Prashanth Manja

USC ID: 3073150764

ML- Assignment -3

Data Imputation technique to deal with the missing values in the data set:

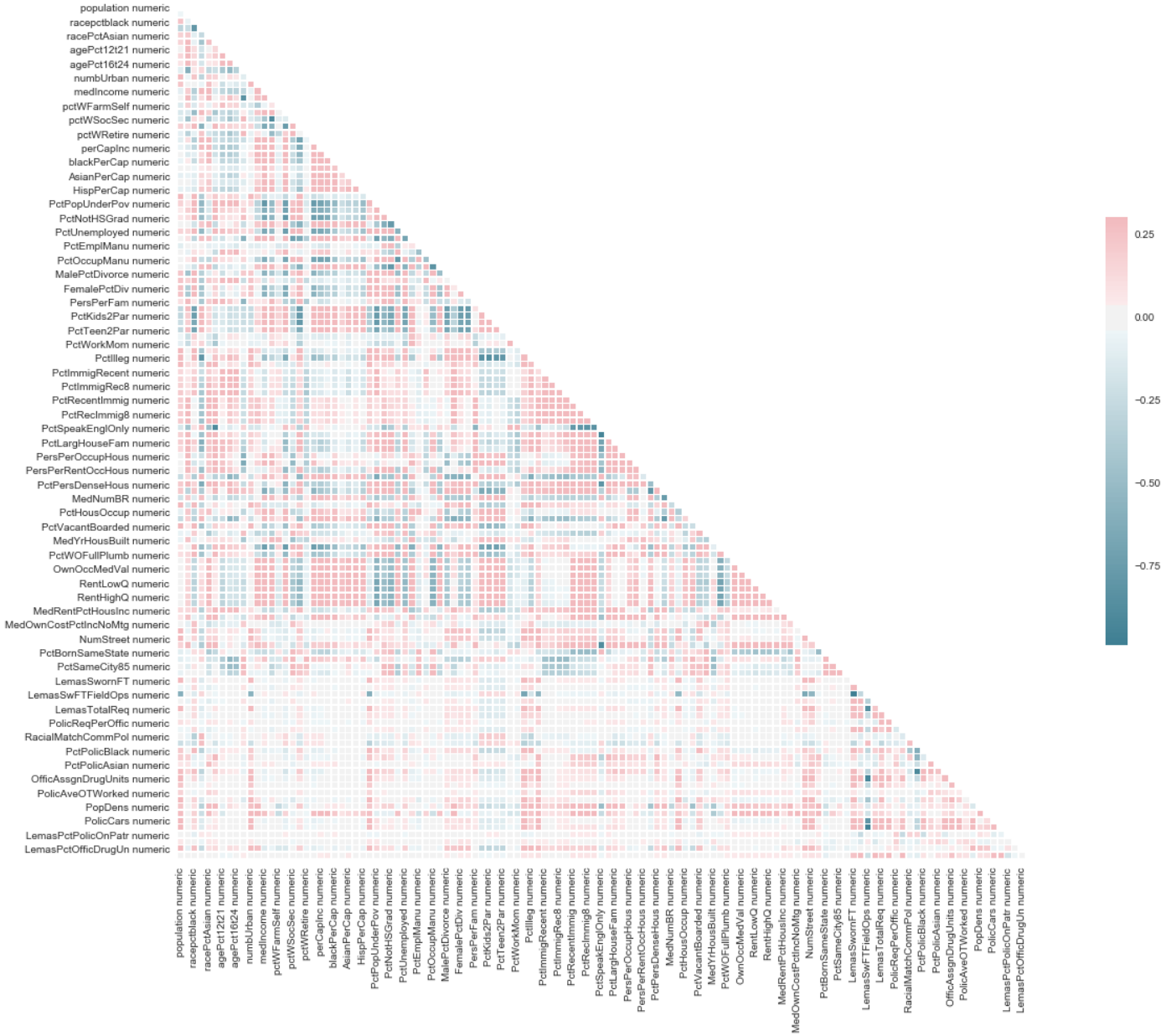
Imputation is replacing missing values with substitute values.

The following are common methods:

- Mean: the mean of the observed values for that variable
- Substitution: the value from a new individual who was not selected to be in the sample
- Hot deck: a randomly chosen value from an individual who has similar values on other variables
- Cold deck: a systematically chosen value from an individual who has similar values on other variables
- Regression: the predicted value obtained by regressing the missing variable on other variables
- Stochastic regression: the predicted value from a regression plus a random residual value.
- Interpolation and extrapolation: an estimated value from other observations from the same individual.

Correlation Matrix – 1:

Correlation Matrix – 2:

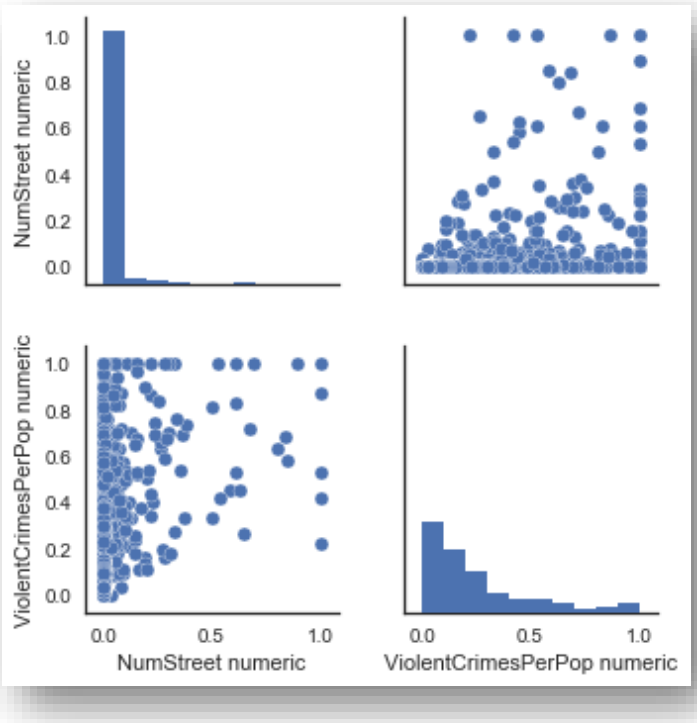


Coefficient of Variation of 11 features with highest CV :

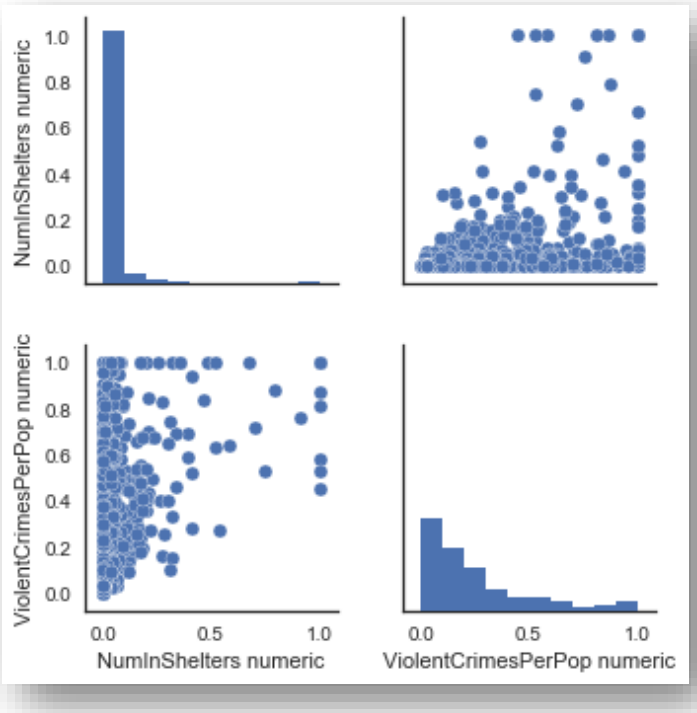
	x		y
0	NumStreet	numeric	4.291487
1	NumInShelters	numeric	3.469791
2	NumIlleg	numeric	3.057941
3	NumImmig	numeric	2.925656
4	LemasPctOfficDrugUn	numeric	2.552092
5	NumUnderPov	numeric	2.341660
6	population	numeric	2.240355
7	numbUrban	numeric	2.037780
8	HousVacant	numeric	1.967809
9	LandArea	numeric	1.644857
10	racePctHisp	numeric	1.611552

Scatter Plots:

