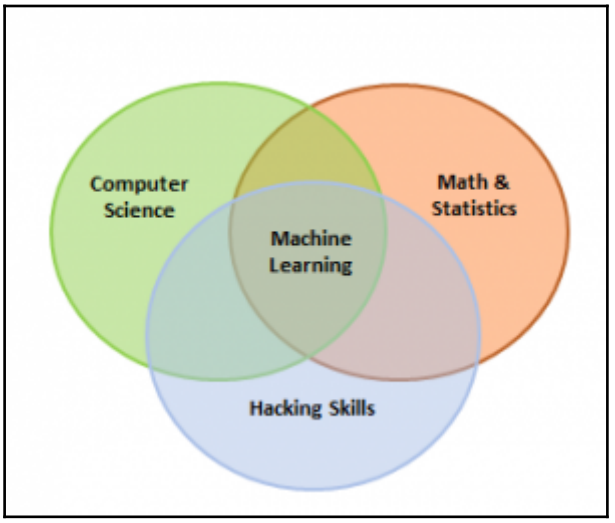
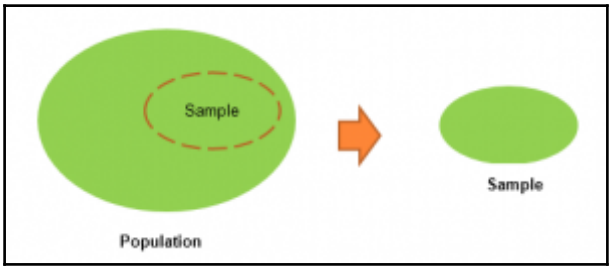
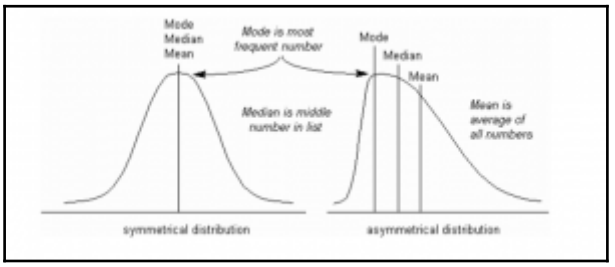
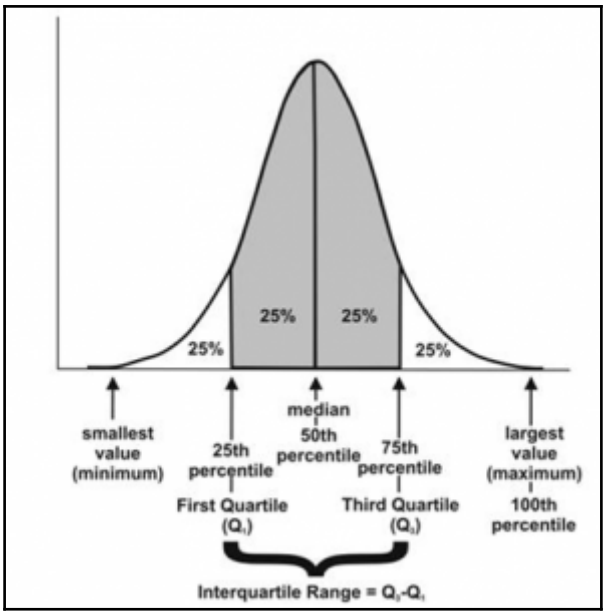
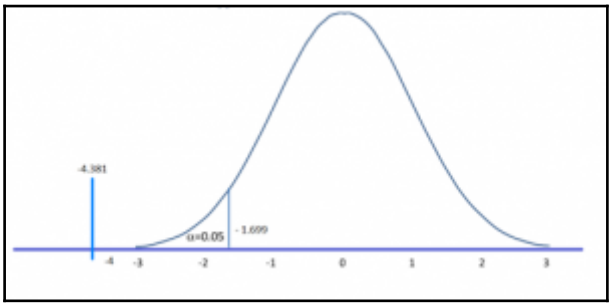
Chapter 1: Journey from Statistics to Machine Learning

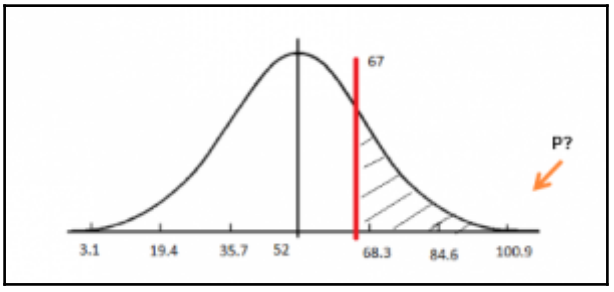


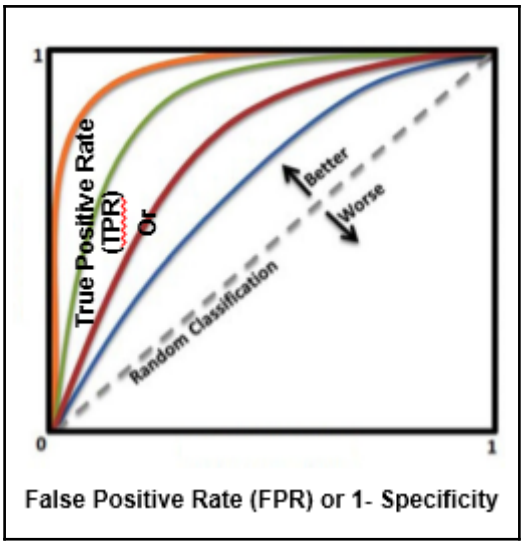


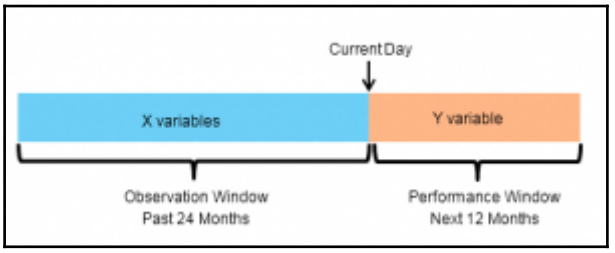


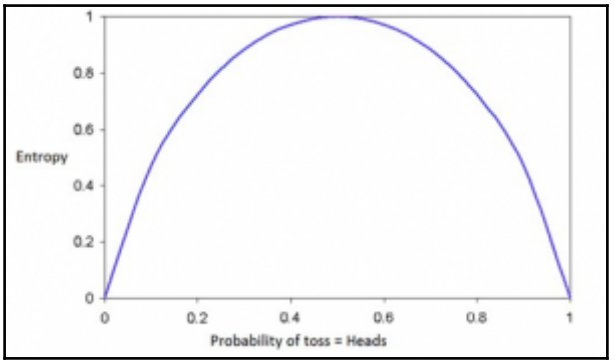

25%
25%
25%
25%
smallest value (minimum)
25th percentile
First Quartile (Q_1)
median
50th percentile
75th percentile
Third Quartile (Q_3)
largest value (maximum)
100th percentile
Interquartile Range = Q_3-Q_1

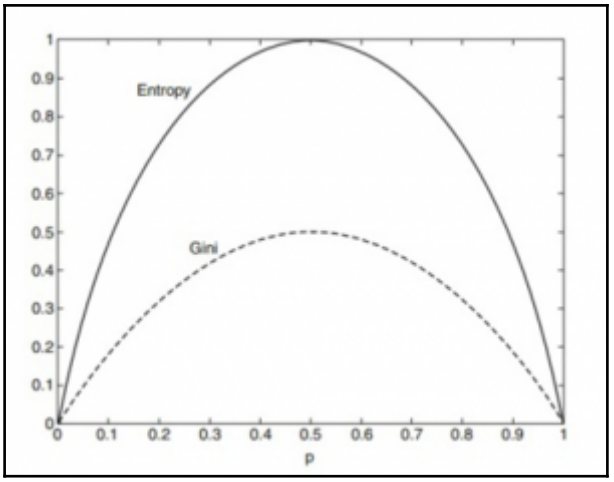

-4.381
α=0.05
-1.699
-4
-3
-2
-1
0
1
2
3

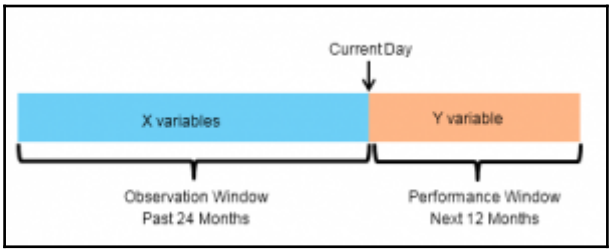

67
P?
3.1
19.4
35.7
52
68.3
84.6
100.9

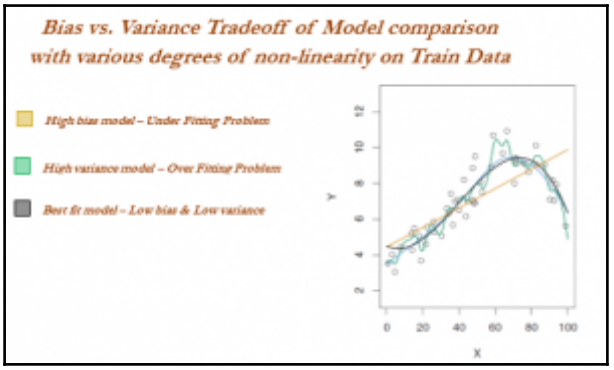

1
True Positive Rate
(TPR)
Or
Better
Worse
Random Classification
0
1
False Positive Rate (FPR) or 1- Specificity

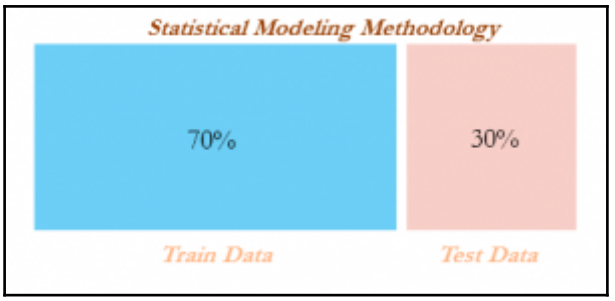

Current Day
X variables
Y variable
Observation Window
Past 24 Months
Performance Window
Next 12 Months

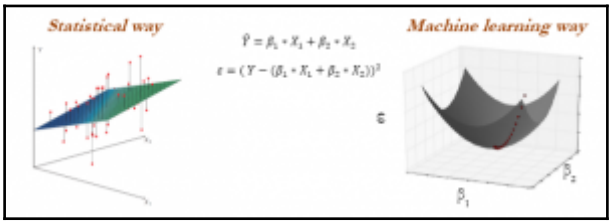

1
0.8
0.6
Entropy
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Probability of toss = Heads

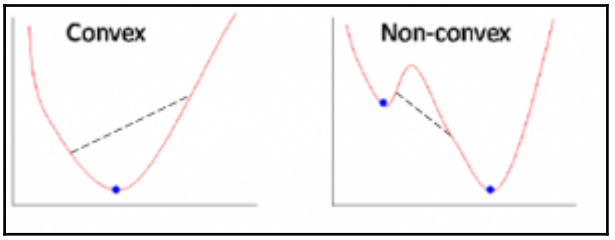

Entropy
Gini
p

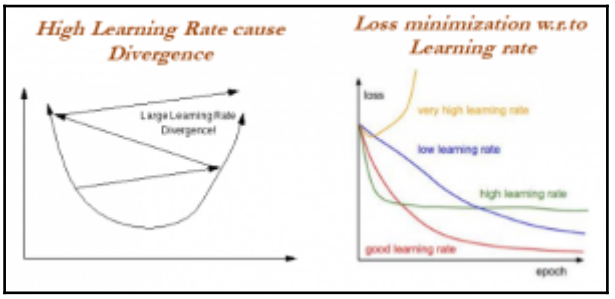

Current Day
X variables
Y variable
Observation Window
Past 24 Months
Performance Window
Next 12 Months

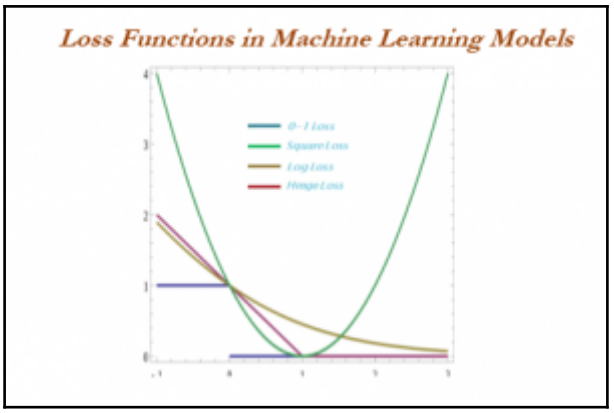

Bias vs. Variance Tradeoff of Model comparison
with various degrees of non-linearity on Train Data
High bias model – Under Fitting Problem
High variance model – Over Fitting Problem
Best fit model – Low bias & Low variance
Y
X

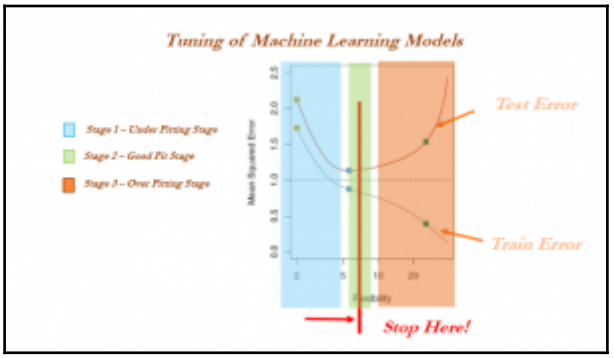

Statistical Modeling Methodology
70%
30%
Train Data
Test Data

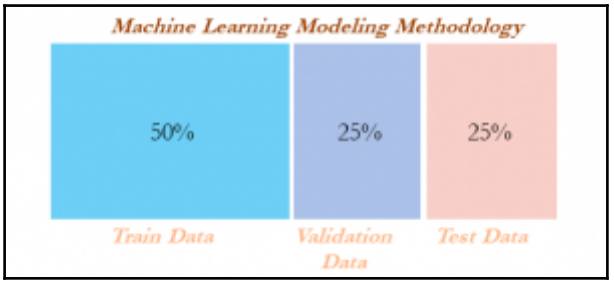

Statistical way
$\hat{Y} = \beta_1 * X_1 + \beta_2 * X_2$
$\varepsilon = (Y - (\beta_1 * X_1 + \beta_2 * X_2))^2$
Machine learning way
ε
β_1
β_2

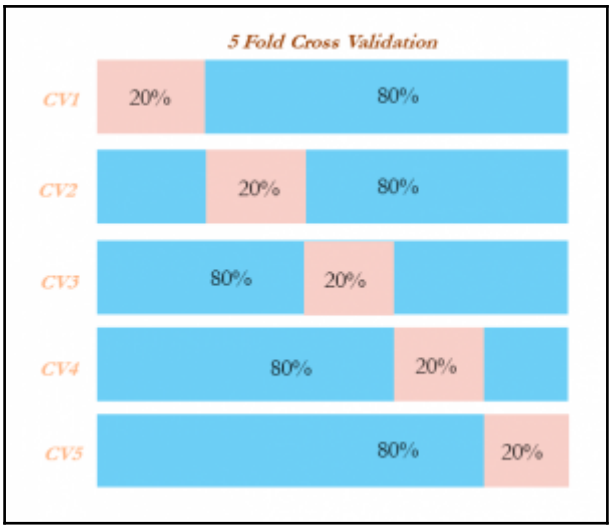

Convex
Non-convex

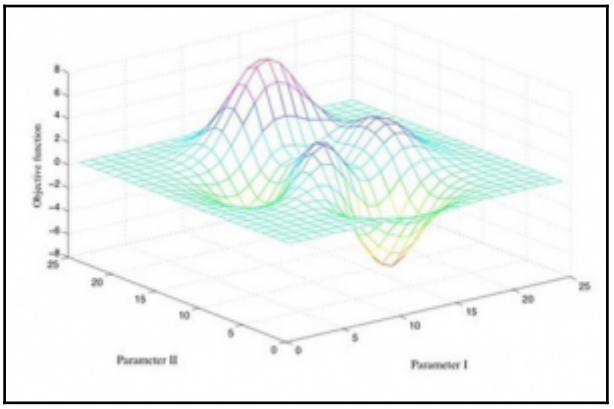

High Learning Rate cause Divergence
Large Learning Rate Divergence!
Loss minimization w.r.to Learning rate
loss
very high learning rate
low learning rate
high learning rate
good learning rate
epoch


Loss Functions in Machine Learning Models
0-1 Loss
SquareLoss
LogLoss
HingeLoss

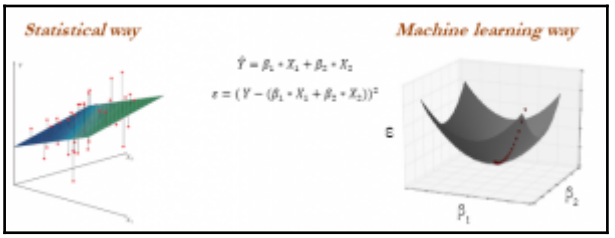

Tuning of Machine Learning Models
Stage 1 – Under Fitting Stage
Stage 2 – Good Fit Stage
Stage 3 – Over Fitting Stage
Test Error
Train Error
Stop Here!

