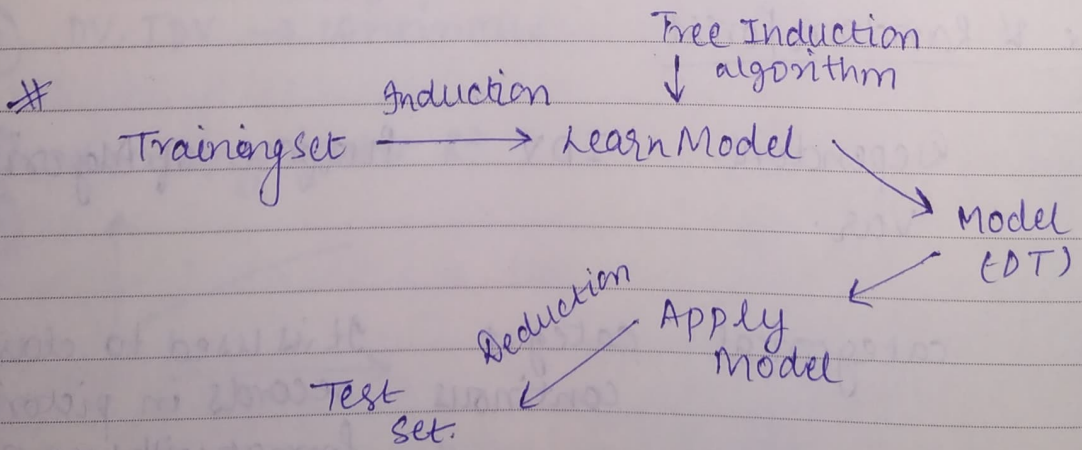
1. Decision Tree Classification
2. Decision Tree Prediction
3. Variable Type & split Type
4. Measure of Node Impurity.
5. Decision Tree Induction Algorithm - ID3, CART
6. Practical Issues of Decision Tree
7. Ensemble method - Random Forest.

only algo → DT → used as classification as well as prediction.
(target var → discrete (categorical))

- We can generate more than 1 DT for a dataset. But which node is having lesser GINI index, that node is taken as classification.

Based on GINI index - we classify DT.

#



– How to specify Test condⁿ ?

Depends on attribute type

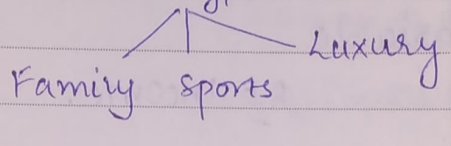
Nominal
Ordinal
Continous

Depends on no. of ways to split

2 way split (Binary split)
Multi-way split

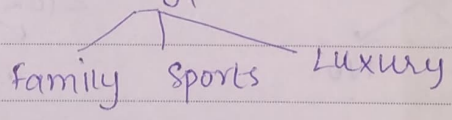
• Nominal Scale → used for Identification purpose.
defining in distinct (different) way.

Ex.



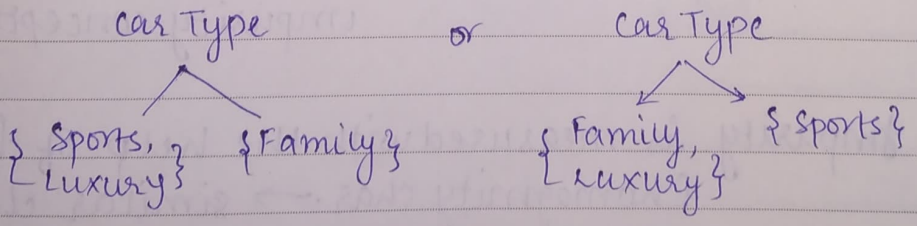
Multiway split → more than 2 way is called multiway split
→ Use as many partitions as distinct values.

Ex.



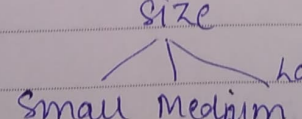
Binary split → only 2 categories are these.
(Divides values into 2 subsets, Need to find optimal partitioning)

Ex



• Ordinal Scale → ranking the object.

Ex.



Ordinal Multiway → 2 or more partitions.

Ex. Size → small, medium, large

Ordinal Binary split → 2 categories/subset.

Ex. Size → {small, medium} {large}

or Size → {med, large} {small}

etc..

- Continous Variable

variables are not defined pre.

It can be any value

Ex Income.

Income >80k → Yes / No

- Binary split

Income → <10k, 10k,25k, 25k,50k, 50k,80k, >80K

- Multiway split

Which is best split?

