ASM file

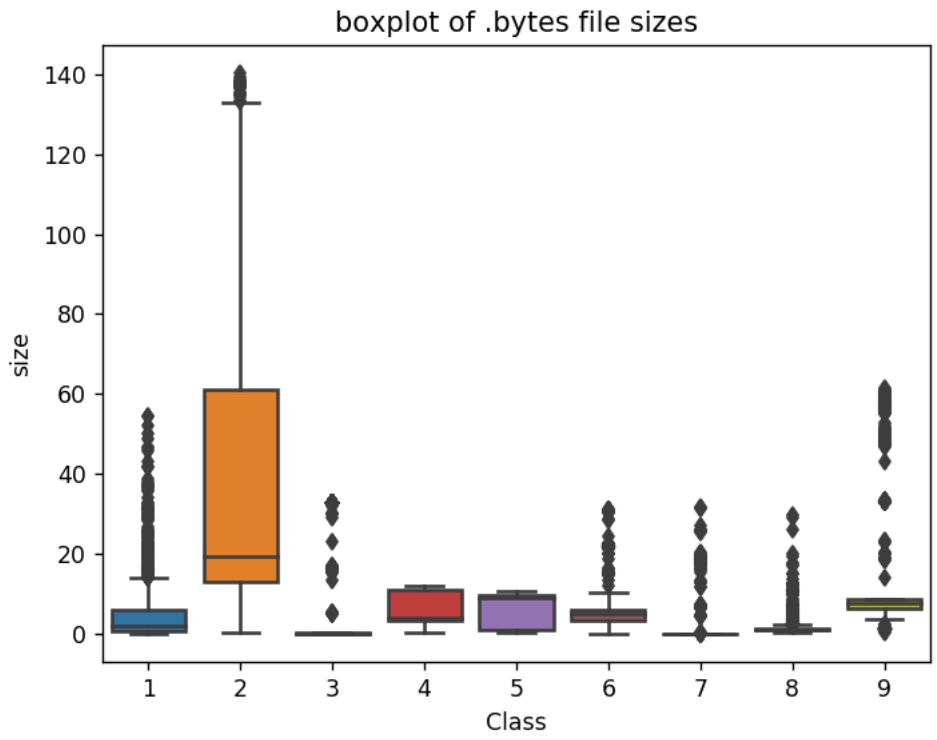
Created	@Jun 10, 2021 11:17 AM
Tags	

Exploratory Data Analysis

File Size as feature

ID	File Name	Size
0	01azqd4InC7m9JpocGv5	56.229886
1	01IsoiSMh5gxyDYTl4CB	13.999378
2	01jsnpXSAlgw6aPeDxrU	8.507785
3	01kcPWA9K2BOxQeS5Rju	0.078190
4	01SuzwMJEIXsK7A8dQbl	0.996723

Box plot of file size as feature



Copy of Bag of Words for ASM File

Property	File Name	HEADER:	.text:	.Pav:	.idata:	.data:	.bss:	.rdata:	.edata:	.rsrc:	...	edx	esi	eax	e
0	01kcPWA9K2BOxQeS5Rju	19	744	0	127	57	0	323	0	3	...	18	66	15	4
1	1E93CpP60RHFNiT5Qfvn	17	838	0	103	49	0	0	0	3	...	18	29	48	8

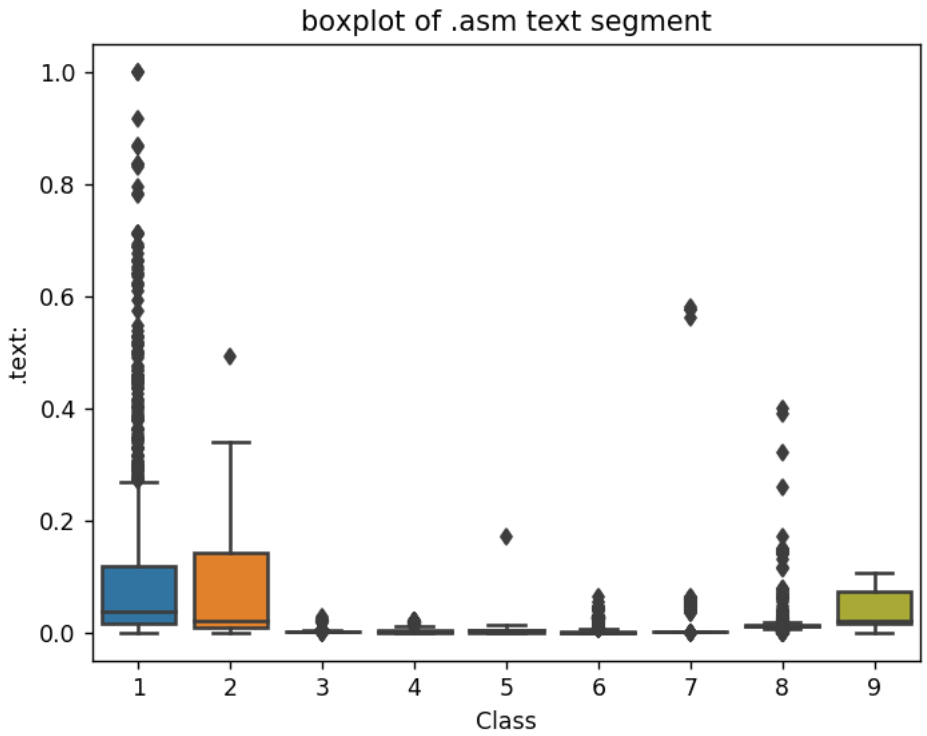
Property	File Name	HEADER:	.text:	.Pav:	.idata:	.data:	.bss:	.rdata:	.edata:	.rsrc:	...	edx	esi	eax	e
2	3ekVow2ajZHbTnBcsDfX	17	427	0	50	43	0	145	0	3	...	13	42	10	6
3	3X2nY7iQaPBlWDrAZqJe	17	227	0	43	19	0	0	0	3	...	6	8	14	7
4	46OZzdsSKDCFV8h7XWxf	17	402	0	59	170	0	0	0	3	...	12	9	18	2

Copy of Combining Bag of Words and File size as Features

ID	File Name	HEADER:	.text:	.Pav:	.idata:	.data:	.bss:	.rdata:	.edata:	.rsrc:	...	esi	eax	ebx	ecx	:
0	01kcPWA9K2BOxQeS5Rju	19	744	0	127	57	0	323	0	3	...	66	15	43	83	(
1	1E93CpP60RHFNiT5Qfvn	17	838	0	103	49	0	0	0	3	...	29	48	82	12	(
2	3ekVow2ajZHbTnBcsDfX	17	427	0	50	43	0	145	0	3	...	42	10	67	14	(
3	3X2nY7iQaPBlWDrAZqJe	17	227	0	43	19	0	0	0	3	...	8	14	7	2	(
4	46OZzdsSKDCFV8h7XWxf	17	402	0	59	170	0	0	0	3	...	9	18	29	5	(