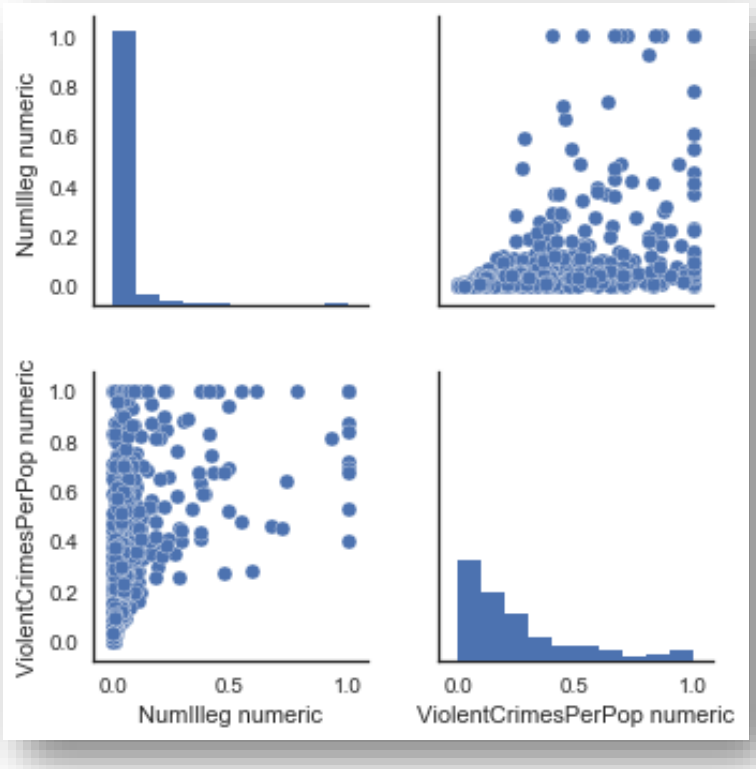
1.NumStreet numeric



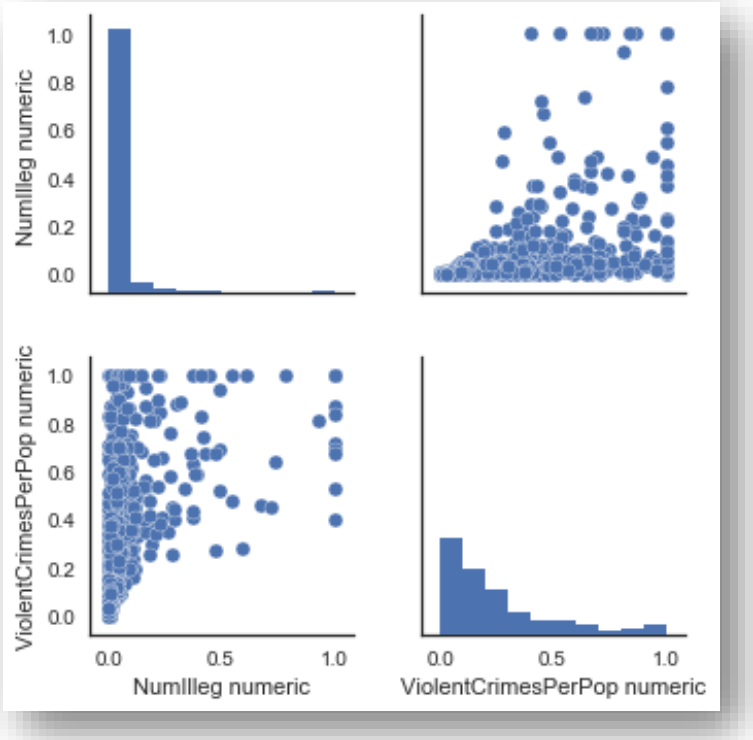
2. NumInShelters numeric



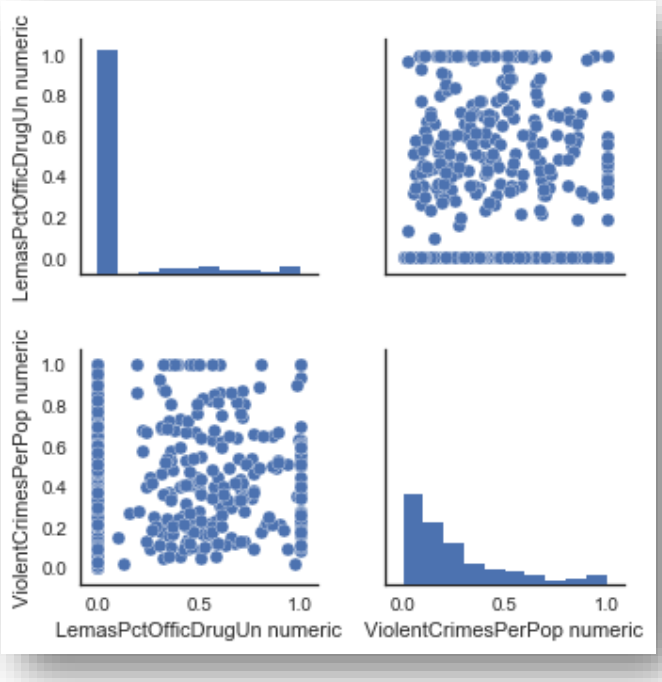
3. **NumIlleg numeric**



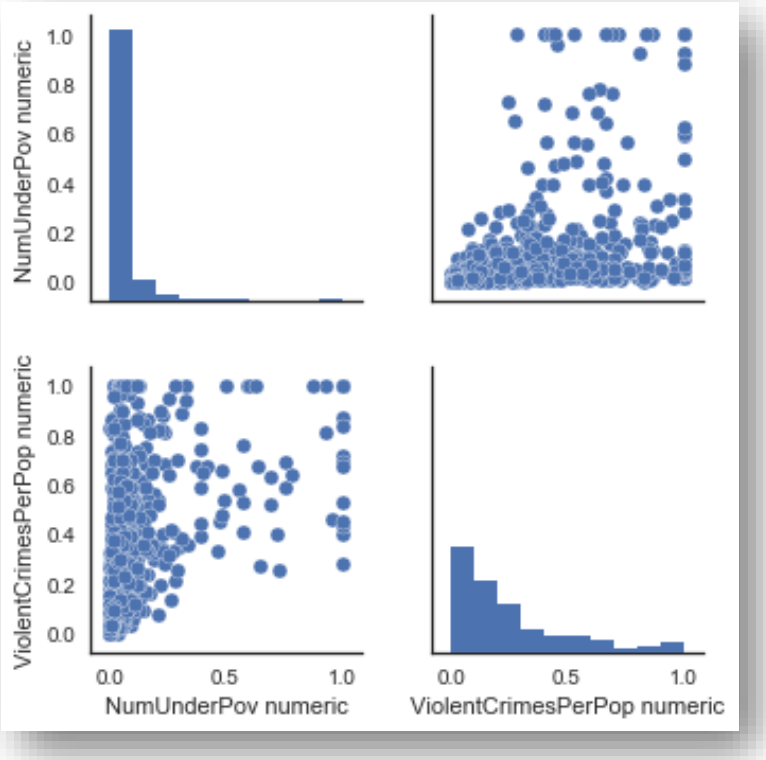
4. **NumImmig numeric**



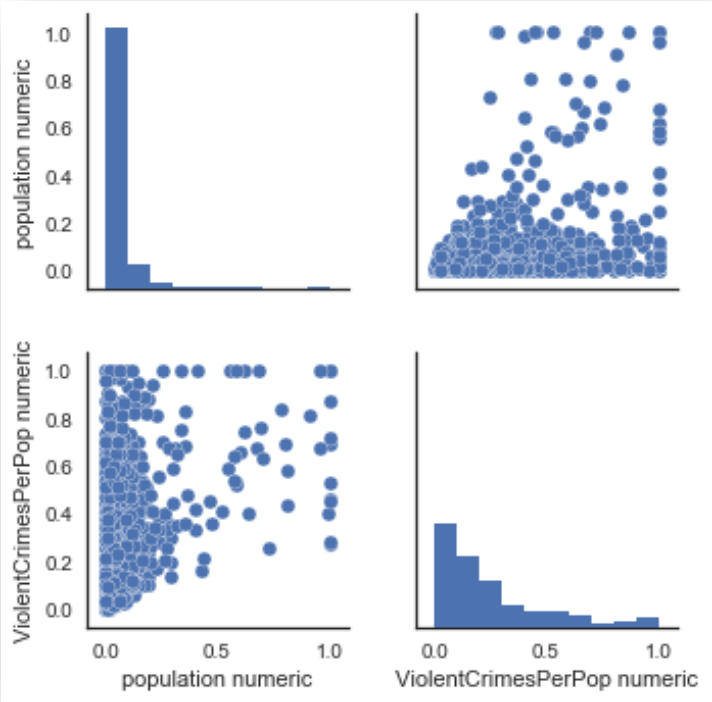
5. **LemasPctOfficDrugUn numeric**



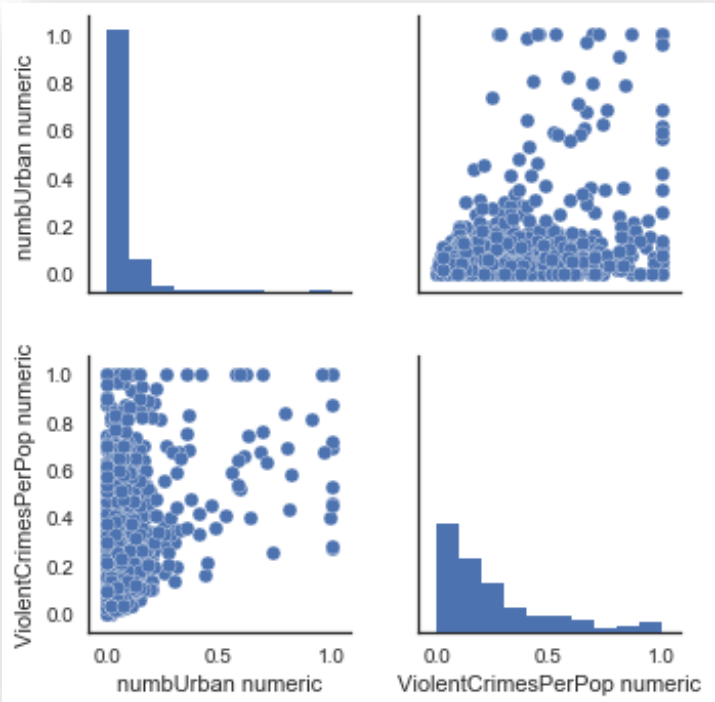
6. **NumUnderPov numeric**



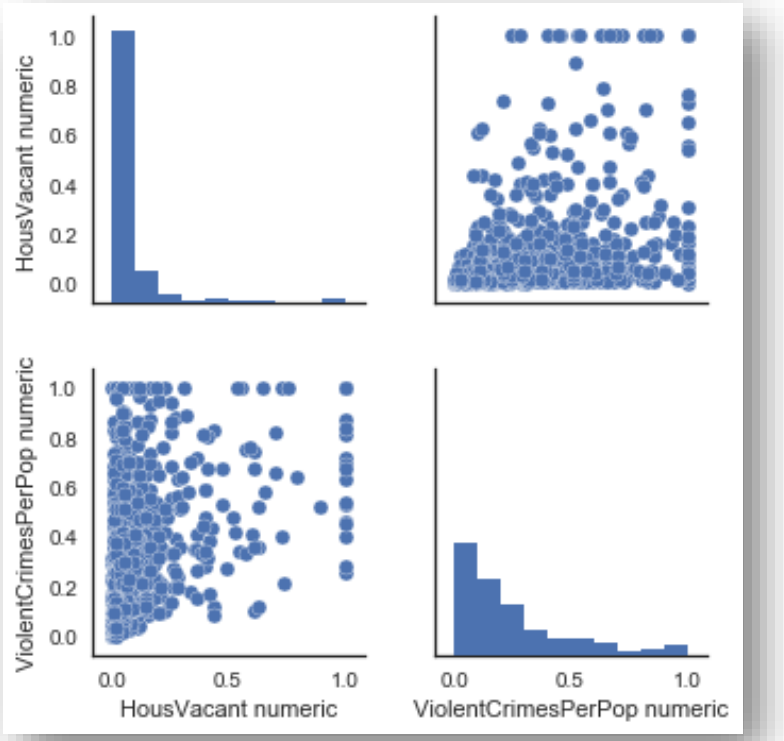
7. **population numeric**



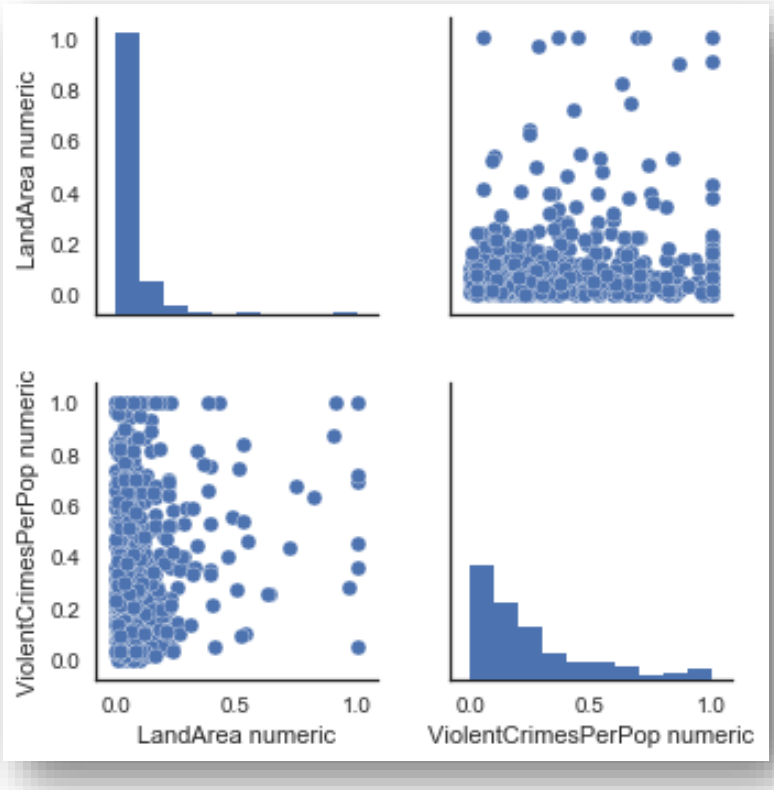
8. **numbUrban numeric**



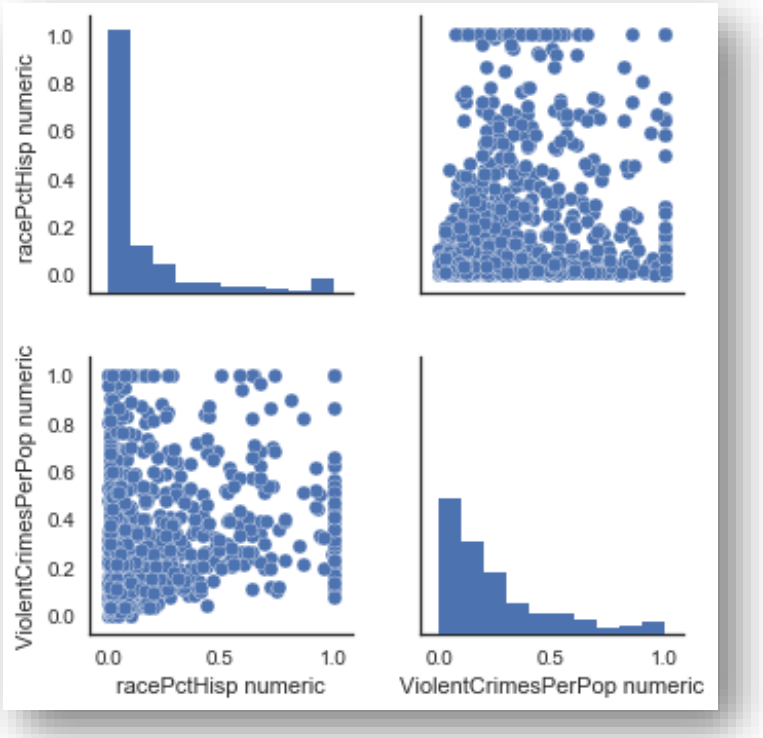
9. **HousVacant numeric**



10. **LandArea numeric**

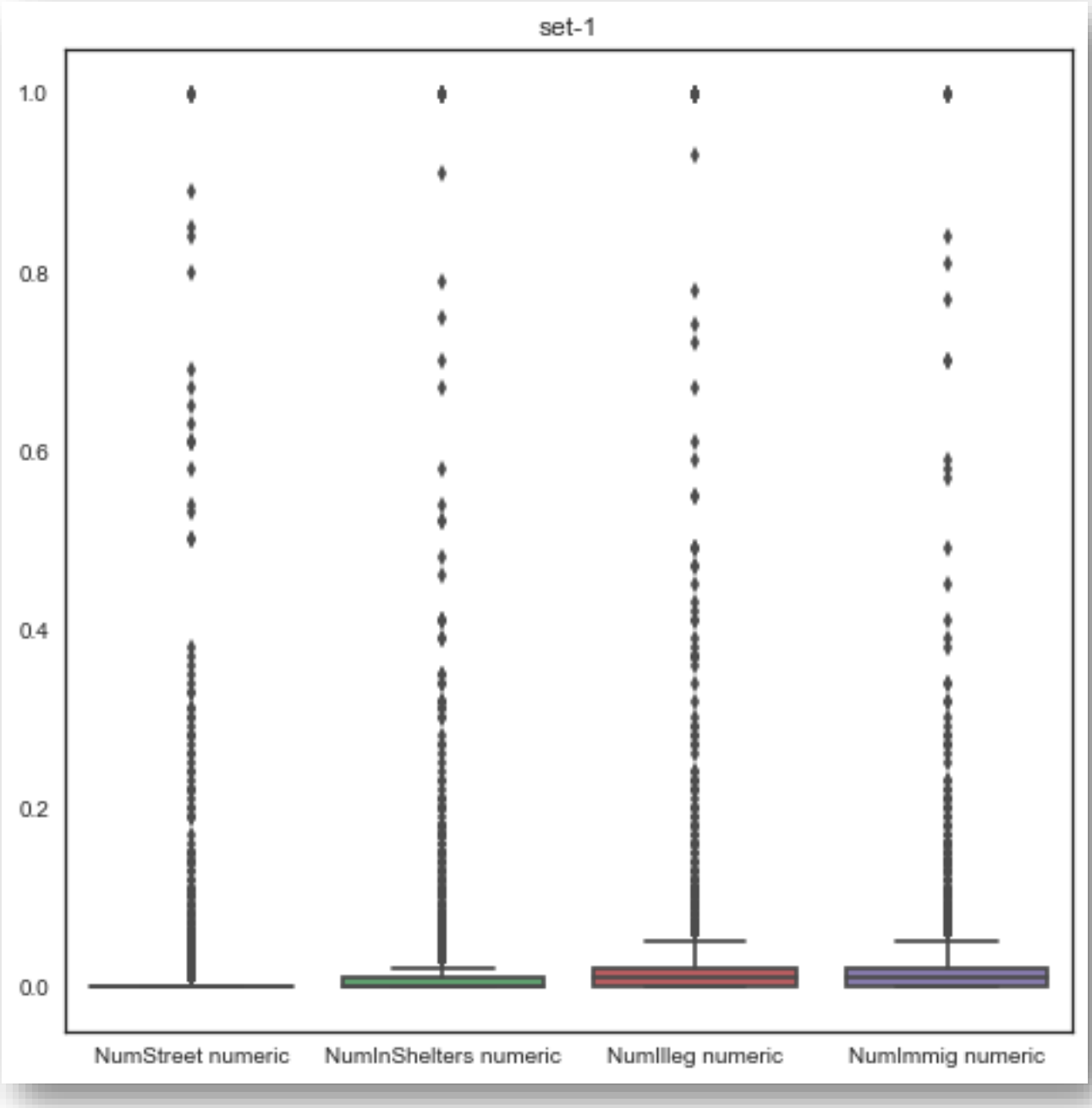


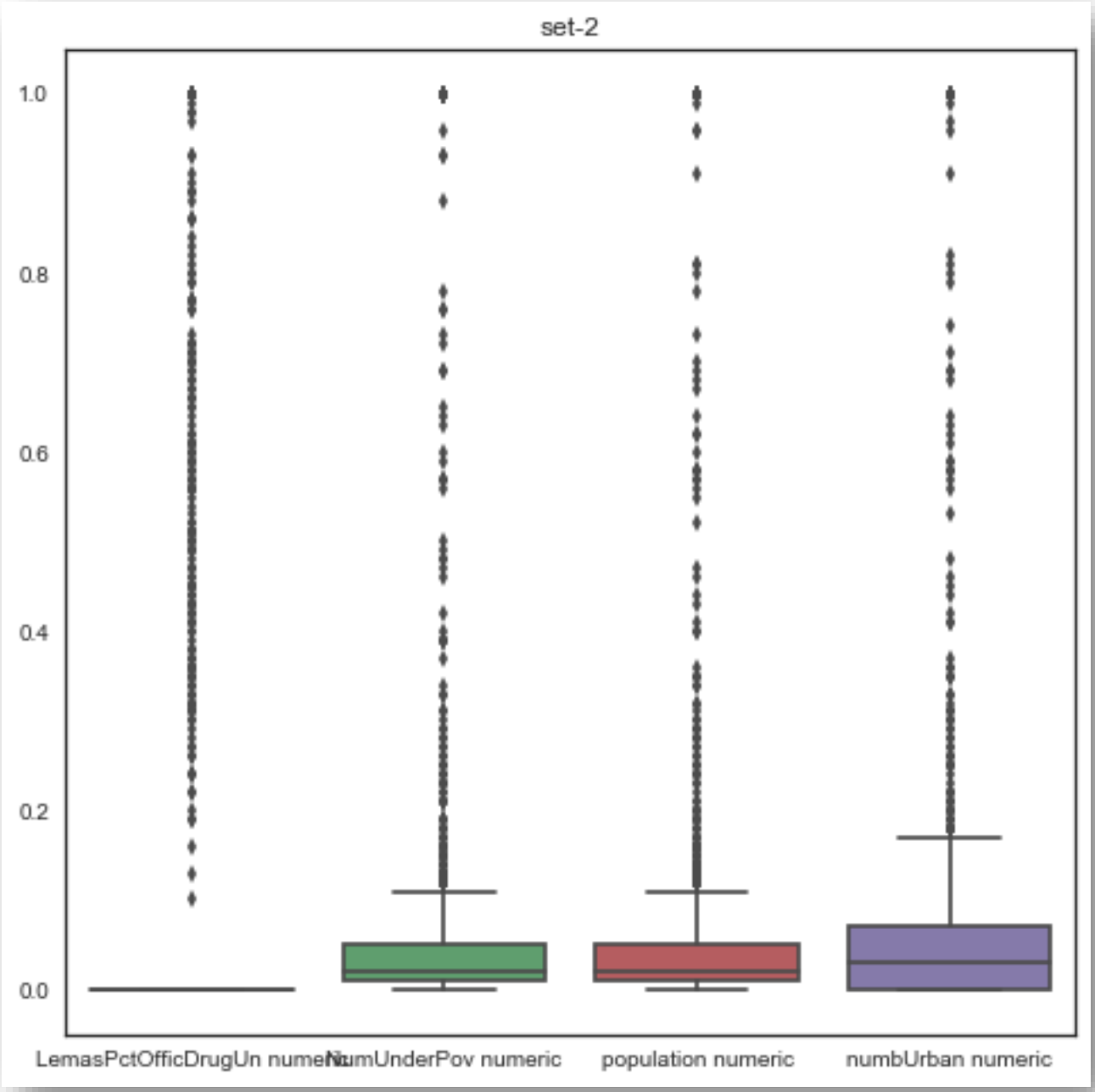
11. **racePctHisp numeric**

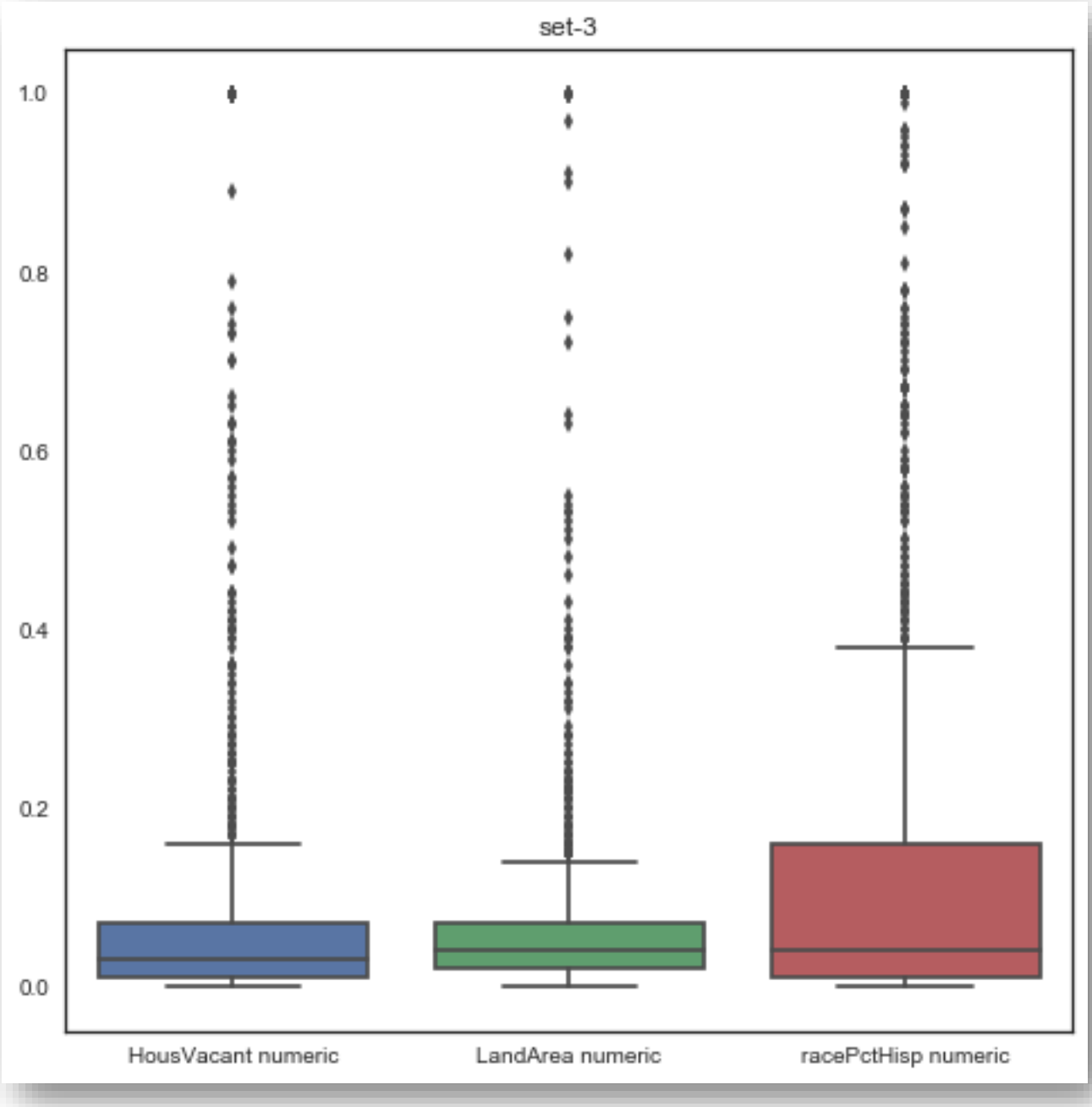


linear relationship between X and Y exists when the pattern of X– and Y-values resembles a line, either uphill (with a positive slope) or downhill (with a negative slope), but looking at the pair wise-scatter plots we can't conclude anything concert about the significance about any feature solely based on scatter plots.

Box-Plots:


set-1
1.0
0.8
0.6
0.4
0.2
0.0
NumStreet numeric
NumInShelters numeric
NumIlleg numeric
NumImmig numeric


set-2
1.0
0.8
0.6
0.4
0.2
0.0
LemasPctOfficDrugUn numeric
NumUnderPov numeric
population numeric
numbUrban numeric


set-3
1.0
0.8
0.6
0.4
0.2
0.0
HousVacant numeric
LandArea numeric
racePctHisp numeric

Linear Regression:
Linear model using least squares to the training set and report the test error.
Mean squared error: **0.79**

RIDGE REGRESSION:

Best value for alpha chosen over cross-validation is: **0.0466301673441609** (**.046)**

Ridge Intercept: **[0.43699573]**
Mean Square error: ViolentCrimesPerPop numeric **0.01803**
Ridge Regression Score: **0.6208609551305065**

LASSO Regression:

LASSO INTERCEPT**: [0.42228093]**
MEAN SQUARE ERROR: **0.08361866691992706**
LASSO SCORE: **0.6262327573680131**

LASSO Regression with Normalised Features:

LASSO INTERCEPT**:[0.42875423]**
MEAN SQUARE ERROR**:0.08348299733538181**
LASSO SCORE**:0.6251347416963564**

PCR :

pca. variance_ratio:

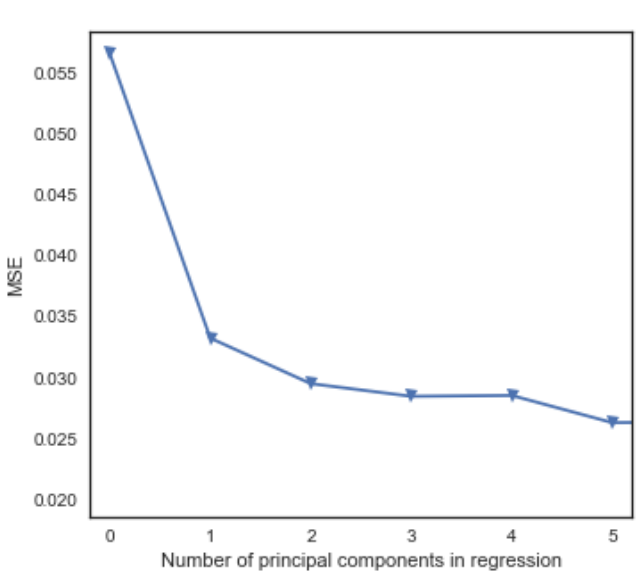
```
array([ 21.21,  35.39,  44.18,  51.08,  56.83,  60.81,  64.17,  67.26,
        69.81,  71.86,  73.56,  75.03,  76.44,  77.78,  79.02,  80.12,