We can't determine. ∴ uses node impurity concept.

- Node impurity is measured with the help of homogeneous (homoginity char. → similar characteristics).

Ex. C0 → 5, C1 → 5 } hetrogeneous.

Non-homogeneous

(High degree of impurity)

Ex. C0 : 9, C1 : 1 } homogeneous

Low degree of impurity

We can measure node impurity →
- Gini Index
- Entropy.
- misclassification error.

GINI Index:

$$(t) = 1 - \sum_j [p(j/t)]^2$$

ranges b/w 0 to 0.5

prob. of geting specific event / total no. of obs.

- smaller gini index → more purity (similarity is more)

Ex:

C1 : 0	C1 : 1	C1 : 2	C1 : 3
C2 : 6	C2 : 5	C2 : 4	C2 : 3
Gini = 0.00	Gini = 0.278	Gini = 0.444	Gini = 0.500

(homogenity is high).

$P(C1) = 0/6 = 0$

$P(C2) = 6/6 = 1$

$Gini = 1 - P(C1)^2 - P(C2)^2$

$= 1 - 0 - 1$

$= 0$

Ex Multiway Split.

	Family	Sport	Luxury
C1	1	2	1
C2	4	1	1

Gini = 0.393.

- use count matrix to make decision

Entropy:

$$(t) = -\sum_j p(j/t) \log p(j/t).$$

ranges from 0 to 1.

Ex: C1 : 1, C2 : 5

$P(C1) = 1/6$ $P(C2) = 5/6$

$Entropy = -(1/6)\log_2(1/6) - (5/6)\log_2(1/6)$

$= 0.65$

27 Wednesday April 2011

Week 17 ▪ 117-248

entropy → ↓ (low) homogenity is more.
entropy → high (↑) homogenity is ~~strigth~~ less

April 2011

Wk	Mo	Tu	We	Th	Fr	Sa	Su
13					1	2	3
14	4	5	6	7	8	9	10
15	11	12	13	14	15	16	17
16	18	19	20	21	22	23	24
17	25	26	27	28	29	30	

entropy is minimized when all values of target are same.
entropy is maximized when there is equal chance of all values for target attribute. (ie result is random).

purity.

0 → impurity is less (homogenity is more)
high → impurity is high (purity is less).

Mis

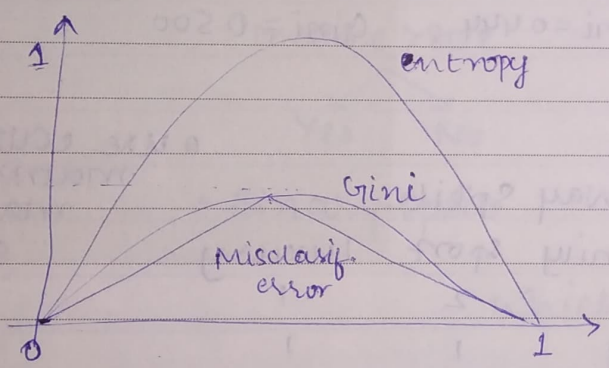
- Classification Error:

Error (t) = $1 - \max_i P(i/t)$.

ranges from 0 to 0.5

Ex

C1 : 1 P(C1) = 1/6 P(C2) = 5/6
C2 : 5 Error = 1 − max (1/6, 5/6)
= 1 − 5/6
= 1/6



Out of these 3,
Gini index gives higher accuracy.

- Stoping Criteria for Tree Induction.

• If node is having same characteristics, we can terminate node.

ppt.

1.) stop expanding a node when all the records belongs to same class.
2.) Stop expanding a node when all records have similar attribute values.
3.) Early termination.

May 2011

Wk	Mo	Tu	We	Th	Fr	Sa	Su
17/22	30	31					1
18	2	3	4	5	6	7	8
19	9	10	11	12	13	14	15
20	16	17	18	19	20	21	22
21	23	24	25	26	27	28	29

- Binary split → only 2
- Categorical split → More than 2 ie (Low, Med., High)
- continous split
  ↓
  Income, Age

Thursday
April 2011
28

Week 17 ▪ 118-247

ID3 (Iterative Dichotomous).
two ie it will do binary split

CART (Classification & Regression Trees).
- Developed by:
  Breiman, Friedman, Olshen,
  Stone in early 80's

Binary → ID3.
cart { categorical, continous

Watch again.
1:55:06 time.

Continous target: → Use sum of squased errors.
(Regression Trees)

categorical target → choice of entropy, Gini measure, 'error rate', information Gain, 'twoing' splitting rule.
(Classification Trees)

Difficulties in DT.