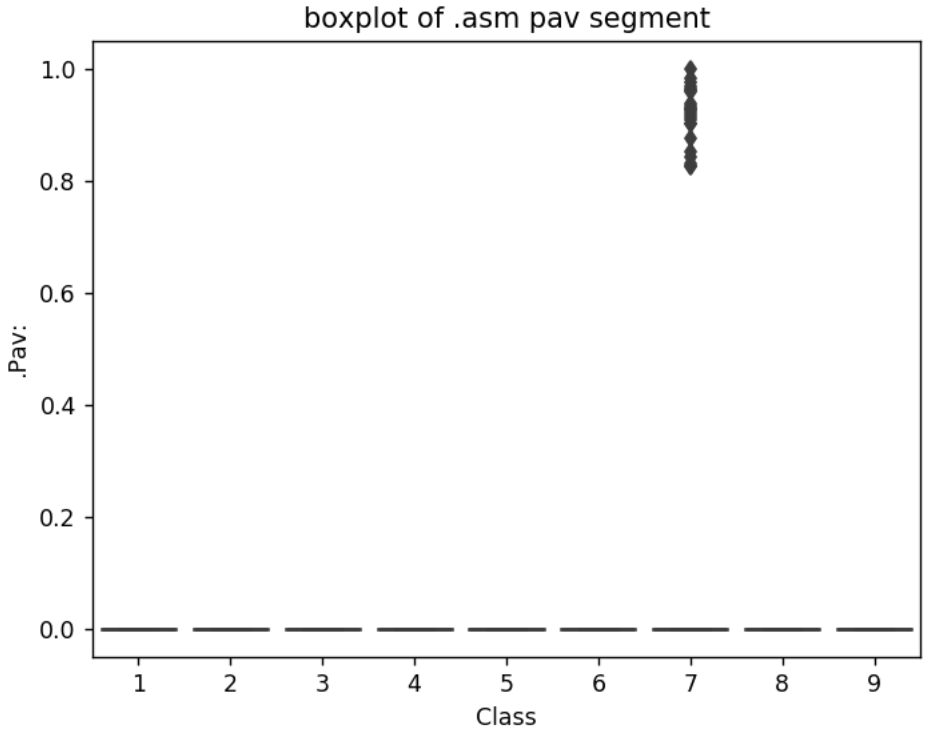
Univariate Analysis

.text vs Class



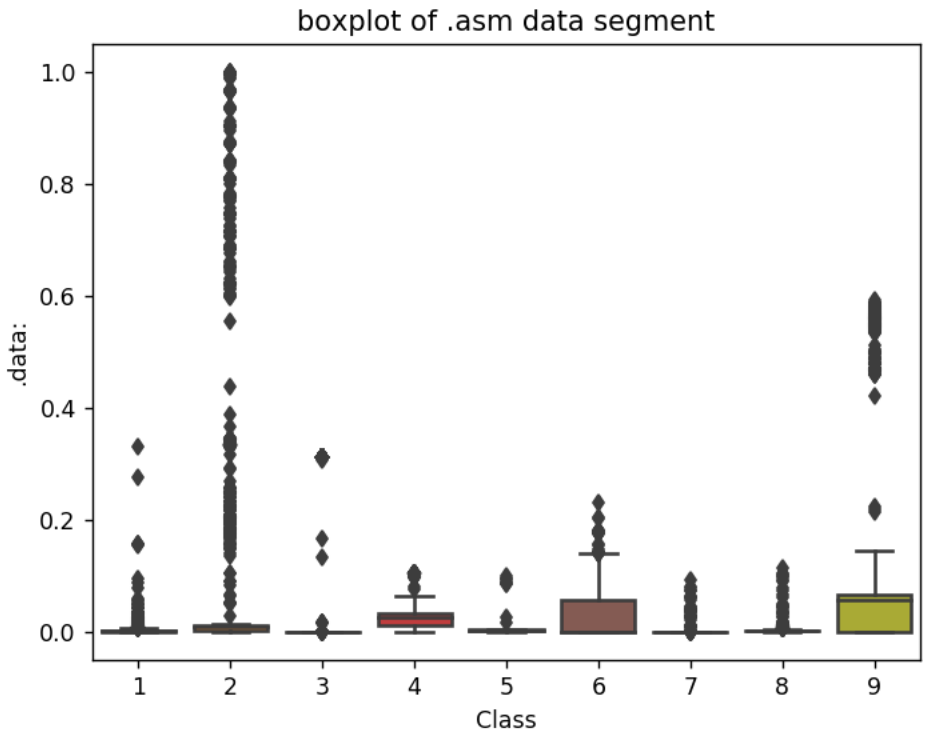

```
The plot is between Text and class
Class 1,2 and 9 can be easly separated
```

.Pav vs Class



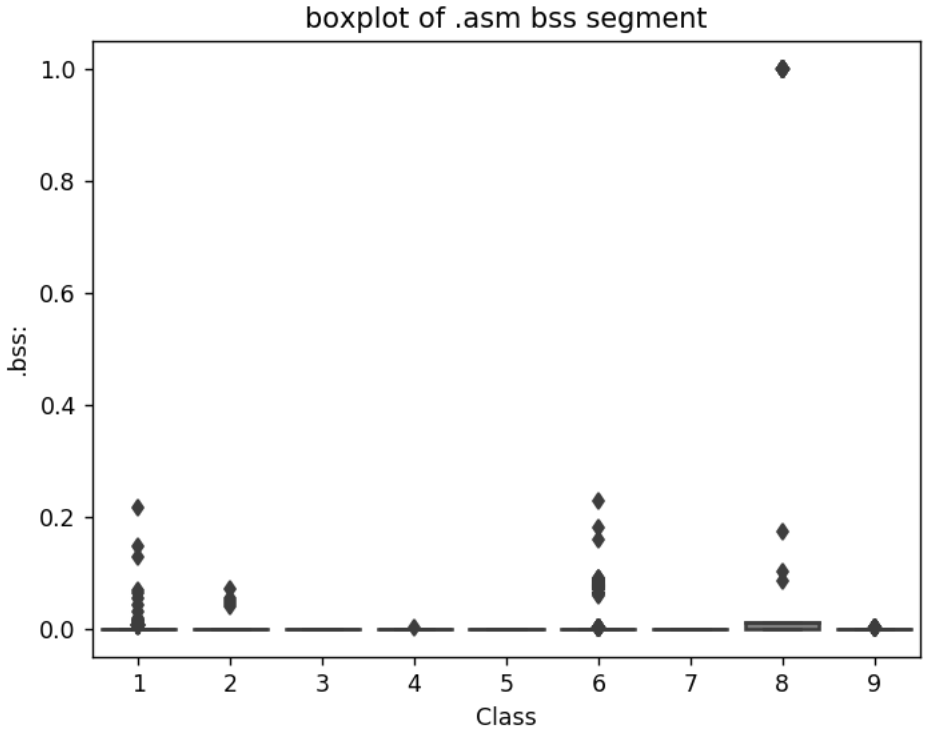

```
No useful information
```

.data vs Class



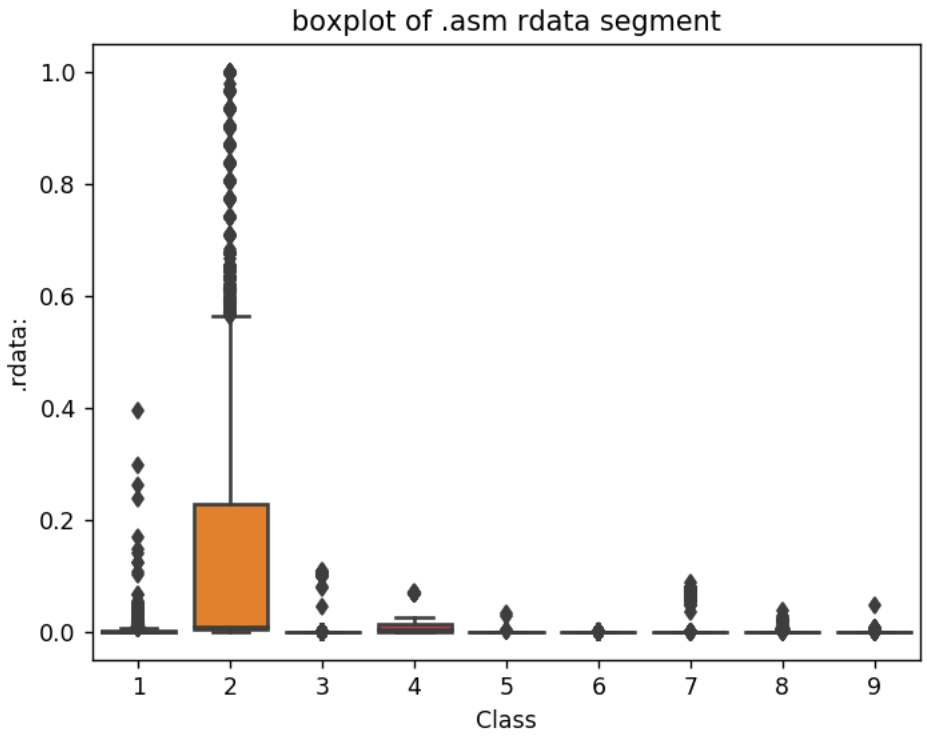

```
The plot is between data segment and class label
class 6 and class 9 can be easily separated from given points
```