        81.18,  82.09,  82.95,  83.8 ,  84.6 ,  85.35,  86.08,  86.79,
        87.47,  88.1 ,  88.7 ,  89.27,  89.8 ,  90.28,  90.76,  91.22,
        91.65,  92.06,  92.46,  92.84,  93.2 ,  93.55,  93.88,  94.2 ,
        94.51,  94.8 ,  95.08,  95.33,  95.57,  95.81,  96.02,  96.23,
        96.43,  96.61,  96.79,  96.96,  97.12,  97.28,  97.43,  97.57,
        97.7 ,  97.83,  97.96,  98.07,  98.18,  98.29,  98.39,  98.48,
        98.57,  98.65,  98.73,  98.81,  98.88,  98.95,  99.01,  99.07,
        99.13,  99.18,  99.23,  99.28,  99.33,  99.37,  99.41,  99.45,
        99.49,  99.53,  99.56,  99.59,  99.62,  99.65,  99.68,  99.71,
        99.74,  99.76,  99.78,  99.8 ,  99.82,  99.84,  99.86,  99.88,
        99.89,  99.9 ,  99.91,  99.92,  99.93,  99.94,  99.95,  99.96,
        99.97,  99.98,  99.99, 100.  , 100.  , 100.  , 100.  , 100.  ,
```

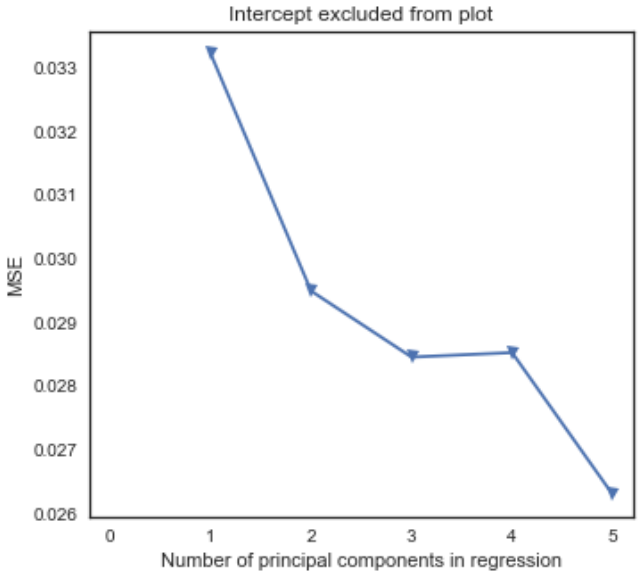
```
100.  , 100.  , 100.  , 100.  , 100.  , 100.  , 100.  , 100.  ,
100.  , 100.  ])
```

The initial 50% of the components explain most of the variance in the data

Number of principal components in regression:

(we can compare MSE as we change the 'M' -number of Principal components)





2. Tree-Based Methods:

The following are common methods:

- Mean: the mean of the observed values for that variable
- Substitution: the value from a new individual who was not selected to be in the sample
- Hot deck: a randomly chosen value from an individual who has similar values on other variables
- Cold deck: a systematically chosen value from an individual who has similar values on other variables
- Regression: the predicted value obtained by regressing the missing variable on other variables
- Stochastic regression: the predicted value from a regression plus a random residual value.
- Interpolation and extrapolation: an estimated value from other observations from the same individual.

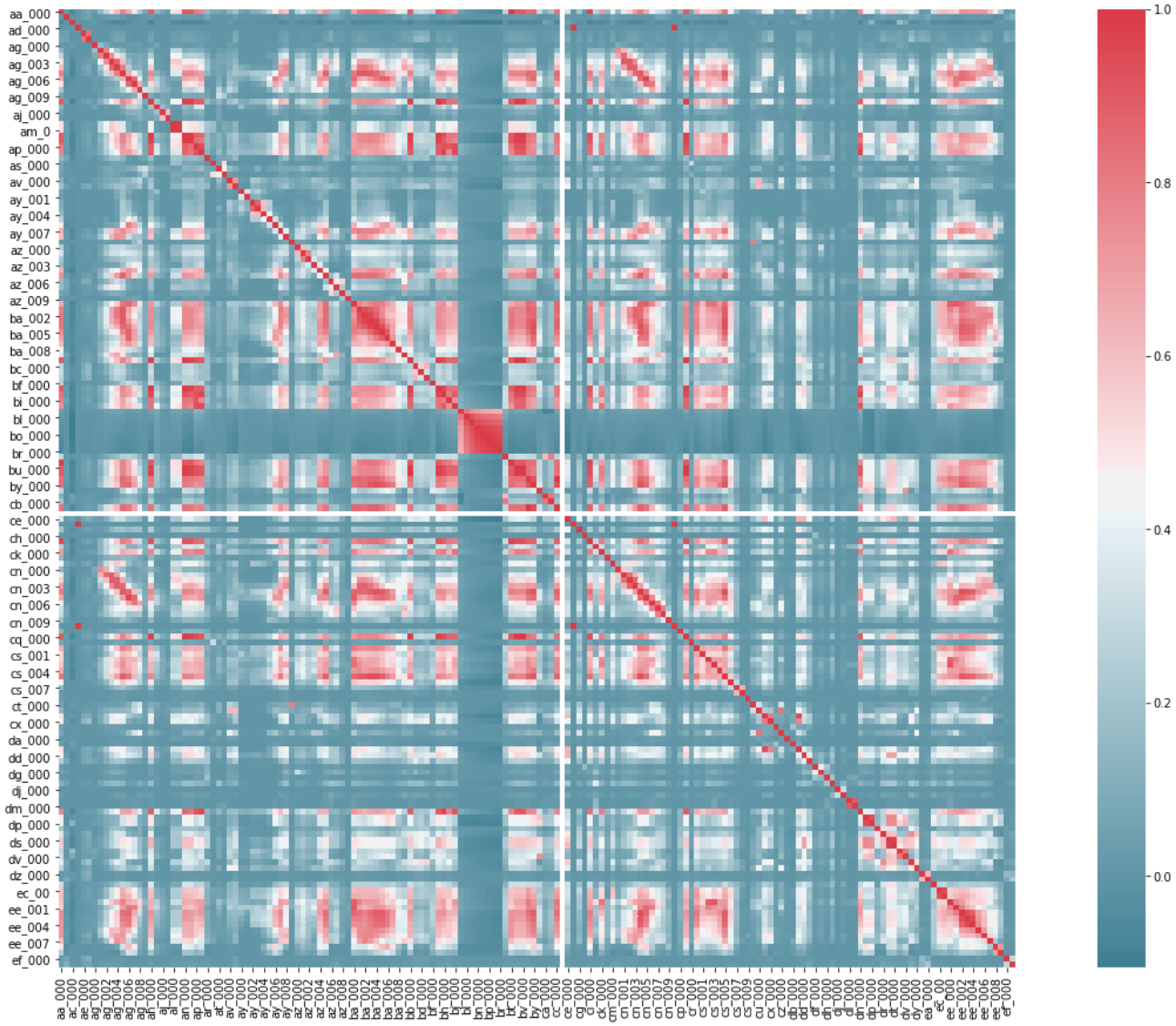
Coefficient of Variation of 13 features with highest CV :

```
cf_000 244.885475829
co_000 244.505347484
ad_000 244.320779727
cs_009 237.928570953
dh_000 123.215070412
dj_000 117.493246022
ag_000 92.9169807183
as_000 87.3317717911
ay_009 84.7330284819
ak_000 80.4243051982
az_009 77.8378956312
ch_000 77.4532116829
au_000 68.8821769233
```