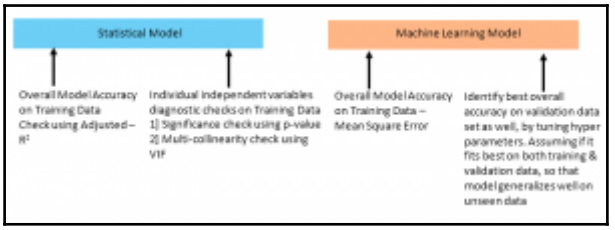

Machine Learning Modeling Methodology
50%
25%
25%
Train Data
Validation Data
Test Data

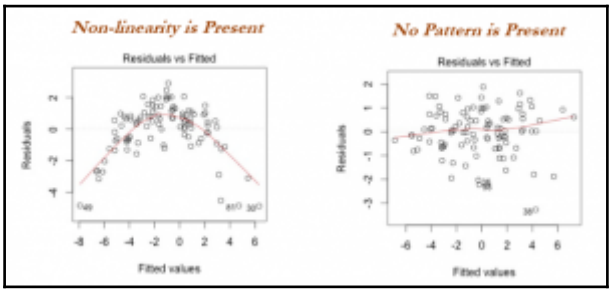

5 Fold Cross Validation
CV1
20%
80%
CV2
20%
80%
CV3
80%
20%
CV4
80%
20%
CV5
80%
20%

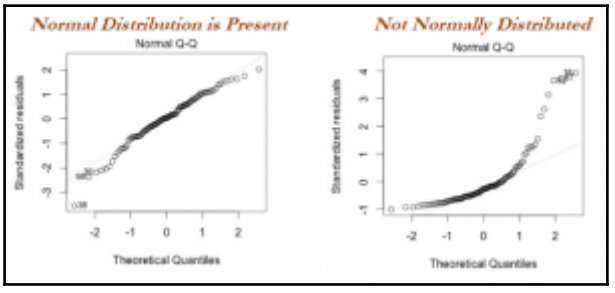

Objective function
Parameter II
Parameter I

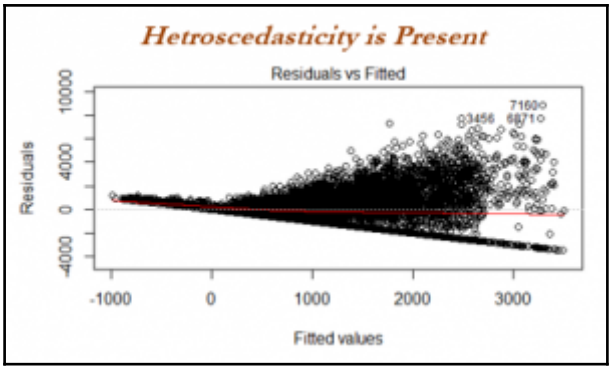
Chapter 2: Parallelism of Statistics and Machine Learning

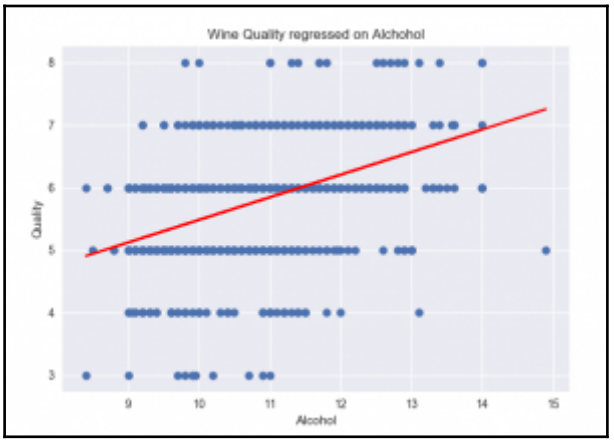




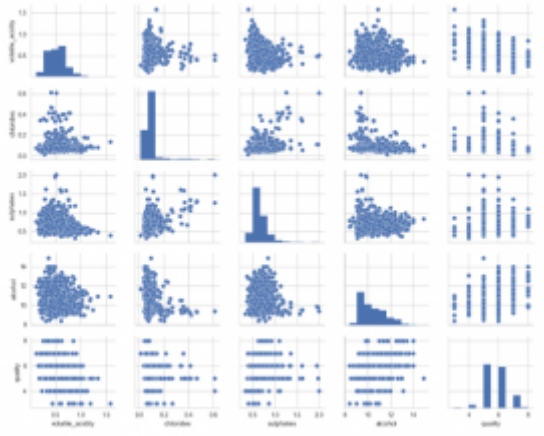





Hetroscedasticity is Present
Residuals vs Fitted
Residuals
Fitted values
7160
3456
6871
10000
4000
0
-4000
-1000
0
1000
2000
3000

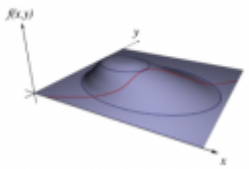

Wine Quality regressed on Alchohol
Quality
Alcohol
8
7
6
5
4
3
9
10
11
12
13
14
15

Pair plots of Wine Quality

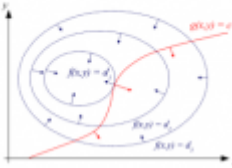


Heat map of Correlation coefficients

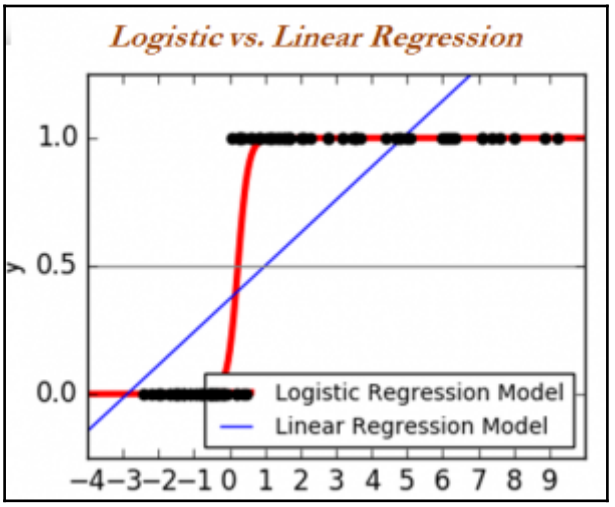
Lagrangian 3D format



Lagrangian 2D format



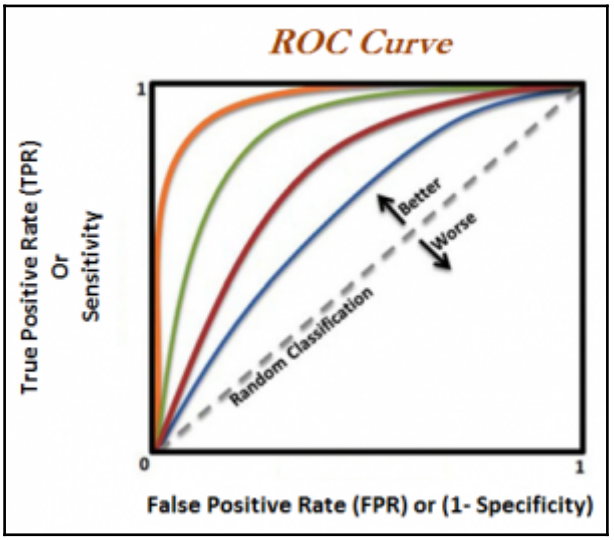
Chapter 3: Logistic Regression Versus Random Forest



Information Value	Predictive Power
<0.02	Useless for prediction
0.02 to 0.1	Weak predictor
0.1 to 0.3	Medium predictor
0.3 to 0.5	Strong predictor
> 0.5	Suspicious or too good predictor

Range	Bin Number	Events	Non Events	% of Events (E) (Events/Total Events)	% of Non-Events (NE) (Non Events/Total Non Events)	(E) - (NE)	WOE = Ln(E/NE)	IV = WOE * (E-NE)
0-50	1	40	394	5%	5%	0.003	0.062	0.0002
51-100	2	68	900	9%	12%	-0.024	-0.234	0.0056
101-150	3	78	984	10%	13%	-0.025	-0.188	0.0040
151-200	4	102	1198	14%	15%	-0.016	-0.111	0.0018
201-250	5	108	1218	15%	16%	-0.011	-0.074	0.0008
251-300	6	110	1264	15%	15%	-0.001	-0.013	0.0000
301-350	7	82	772	11%	10%	0.011	0.107	0.0012
351-400	8	46	330	6%	4%	0.019	0.379	0.0070
401-450	9	42	368	6%	5%	0.009	0.179	0.0017
>451	10	68	470	9%	6%	0.031	0.416	0.0129
	Total	744	7704				I.V.Total	0.0356

	Predicted: YES	Predicted: NO
Actual: YES	TP (True Positive)	FN (False Negative)
Actual: NO	FP (False Positive)	TN (True Negative)

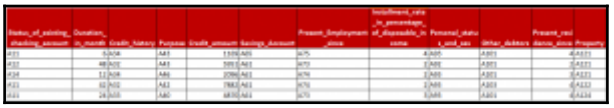


Actual	Predicted
1	0.92
0	0.34
0	0.12
1	0.4
1	0.64
0	0.82
1	0.84

Actual	Predicted
1	0.92
1	0.4
1	0.64
1	0.84

Actual	Predicted
0	0.34
0	0.12
0	0.82

Actual	Predicted	Actual	Predicted	Concordant pair	Discordant pair
1	0.92	0	0.34	✓	
1	0.92	0	0.12	✓	
1	0.92	0	0.82	✓	
1	0.4	0	0.34	✓	
1	0.4	0	0.12	✓	
1	0.4	0	0.82		✓
1	0.64	0	0.34	✓	
1	0.64	0	0.12	✓	
1	0.64	0	0.82		✓
1	0.84	0	0.34	✓	
1	0.84	0	0.12	✓	
1	0.84	0	0.82	✓	

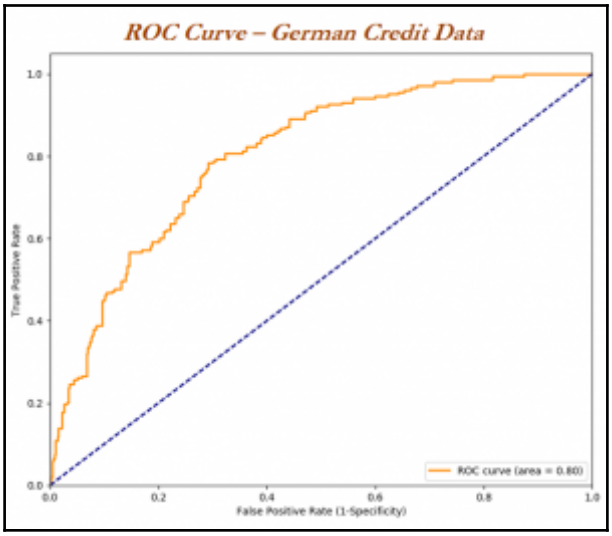


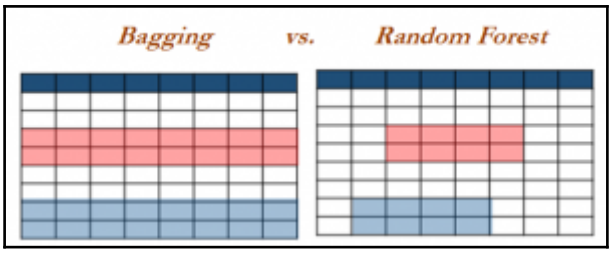
Property	Age_in_years	Other_installment_plans	Housing	Number_of_existing_credits_at_this_bank	Job	Number_of_People_being_liable_to_provide_maintenance_for	Telephone	Foreign_worker	class
A121	67	A143	A152	2	A173	1	A192	A201	1
A121	22	A143	A152	1	A173	1	A191	A201	2
A121	49	A143	A152	1	A172	2	A191	A201	1
A122	45	A143	A153	1	A173	2	A191	A201	1
A124	53	A143	A153	2	A173	2	A191	A201	2

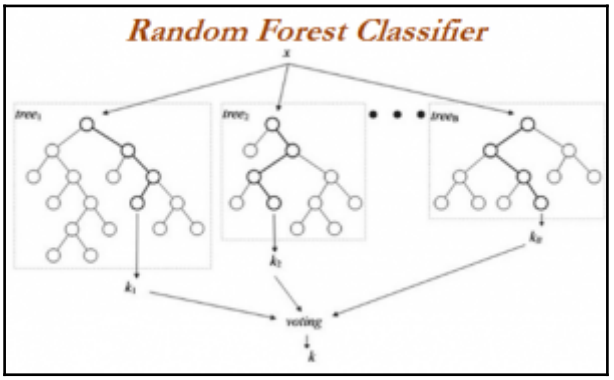
Credit_History- Information Value						
	Total	bad	good	bad_per	good_per	I_V
A30	46	27	19	0.091525	0.026685	0.079915
A31	44	26	18	0.088136	0.025281	0.078495
A32	533	161	372	0.545763	0.522472	0.001016
A33	90	32	58	0.108475	0.081461	0.007737
A34	294	49	245	0.166102	0.344101	0.129643
					Total_IV	**0.296806**

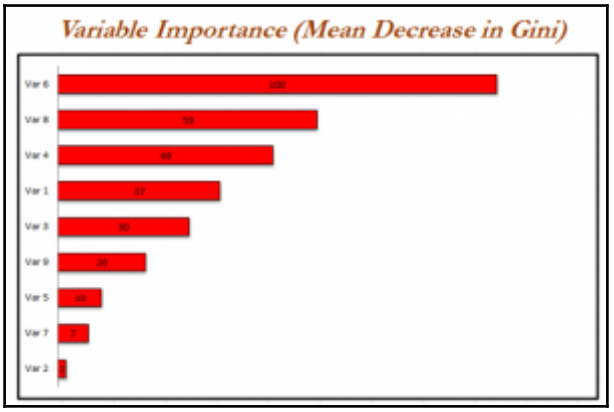
Duration_in_Months- Information Value						
	Total	bad	good	bad_per	good_per	I_V
[1,101.6]	101	10	91	0.033898	0.127809	0.124636
(101.6,202.2]	101	16	85	0.054237	0.119382	0.051397
(202.2,302.8]	100	25	75	0.084746	0.105337	0.004479
(302.8,403.4]	101	29	72	0.098305	0.101124	0.00008
(403.4,504]	101	21	80	0.071186	0.11236	0.018791
(504,604.6]	100	38	62	0.128814	0.087079	0.016341
(604.6,705.2]	101	30	71	0.101695	0.099719	0.000039
(705.2,805.8]	100	32	68	0.108475	0.095506	0.001651
(805.8,906.4]	101	45	56	0.152542	0.078652	0.048946
(906.4,1007]	101	49	52	0.166102	0.073034	0.076472
					Total_IV	0.342832

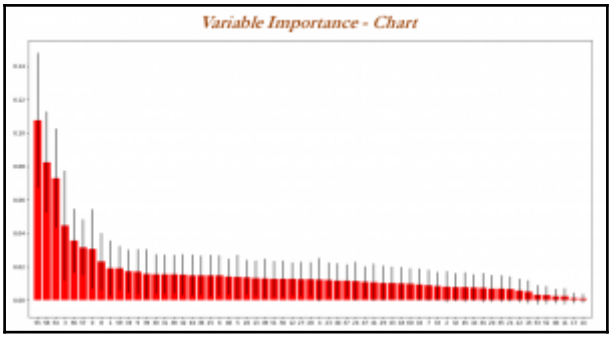
Variable	p-value	VIF
purpose_A46	0.937	1.279
Per_stat_sx_A93	0.000	6.177