.bss vs Class



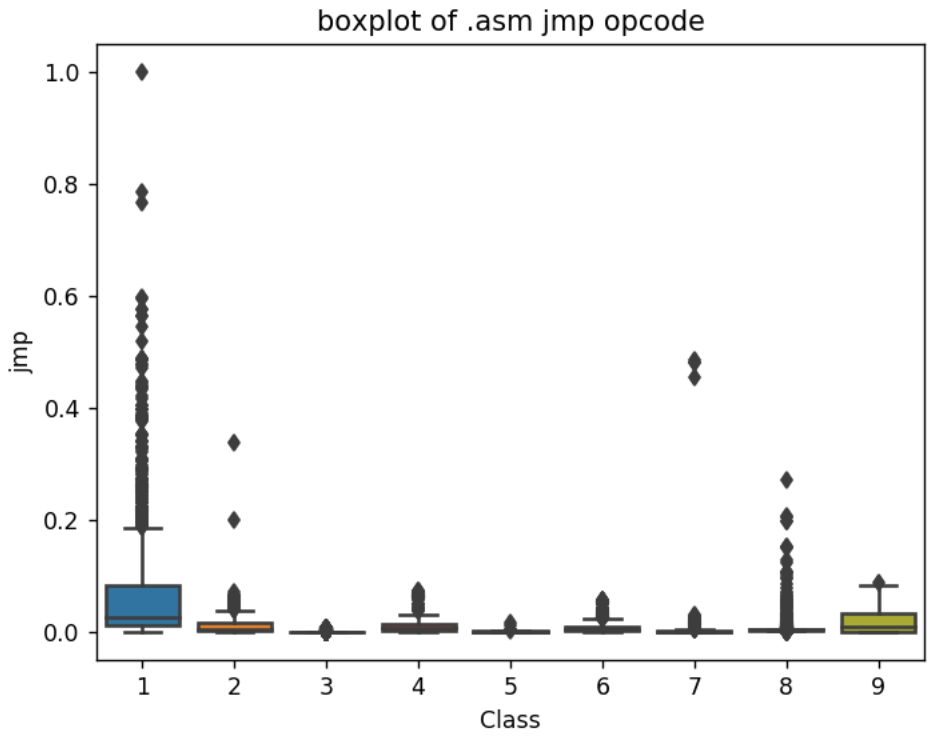

```
plot between bss segment and class label
very less number of files are having bss segment
```

.rdata vs Class



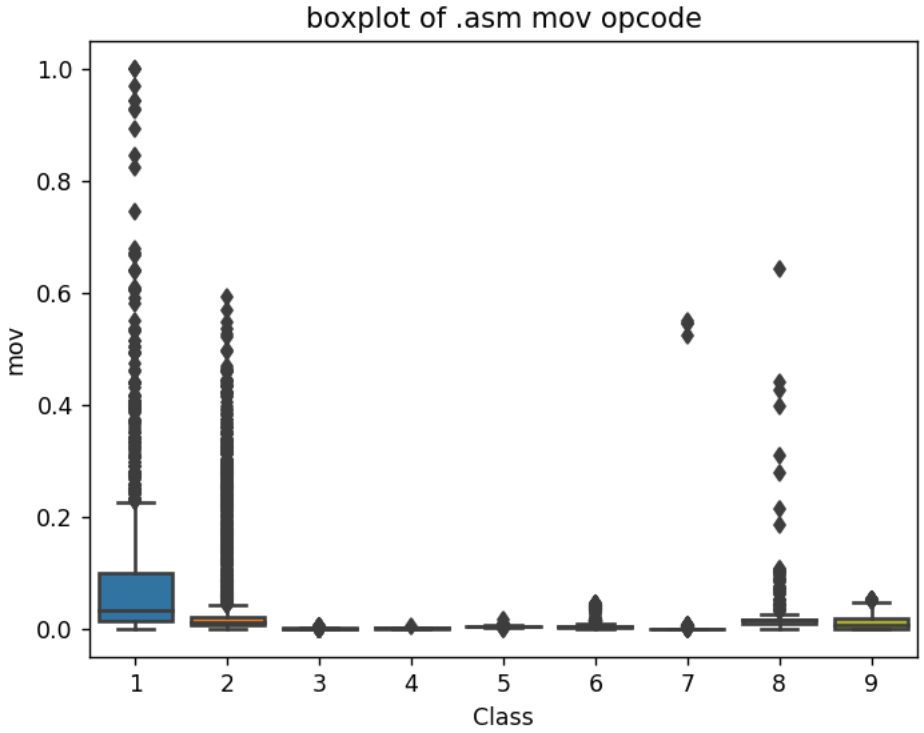

```
Plot between rdata segment and Class segment
Class 2 can be easily separated 75 pecentile files are having 1M rdata lines
```

jmp vs Class



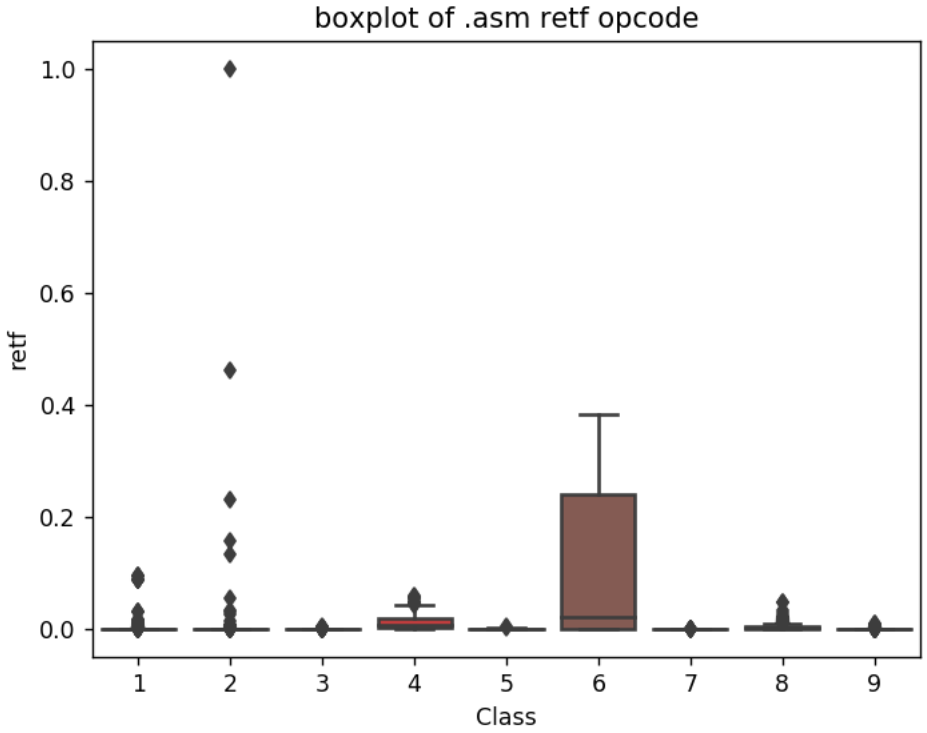

```
plot between jmp and Class label
Class 1 is having frequency of 2000 approx in 75 perentile of files
```

mov vs Class



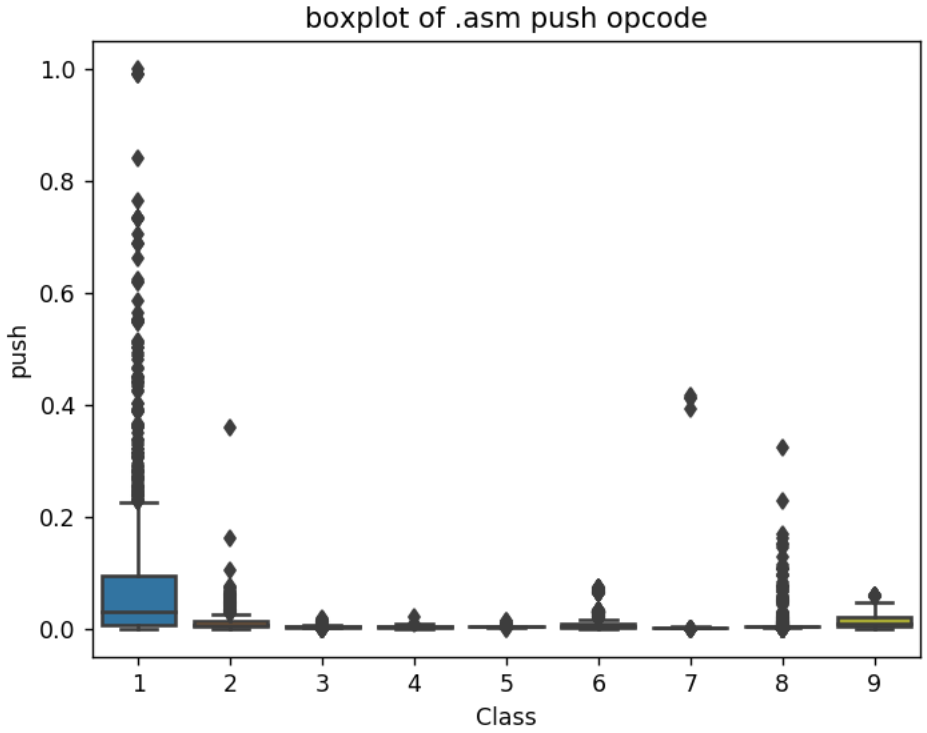

```
plot between Class label and mov opcode
Class 1 is having frequency of 2000 approx in 75 perentile of files
```

retf vs Class




```
plot between Class label and retf
Class 6 can be easily separated with opcode retf
The frequency of retf is approx of 250.
```

