Status Report: Gaussian Maximum Likelihood Classifier

Eileanor LaRocco • 10.23.2024

Overview

Progress

- Updated code from last week to train/predict on 80/20 and 60/40 sampling for the four datasets: DC, polymers, indian pines, upwins
- Evaluated using accuracy, balanced accuracy, f1 score, and confusion matrix
- Ran lazy predict on the indian pines and upwins datasets and sklearn lda on the dc and polymers datasets

Issues/Modifications

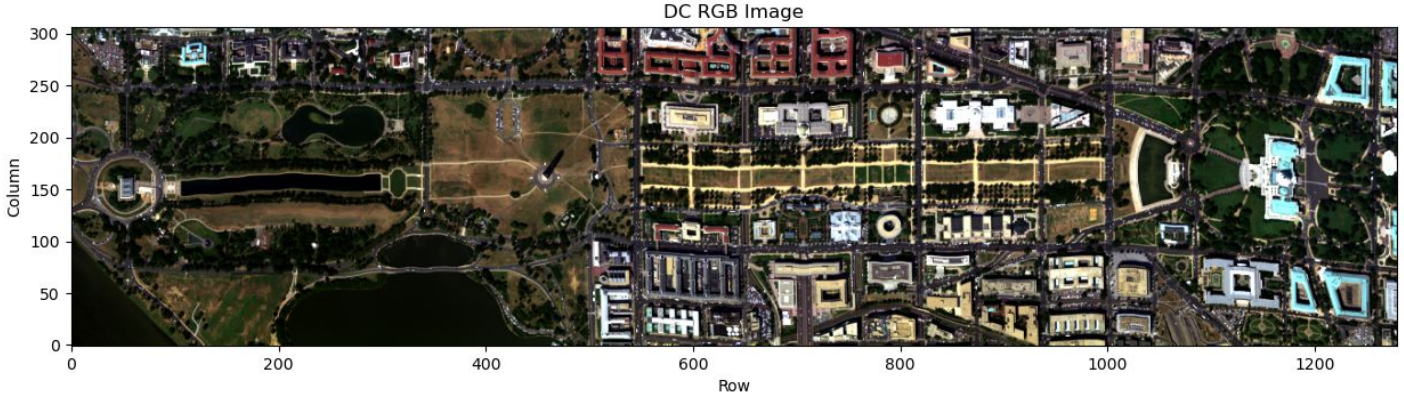
- Lazy classifier bug
  - work around: cloned repo
- Not enough memory to run lazy classifier on DC and Polymers datasets
  - work around: ran sklearn LDA for comparison
- ROC score does not make sense as we're thresholding by MD not predicting class probabilities and is not useful in terms of our evaluation

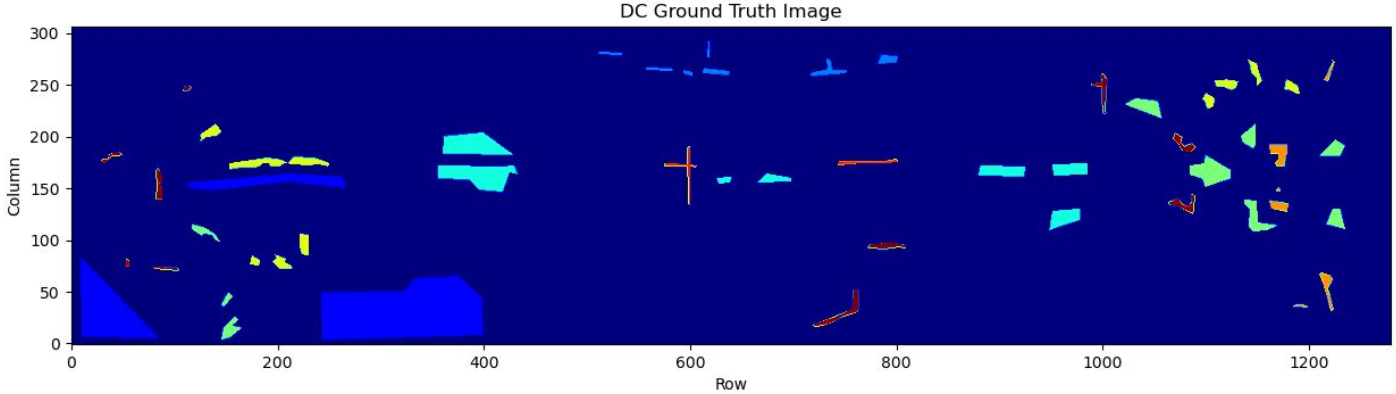
Next Steps

- Threshold by MD
- Solve memory issue moving forward

Results

DC Dataset


DC RGB Image
Column
Row
0
50
100
150
200
250
300
0
200
400
600
800
1000
1200


DC Ground Truth Image
Column
Row
0
50
100
150
200
250
300
0
200
400
600
800
1000
1200