Bagging
vs.
Random Forest


Random Forest Classifier
x
tree1
tree2
treeB
k1
k2
kB
voting
k


Variable Importance (Mean Decrease in Gini)
Var 6
Var 8
Var 4
Var 1
Var 3
Var 9
Var 5
Var 7
Var 2



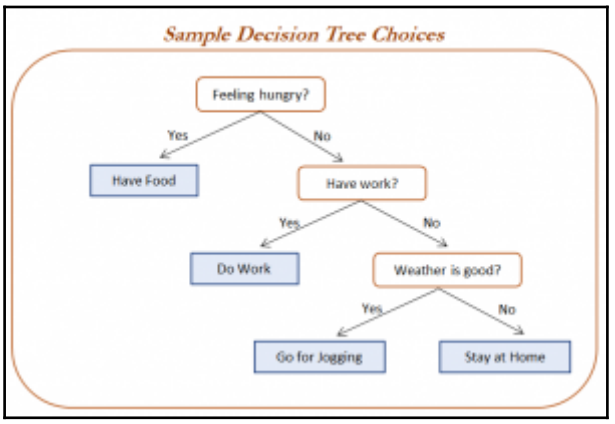
Logistic Regression : Summary

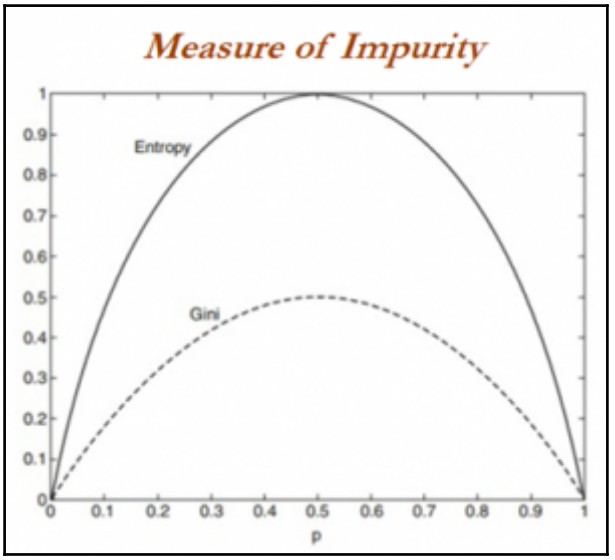
Variable	Co-efficient	p-value
const	-1.1760	0
status_exs_acnt_A14	-1.7760	0
dred_hist_A34	-0.9757	0
per_stat_sx_A95	-0.7401	0
installment_rate_in_percentage_of_disposable_income	0.3765	0
purpose_A40	1.3000	0.002
property_A124	0.7256	0.007
status_exs_acnt_A11	1.1014	0.009
oth_inst_oth_A143	1.1010	0.009
purpose_A43	-0.6300	0.011
Credit_amount	0.0003	0.026
Number_of_existing_credits_at_this_bank	0.4676	0.04
dred_hist_A31	0.9028	0.042
Duration_in_month	0.0185	0.057

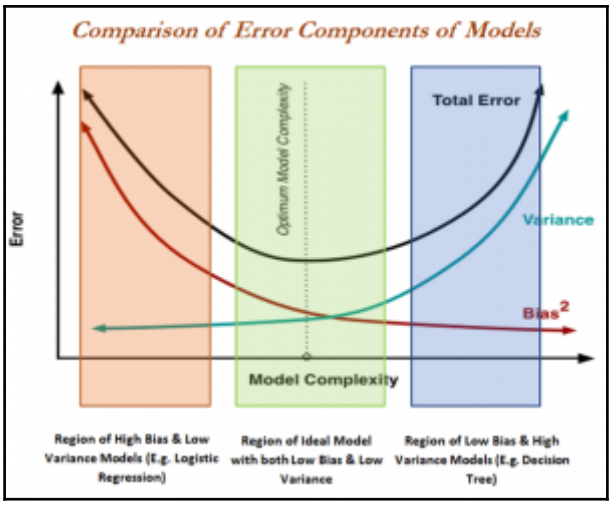
Variable Importance : Random Forest

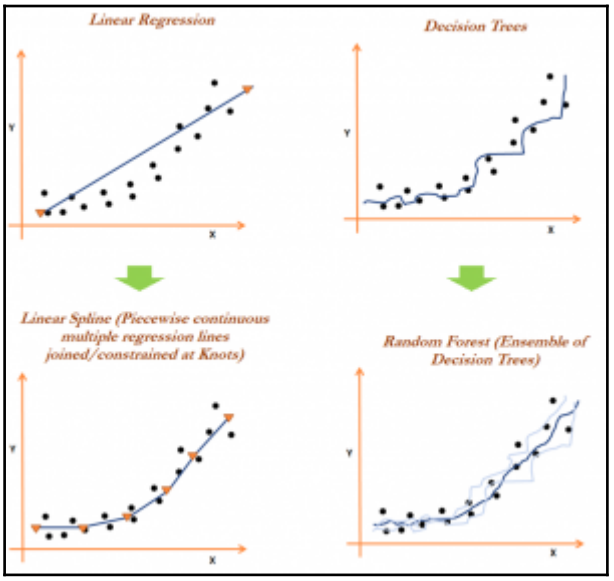
Feature Number	Variable Name	Mean Decrease in Gini
55	Credit_amount	0.0075
59	Age_in_years	0.0059
54	Duration_in_month	0.0059
1	status_exs_acnt_A34	0.0048
56	Installment_rate_in_percentage_of_disposable_income	0.0031
57	Present_residence_since	0.0030
0	status_exs_acnt_A11	0.0027
3	dred_hist_A31	0.0025
5	status_exs_acnt_A12	0.0025
58	Number_of_existing_credits_at_this_bank	0.0025
39	ban_acc_A61	0.0017
9	purpose_A40	0.017
49	property_A124	0.0007
30	per_stat_sx_A95	0.0005
31	per_stat_sx_A93	0.0005
50	property_A121	0.0005
42	oth_inst_oth_A143	0.0002
44	housing_A152	0.0049
36	property_A123	0.0049
20	pre_emp_ems_A73	0.0047
6	dred_hist_A32	0.0047
46	job_A173	0.004
7	dred_hist_A33	0.0039
38	pre_emp_ems_A75	0.0037
13	purpose_A41	0.0035
45	job_A174	0.0035
51	Telephn_A192	0.0038
10	Telephn_A191	0.0027
12	purpose_A42	0.0027
27	pre_emp_ems_A76	0.0027
28	pre_emp_ems_A74	0.0026
4	dred_hist_A30	0.0023
33	ban_acc_A65	0.0022
40	oth_inst_oth_A141	0.0015

Chapter 4: Tree-Based Machine Learning Models

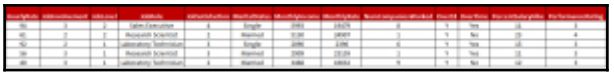


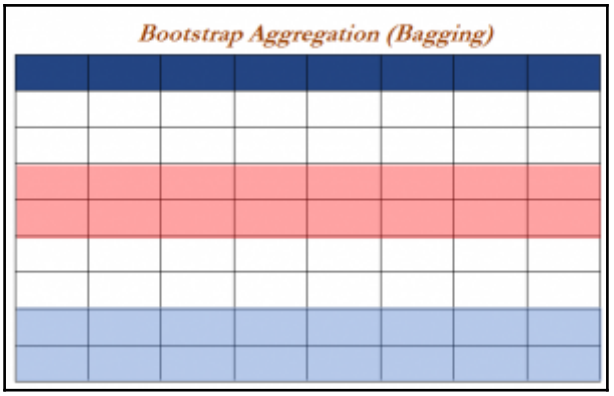


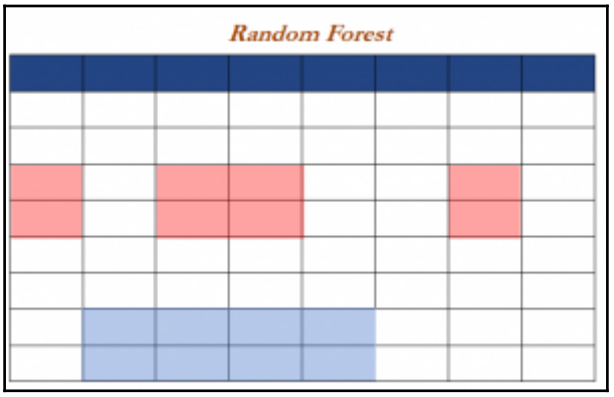


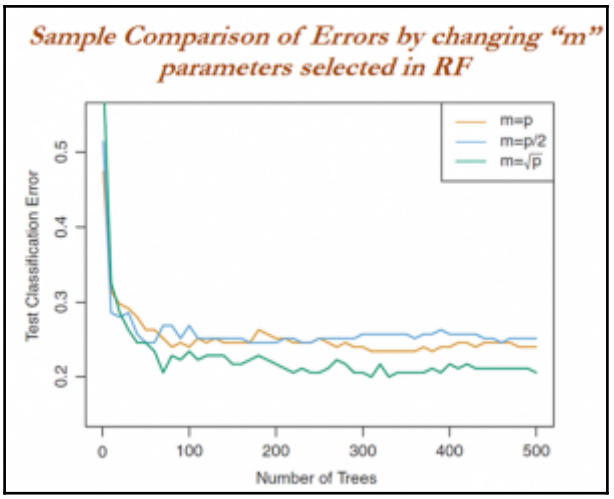


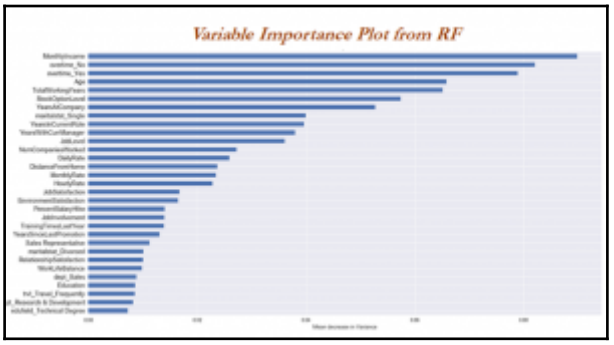
Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender
41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female
49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male
37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male
33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female
27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male

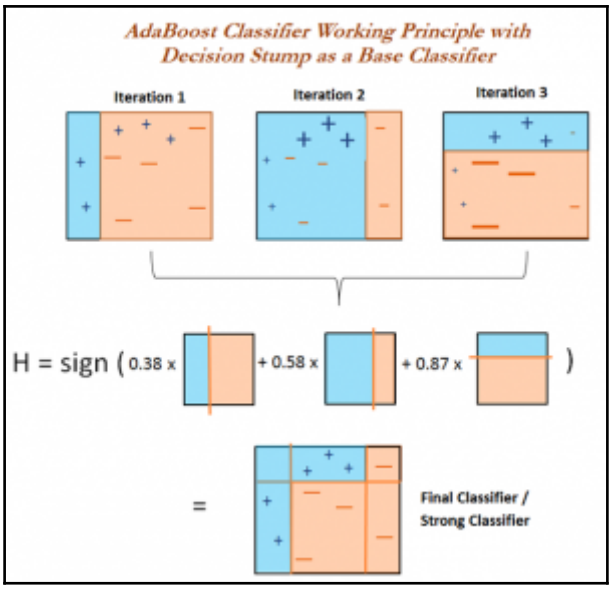


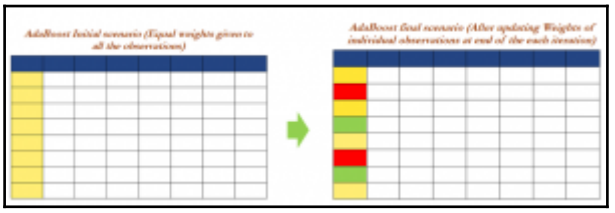

Bootstrap Aggregation (Bagging)

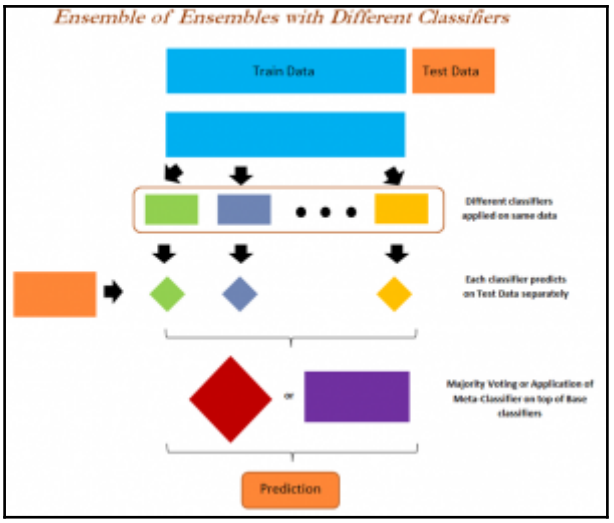

Random Forest

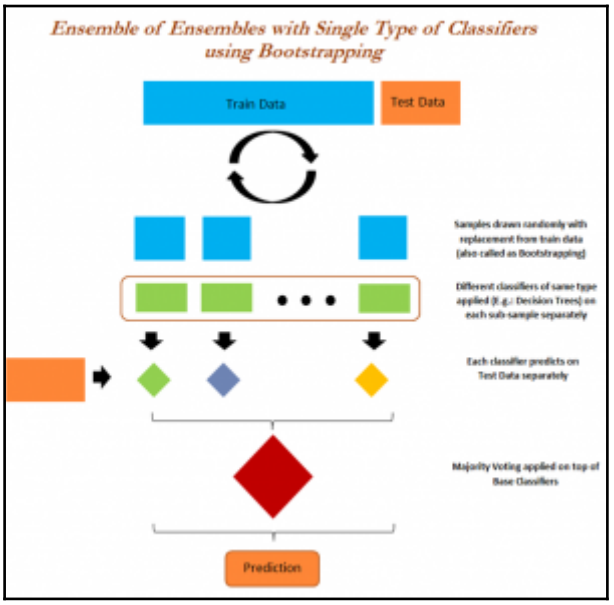

Sample Comparison of Errors by changing "m" parameters selected in RF
m=p
m=p/2
m=√p
Test Classification Error
0.2
0.3
0.4
0.5
0
100
200
300
400
500
Number of Trees


Variable Importance Plot from RF

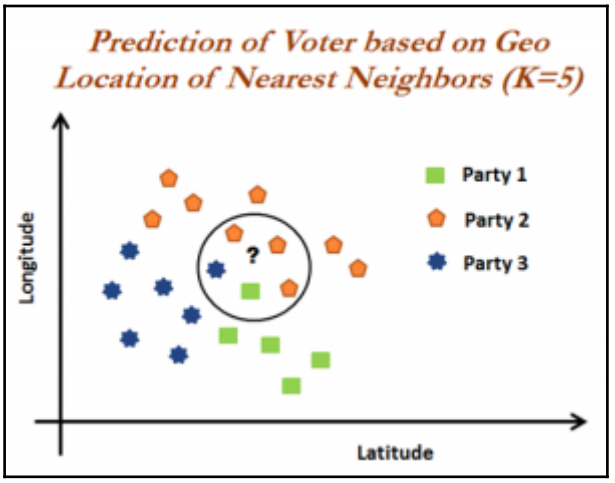

AdaBoost Classifier Working Principle with Decision Stump as a Base Classifier
Iteration 1
Iteration 2
Iteration 3
H = sign (0.38 x
+ 0.58 x
+ 0.87 x
)
=
Final Classifier / Strong Classifier