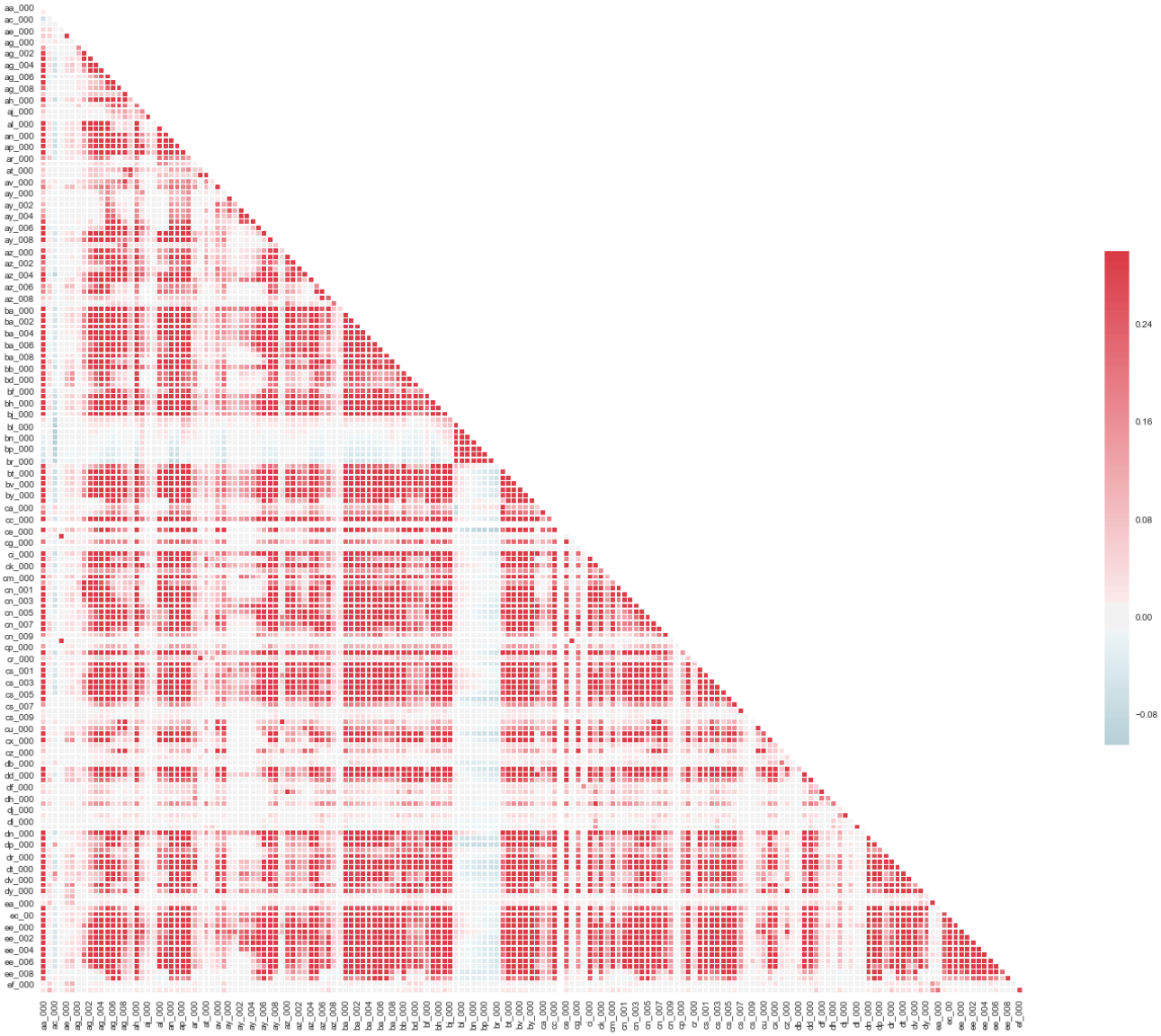
The training dataset is imbalanced because 98.33% of data are classified as negative

class **–"negative" =** 59000
class **–"Positive" =** 1000

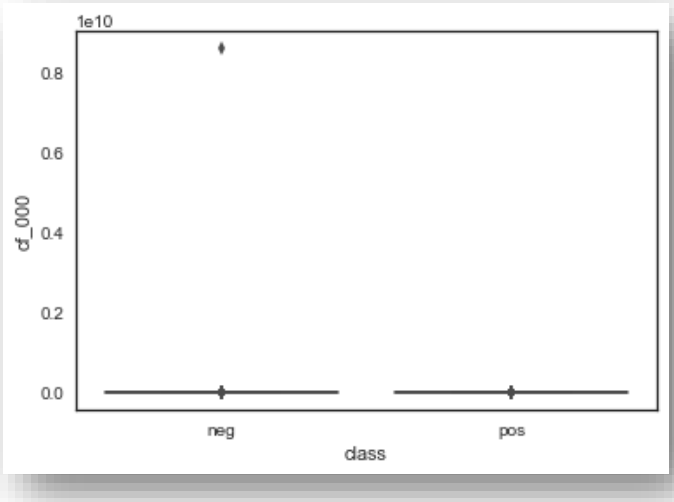
Correlation Matrix – 1:

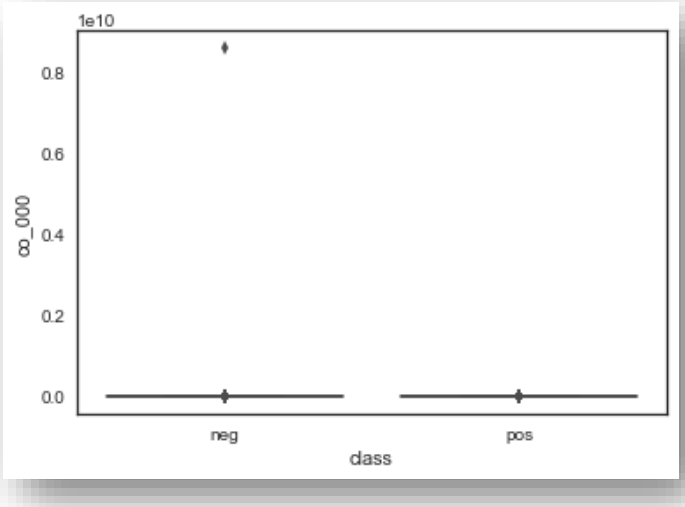


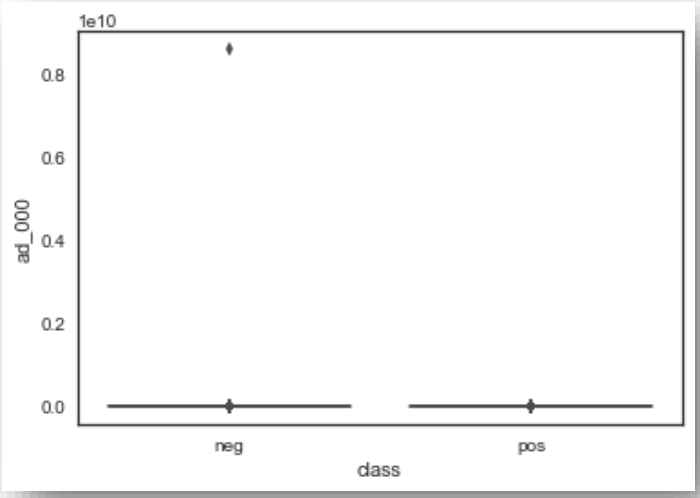
Correlation Matrix – 2:

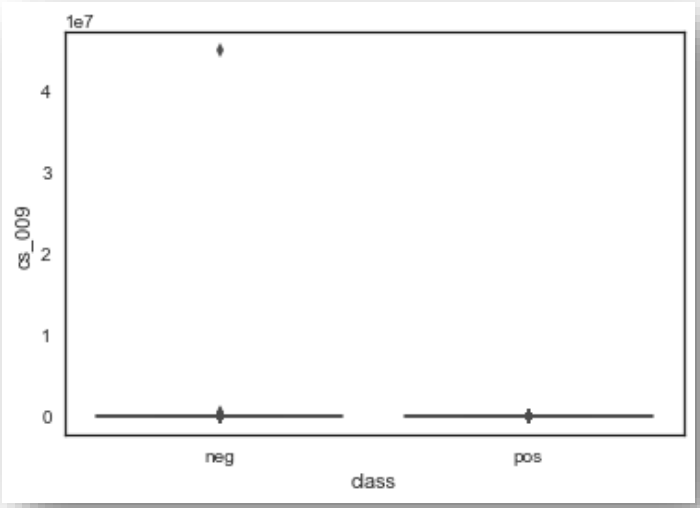


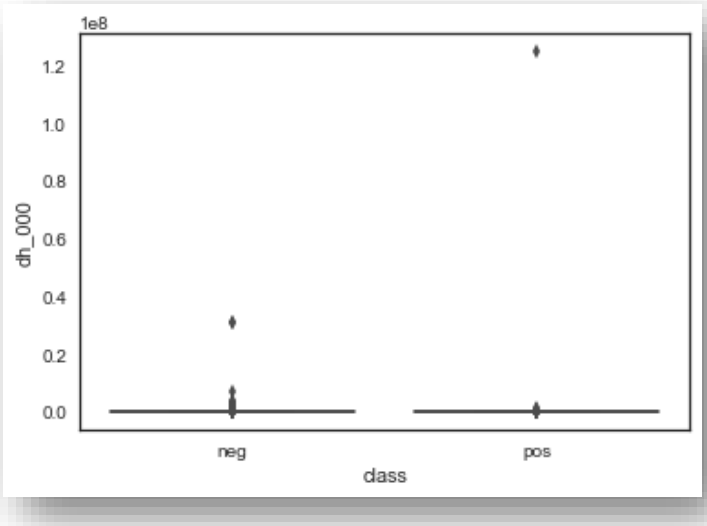
Box-Plots:

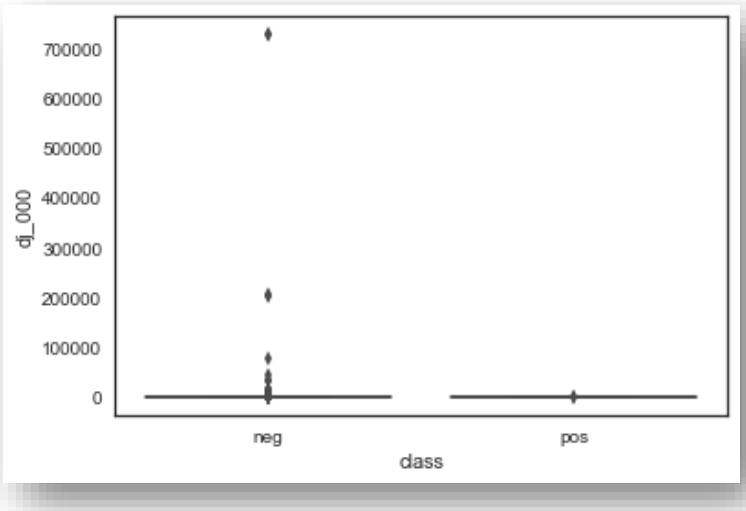

1e10
0.8
0.6
0.4
0.2
0.0
cf_000
neg
pos
class

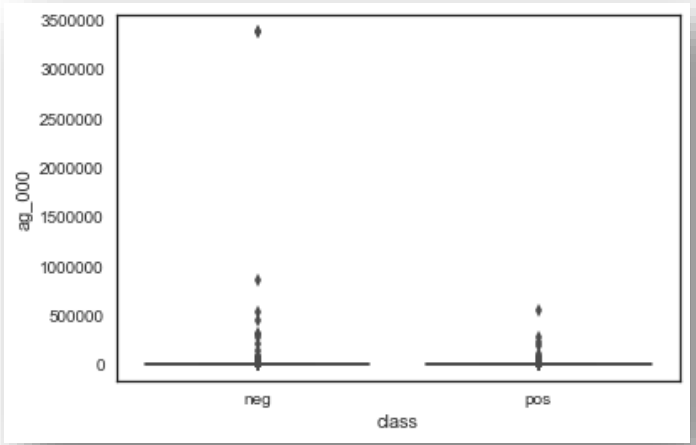

1e10
0.8
0.6
0.4
0.2
0.0
co_000
neg
pos
class

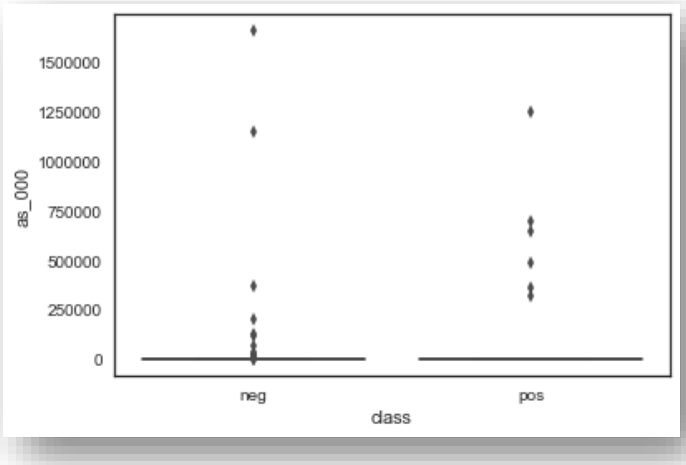

1e10
0.8
0.6
0.4
0.2
0.0
ad_000
neg
pos
class

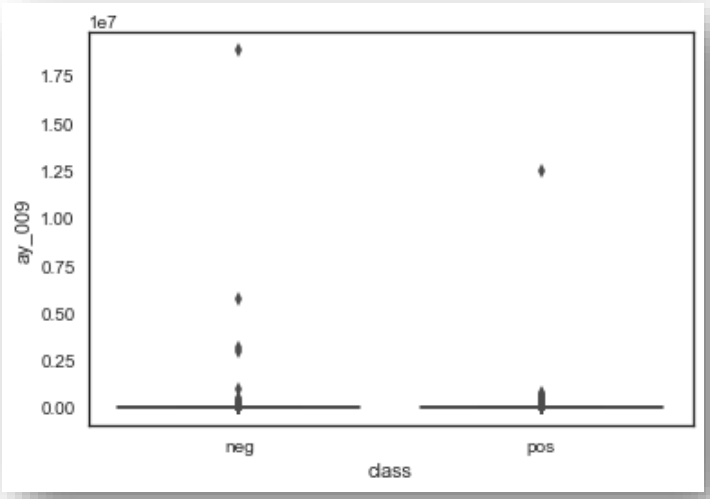

1e7
4
3
2
1
0
cs_009
neg
pos
class

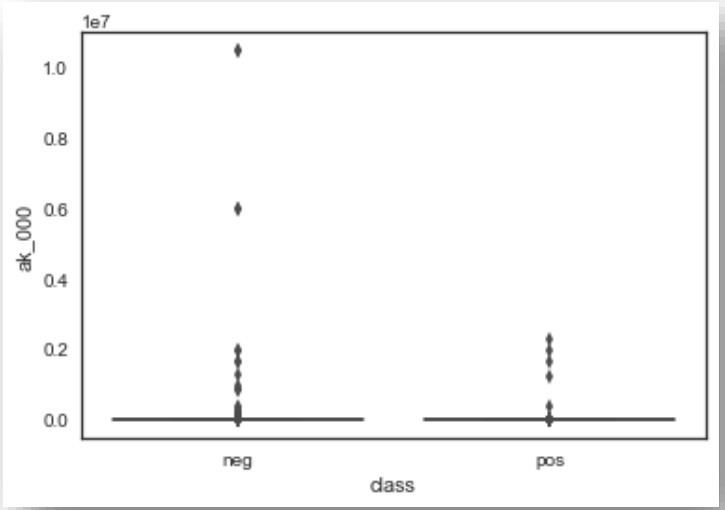

1e8
1.2
1.0
0.8
0.6
0.4
0.2
0.0
dh_000
neg
pos
class

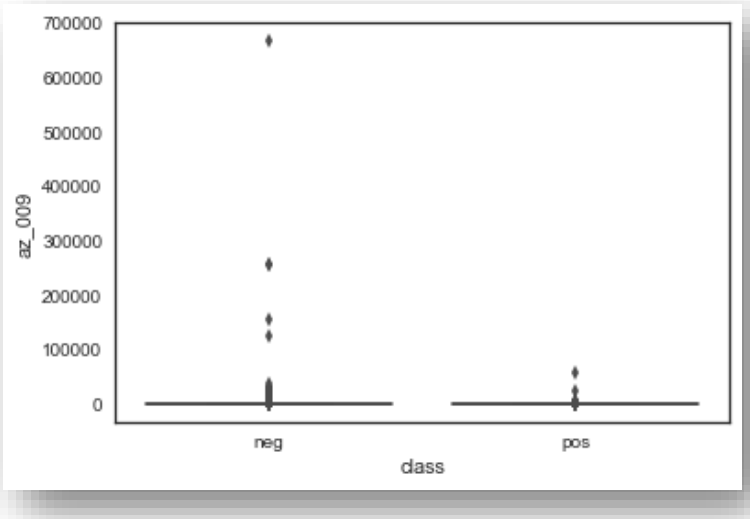

700000
600000
500000
400000
300000
200000
100000
0
dj_000
neg
pos
class

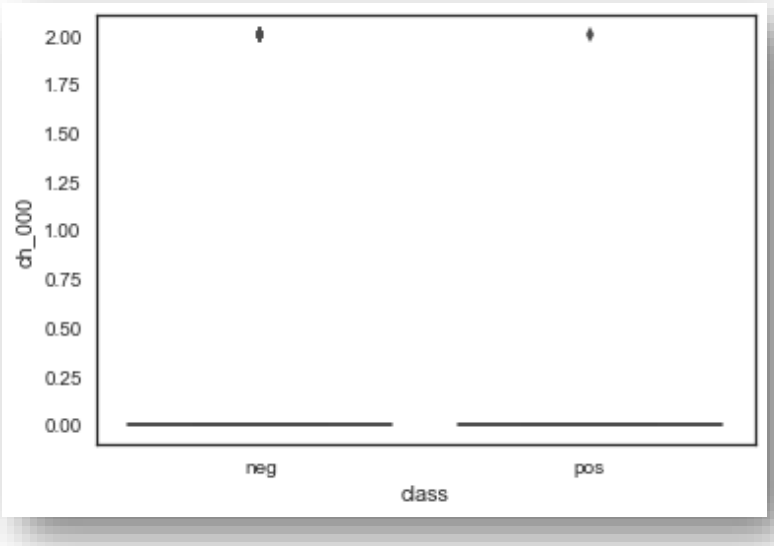

3500000
3000000
2500000
2000000
1500000
1000000
500000
0
ag_000
neg
pos
class

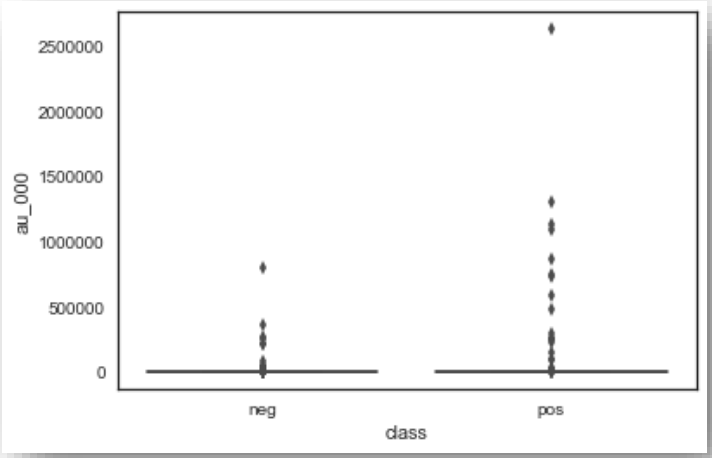

1500000
1250000
1000000
750000
500000
250000
0
as_000
neg
pos
class


1e7
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
ay_009
neg
pos
class


1e7
1.0
0.8
0.6
0.4
0.2
0.0
ak_000
neg
pos
class


700000
600000
500000
400000
300000
200000
100000
0
az_009
neg
pos
class


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
ch_000
neg
pos
class


2500000
2000000
1500000
1000000
500000
0
au_000
neg
pos
class

Scatterplots show possible associations or relationships between two variables. Howeve r, just because graph or chart shows something is going on, it doesn't mean that a caus e-and-effect relationship exists. Also looking at the plots we can conclude that the data i s skewed to "negative" class i.e class.

Random Forest :

When using Random Forest Classifier a useful setting is **class weight=balanced** , wherein classes are automatically weighted inversely proportional to how frequently they appear in the data.

class_weight : "balanced", "balanced_subsample","None"

"Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one. For multi-output problems, a list of dicts can be provided in the same order as the columns of y.

Balanced mode :

It uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`

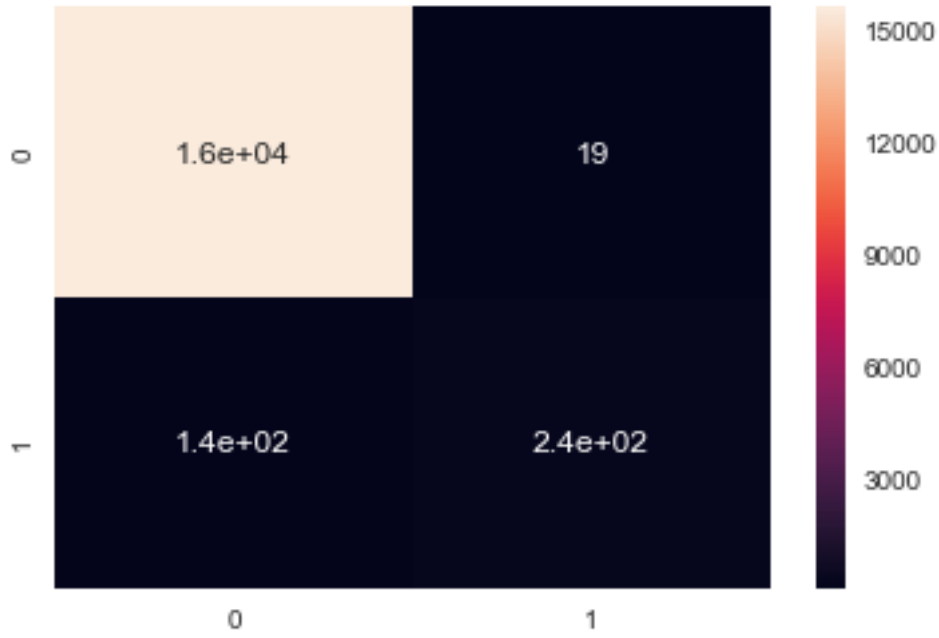
Balanced_subsample :

This mode is the same as "balanced" except that weights are computed based on the bootstrap sample for every tree grown.

For multi-output, the weights of each column of y will be multiplied.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

1. **Confusion Matrix:**



Confusion Matrix:

```
[[15606    19]
 [  137   238]]
```

	precision	recall	f1-score	support
neg	0.99	1.00	1.00	15625
pos	0.93	0.63	0.75	375
avg / total	0.99	0.99	0.99	16000

Out-of-the-Bag Error: **0.99155**

Accuracy score : 0.99025

Feature Importance:

```
[('aa_000', 0.009318908129270562),
 ('ab_000', 0.0009236356048983102),
 ('ac_000', 0.001355457336623017),
```

('ad_000', 0.0013194887041613368),
('ae_000', 8.80778685011355e-05),
('af_000', 0.0003634820754740484),
('ag_000', 0.0008569023189610069),
('ag_001', 0.015601888585829981),
('ag_002', 0.04290873257260361),
('ag_003', 0.013395047375958934),
('ag_004', 0.007939392406223596),
('ag_005', 0.0032186824141868 58),
('ag_006', 0.00885648594997327),
('ag_007', 0.007575813938176852),
('ag_008', 0.004960637349898439),
('ag_009', 0.002420063500586411),
('ah_000', 0.01751202278866934),
('ai_000', 0.0083728383736124 63),
('aj_000', 0.0014213794602381633),
('ak_000', 0.00018321265639883842),
('al_000', 0.02734040294498698),
('am_0', 0.025160104464917628),
('an_000', 0.004994319518192125),
('ao_000', 0.0050054846801116885),
('ap_000', 0.003576231962673932),
('aq_000', 0.048454692819530285),
('ar_000', 0.0012616763321520864),
('as_000', 0.0001969545510320934),
('at_000', 0.0032555652343888449),
('au_000', 0.00023671542706175255),
('av_000', 0.002628264388103685),
('ax_000', 0.002379585151930219),
('ay_000', 0.005430818084410801),
('ay_001', 0.0040798969830602725),
('ay_002', 0.0011528501940856032),
('ay_003', 0.008359344635105289),
('ay_004', 0.005011251517606147),
('ay_005', 0.01213055480170663),
('ay_006', 0.011773876896755653),
('ay_007', 0.006796451582717251),
('ay_008', 0.00765415334597796),
('ay_009', 0.008894270508724761),
('az_000', 0.0020414952642752973),
('az_001', 0.0035250018259142765),
('az_002', 0.005985730689115252),
('az_003', 0.006056412710101952),
('az_004', 0.002952764008140937),
('az_005', 0.004260625696201491),
('az_006', 0.002936398862568741),
('az_007', 0.0031424023458056956),
('az_008', 0.0012212198748640516),
('az_009', 0.0012893476585084955),
('ba_000', 0.004233172232680201),
('ba_001', 0.0039620769602695316),
('ba_002', 0.002430363946568795),
('ba_003', 0.005561336911113694),
('ba_004', 0.004175632842835701),
('ba_005', 0.008111419483360529),
('ba_006', 0.002886377028417152),
('ba_007', 0.003929163556578083),
('ba_008', 0.003708758384871602),

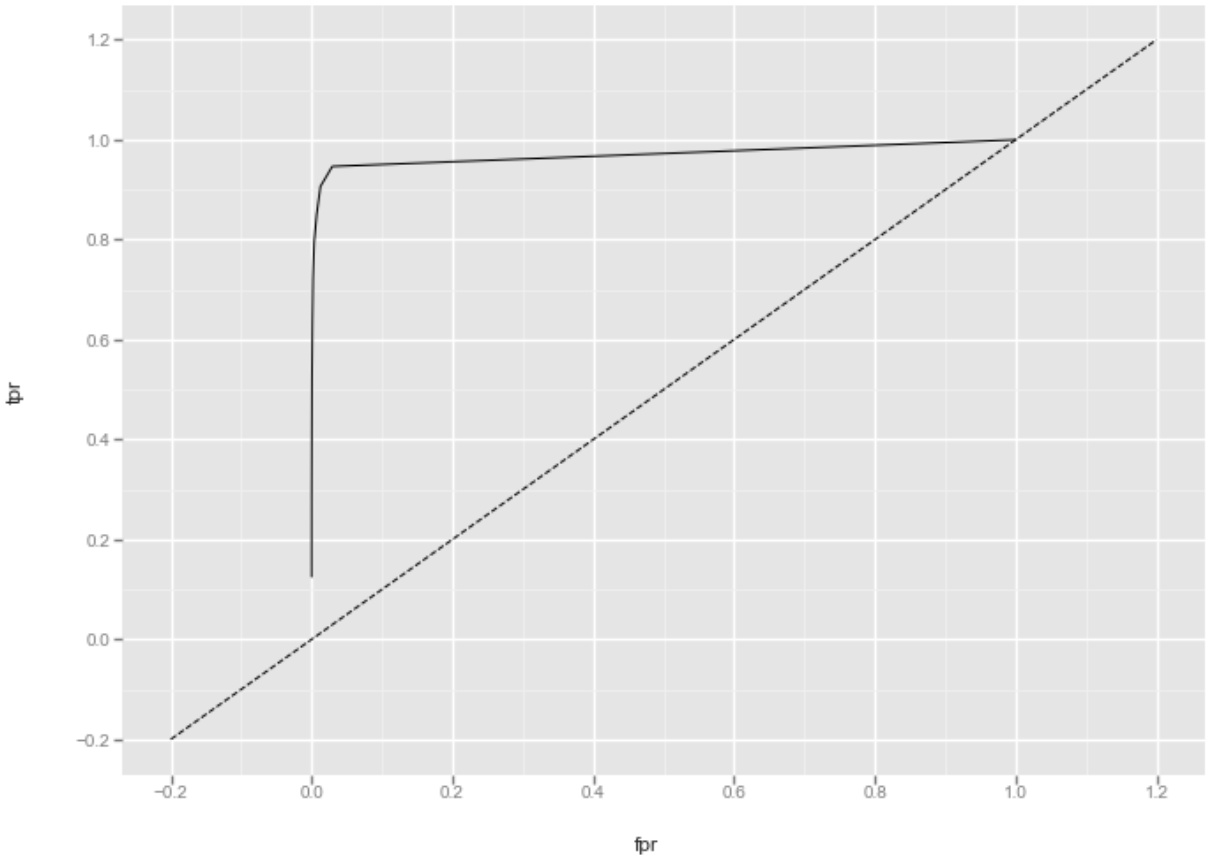
('ba_009', 0.0026592783293629004),
('bb_000', 0.03102767178784982),
('bc_000', 0.005152181909896444),
('bd_000', 0.002329569518708152),
('be_000', 0.0017905357626537039),
('bf_000', 0.0035549061342826913),
('bg_000', 0.024899533329828975),
('bh_000', 0.003428334158280364),
('bi_000', 0.005137884611398745),
('bj_000', 0.03844747871235034),
('bk_000', 0.0063038013838910 86),
('bl_000', 0.004232615945935017),
('bm_000', 0.005764730438581776),
('bn_000', 0.0029002443716286347),
('bo_000', 0.0027697180965063207),
('bp_000', 0.0024856575244364856),
('bq_000', 0.0022265485032869434),
('br_000', 0.002488443320011377),
('bs_000', 0.0032906973359567828),
('bt_000', 0.005608438606072504),
('bu_000', 0.04937711176089958),
('bv_000', 0.0027708922911075136),
('bx_000', 0.005883177969718134),
('by_000', 0.024268151489269186),
('bz_000', 0.0014313818783468434),
('ca_000', 0.003246506249157856),
('cb_000', 0.0029519910960847553),
('cc_000', 0.0034274729631567453),
('cd_000', 0.0),
('ce_000', 0.0015488639895899971),
('cf_000', 0.0011723630815291583),
('cg_000', 0.001520373216182779),
('ch_000', 0.0),
('ci_000', 0.008663665564850143),
('cj_000', 0.00787447608171449),
('ck_000', 0.010228517170745573),
('cl_000', 0.009428225695657792),
('cm_000', 0.0015540796099202422),
('cn_000', 0.014567529335844171),
('cn_001', 0.0055094219353808305),
('cn_002', 0.008698166828810092),
('cn_003', 0.004612329232826145),
('cn_004', 0.0067287979019987985),
('cn_005', 0.005752161438351083),
('cn_006', 0.005340495925461259),
('cn_007', 0.005049941099335332),
('cn_008', 0.004777894886283289),
('cn_009', 0.0028029937029340752),
('co_000', 0.0012576678608898738),
('cp_000', 0.0022312223250358933),
('cq_000', 0.002837887690705845),
('cr_000', 0.0),
('cs_000', 0.004003034661661827),
('cs_001', 0.003757497295182089),
('cs_002', 0.026309654475776212),
('cs_003', 0.004158819884613459),
('cs_004', 0.003551431206219118),
('cs_005', 0.004892132908280976),