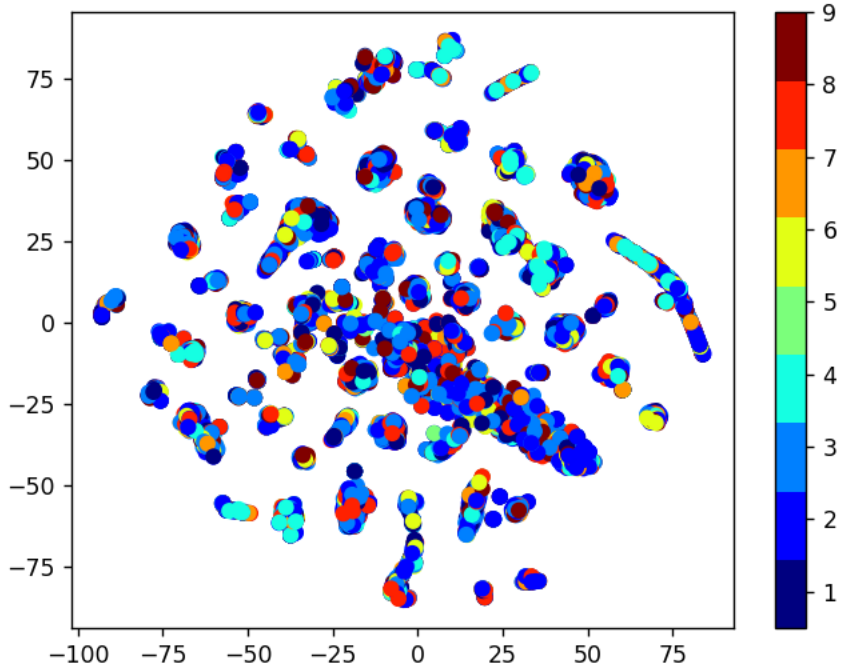
push vs Class



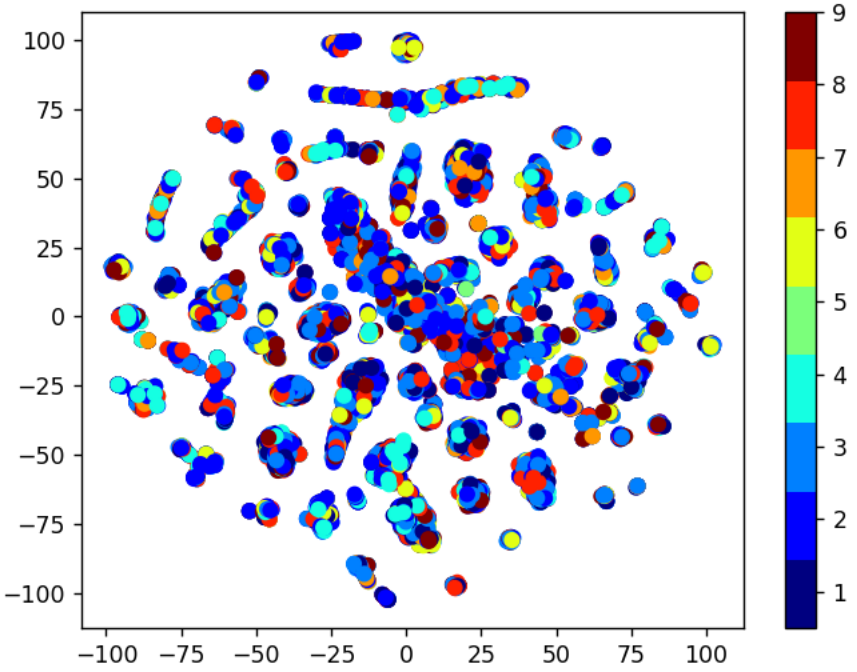

```
plot between push opcode and Class label
Class 1 is having 75 precentile files with push opcodes of frequency 1000
```

Multivariate Analysis

Perplexity = 50



Perplexity = 30



Conclusion of EDA

- We have taken only 52 features from asm files (after reading through many blogs and research papers)
- The univariate analysis was done only on few important features.
- Take-aways

  1. Class 3 can be easily separated because of the frequency of segments, opcodes and keywords being less

  2. Each feature has its unique importance in separating the Class labels.

Train and test Split

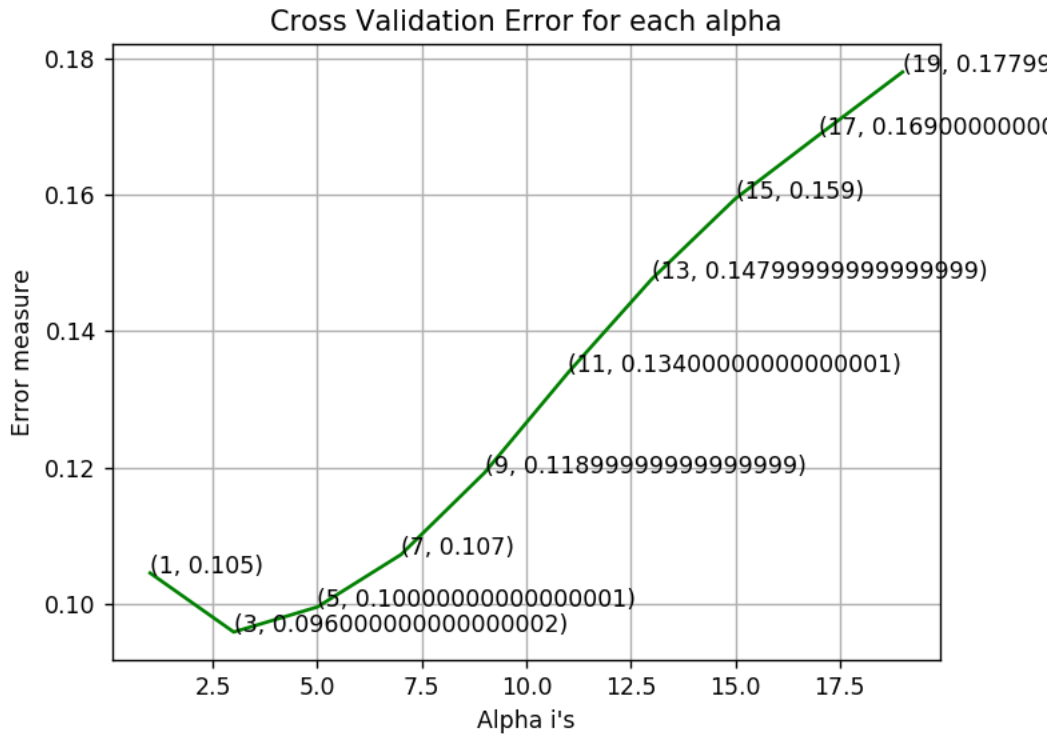
Number of data points in train data: 6955
Number of data points in test data: 2174
Number of data points in cross validation data: 1739