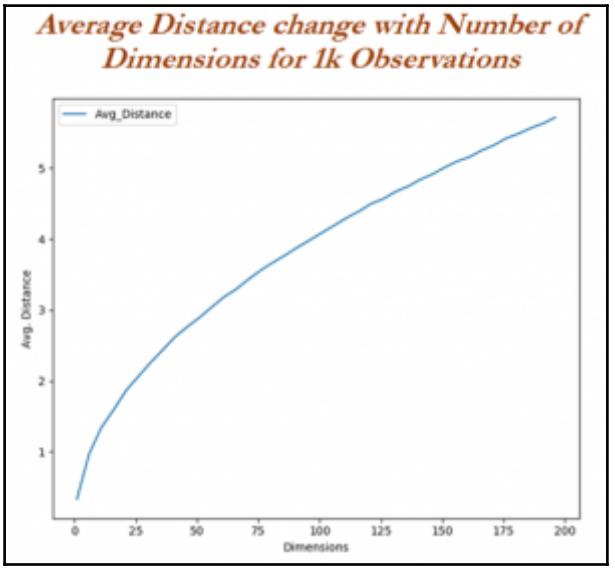


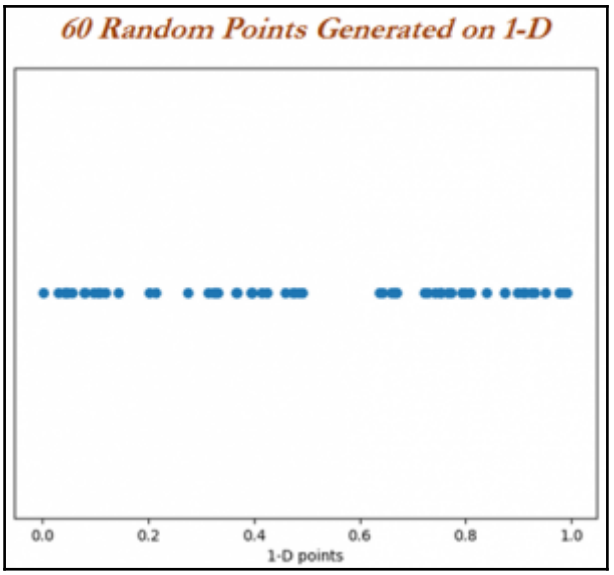

Ensemble of Ensembles with Different Classifiers
Train Data
Test Data
Different classifiers applied on same data
Each classifier predicts on Test Data separately
or
Majority Voting or Application of Meta-Classifier on top of Base classifiers
Prediction

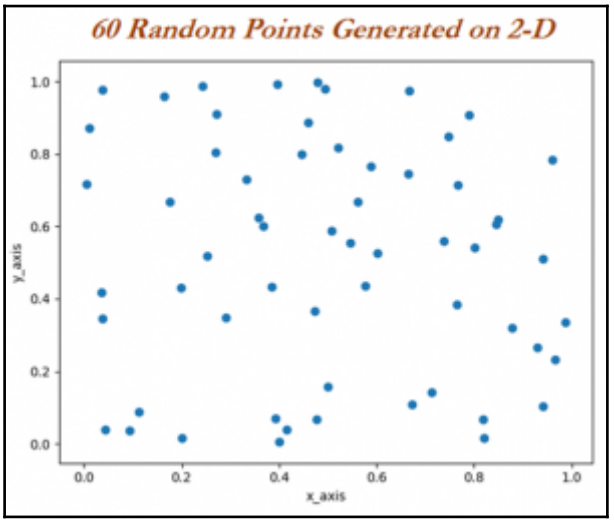

Ensemble of Ensembles with Single Type of Classifiers using Bootstrapping
Train Data
Test Data
Samples drawn randomly with replacement from train data (also called as Bootstrapping)
Different classifiers of same type applied (E.g.: Decision Trees) on each sub-sample separately
Each classifier predicts on Test Data separately
Majority Voting applied on top of Base Classifiers
Prediction

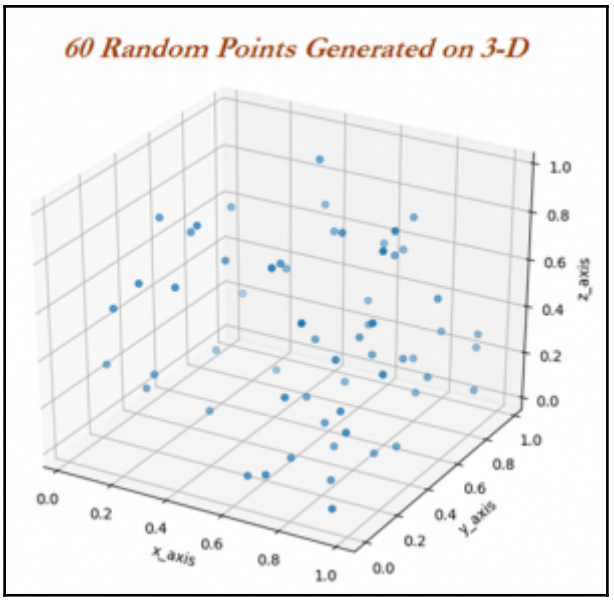
Chapter 5: K-Nearest Neighbors and Naive Bayes



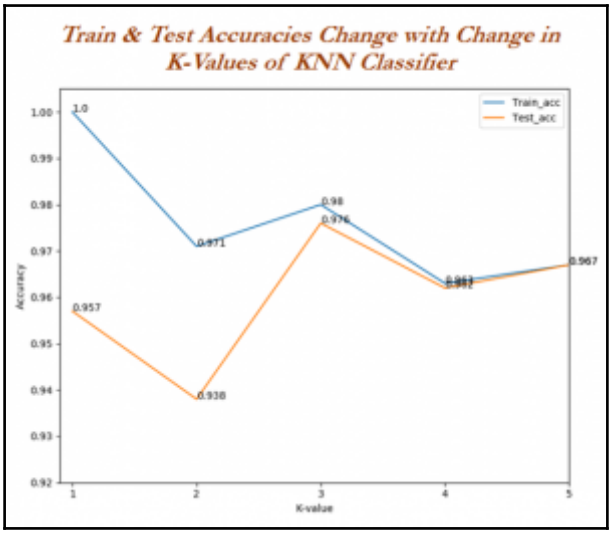


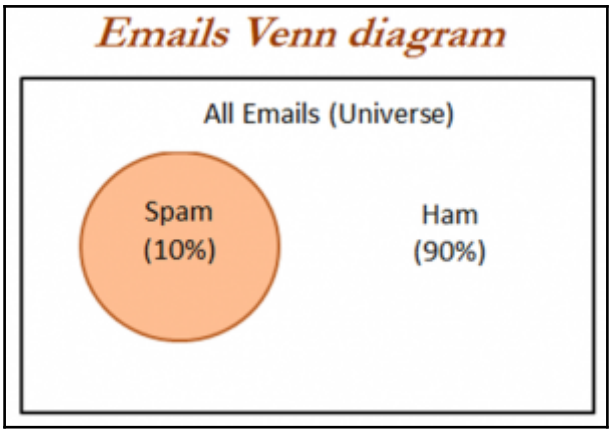

60 Random Points Generated on 1-D
0.0
0.2
0.4
0.6
0.8
1.0
1-D points

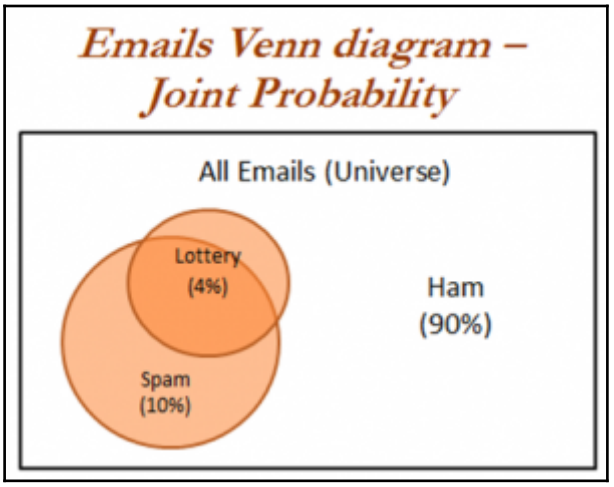

60 Random Points Generated on 2-D
1.0
0.8
0.6
0.4
0.2
0.0
y_axis
0.0
0.2
0.4
0.6
0.8
1.0
x_axis

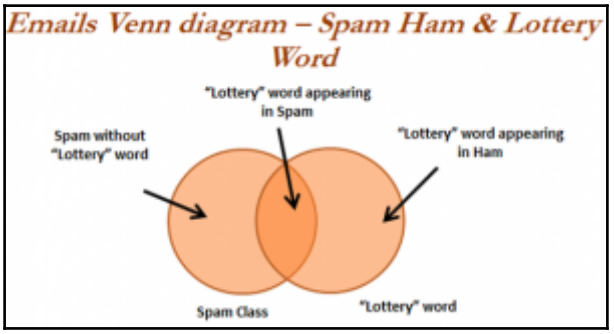


ID Number	Clump Thickness	Unif Cell Size	Unif Cell Shape	Marg Adhesion	Single Epith Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2
1016277	6	8	8	1	3	4	3	7	1	2
1017023	4	1	1	3	2	1	3	1	1	2




Emails Venn diagram
All Emails (Universe)
Spam
(10%)
Ham
(90%)


Emails Venn diagram –
Joint Probability
All Emails (Universe)
Lottery
(4%)
Spam
(10%)
Ham
(90%)


Emails Venn diagram – Spam Ham & Lottery Word
"Lottery" word appearing in Spam
Spam without "Lottery" word
"Lottery" word appearing in Ham
Spam Class
"Lottery" word

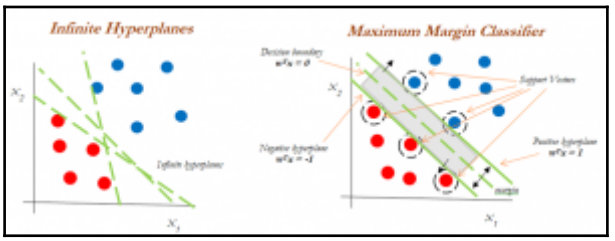
Word Frequency & Likelihood of Lottery with Spam & Ham

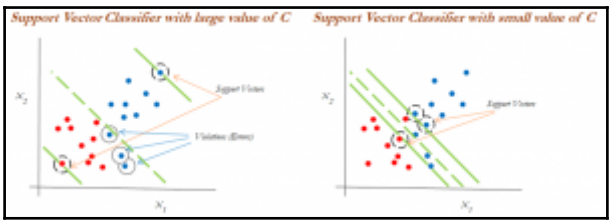
	Lottery		
Frequency	yes	no	Total
spam	3	19	22
ham	2	76	78
Total	5	95	100

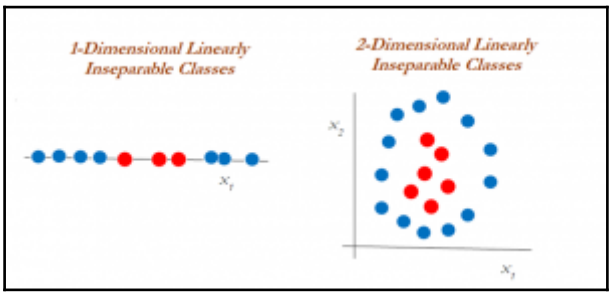
	Lottery		
Likelihood	yes	no	Total
spam	3/22	19/22	22
ham	2/78	76/78	78
Total	5/100	95/100	100

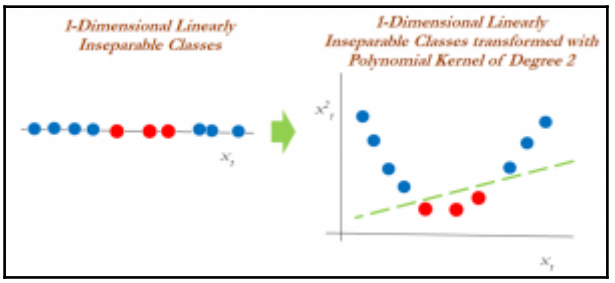
	Lottery (W1)		Million (W2)		Unsubscribe (W3)		
Likelihood	yes	no	yes	no	yes	no	Total
spam	3/22	19/22	11/22	11/22	13/22	9/22	22
ham	2/78	76/78	15/78	63/78	21/78	57/78	78
Total	5/100	95/100	26/100	74/100	34/100	66/100	100

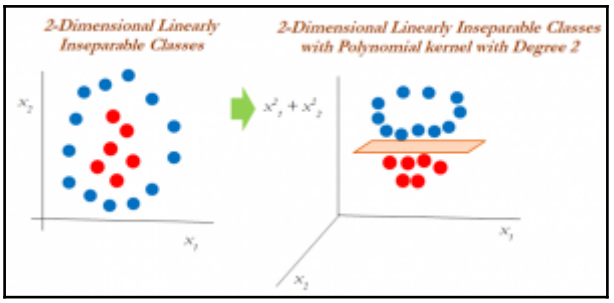
Chapter 6: Support Vector Machines and Neural Networks

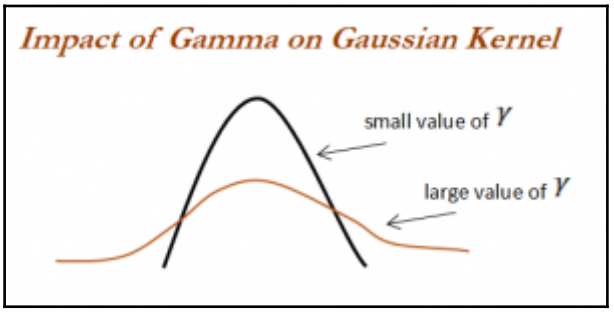


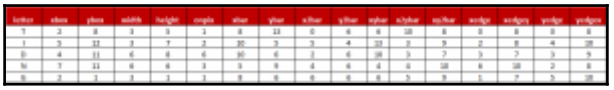




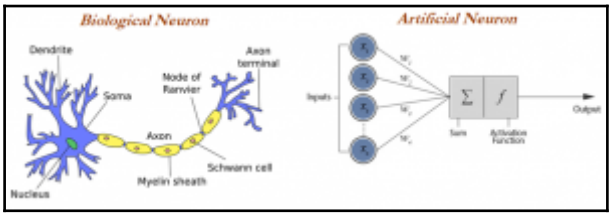


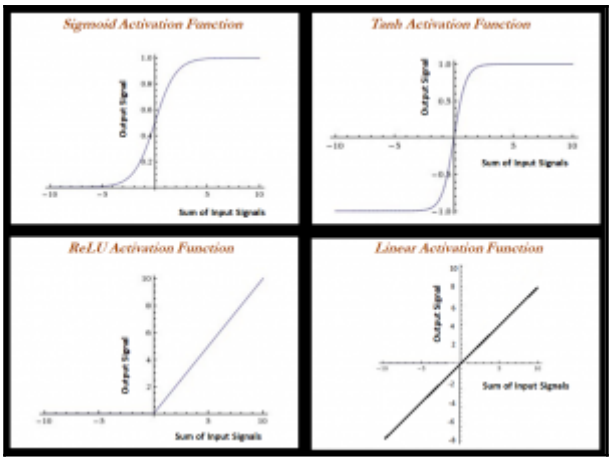


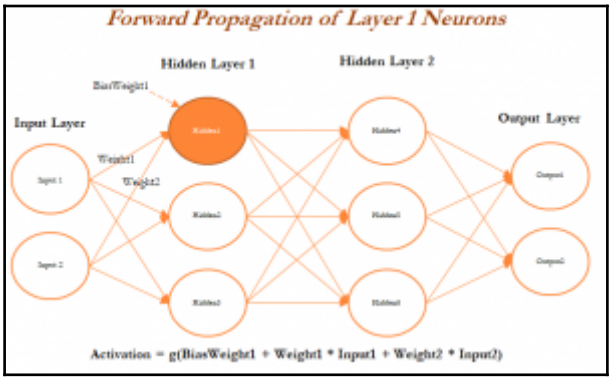


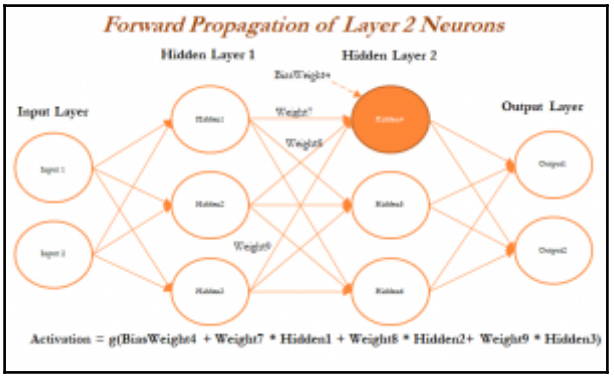


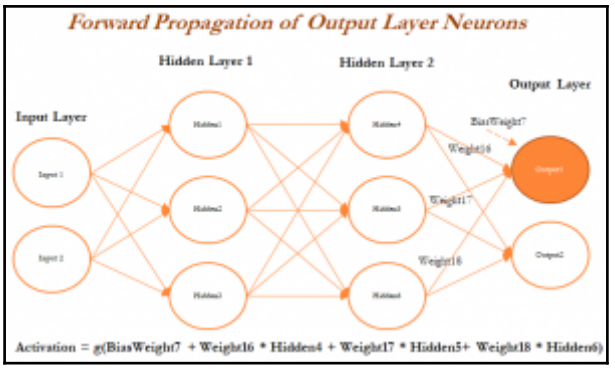
7	2	8	3	5	1	8	13	0	6	6	10	8	0	8	0	8
1	5	11	1	7	2	10	3	5	4	10	3	9	2	8	4	10
0	4	11	6	8	6	10	6	3	6	10	5	7	5	7	5	9
6	7	11	6	6	3	5	9	4	6	4	6	10	6	10	2	8
6	2	3	3	3	3	9	6	6	6	6	5	9	3	7	5	10