- Underfitting & overfitting.
- Missing values
- costs of classification.

Day-23.

- Overfitting → If you are ting more var. (have more IDV) it is difficult to classify records.
  We go for random forest algo.

Q: How to avoid overfitting? →

1.) Stop growing when data split not statistically significant
2.) Grow full tree then post-prune
3.) Use ensemble of DT.

• Pre-pruning → Before generation of DT, you can resolve overfitting

Random Forest → ~~a sample~~ ensemble DT.
→ identify imp. indepⁿ var. for DT algo.

• Pruning (Minimizing tree).
cut the tree.
(control growing of tree - either it is depth wise or width wise)

Two types of Pruning.

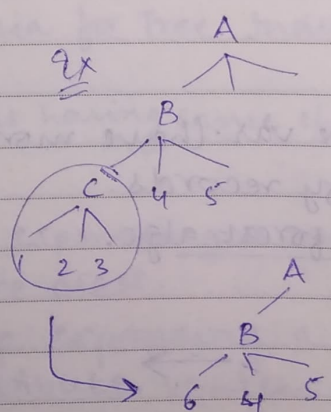
Pre-Pruning (forward pruning).
Before generation of DT, if you are minimizing the tree.
• We go for Random Forest

Post-Pruning (backward pruning)
once tree is generated, how to classify the records.
• Subtree Replacement
• Subtree Raising.

(it increases the accuracy of tree).

Subtree replacement.

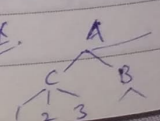
Ex



If tree is growing more, identify which node is growing more & that node is summarized.

If tree is raising more, the entire node is attached to main node.

Ex.



Handling Missing Value.

- Missing value affect DT construction in 3 different ways.
  1.) Affect how impurity measures are computed
  2.) Affect how to distribute instance with missing value to child nodes.
  3.) Affect how a test instance with missing value is classified.

If missing data is less, we can substitute it

If missing data is more, we can del entire col.

Ex 100 obsr.

1. 15-20 missing → fill it
2. 50-80 missing in a coln (del. that coln).

how to fill missing value.

- var. is continous → fill by mean (avg.)
- var. is categorical → fill by mode.

9 10 11 12 1 2 3 4 5 6

- Fast at clasifying unknown record
- Easy to interpret

computation

Strengths (Advantage of DT).

- Use for generating rules.
- Perform classification without much
- IDV can be categorical or continous.
- Provide clear inoication of which field are most imp. for prediction or classification.

Weaknees (Disadvantage of DT)

- Prediction / Depn var. (DV) should be categorical. (Not suitaible for continous var.)
- Perform poorly with many class (categories), & small data.
-

continous var → changed to ranges.
ex DV → Sales / Income
<10K, 10-15K, 15-20K...
converted to categories then apply Regressor DT.

May
Wk	Mo	Tu	We	Th	Fr	Sa	Su
17/22	30	31					1
18	2	3	4	5	6	7	8
19	9	10	11	12	13	14	15
20	16	17	18	19	20	21	22
21	23	24	25	26	27	28	29

2011

Random Forest.

- which IDV is having more depth in tree.
  more easy to classify.
- find most imp. variable in DT.

Practical Implementation.

Dataset used → Titanic.

```
train = pd.read_csv('train.csv')
```

ex if Age coln has missing value,
ex train['Age'].mean # 32.69985

```
new_age = np.where(train['Age'].isnull(), 32, train['Age'])
train['Age'] = new_age.
```

isnull() → check null value
32 → Replace by 32
train['Age'] → Put in this coln.

text → category.

- LabelBinarizer (only 2 category).
- LabelEncoder (2 category, or more.)

```
from sklearn import tree
from sklearn import preprocessing.

lab = preprocessing.LabelEncoder()
sex = lab.fit_transform(train['sex'])

model = tree.DecisionTreeClassifier()
model.fit(X=pd.DataFrame(sex),
          y = train['survived'])
```

DV is categorical. (if it is continous go for DTRegressor)

DT always support dot file.

```
with open('DT1.dot','w') as f:
    f = tree.export_graphviz(model, feature_names=['sex'],
                             out_file = f);
```

('DT1.dot' — user defined name of file; 'w' — write; IDV; export_graphviz — interface; out_file = f — where to store this model.)

copy the code from the file (opened in Notepad)

web.graphviz.com (interface)
→ use for generating DT.
→ paste that code in this site & generate graph.

```
model2 = pd.DataFrame([sex, train['Pclass']]).T
model.fit(X = model2, y = train['survived'])

with open('DT2.dot','w') as f:
    f = tree.export_graphviz(model, feature_names = ['sex','Pclass'],
                             out_file = f);
```