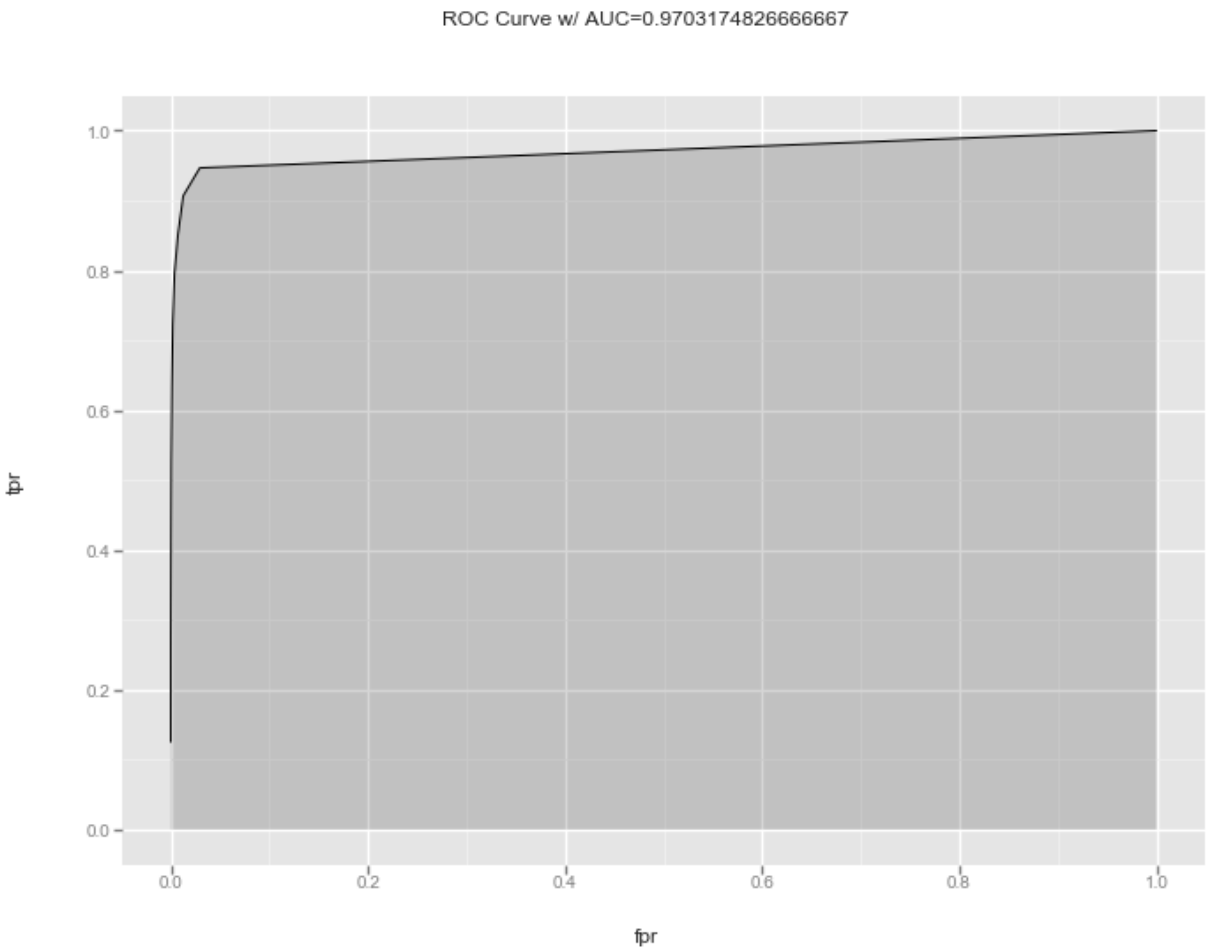
```
('cs_006', 0.0047948823543712965),
('cs_007', 0.0039300894879102666),
('cs_008', 0.0014175218247629630),
('cs_009', 0.00044089423804474147),
('ct_000', 0.0006858827739323633),
('cu_000', 0.0015615726855156369),
('cv_000', 0.0017140397670040845),
('cx_000', 0.0022576060746049694),
('cy_000', 0.0003665836274398399),
('cz_000', 0.0021853249961859550),
('da_000', 0.00026612419867482015),
('db_000', 0.0007964874573564758),
('dc_000', 0.0019585024662133750),
('dd_000', 0.0017774836225060403),
('de_000', 0.0051324975818844590),
('df_000', 0.0020883190640818713),
('dg_000', 0.0027859121929732465),
('dh_000', 0.0006093865961139146),
('di_000', 0.0008110268222274588),
('dj_000', 0.00010007647913553802),
('dk_000', 0.00012253652335834865),
('dl_000', 9.536867696501665e-05),
('dm_000', 0.0),
('dn_000', 0.0035819452519204890),
('do_000', 0.0031537213386265543),
('dp_000', 0.0020773448611141254),
('dq_000', 0.0015399055781129285),
('dr_000', 0.0027833293106119426),
('ds_000', 0.0024680404641441835),
('dt_000', 0.0024267175352006226),
('du_000', 0.0034735743296273568),
('dv_000', 0.0029224421094531320),
('dx_000', 0.0017485539315109096),
('dy_000', 0.0033624021257042795),
('dz_000', 0.0003100397850725192),
('ea_000', 7.570759229077746e-05),
('eb_000', 0.0017213512592228353),
('ec_00', 0.0033704070468199864),
('ed_000', 0.0034978737429054206),
('ee_000', 0.0065537155839382630),
('ee_001', 0.0031282488419372950),
('ee_002', 0.0071211618222731090),
('ee_003', 0.0033135620469885530),
('ee_004', 0.0046436063123677885),
('ee_005', 0.0561219228047952800),
('ee_006', 0.0036364898669561120),
('ee_007', 0.0072868211480786320),
('ee_008', 0.0045662552298432240),
('ee_009', 0.0036150409641679745),
('ef_000', 4.403316586459756e-05),
('eg_000', 9.76631275792836e-05)]
```

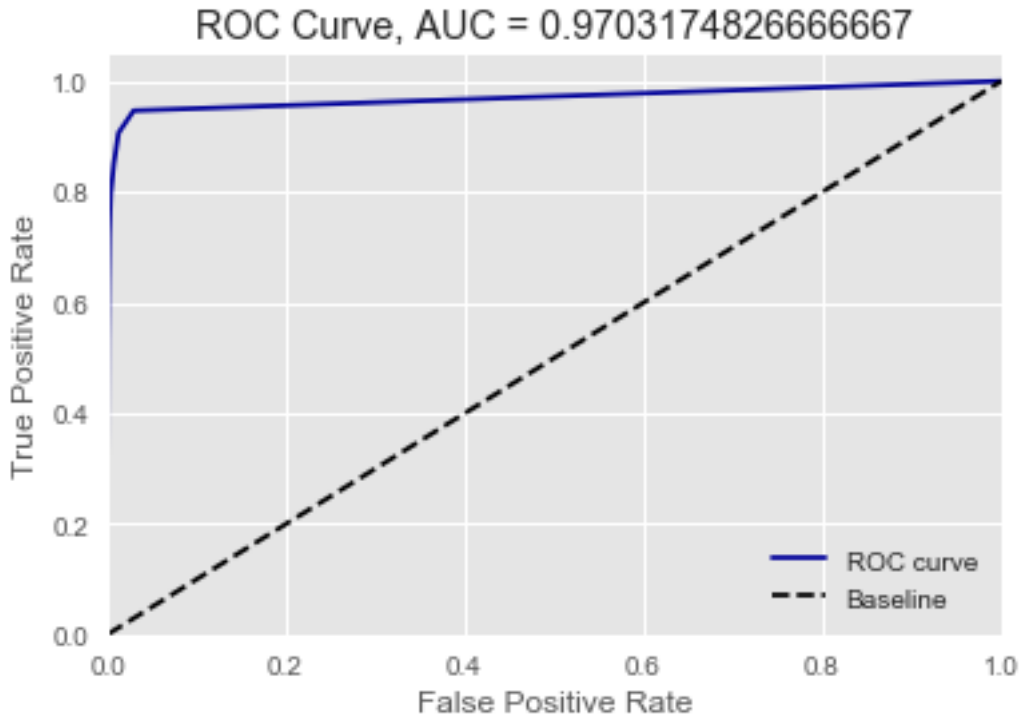
TOP FEATURES WITH THERE IMPORTANCE TO THE MODEL: (top 20)

```
bj_000 0.07073877157330911
ee_005 0.04967770912002216
bu_000 0.04391978285406852
ag_003 0.03784372261338807
dn_000 0.03255303383532958
by_000 0.02961721328445659
bb_000 0.029005835866274653
ap_000 0.02822963009571079
ag_002 0.02707907370679292
ba_004 0.025250935083267485
bv_000 0.018965023888642067
al_000 0.017927421262339815
am_0 0.017106970558189116
cn_001 0.012908966189349416
ag_001 0.01257106438729883
ck_000 0.012260947909714767
ay_006 0.012172544622231152
cn_000 0.012107780189836903
ay_009 0.012073467330672823
az_001 0.009923043515639132
az_000 0.009746146482748023
ay_004 0.009307095393530388
```

ROC & AUC :




ROC Curve w/ AUC=0.9703174826666667
tpr
fpr
0.0
0.2
0.4
0.6
0.8
1.0


ROC Curve, AUC = 0.9703174826666667
True Positive Rate
False Positive Rate
ROC curve
Baseline
0.0
0.2
0.4
0.6
0.8
1.0

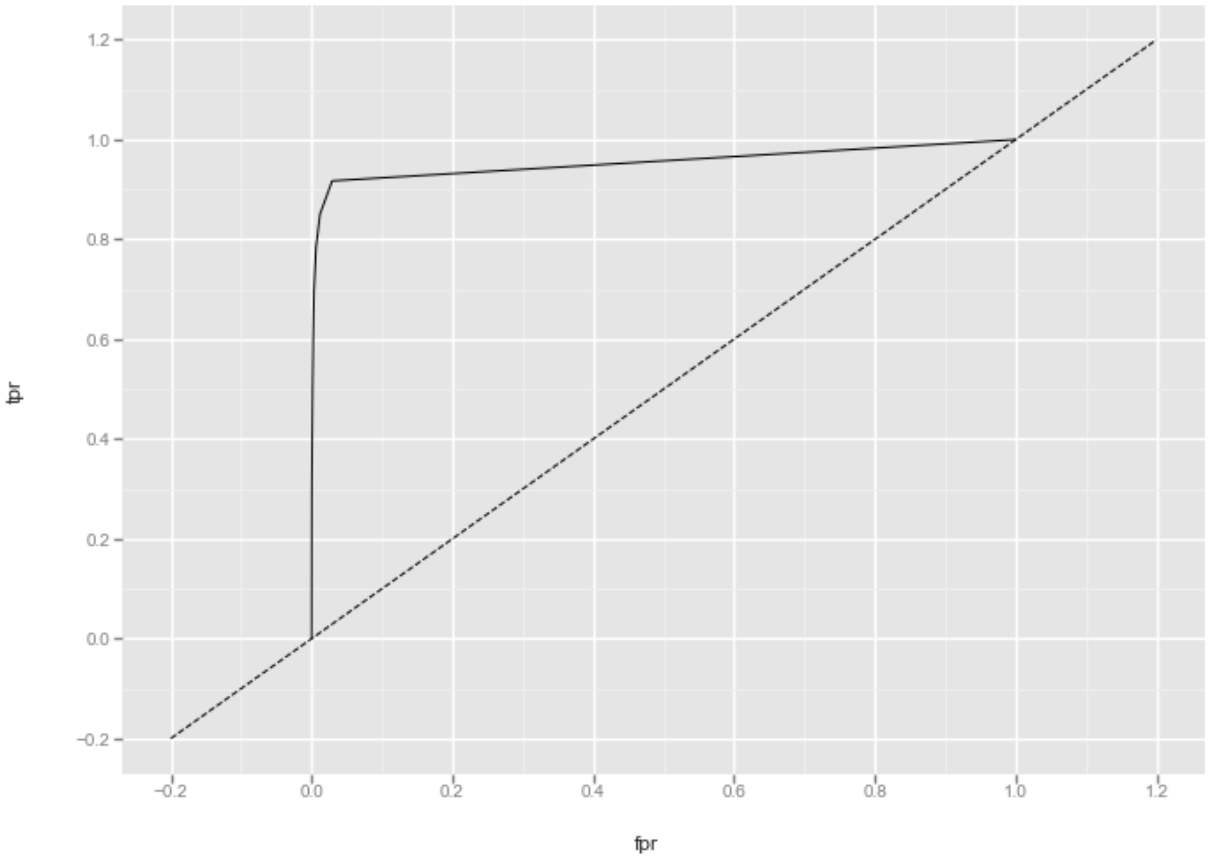
Random-Forest for handling Class-Imbalancing:

```
array([[15610,    15],
       [  175,   200]]
dtype=int64)
```