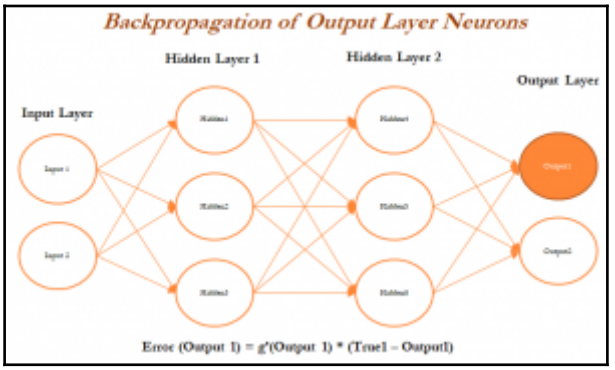


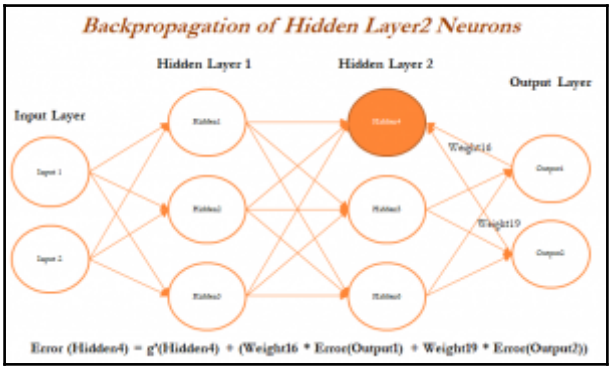

Sigmoid Activation Function
Output Signal
Sum of Input Signals
Tanh Activation Function
Output Signal
Sum of Input Signals
ReLU Activation Function
Output Signal
Sum of Input Signals
Linear Activation Function
Output Signal
Sum of Input Signals


Forward Propagation of Layer 1 Neurons
Hidden Layer 1
Hidden Layer 2
Input Layer
Output Layer
Weight1
Weight2
Activation = g(BiasWeight1 + Weight1 * Input1 + Weight2 * Input2)

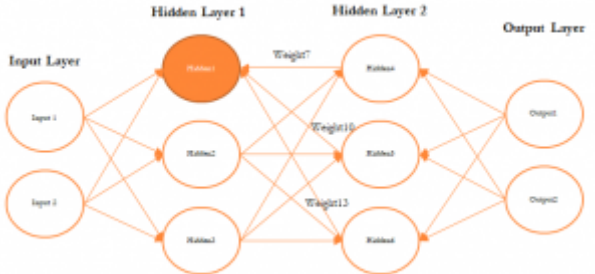

Forward Propagation of Layer 2 Neurons
Hidden Layer 1
Hidden Layer 2
Input Layer
Output Layer
Weight6
Activation = g(BiasWeight4 + Weight7 * Hidden1 + Weight8 * Hidden2+ Weight9 * Hidden3)


Forward Propagation of Output Layer Neurons
Hidden Layer 1
Hidden Layer 2
Output Layer
Input Layer
BiasWeight7
Weight16
Weight17
Weight18
Activation = g(BiasWeight7 + Weight16 * Hidden4 + Weight17 * Hidden5+ Weight18 * Hidden6)


Backpropagation of Output Layer Neurons
Hidden Layer 1
Hidden Layer 2
Output Layer
Input Layer
Error (Output 1) = g'(Output 1) * (True1 – Output1)

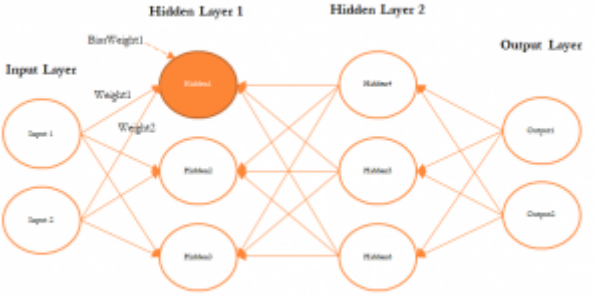

Backpropagation of Hidden Layer2 Neurons
Hidden Layer 1
Hidden Layer 2
Output Layer
Input Layer
Weight16
Weight19
Error (Hidden4) = g'(Hidden4) * (Weight16 * Error(Output1) + Weight19 * Error(Output2))

Backpropagation of Hidden Layerl Neurons



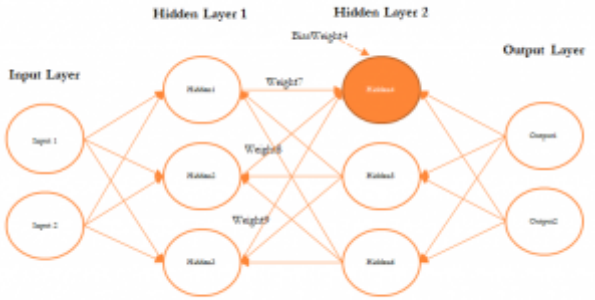
Error (Hidden1) = g'(Hidden1) * (Weight7 * Error(Hidden4) + Weight10 * Error(Hidden5) + Weight13*Error(Hidden6))

Updating Weights of Hidden Layer 1 – Input Layer during Backpropagation



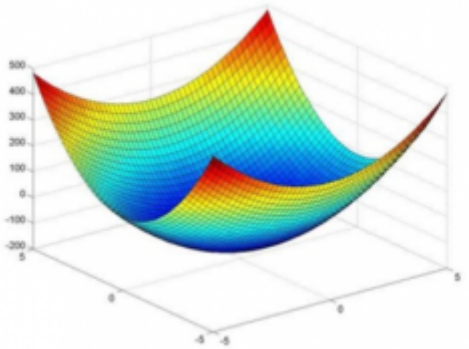
BiasWeight1= BiasWeight1 + a * Error(Hidden1) * 1
Weight1 = Weight1 + a * Error(Hidden1) * Input1
Weight2 = Weight2 + a * Error(Hidden1) * Input2

Updating Weights of Hidden Layer 1 – Hidden Layer 2 during Forward Propagation

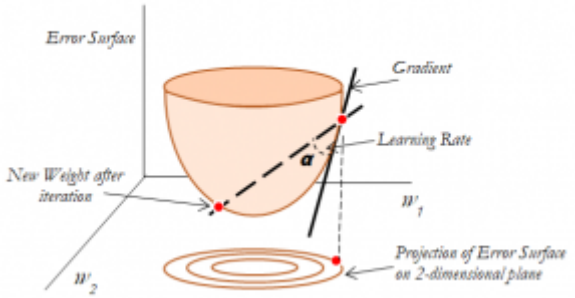


BiasWeight4= BiasWeight4 + a * Error(Hidden4) * 1
Weight7 = Weight7 + a * Error(Hidden4) * Hidden1
Weight8 = Weight8 + a * Error(Hidden4) * Hidden2
Weight9 = Weight9 + a * Error(Hidden4) * Hidden3

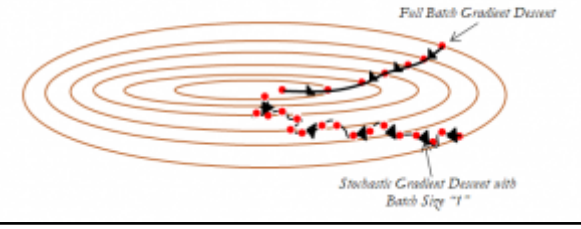
Optimization of Error Surface

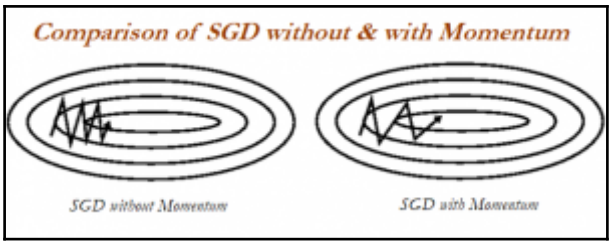


Stochastic Gradient Descent with Batch size "1"

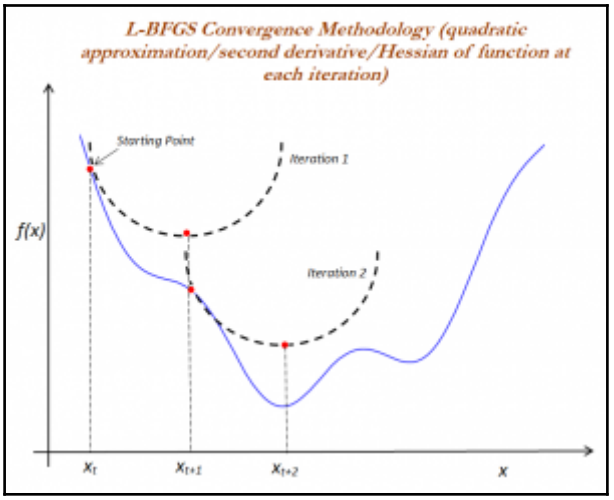


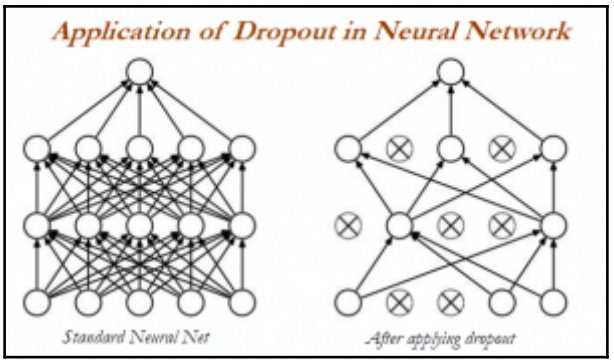
Comparison of Convergence between Stochastic Gradient Descent with Batch size "1" & Full Batch Gradient Descent

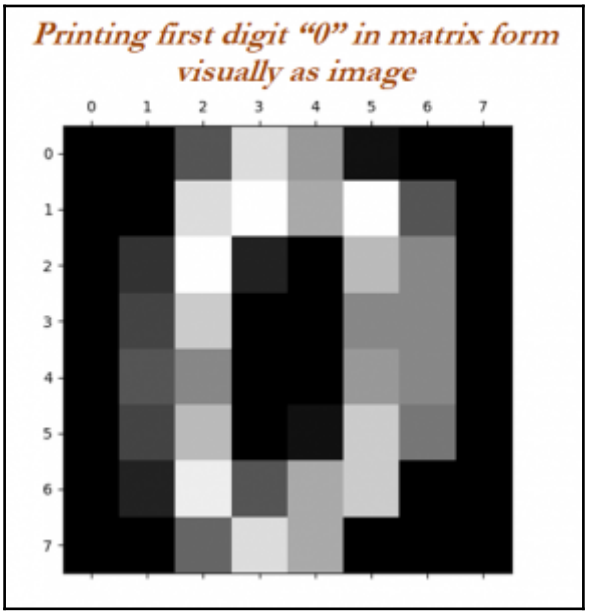


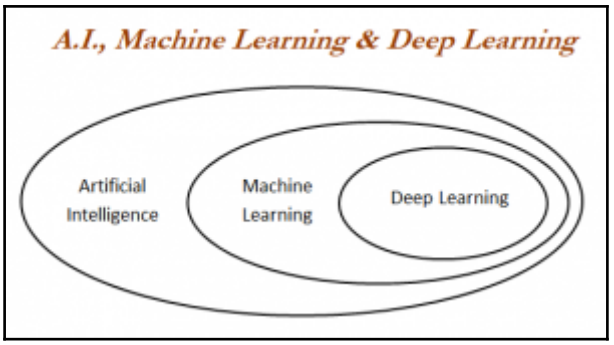

Comparison of SGD without & with Momentum
SGD without Momentum
SGD with Momentum

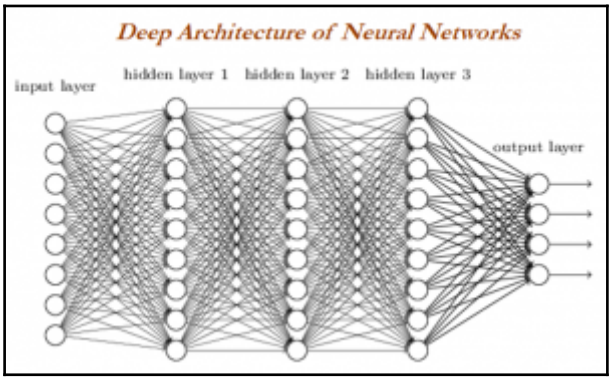

Nesterov Momentum Update

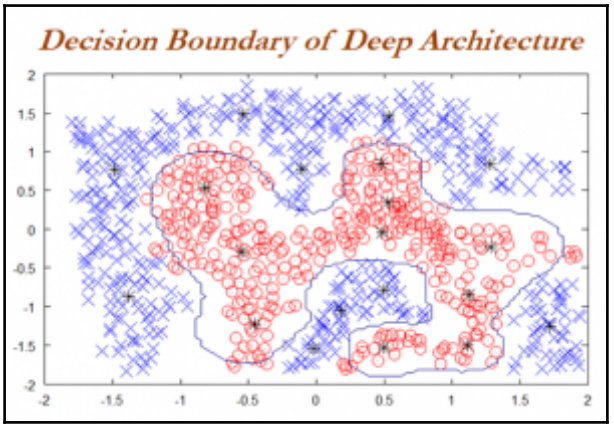

L-BFGS Convergence Methodology (quadratic approximation/second derivative/Hessian of function at each iteration)
Starting Point
Iteration 1
Iteration 2
f(x)
x
x_t
x_{t+1}
x_{t+2}

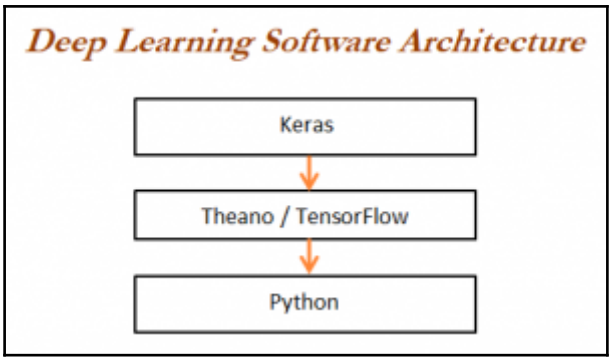

Application of Dropout in Neural Network
Standard Neural Net
After applying dropout


Printing first digit "0" in matrix form visually as image
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7

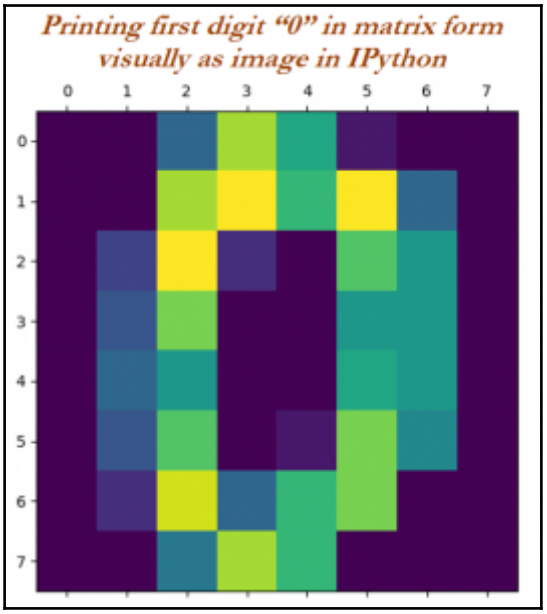

A.I., Machine Learning & Deep Learning
Artificial Intelligence
Machine Learning
Deep Learning