Machine Learning Model

K-Nearest Neighbors

Hyperparameter search

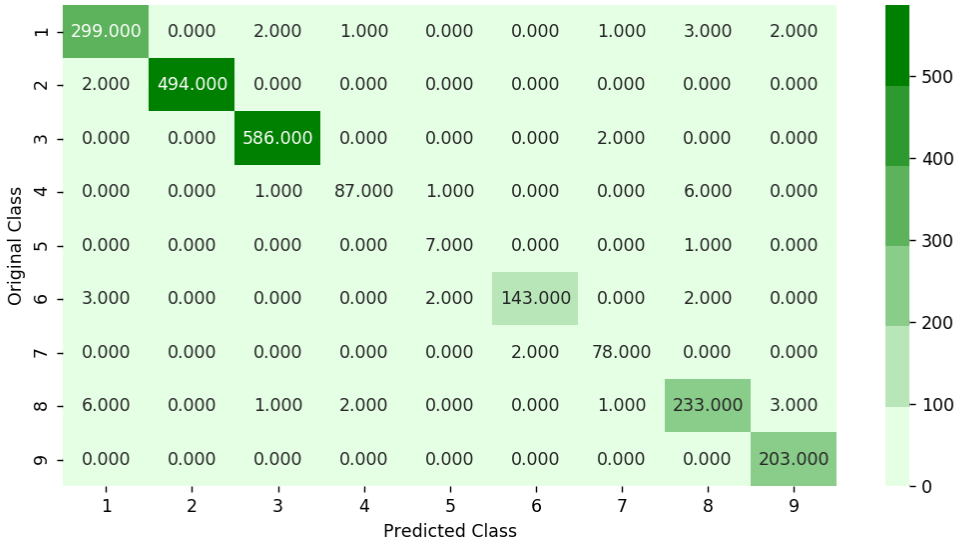
log_loss for k = 1 is 0.104531321344
log_loss for k = 3 is 0.0958800580948
log_loss for k = 5 is 0.0995466557335
log_loss for k = 7 is 0.107227274345
log_loss for k = 9 is 0.119239543547
log_loss for k = 11 is 0.133926642781
log_loss for k = 13 is 0.147643793967
log_loss for k = 15 is 0.159439699615
log_loss for k = 17 is 0.16878376444
log_loss for k = 19 is 0.178020728839



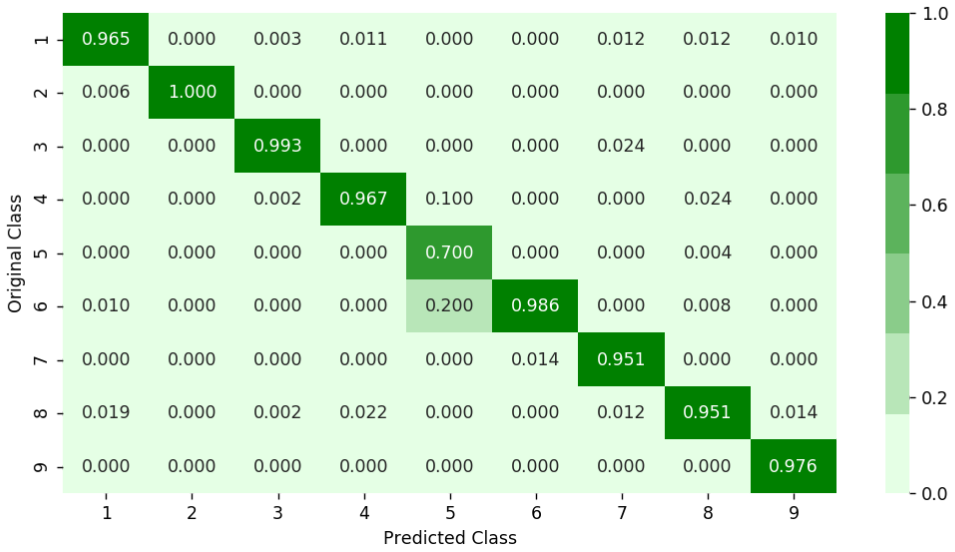
Results from the Best Model

log loss for train data 0.048
log loss for cv data 0.096
log loss for test data 0.090
Accuracy 97.98

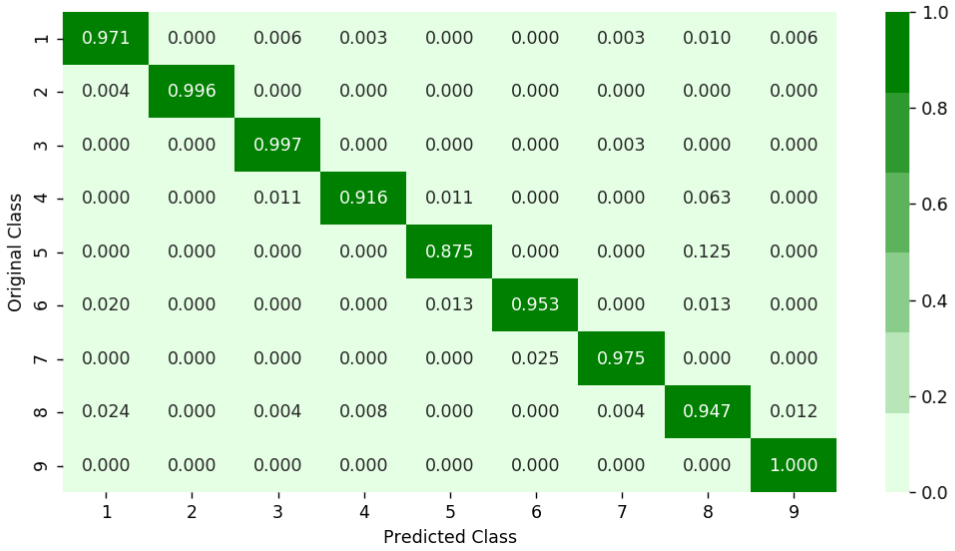
Confusion Matrix



Precision Matrix



Recall Matrix

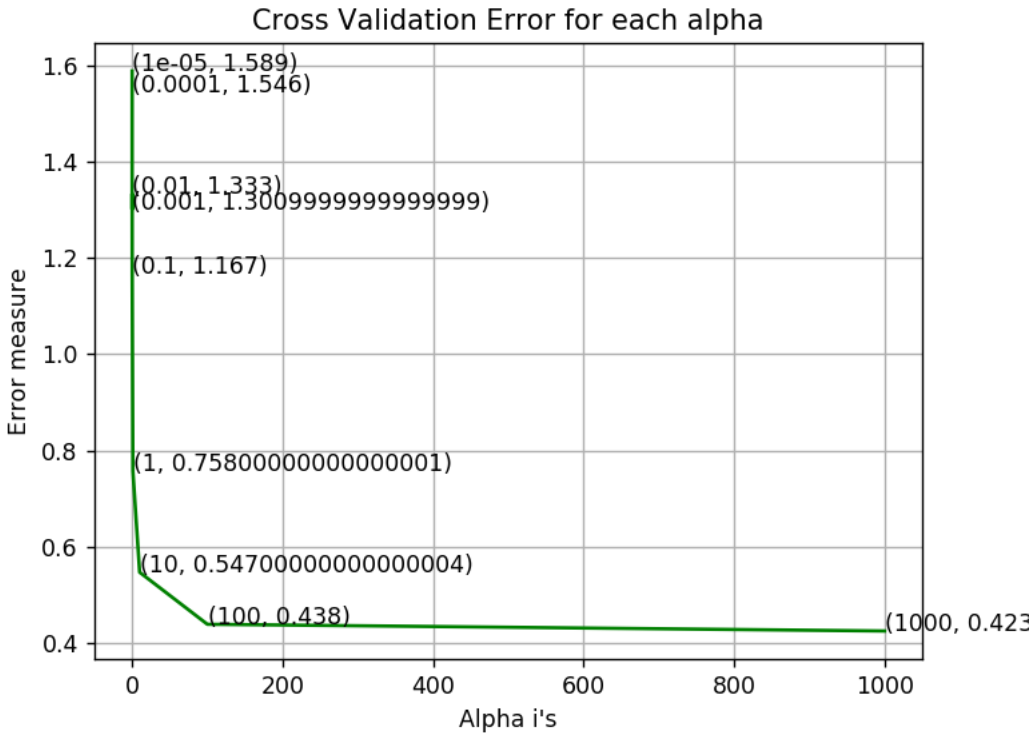


Logistic Regression

Hyperparameter search

log_loss for c = 1e-05 is 1.58867274165
log_loss for c = 0.0001 is 1.54560797884
log_loss for c = 0.001 is 1.30137786807
log_loss for c = 0.01 is 1.33317456931
log_loss for c = 0.1 is 1.16705751378