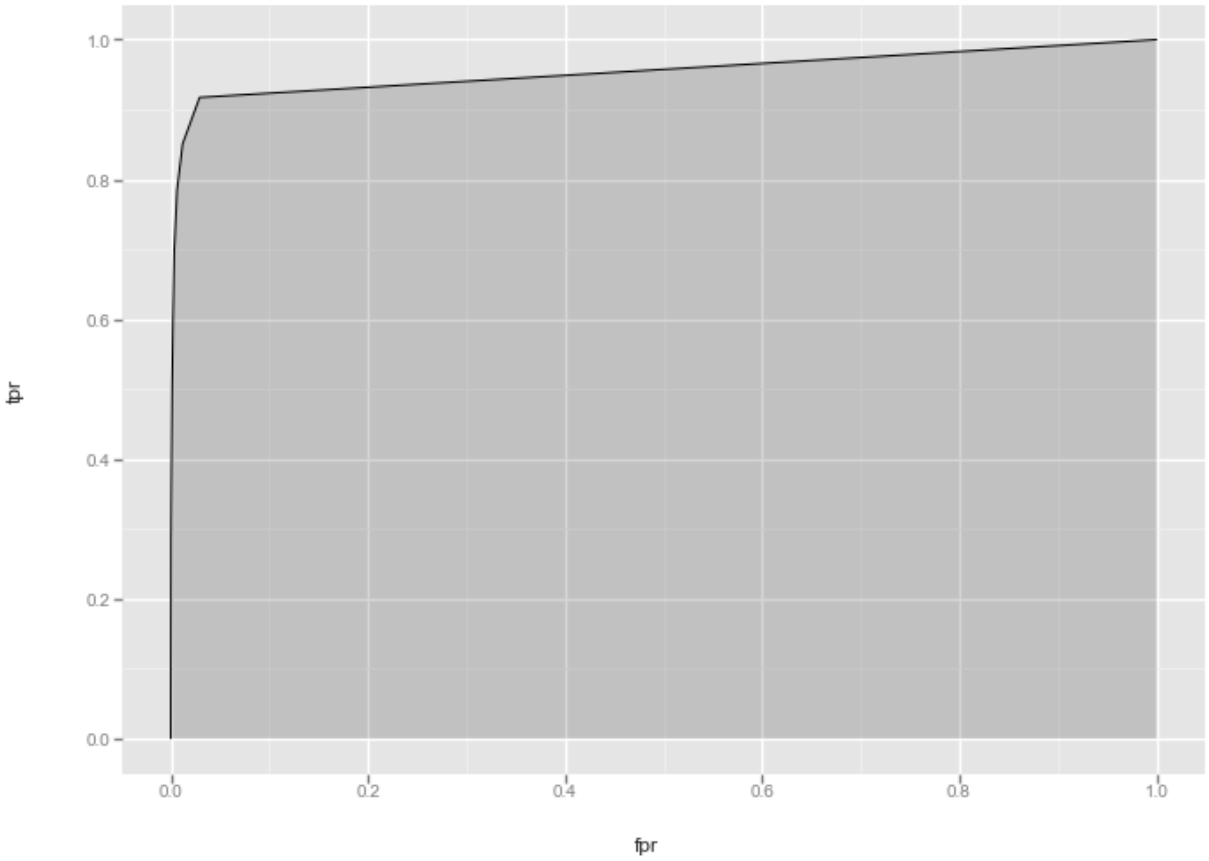
So we observe that after handling class-misbalance the results improve.
With parameters set: **class_weight='balanced_subsample' / class_weight='balanced'**

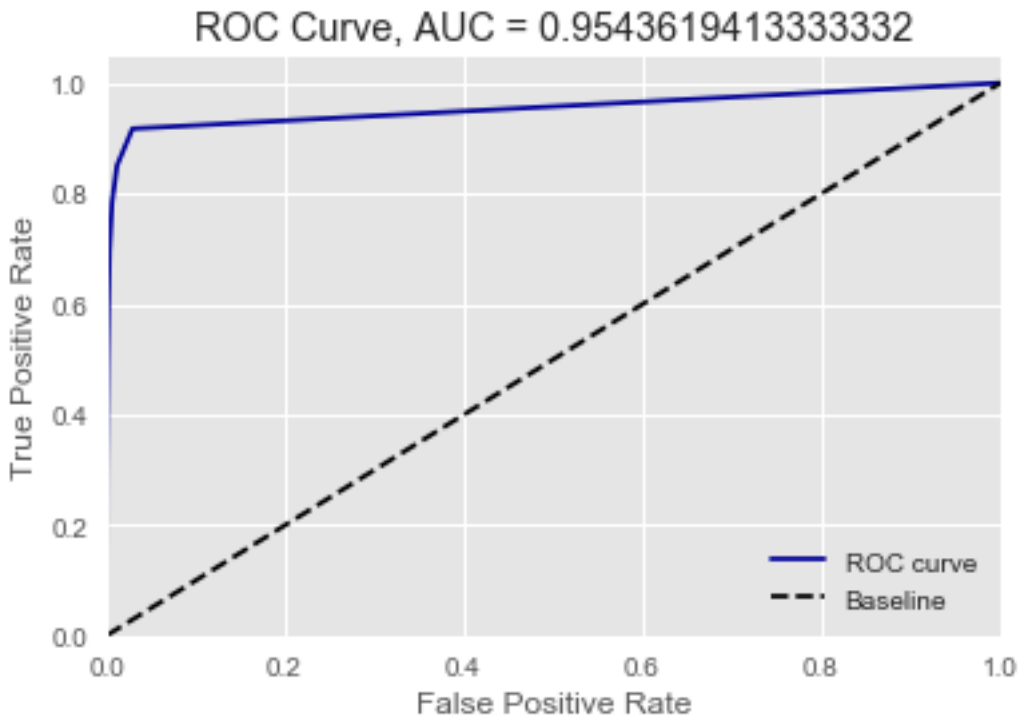
```
oob_score_ : 0.9898666666666667
accuracy :   0.9869375
```

	precision	recall	f1-score	support
neg	0.99	1.00	1.00	15625
pos	0.93	0.63	0.75	375
avg / total	0.99	0.99	0.99	16000



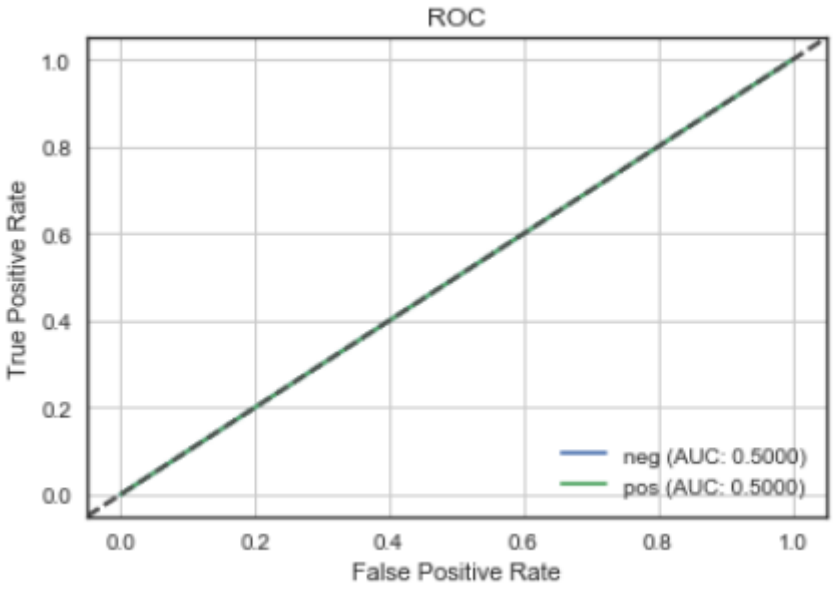
ROC Curve w/ AUC=0.9543619413333332





2.e Model Trees:

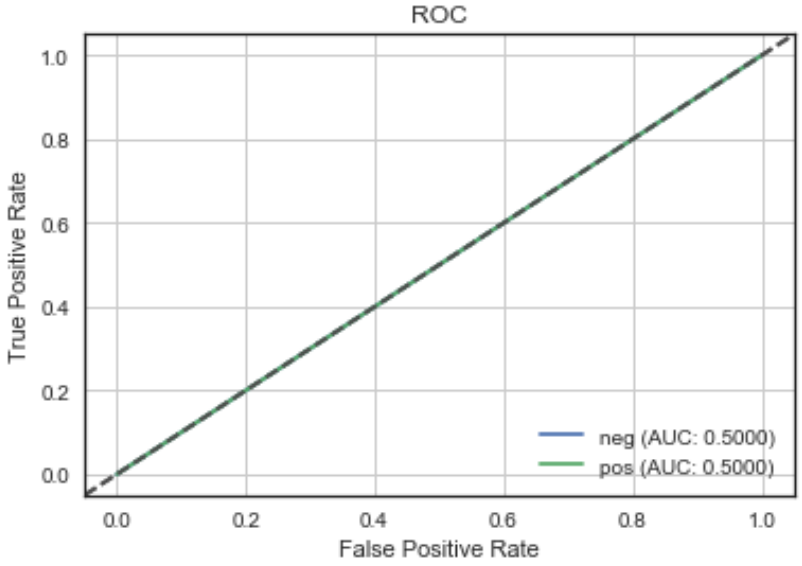
```
Correctly Classified Instances          10               90.9091 %
Incorrectly Classified Instances         1                9.0909 %
Kappa statistic                          0
Mean absolute error                      0.5
Root mean squared error                  0.5
Relative absolute error                220      %
Root relative squared error            159.2243 %
Total Number of Instances               11

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.909      1.000    0.952      ?        0.500     0.909     neg
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.091     pos
Weighted Avg.    0.909    0.909    ?          0.909    ?          ?        0.500     0.835

=== Confusion Matrix ===

  a  b   <-- classified as
 10  0 |  a = neg
  1  0 |  b = pos

areaUnderPRC/0: 0.909090909091
weightedAreaUnderPRC: 0.834710743802
areaUnderROC/1: 0.5
weightedAreaUnderROC: 0.5
avgCost: 0.0
totalCost: 0.0
confusionMatrix: [[10.  0.]
 [ 1.  0.]]
```



2.f SMOTE

```
Resampled dataset shape Counter({'neg': 59000, 'pos': 59000})
```

```
Correctly Classified Instances          12               92.3077 %
Incorrectly Classified Instances         1                7.6923 %
Kappa statistic                          0
Mean absolute error                      0.5
Root mean squared error                  0.5
Relative absolute error                248.5294 %
Root relative squared error            172.239  %
Total Number of Instances               13

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.923      1.000    0.960      ?        0.500     0.923     neg
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.077     pos
Weighted Avg.    0.923    0.923    ?          0.923    ?          ?        0.500     0.858

=== Confusion Matrix ===

  a  b   <-- classified as
 12  0 |  a = neg
  1  0 |  b = pos

areaUnderPRC/0: 0.923076923077
weightedAreaUnderPRC: 0.85798816568
areaUnderROC/1: 0.5
weightedAreaUnderROC: 0.5
avgCost: 0.0
totalCost: 0.0
confusionMatrix: [[12.  0.]
 [ 1.  0.]]
```



Before Using SMOTE:

```
Correctly Classified Instances                  90.9091 %
Incorrectly Classified Instances                 9.0909 %
```

After Using SMOTE:

```
Correctly Classified Instances                  92.3077 %
Incorrectly Classified Instances                 7.6923 %
```

classmate
Date
Page

3. ISLR - 6.8.3

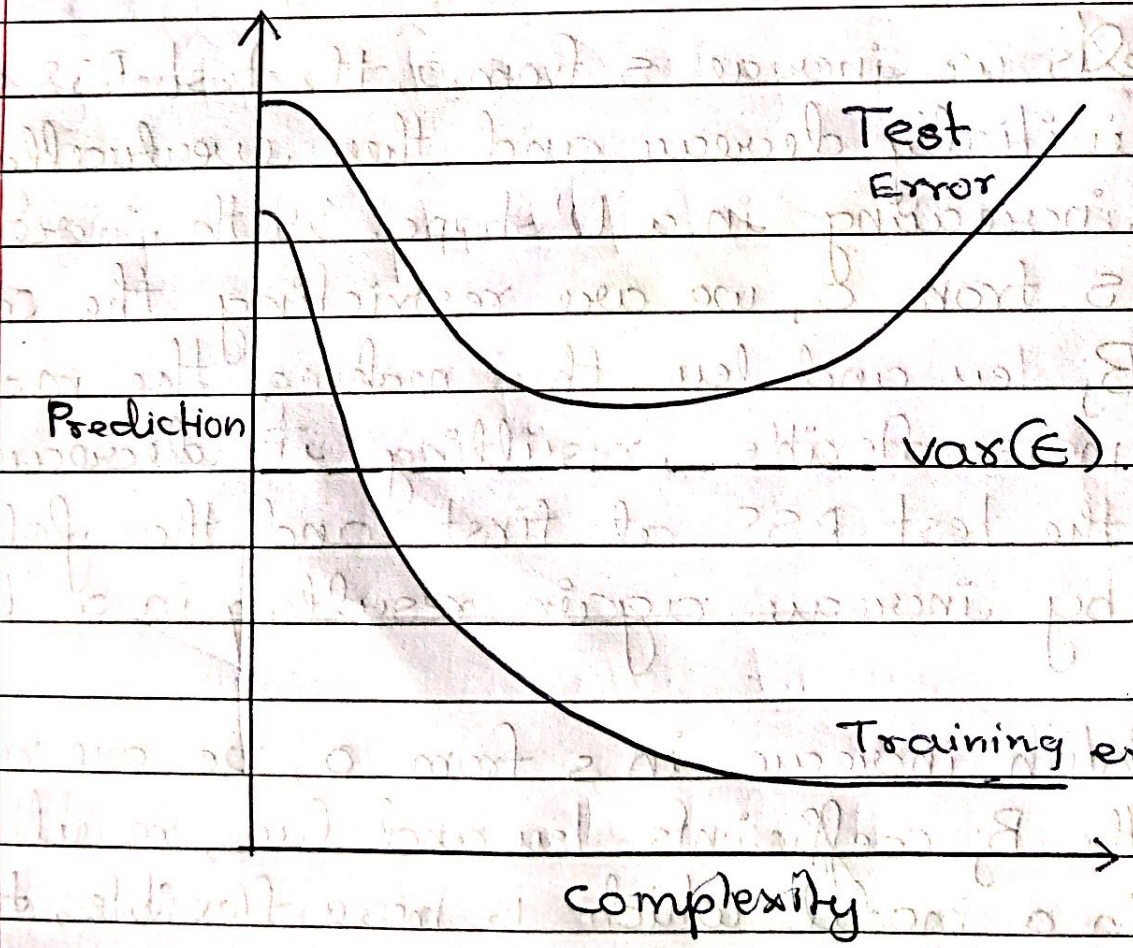
(a) As we increase s from 0, the training RSS will steadily decrease, because we are restricting the 'B_j' coefficients less and less, thus making the model flexible which in turn results in steady decrease in the training RSS. With increase 's' from 0, all the β's increase from 0 to their least square estimate value.

(b) As we increase s from 0, the test RSS will initially decrease and then eventually start increasing in a 'U' shape. With increase in s from 0, we are restricting the coefficients B_j less and less, thus making the model more flexible, resulting in decrease in the test RSS at first and then followed by increase again resulting in a 'U' shape.

(c) With increase in s from 0, we are restricting the B_j coefficients less and less, resulting in a model which is more flexible, thus we observe a steady increase in variance.

(d) As we increase 's' from 0, $(Bias)^2$ steadily decrease, i.e because as we increase s from 0, we are restricting the B_j coefficients less and resulting in a model which is more flexible and which results in decrease in the bias.

(e) With increase in 's' from 0, the var(ϵ) remains constant, because we know that 'ϵ' is independent of the model and 's'



4. ISLR - 6.8.5.

(a) As per the data in hand, we are given with, $x_{11} = x_{12} = x_1$ & $x_{21} = x_{22} = x_2$.