Deep Architecture of Neural Networks
input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

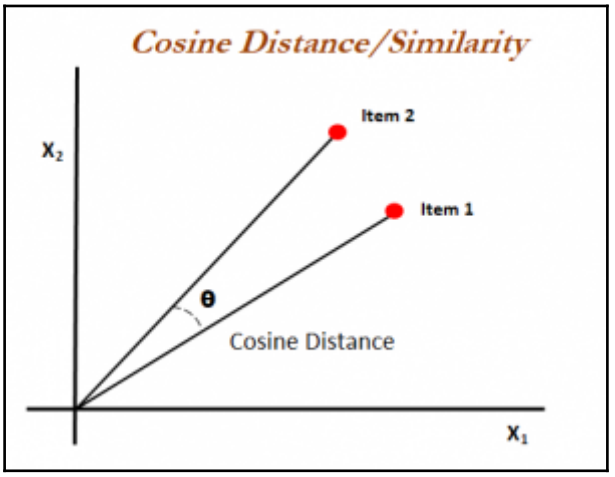

Decision Boundary of Deep Architecture
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

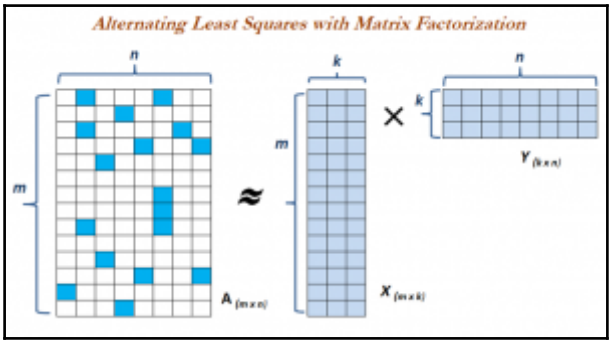



```
[global]
floatX = float32
device = gpu

[nvcc]
flags=-LC:\Users\prata\Anaconda2\libs
compiler_bindir=C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\bin

[lib]
cnmem = 0.8
```



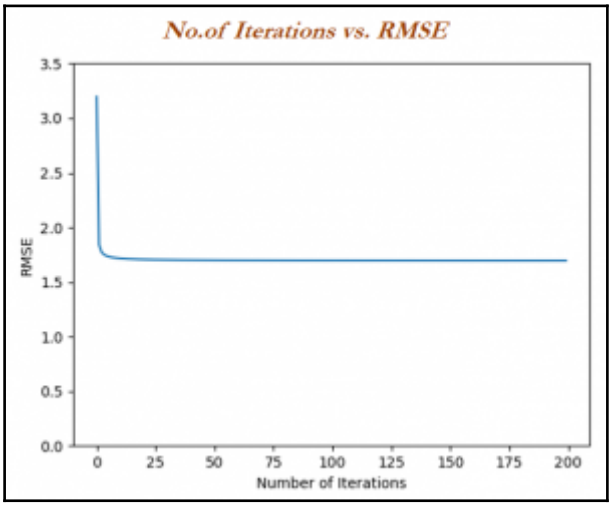
Chapter 7: Recommendation Engines



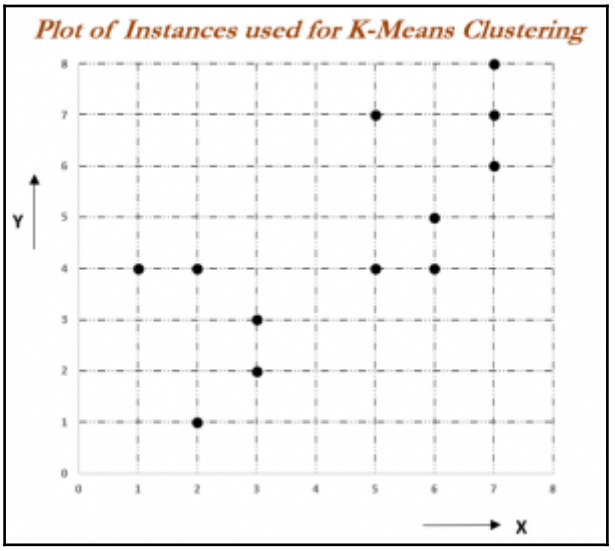


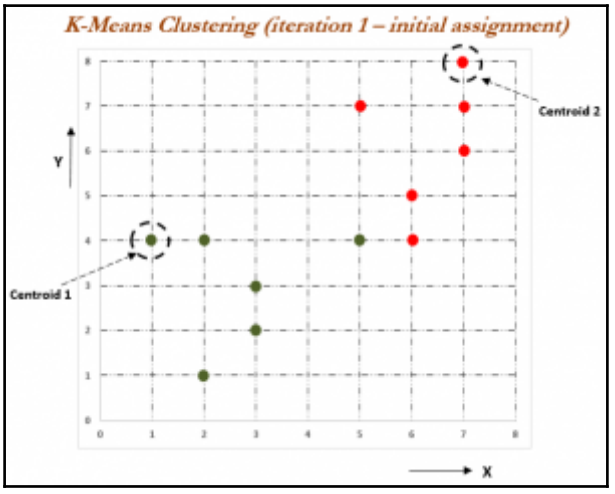
userId	movieId	rating	timestamp
1	31	2.5	1260759144
1	1029	3	1260759179
1	1061	3	1260759182
1	1129	2	1260759185
1	1172	4	1260759205

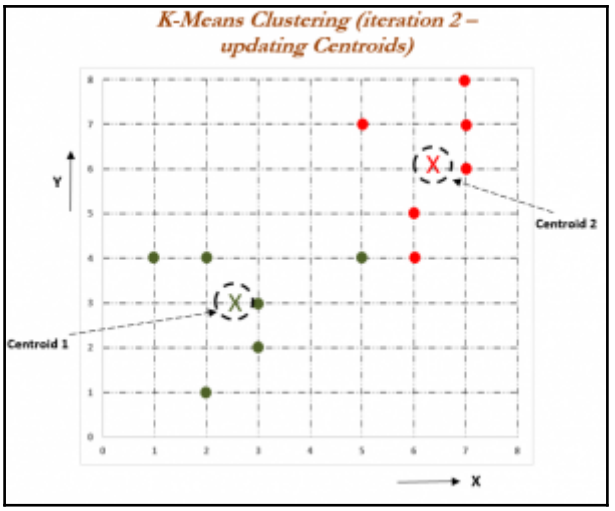
movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy

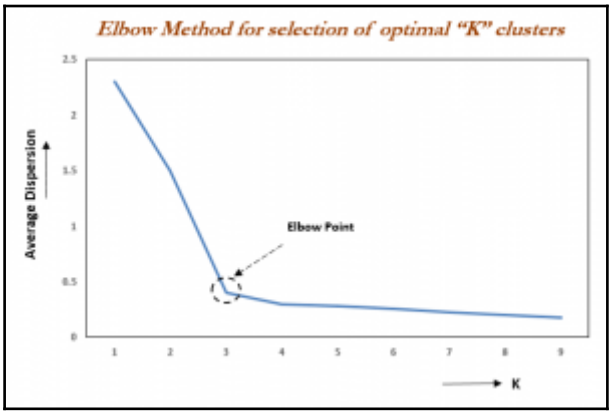


Chapter 8: Unsupervised Learning

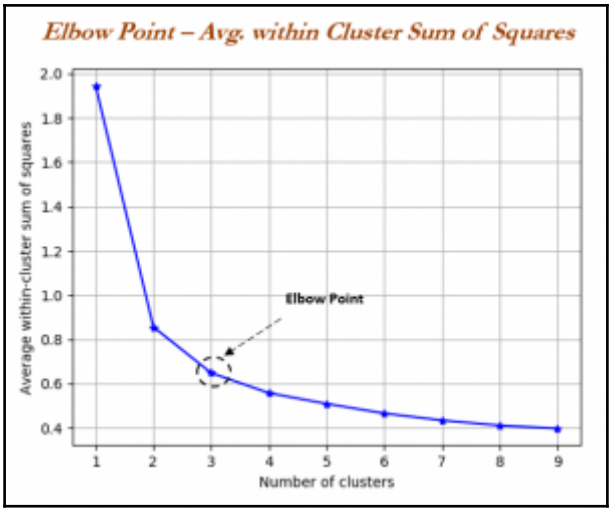


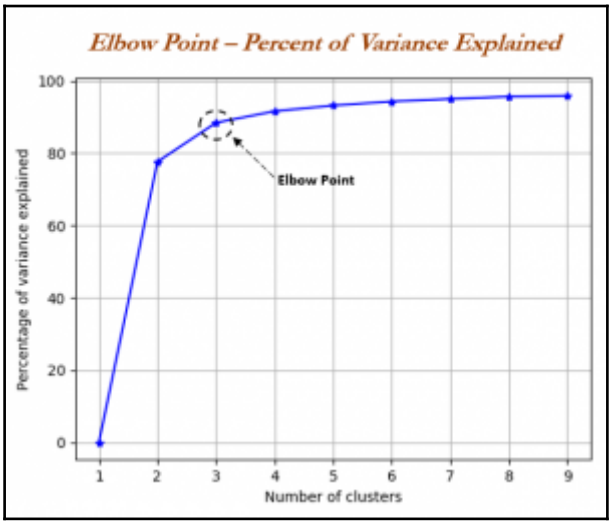


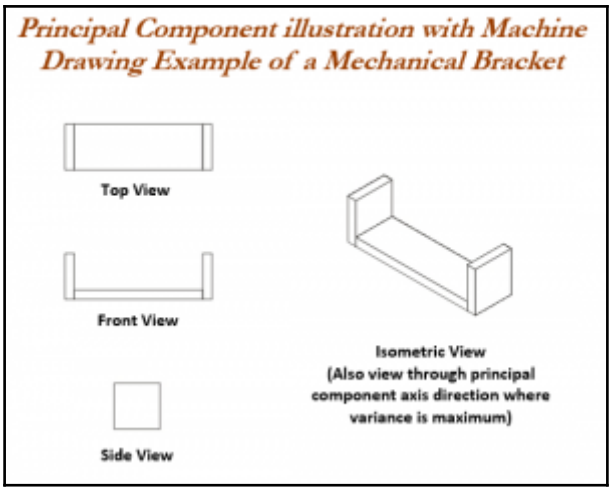


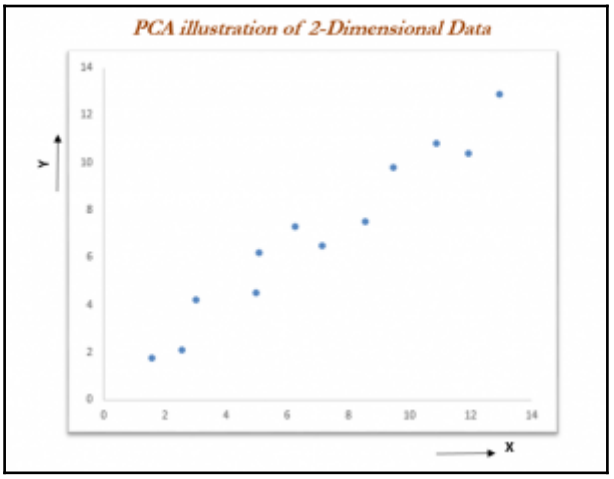


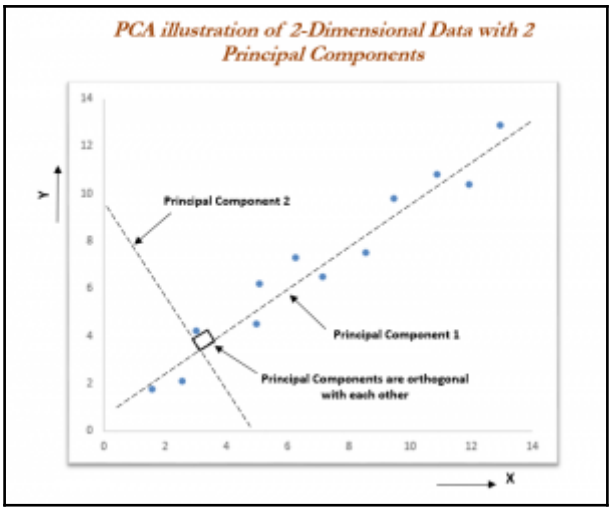
sepal_length	sepal_width	petal_length	petal_width	class
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa

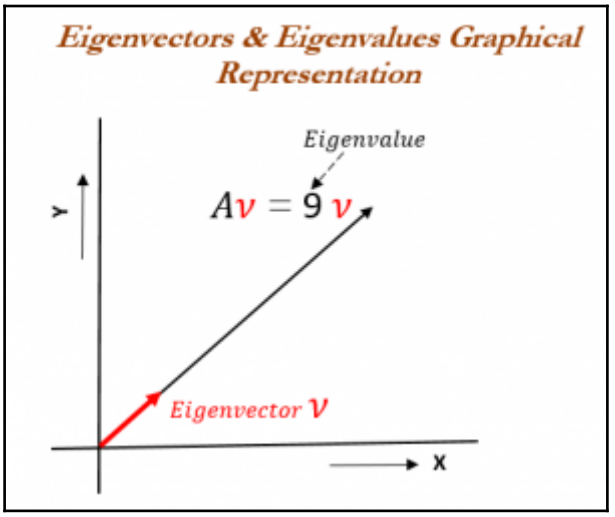

Elbow Point – Avg. within Cluster Sum of Squares
Average within-cluster sum of squares
Number of clusters
Elbow Point
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
1
2
3
4
5
6
7
8
9

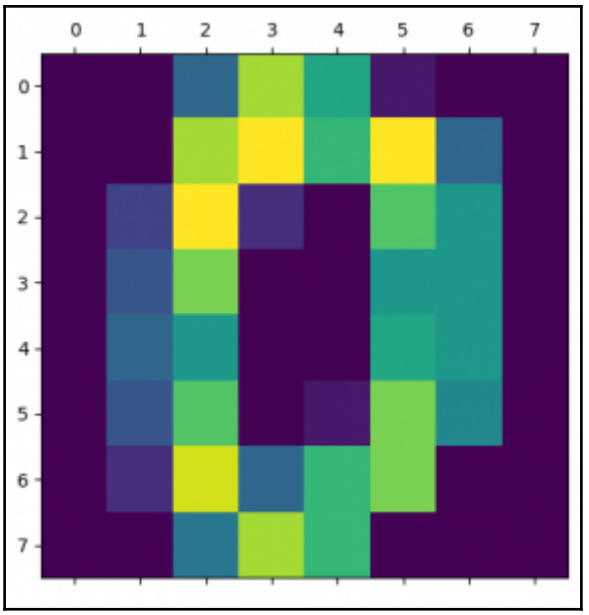

Elbow Point – Percent of Variance Explained
Percentage of variance explained
Number of clusters
Elbow Point
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9

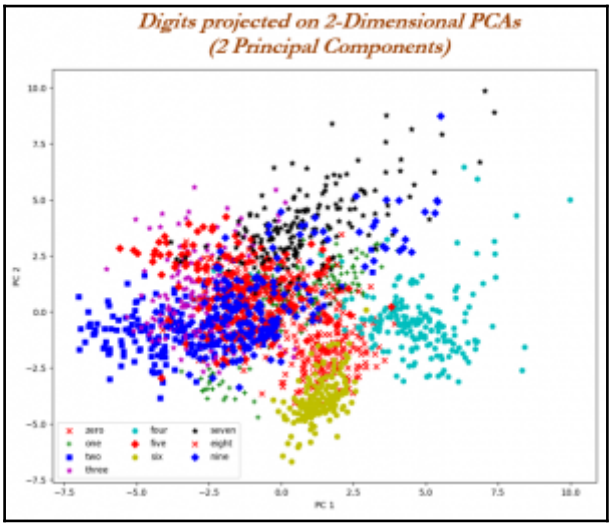

Principal Component illustration with Machine Drawing Example of a Mechanical Bracket
Top View
Front View
Side View
Isometric View
(Also view through principal component axis direction where variance is maximum)