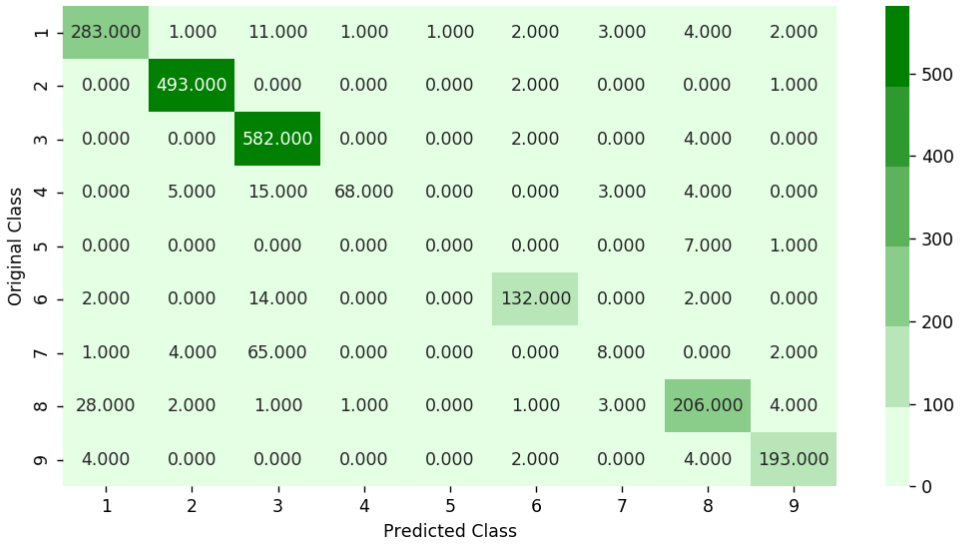
log_loss for c = 1 is 0.757667807779
log_loss for c = 10 is 0.546533939819
log_loss for c = 100 is 0.438414998062
log_loss for c = 1000 is 0.424423536526



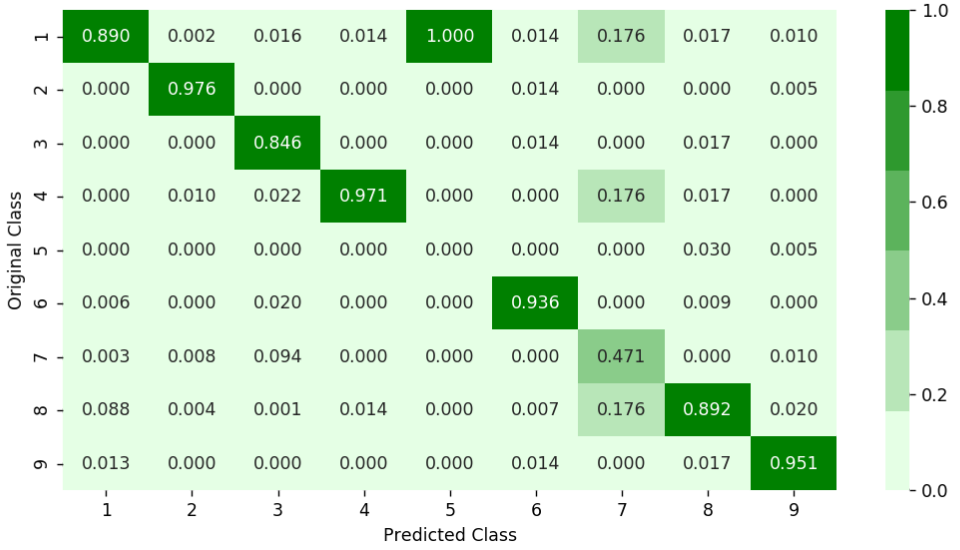
Results from the Best Model

log loss for train data 0.40
log loss for cv data 0.42
log loss for test data 0.42
Number of misclassified points 90.39