→ the Ridge Regression seeks to minimize thus resulting in :

$$(y_1 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_1)^2 + (y_2 - \hat{\beta}_1 x_2 - \hat{\beta}_2 x_2)^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2)$$

(b) from the above solution (a), we can further derive that : (i.e when $\beta_1 = \beta_2 = 0$)

$$\hat{\beta}_1(x_1^2 + x_2^2 + \lambda) + \hat{\beta}_2(x_1^2 + x_2^2) = y_1 x_1 + y_2 x_2 \quad - (1)$$

and

$$\hat{\beta}_1(x_1^2 + x_2^2) + \hat{\beta}_2(x_1^2 + x_2^2 + \lambda) = y_1 x_1 + y_2 x_2 \quad - (2)$$

By Substracting (1) and (2) we get.

$$\hat{\beta}_1 = \hat{\beta}_2$$

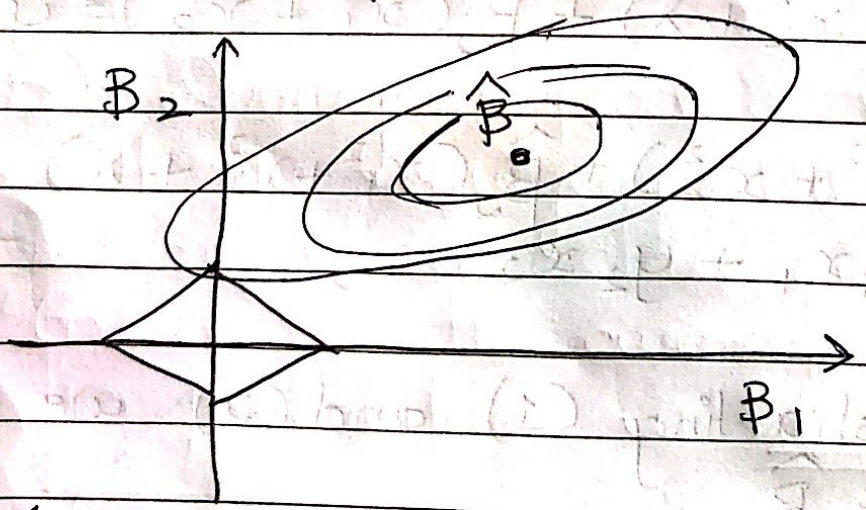
(c) When: $x_{11} = x_{12} = x_1$ & $x_{21} = x_{22} = x_2$
Lasso Regression aims to minimize resulting in:

$$(y_1 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_1)^2 + (y_2 - \hat{\beta}_1 x_2 - \hat{\beta}_2 x_2)^2 + \lambda(|\hat{\beta}_1| + |\hat{\beta}_2|).$$

(d) We can use 'Lasso Problem' in a different way i.e.

$$(y_1 - \hat{\beta}_1 x_1 - \hat{\beta}_2 x_1)^2 + (y_2 - \hat{\beta}_1 x_2 - \hat{\beta}_2 x_2)^2$$

subject to $|\hat{\beta}_1| + |\hat{\beta}_2| \leq s$



The lasso contraits take the form of a diamond centred at the origin of the plane β_1, β_2.

Date
Page

By using the data given the question we have

$(x_{11} = x_{12} = x_1)$ & $(x_{21} = x_{22} = x_2)$ & $x_1 + x_2 = 0$ & $y_1 + y_2 = 0$ so we have to minimize

$$2\left[y_1 - (\hat{\beta}_1 + \hat{\beta}_2)x_1\right]^2 > 0 \quad \text{— (1)}$$

so,

Solution to this (1) is : $\hat{\beta}_1 + \hat{\beta}_2 = y_1/x_1$ — (2)

This (2) is parallel to the edge of the diamond

So now, the solution to the 'Lasso Optimization Problem' are contours of the function.

$$\left[y_1 - (\hat{\beta}_1 + \hat{\beta}_2)x_1\right]^2 \quad \text{— (3)}$$

i.e when (3) intersets the diamond :

Entire edge : $\hat{\beta}_1 + \hat{\beta}_2 = s$ also is the edge : $\hat{\beta}_1 + \hat{\beta}_2 = -s$

Thus we come to the understanding that 'Lasso Optimization Problem' has a large set of solutions instead of a unique one

$\{(\hat{\beta}_1, \hat{\beta}_2) : \hat{\beta}_1 + \hat{\beta}_2 = s$ with $\hat{\beta}_1, \hat{\beta}_2 > 0$

Date
Page

$\beta \rightarrow (\hat{\beta}_1 + \hat{\beta}_2 = -5 \text{ with } \hat{\beta}_1, \hat{\beta}_2 \leq 0)$

5. ISLR - 8.4.5 :

We have : p = c(0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75)

2 Approaches :

(A). Majority Approach :

sum (p > = 0.5) > sum (p < 0.5)

ie here we classify 'X' as 'Red' since it occurs most frequently among the 10 predictions.

(6 for Red vs 4 for Green).

(B). Average Approach : mean (p)

⇒ with the average probability approach, will result in classifying 'X' as 'Green' as the average of the 10 probabilities is 0.45

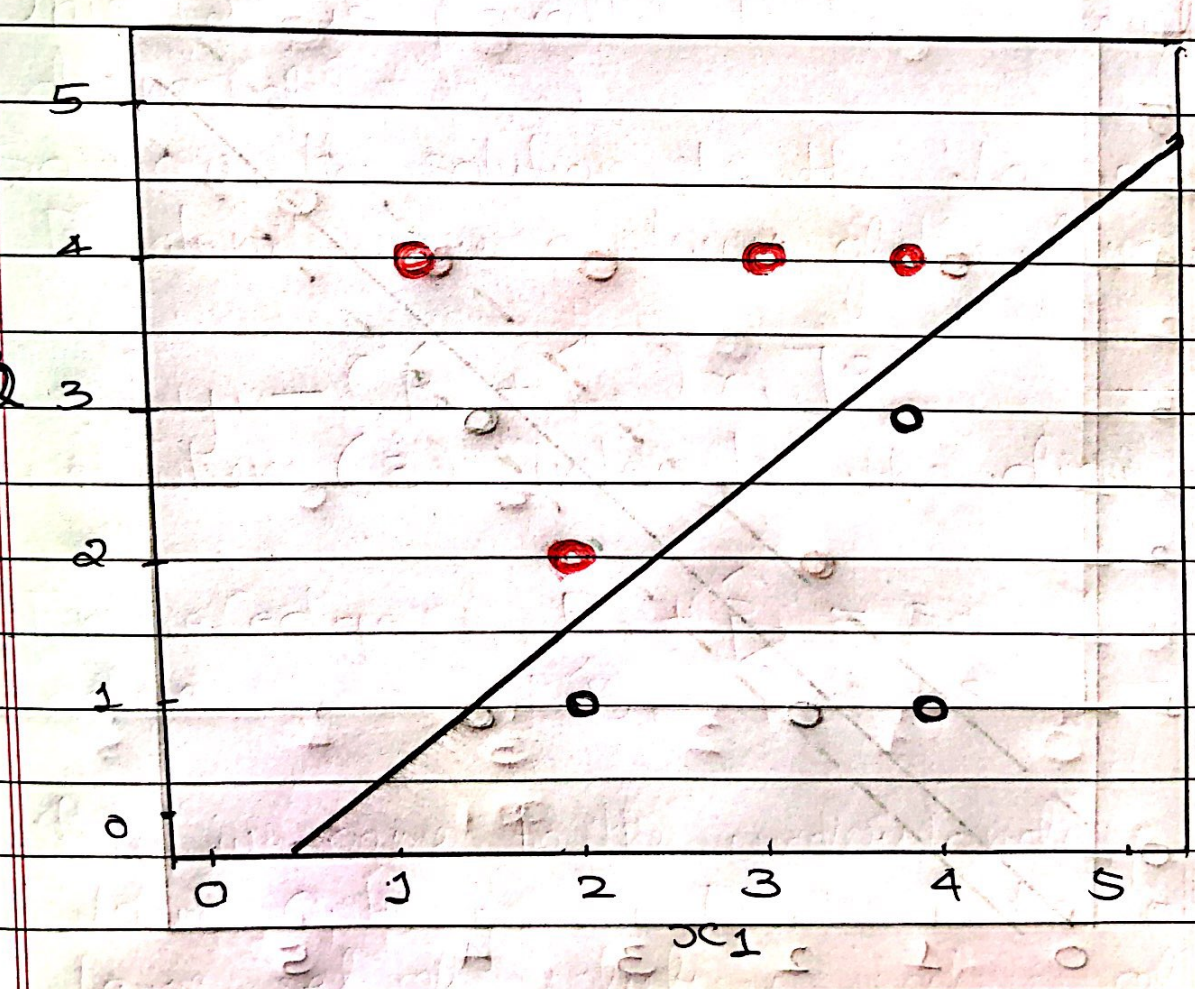
6 ISLR - 9.7.3

(a) $x_1 = c(3, 2, 4, 1, 2, 4, 4)$

$x_2 = c(4, 2, 4, 4, 1, 3, 1)$

colors = c("red", "red", "red", "red", "blue", "blue", "blue")

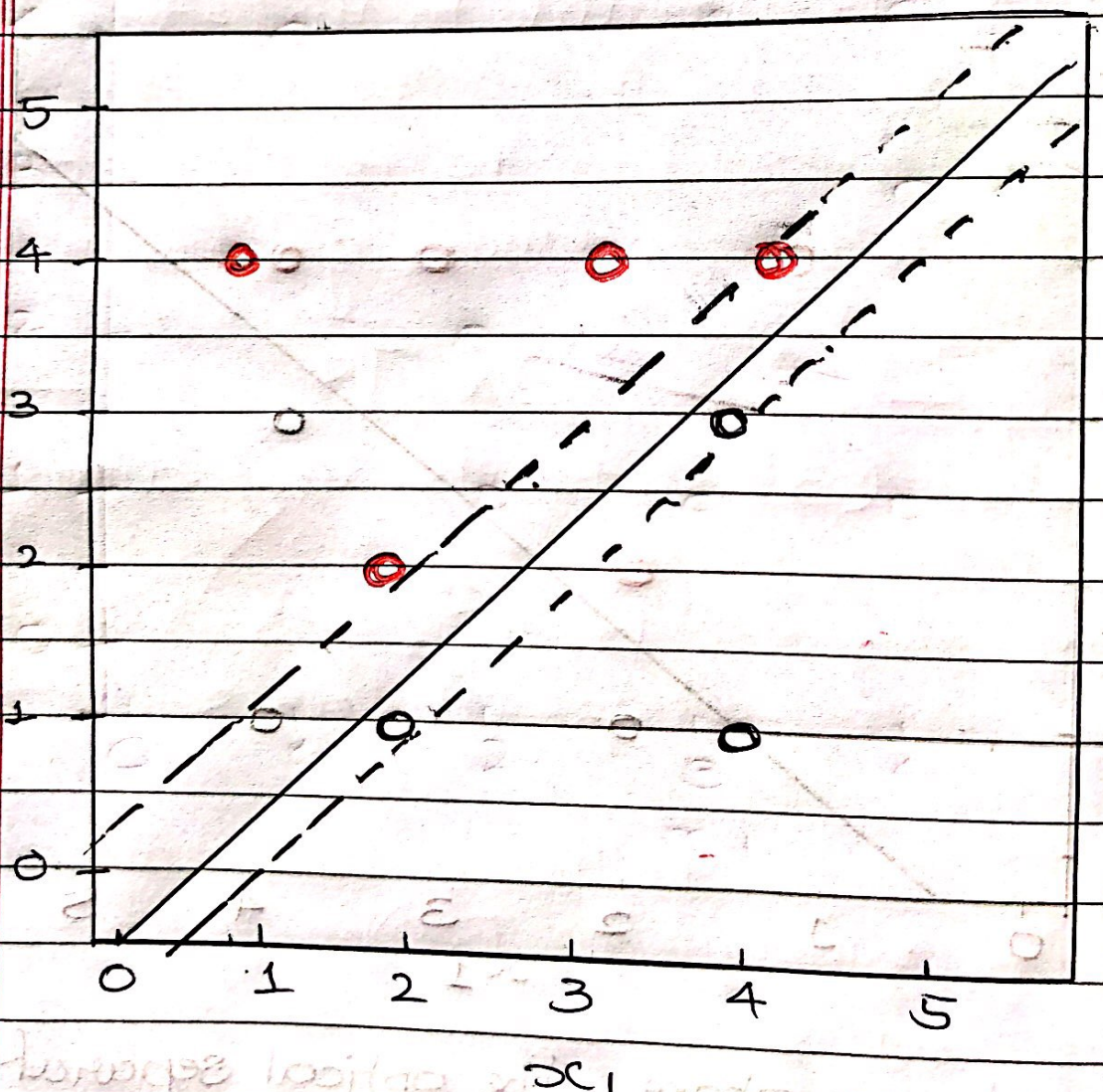
(b) (a) & (b)



As shown above, the optical separating hyperplane has to be between (2,1) and (2,2) & between (4,3) and (4,4).

(c) Classification Rule is:
→ "Classified to class "RED" if $X_1 - X_2 - 0.5 < 0$, and to class "Blue" otherwise.

(d)



The margin is here equal to 1/4.

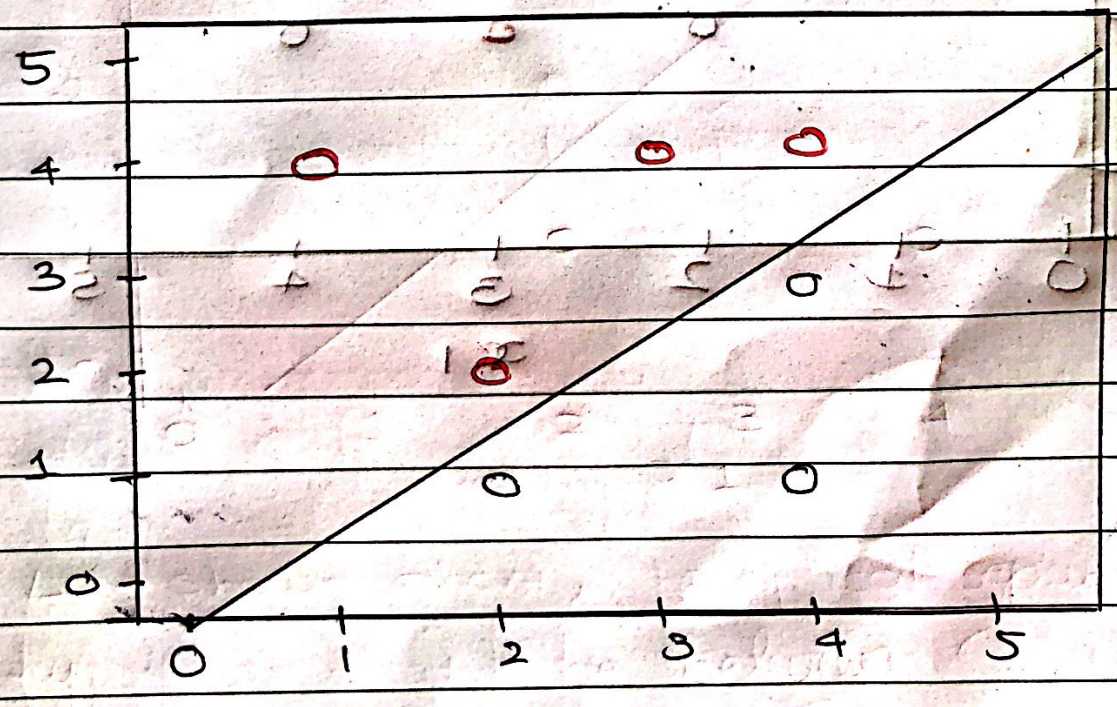
(e) The support vectors are the points (2,1), (2,2), (4,3) and (4,4).

(f) By looking at the plot, we can come to understanding that; if we moved point (4,1), we would end up not making any change to the maximal margin hyperplane, since it is not a support vector.

(g) The equation of the hyperplane which can be written as:

$$X_1 - X_2 - 0.3 = 0$$

This is not the optimal separating hyperpla



(h). When the red point (3,1) is added to the plot, the two classes are obviously not separable.