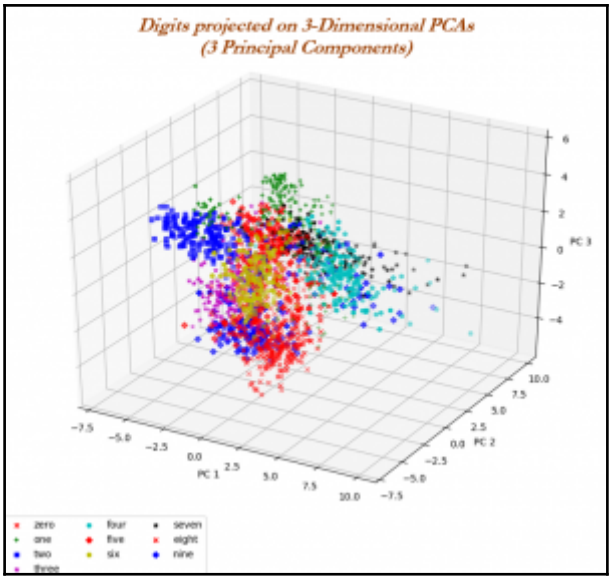

PCA illustration of 2-Dimensional Data
Y
X
0
2
4
6
8
10
12
14

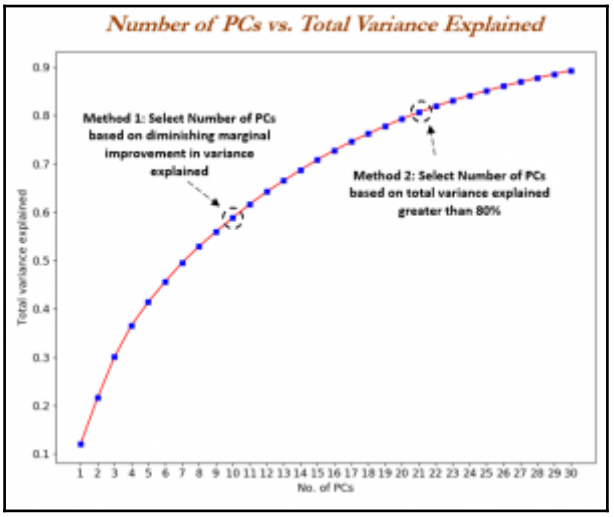

PCA illustration of 2-Dimensional Data with 2 Principal Components
Principal Component 2
Principal Component 1
Principal Components are orthogonal with each other
Y
X
0
2
4
6
8
10
12
14

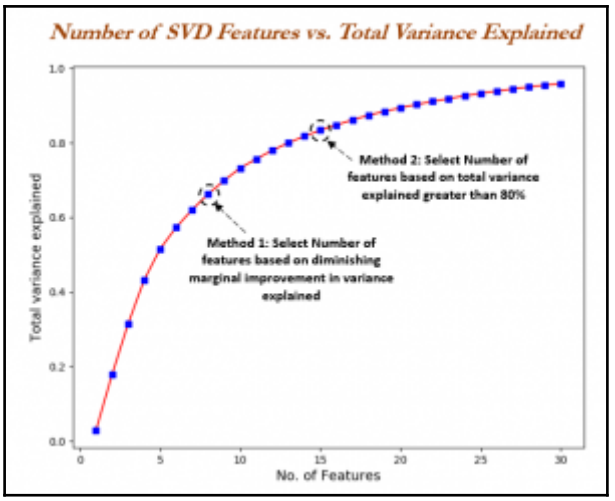

Eigenvectors & Eigenvalues Graphical Representation
Eigenvalue
$Av = 9v$
Eigenvector v
Y
X

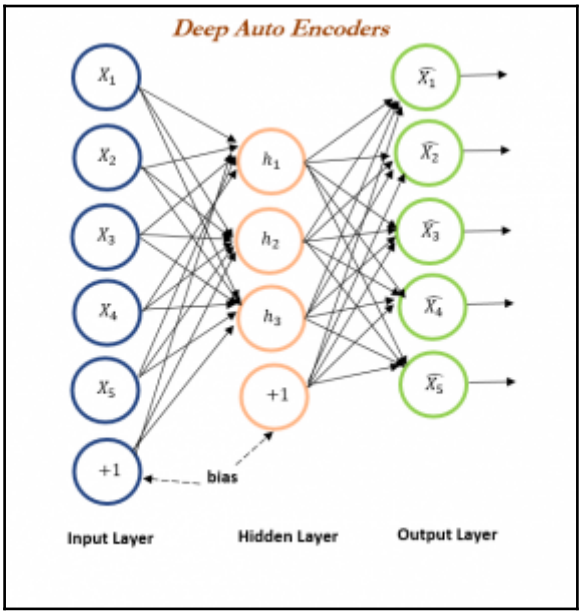

0
1
2
3
4
5
6
7

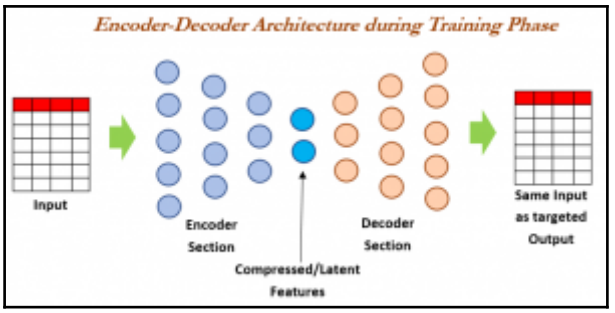

Digits projected on 2-Dimensional PCAs
(2 Principal Components)
PC 1
PC 2
zero
one
two
three
four
five
six
seven
eight
nine

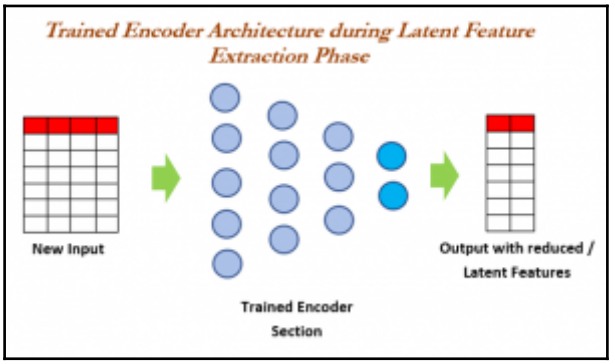

Digits projected on 3-Dimensional PCAs
(3 Principal Components)
PC 1
PC 2
PC 3
zero
one
two
three
four
five
six
seven
eight
nine

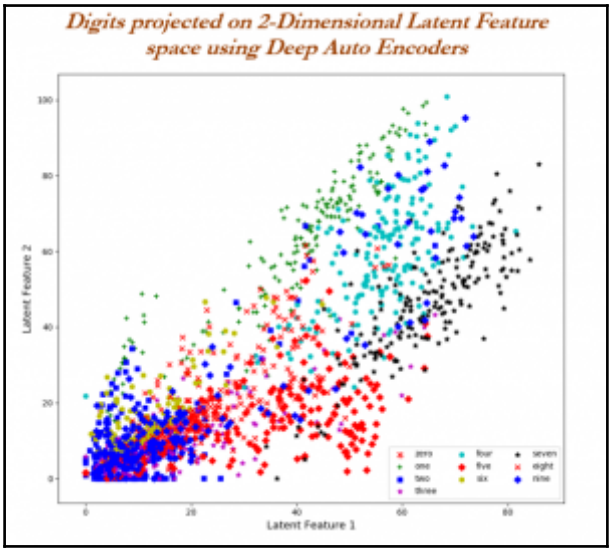

Number of PCs vs. Total Variance Explained
Method 1: Select Number of PCs based on diminishing marginal improvement in variance explained
Method 2: Select Number of PCs based on total variance explained greater than 80%
Total variance explained
No. of PCs

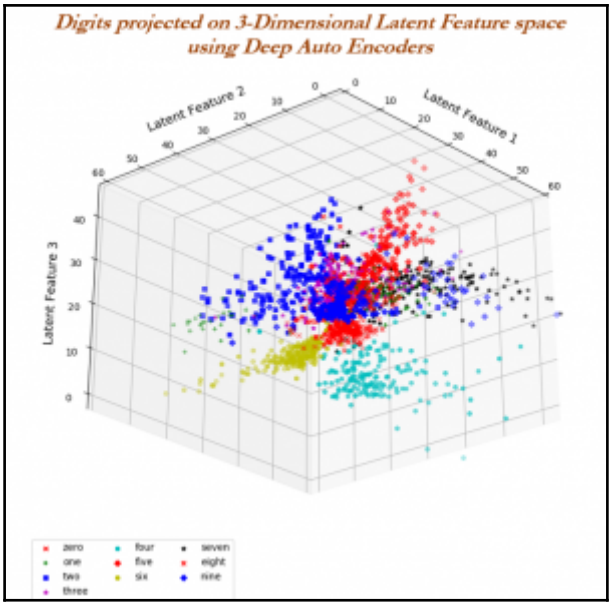

Number of SVD Features vs. Total Variance Explained
Total variance explained
No. of Features
Method 2: Select Number of features based on total variance explained greater than 80%
Method 1: Select Number of features based on diminishing marginal improvement in variance explained


Deep Auto Encoders
bias
Input Layer
Hidden Layer
Output Layer

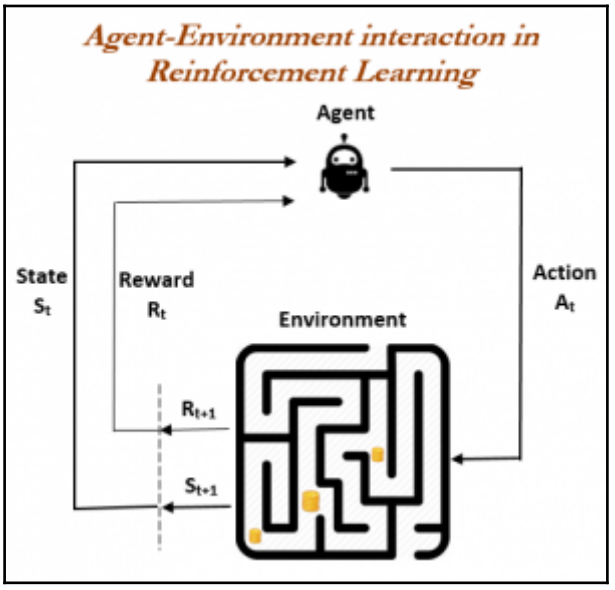

Encoder-Decoder Architecture during Training Phase
Input
Encoder Section
Compressed/Latent Features
Decoder Section
Same Input as targeted Output

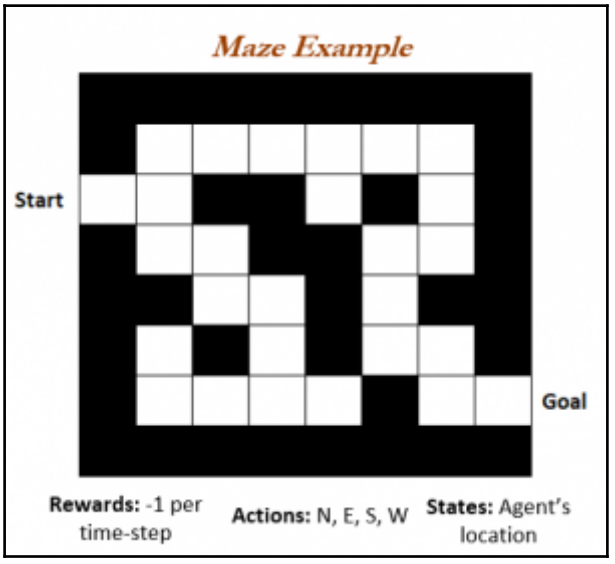

Trained Encoder Architecture during Latent Feature Extraction Phase
New Input
Trained Encoder Section
Output with reduced / Latent Features

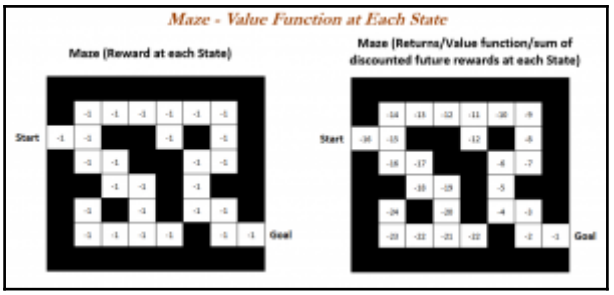

Digits projected on 2-Dimensional Latent Feature space using Deep Auto Encoders
Latent Feature 2
Latent Feature 1
zero
one
two
three
four
five
six
seven
eight
nine

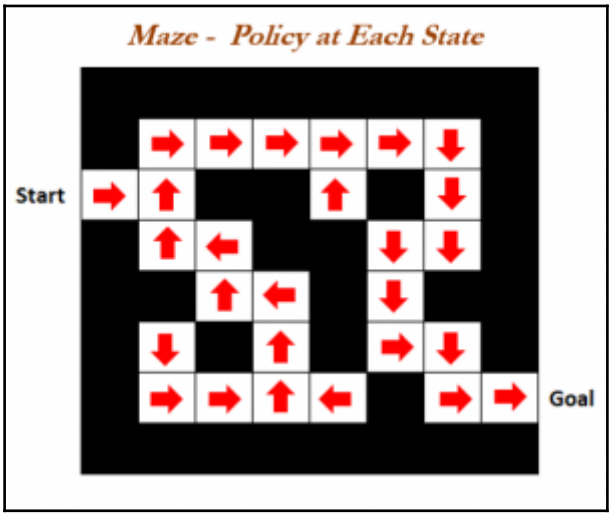

Digits projected on 3-Dimensional Latent Feature space
using Deep Auto Encoders
Latent Feature 1
Latent Feature 2
Latent Feature 3
zero
one
two
three
four
five
six
seven
eight
nine

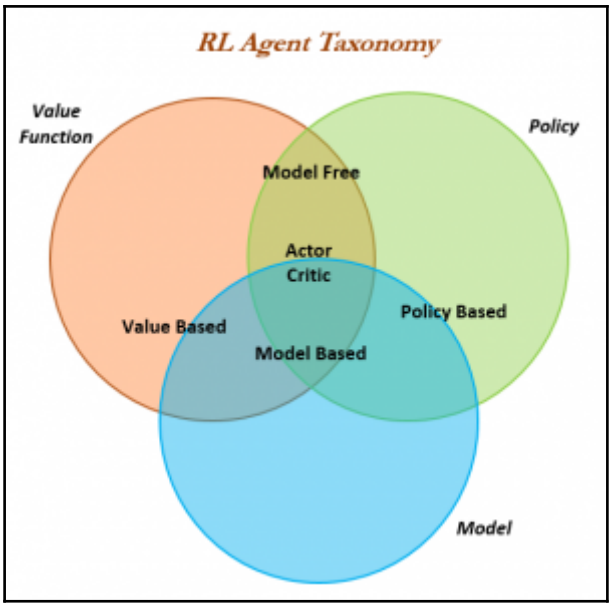
Chapter 9: Reinforcement Learning

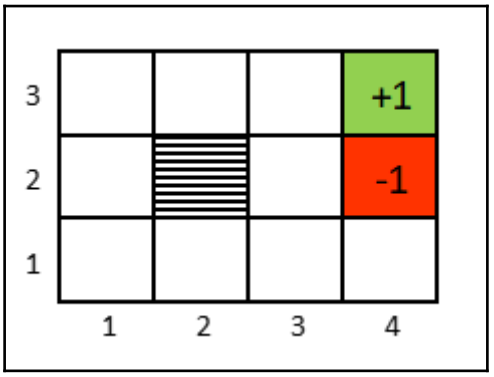


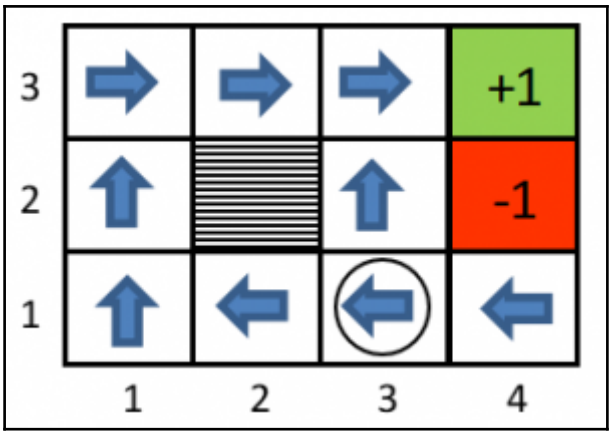


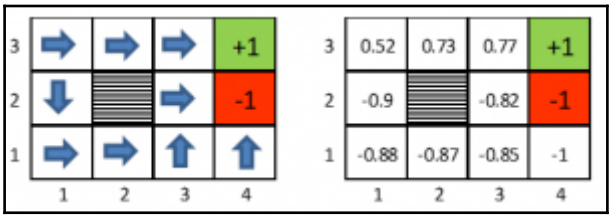

Maze - Value Function at Each State
Maze (Reward at each State)
Start
Goal
Maze (Returns/Value function/sum of discounted future rewards at each State)
Start
Goal