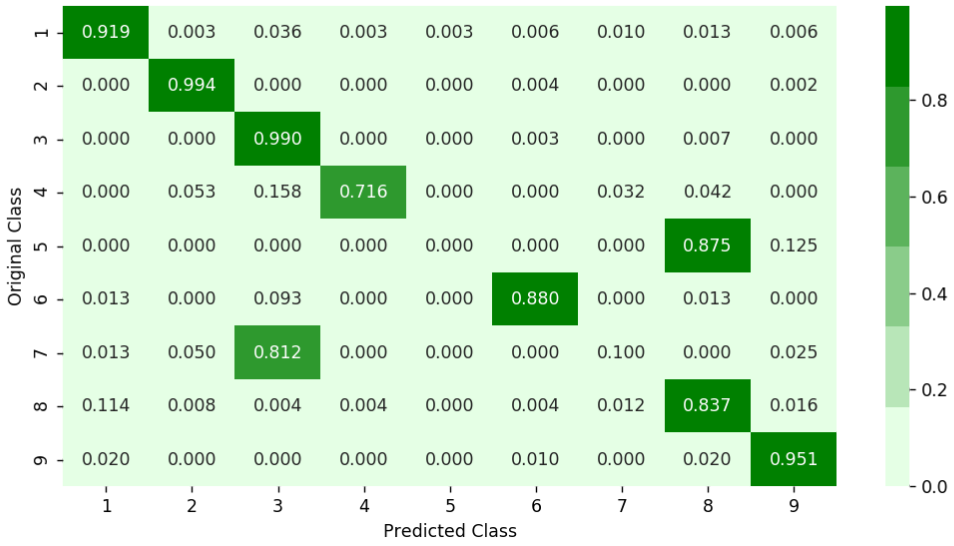
Confusion Matrix



Precision Matrix



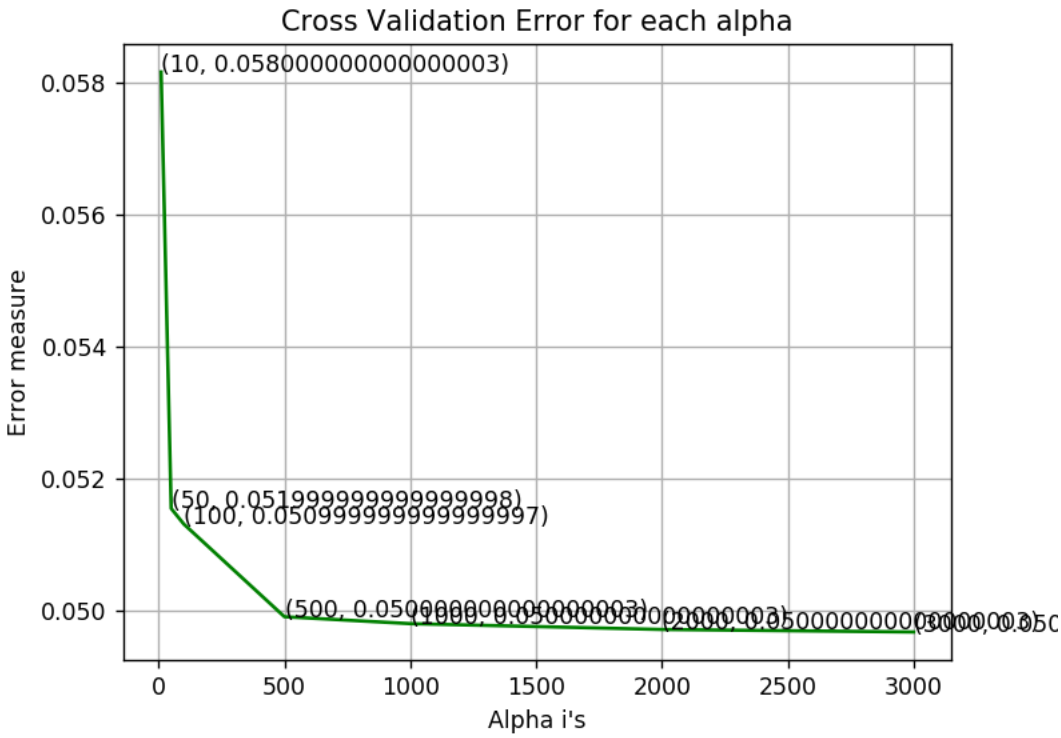
Recall Matrix



Random Forest

Hyperparameter search

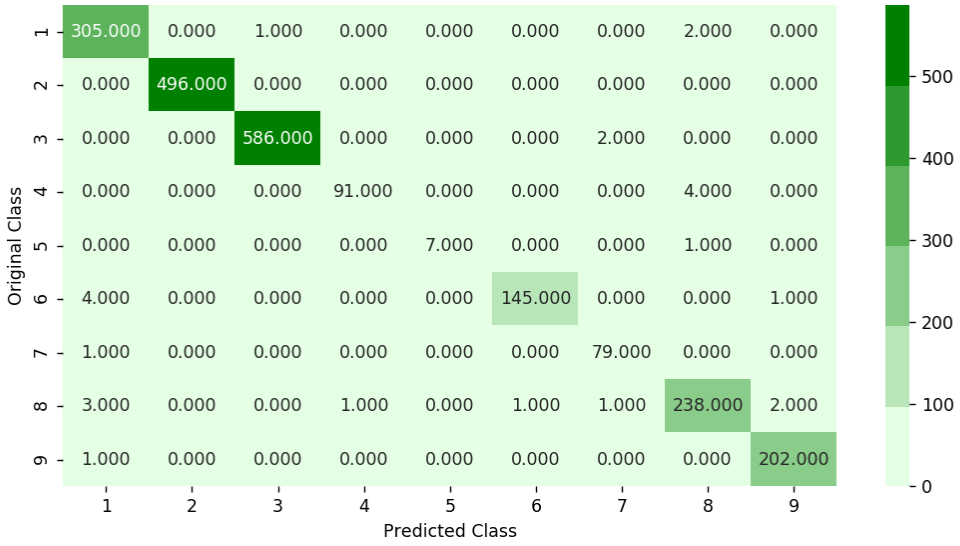
log_loss for c = 10 is 0.0581657906023
log_loss for c = 50 is 0.0515443148419
log_loss for c = 100 is 0.0513084973231
log_loss for c = 500 is 0.0499021761479
log_loss for c = 1000 is 0.0497972474298
log_loss for c = 2000 is 0.0497091690815
log_loss for c = 3000 is 0.0496706817633



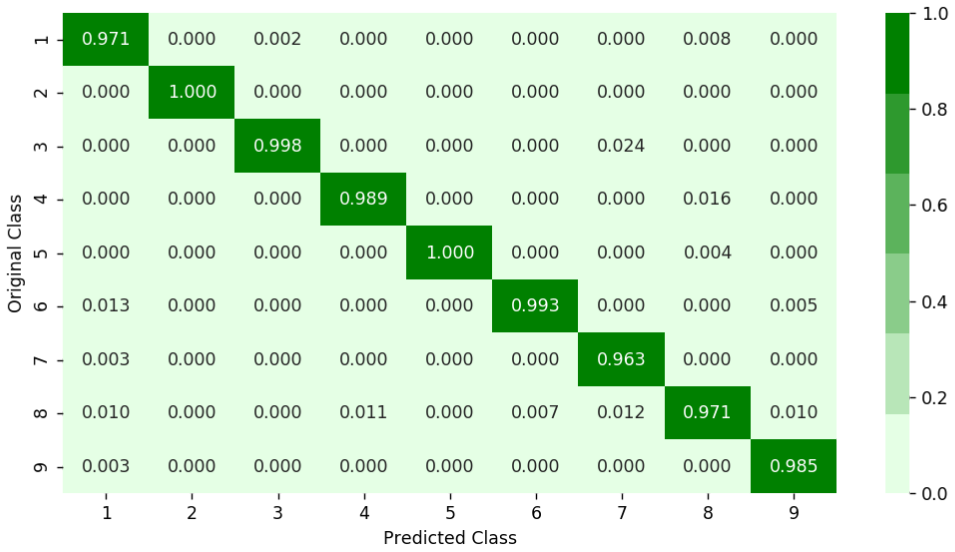
Results from the Best Model

log loss for train data 0.012
log loss for cv data 0.050
log loss for test data 0.057
Accuracy 98.85

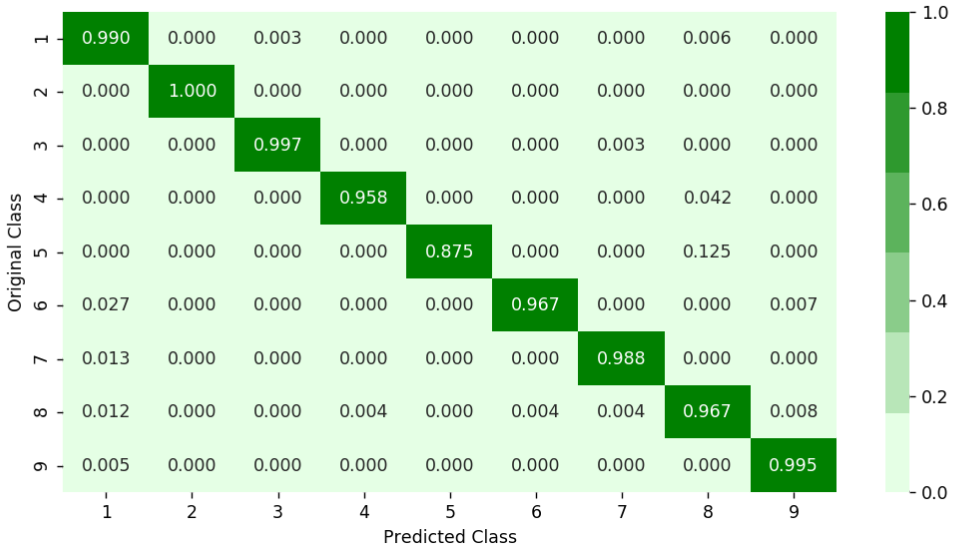
Confusion Matrix



Precision Matrix



Recall Matrix

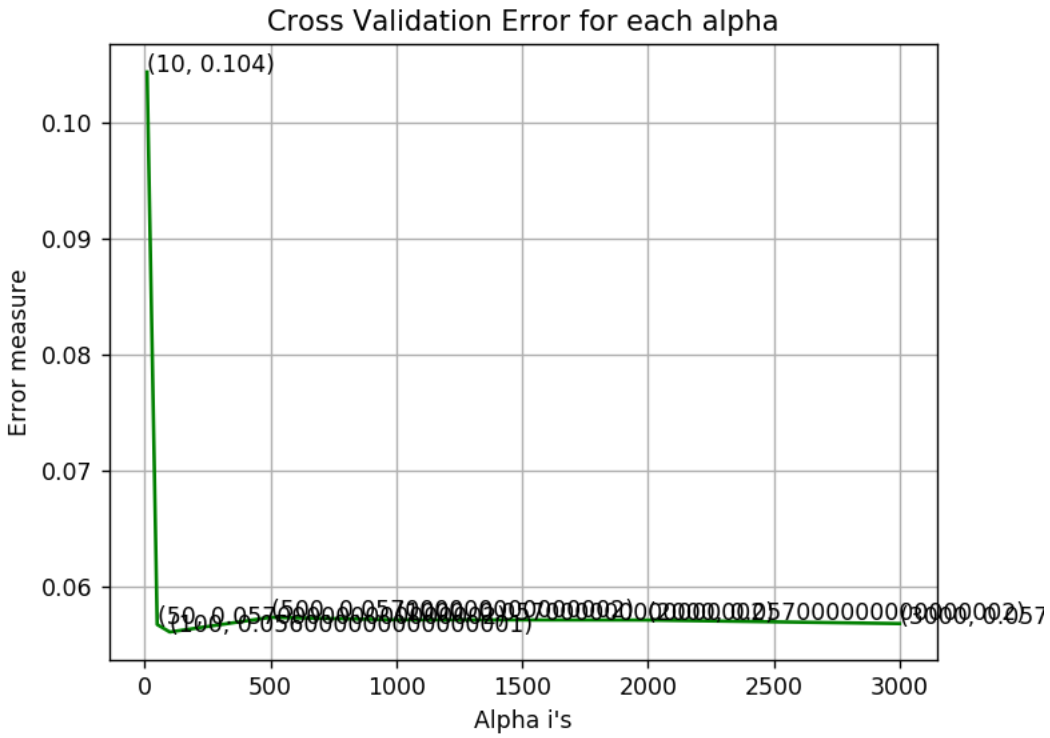


XgBoost Classifier

Hyperparameter search

log_loss for c = 10 is 0.104344888454
log_loss for c = 50 is 0.0567190635611
log_loss for c = 100 is 0.056075038646
log_loss for c = 500 is 0.057336051683
log_loss for c = 1000 is 0.0571265109903