Day-24.

```
model3 = pd.DataFrame([sex, train['Pclass'], train['Age'],
                       train['Fare']]).T
model = tree.DecisionTreeClaseifier(max_depth=8)
model.fit(X = model3, y = train['survived'])

with open('DT3.dot', 'w') as f:
    f = tree.export_graphviz(model, feature_name = ['sex','Pclass',
                             'Age','Fare'], out_file = f);
```

if you add more IDV, tree gets complicated.

```
model.score(x=models, y=train['survived'])      # 89% accurate
```

Q. Accuracy is 89% but not more than 90% why?
To increase accuracy, we need to find imp. variable by Random Forest then again predict. The accuracy will increase.

- Till now we do classification, Now We will predict by DT.

```
test = pd.read_csv('test.csv')
new_age = np.where(test['Age'].isnull(), 28, test['Age'])
test['Age'] = new_age
sex = lab.fit_transform(test['sex'])        # label encoded: text → No.
feature = pd.DataFrame([sex, test['Pclass'], test['Age'],
                        test['Fare']]).T
pred = model.predict(x=feature)

output = pd.DataFrame({'PassengerId': test['PassengerId'],
                       'survived': pred})

output.to_csv('output.csv', index=False);
```

output → DataFrame converted to csv format.

- Random Forest.

```
from sklearn.ensemble import RandomForestClassifier
train.columns                    # Titanic train dataset (imported already).
train['sex'] = lab.fit_transform(train['sex'])      # label encoder.
```

```
train['Embarked'] = lab.fit_transform(train['Embarked'])

rf_model = RandomForestClassifier(n_estimators = 1000,   → iterations.
                                  max_features = 2,   → Binary split it will do.
                                  oob_score = True)
                                  out of bag score,
                                  Based on DT, each & every node, we can find accuracy.
features = ['sex', 'Pclass', 'sibsp',
            'Embarked', 'Age', 'Fare']

rf_model.fit(X=train[features], y=train['Survived'])
print('OOB Accuracy')
print(rf_model.oob_score_);          #→ 80%.

Q. Find imp. features:

for feature, imp in zip(features, rf_model.feature_importances_):
    print(feature, imp);
```

Sex - 0.268 ✓
Pclass - 0.088 }
Sibsp - 0.052 } X
Embarked - 0.034 }
Age - 0.970 ✓
Fare - 0.285 ✓
accuracyscore.

out of 6 :
• these 3 are imp. features

HW
• With these 3 variable, find accuracy. (prediction)
write the rules also.
• Attrition → DT, RF
• Bankloan → DT, RF

Day-25.

Naiive Bayes

→ classification Technique

→ classify records with help of probability.

Classification Algorithm.

Model a classific^n rule directly.	Model the prob. of class membership given I/P data	Make a probabilistic model of data within each class.
Ex. K-NN, SVM, DT, perceptron	Ex. Logistic Regress^n, perceptron with cross-entropy cost	Ex. Naive Bayes, Model Based classifier.

1, 2 are ex. of discriminative or conditional classification

2, 3 are both ex. of probabilistic classification

3 is an ex. of generative classification

- Naiive Bayes.
  - → Both DV, IDV are categorical
    - Numerical value must be binned & converted to categorical.
  - → Can be used with very large datasets.

Probability Basics.

- Prior Prob. : $P(X)$
- conditional Prob: $P(X_1 | X_2)$, $P(X_2 | X_1)$
- Joint prob. : $X = (X_1, X_2)$

  $P(X) = P(X_1, X_2)$

- Relationship : $P(X_1, X_2) = P(X_2|X_1) \cdot P(X_1) = P(X_1|X_2) \cdot P(X_2)$
- Independence: $P(X_2|X_1) = P(X_2)$, $P(X_1|X_2) = P(X_1)$

$$P(X_1, X_2) = P(X_1) \cdot P(X_2)$$

- **Bayesian Rule.**

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

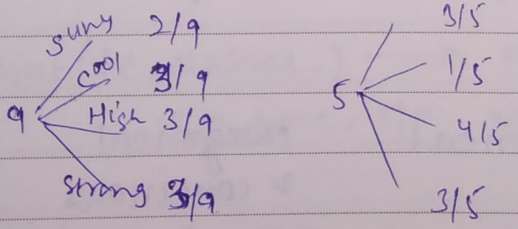
Ex. Dataset:

Day	Outlook	Temp.	Humidity	Wind	Play Badminton
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
⋮					
Day14	Rain	Mild	High	Strong	No

Q. For day (Sunny, cool, High, Strong) x', what's play prediction?

→ There is no such record in dataset (exactly same nhi hai)

Total = 14 Yes = 9 No = 5



$$P(\text{Yes}|x') = [P(\text{Sunny}|\text{Yes}) \cdot P(\text{Cool}|\text{Yes}) \cdot P(\text{High}|\text{Yes}) \cdot P(\text{Strong}|\text{Yes})] \cdot P(\text{Yes})$$

$$= 0.0053$$

$$P(\text{No}|x') = 0.0206$$

Predicted O/P → No ($P(\text{No}) > P(\text{Yes})$)

• It doesn't involve statistical models

9 Monday May 2011

→ conditional independence assumption.

• Advantages of Naiive Bayes.
- Handles purely categorical data.
- Works well with very large datasets.
- Simple & computationally efficient. (easy & fast).

Disadvantages.
- Requires large no. of records
- Problematic when a predictor category is not present in training data.
(Assign 0 prob. of response, ignoring info. in other var.)

Applications.
- Spam classification.
- Medical Diagnosis
- Weather Prean
- Text classification
- Character recognization.

Classification Technique	DV	IDV	Purpose.
Naiive Bayes	Categorical	Categorical	Used to classify record with help of probability.
K-NN (K-Neigheset Neighbour)	Categorical	categorical & continous	Used to classify record with help of Euclidean Distance
SVM (support vector M/c)	Categorical	Categorical & continous	Used to classify records with help of hyper plane.

• SVM is applicable for over-dimensional data.

Practical implementation

Dataset → svmtrain (Titanic dataset)

```
df = pd.read_csv('svmtrain.csv')

from sklearn import preprocessing
from sklearn.cross_validation import train_test_split   ~~test_train~~
```

If it is not working use this ↓

```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
```

NOTE
- accuracyscore → how much % the model is accurate.
- confusion matrix → how many records are classified more accurately

```
from sklearn.metrics import accuracy_score, confusion_matrix
```

→ convert text into numerical.,

```
le = preprocessing.LabelEncoder()
df['Sex'] = le.fit_transform(df['sex'])
```

print(le.classes_) ⟶ gives you category name
eg here, o/p: male, females.

```
y = df['Survived']
x = df.drop(['Survived','PassengerId'], axis=1)
```

col.

. axis=0 (row.)

```
y.count()        # 889

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,
                                                    random_state = 0)

from sklearn.naive_bayes import *
clf = BernoulliNB()
y_pred = clf.fit(X_train, y_train).predict(X_test)

accuracy_score(y_test, y_pred, normalize = True)    # 76.78%
confusion_matrix(y_test, y_pred)
```

→ 132 25
37 73

confusion matrix →

Actual → Training Data in can

(Test) predicted in row

Predicted ↑

		Class of Interest	All other classes
Class of Interest		TP (True +ve)	FP (False +ve)
All other classes		FN (False -ve)	TN (True -ve)

Ex =

		Survived Training 0	Survived Training 1
Survived Test	0	132	65
	1	37	73

Total = 267
correct = 205 (diag.)
Incorrect = 62 (non-diag.)

correct % = 205/267 = 0.76779

Assignment - (Print confusion matrix also)

DV	IDV	Accuracy.
Pclass	other all	? 59%
Gender	"	? 78%
sibsp	"	? 67%
Parch	"	? 74%
Embarked	"	? 69%

· use loop.

(watch it once)

Day-26

- All ✓ Project Description (Day 24/21)

Day-27.

K-Nearest Neighbours.

- Classification Technique
- classify records with help of euclidean distance. find closeness of 2 points

Features.

- All instances correspond to pts. in n-dimensional Euclidean space
- classification is delayed till new instance arrives.
- Target fn may be discrete or real valued.
- classification done by comparing feature vectors of diff. points.

It is instance Based learning
→ Based on characteristics of record we classify the O/P.

Euclidean distance →

Find closeness of 2 records.

$$\sqrt{(x_1-u_1)^2+(x_2-u_2)^2+\ldots+(x_p-u_p)^2}$$

(1st response, 2nd response)

or

$$D(X,Y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

Ex: record 1
Jay:
Age = 35
Salary = 95K
CD = 3

record 2.
Riya:
Age = 41
Salary = 215K
CD = 2

Ex:

$$\text{euclideanalis.} = \sqrt{(35-41)^2+(95000-215000)^2-(3-2)^2}$$

(Age, Salary, CD)