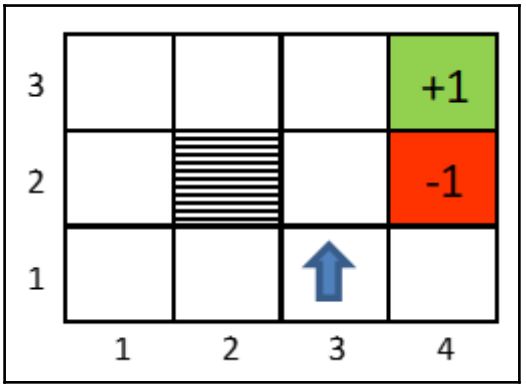

Maze - Policy at Each State
Start
Goal

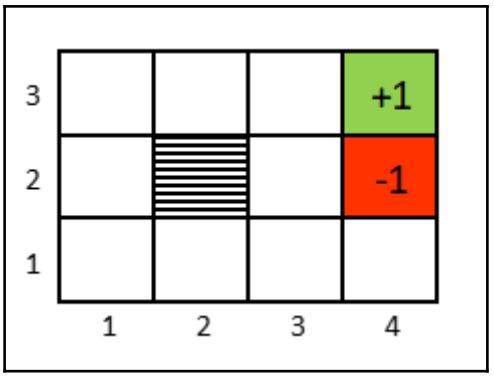

RL Agent Taxonomy
Value Function
Policy
Model Free
Actor Critic
Policy Based
Value Based
Model Based
Model

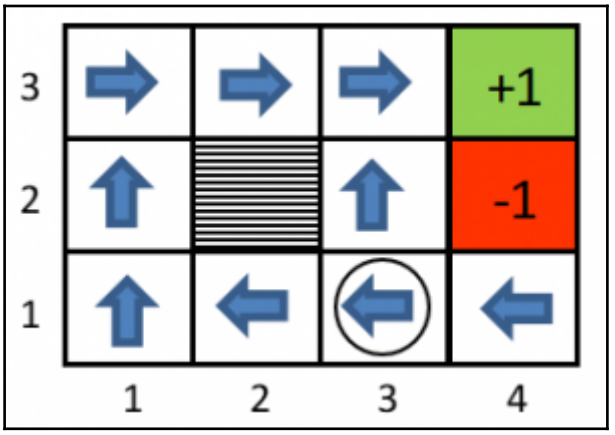

3
2
1
+1
-1
1
2
3
4

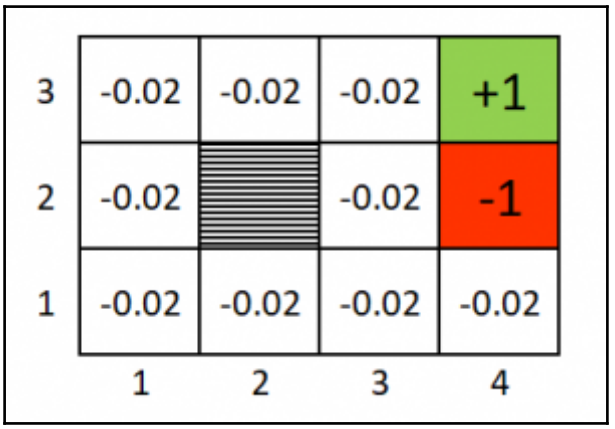

3
2
1
+1
-1
1
2
3
4

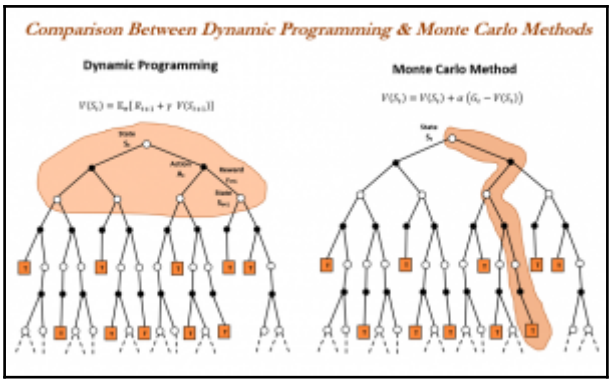

3
2
1
+1
-1
1
2
3
4
0.52
0.73
0.77
+1
-0.9
-0.82
-1
-0.88
-0.87
-0.85
-1

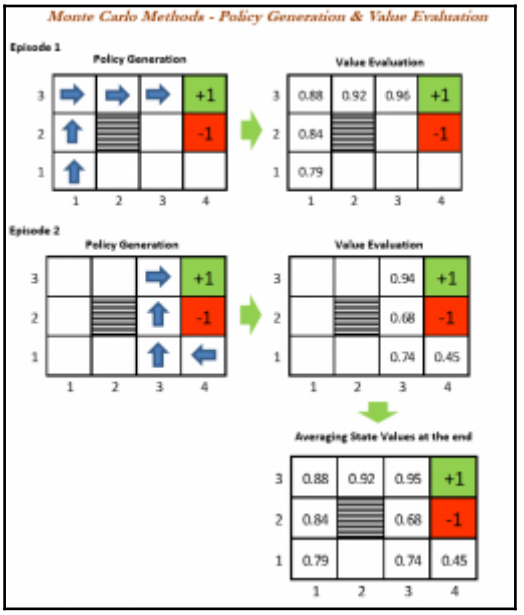

3
2
1
+1
-1
1
2
3
4


3
2
1
+1
-1
1
2
3
4


3
2
1
+1
-1
1
2
3
4


3
2
1
-0.02
-0.02
-0.02
+1
-0.02
-0.02
-1
-0.02
-0.02
-0.02
-0.02
1
2
3
4

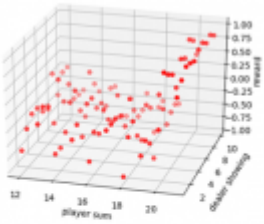

Comparison Between Dynamic Programming & Monte Carlo Methods
Dynamic Programming
Monte Carlo Method


Monte Carlo Methods - Policy Generation & Value Evaluation
Episode 1
Policy Generation
Value Evaluation
Episode 2
Policy Generation
Value Evaluation
Averaging State Values at the end

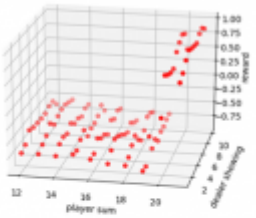
Black Jack Example – Monte Carlo Methods

Approximate state-value functions for Blackjack policy that sticks only on 20 or 21, computed by MC Policy Evaluation

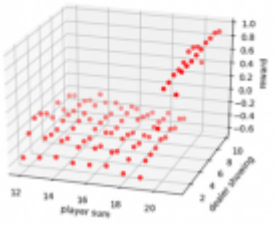
Usable Ace & 10000 Episodes



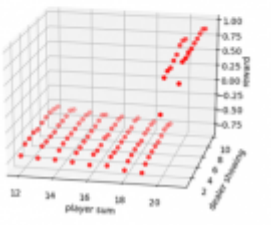
No Usable Ace & 10000 Episodes

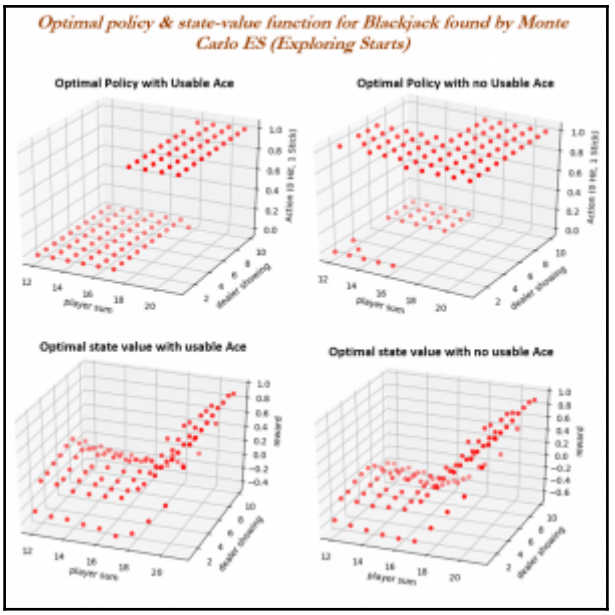


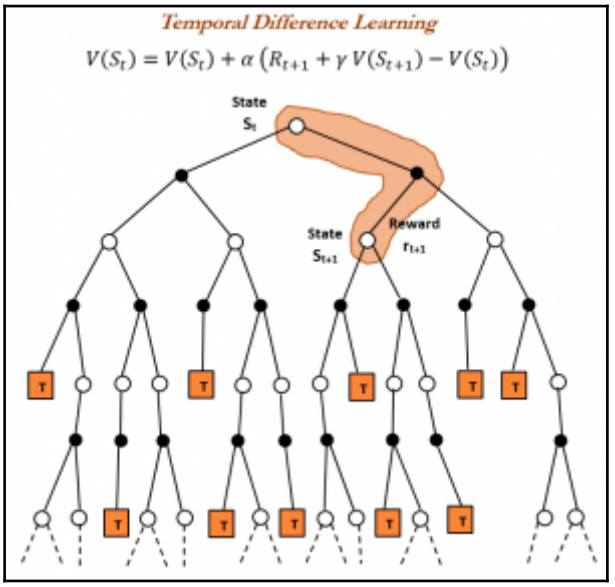
Usable Ace & 500000 Episodes



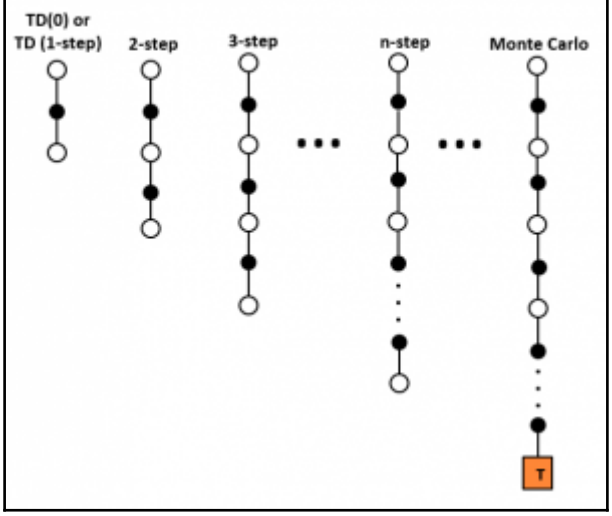
No Usable Ace & 500000 Episodes




Optimal policy & state-value function for Blackjack found by Monte Carlo ES (Exploring Starts)
Optimal Policy with Usable Ace
Optimal Policy with no Usable Ace
Optimal state value with usable Ace
Optimal state value with no usable Ace
player sum
dealer showing
reward

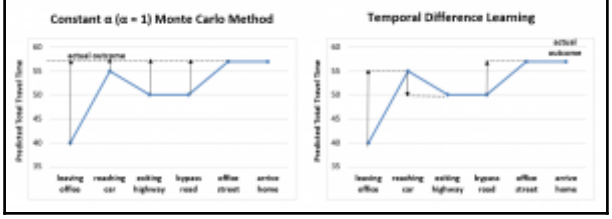

Temporal Difference Learning
$V(S_t) = V(S_t) + \alpha\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$
State
S_t
State
S_{t+1}
Reward
r_{t+1}
T

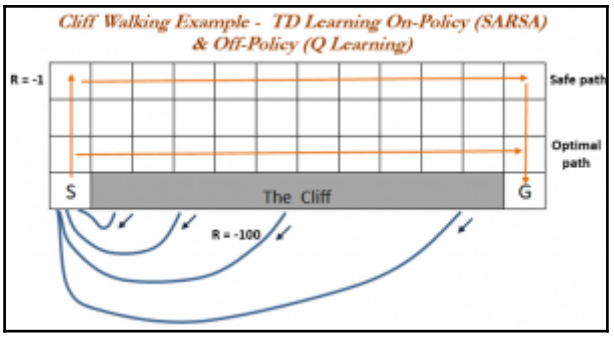
Comparison of TD Learning vs. MC Method

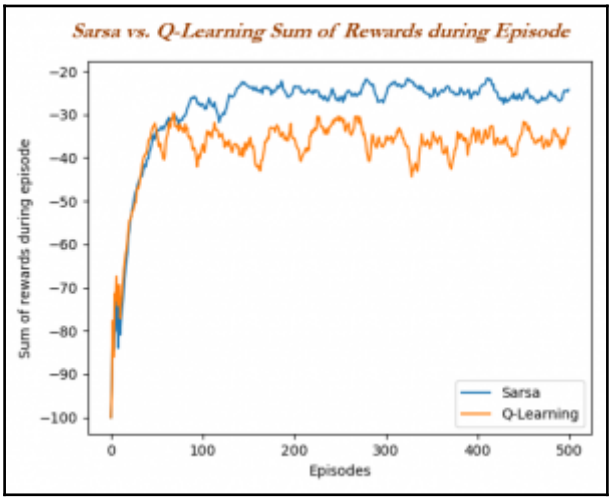


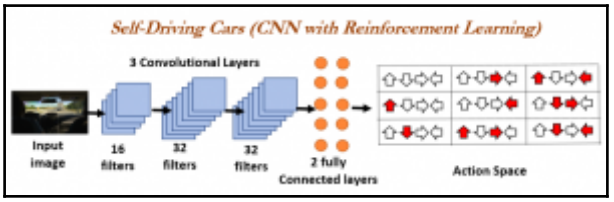
State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving home, Monday at 8 a.m.	0	40	40
reaching car, minister convoy passes by	10	45	55
exiting highway	25	25	50
by pass road, behind bullockcarts	30	20	50
entering office street	50	7	57
arrive office parking space	57	0	57

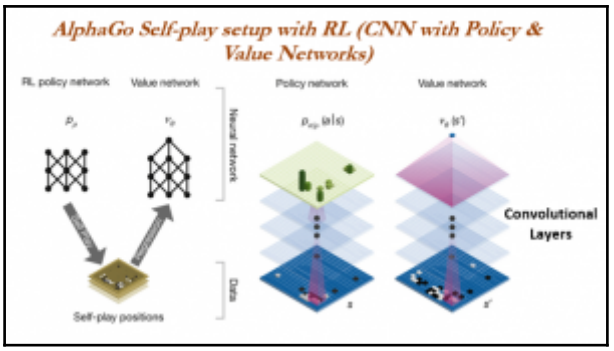
MC Method vs. TD Learning for Office Driving Example

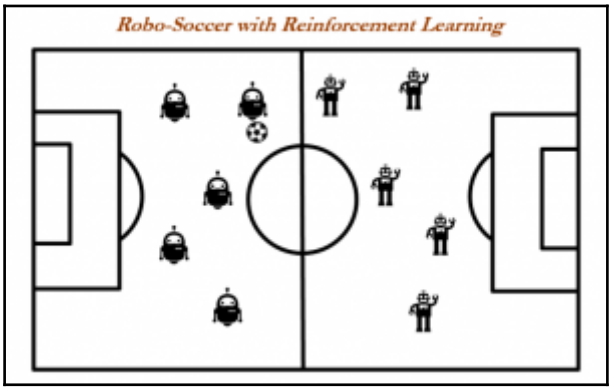



Cliff Walking Example - TD Learning On-Policy (SARSA) & Off-Policy (Q Learning)
R = -1
Safe path
Optimal path
S
The Cliff
G
R = -100


Sarsa vs. Q-Learning Sum of Rewards during Episode
Sum of rewards during episode
Episodes
Sarsa
Q-Learning


Self-Driving Cars (CNN with Reinforcement Learning)
3 Convolutional Layers
Input image
16 filters
32 filters
32 filters
2 fully Connected layers
Action Space


AlphaGo Self-play setup with RL (CNN with Policy & Value Networks)
RL policy network
Value network
Policy network
Value network
Neural network
Data
Convolutional Layers
Self-play positions


Robo-Soccer with Reinforcement Learning