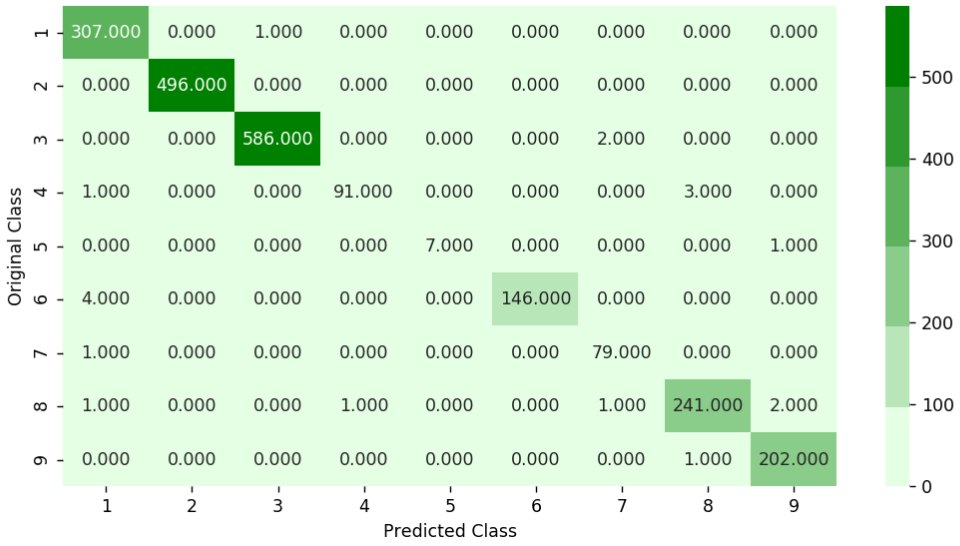
log_loss for c = 2000 is 0.057103406781
log_loss for c = 3000 is 0.0567993215778



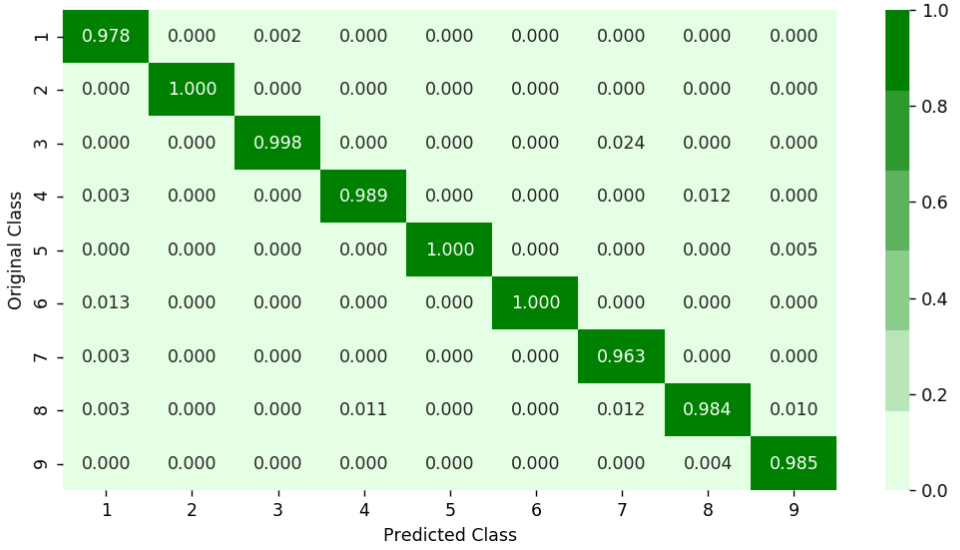
Results from the Best Model

For values of best alpha = 100 The train log loss is: 0.012
For values of best alpha = 100 The cross validation log loss is: 0.056
For values of best alpha = 100 The test log loss is: 0.049
Accuracy 99.13

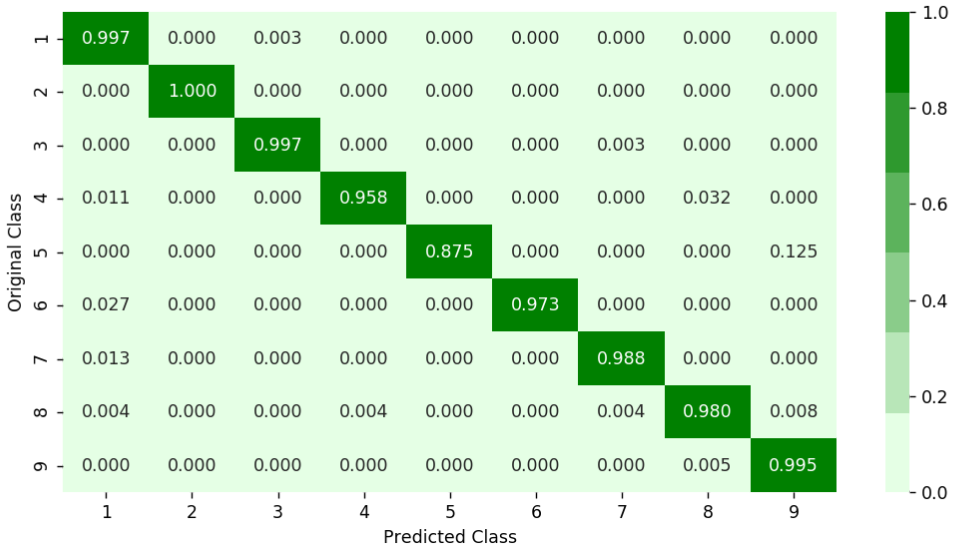
Confusion Matrix



Precision Matrix



Recall Matrix



XgBoost Classifier with best hyperparameters

```
Fitting 3 folds for each of 10 candidates, totalling 30 fits


[Parallel(n_jobs=-1)]: Done   2 tasks      | elapsed:    8.1s
[Parallel(n_jobs=-1)]: Done   9 tasks      | elapsed:   32.8s
[Parallel(n_jobs=-1)]: Done  19 out of  30 | elapsed:  1.1min remaining:   39.3s
[Parallel(n_jobs=-1)]: Done  23 out of  30 | elapsed:  1.3min remaining:   23.0s
[Parallel(n_jobs=-1)]: Done  27 out of  30 | elapsed:  1.4min remaining:    9.2s
[Parallel(n_jobs=-1)]: Done  30 out of  30 | elapsed:  2.3min finished


RandomizedSearchCV(cv=None, error_score='raise',
          estimator=XGBClassifier(base_score=0.5, colsample_bylevel=1, colsample_bytree=1,
       gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=3,
       min_child_weight=1, missing=None, n_estimators=100, nthread=-1,
       objective='binary:logistic', reg_alpha=0, reg_lambda=1,
       scale_pos_weight=1, seed=0, silent=True, subsample=1),
          fit_params=None, iid=True, n_iter=10, n_jobs=-1,
          param_distributions={'learning_rate': [0.01, 0.03, 0.05, 0.1, 0.15, 0.2], 'n_estimators': [100, 200, 500, 1000, 2000], 'max_dep
th': [3, 5, 10], 'colsample_bytree': [0.1, 0.3, 0.5, 1], 'subsample': [0.1, 0.3, 0.5, 1]},
          pre_dispatch='2*n_jobs', random_state=None, refit=True,
          return_train_score=True, scoring=None, verbose=10)
```

Best Parameters

{'subsample': 1, 'n_estimators': 200, 'max_depth': 5, 'learning_rate': 0.15, 'colsample_bytree': 0.5}

Results from the Best Parameter Model

train loss 0.010
cv loss 0.050
test loss 0.048 Accuracy 99.16