DC Dataset - Model Comparison

80/20 Split

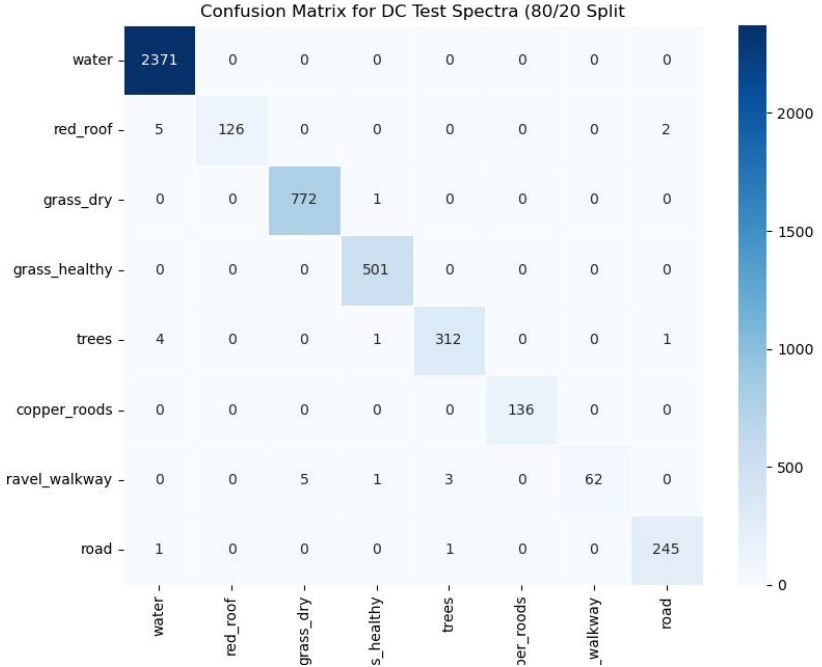
Model	Accuracy	Balanced Accuracy	F1 Score
Sklearn LDA	0.8004	0.6766	0.8574
Our Model	0.9945	0.9740	0.9944

60/40 Split

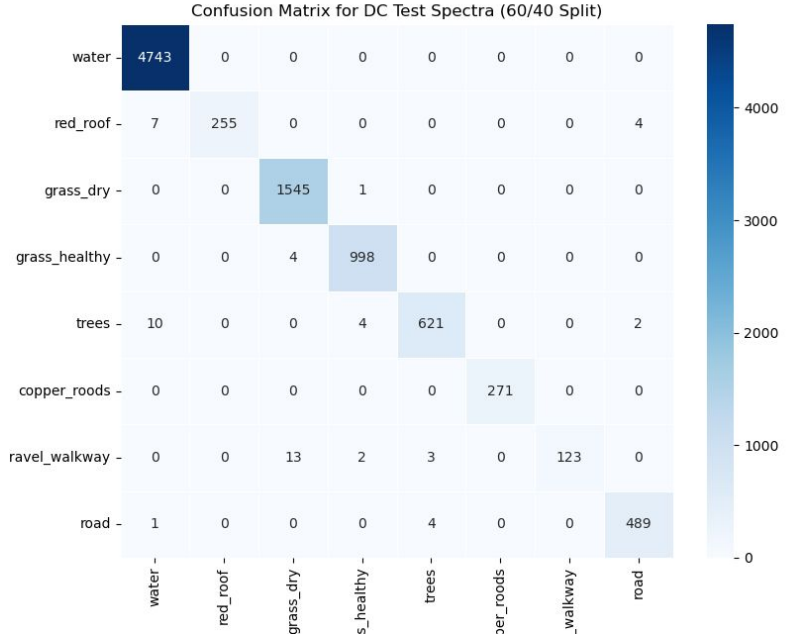
Model	Accuracy	Balanced Accuracy	F1 Score
Sklearn LDA	0.8011	0.6785	0.8577
Our Model	0.9940	0.9739	0.9939

DC Dataset - Confusion Matrices

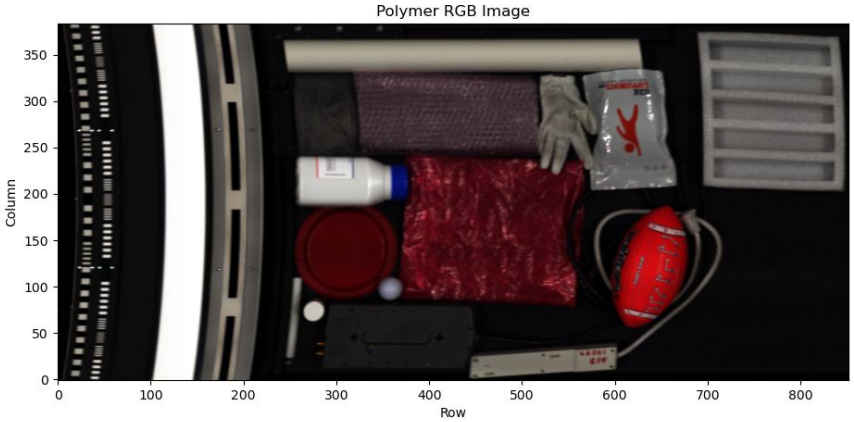
80/20 Split

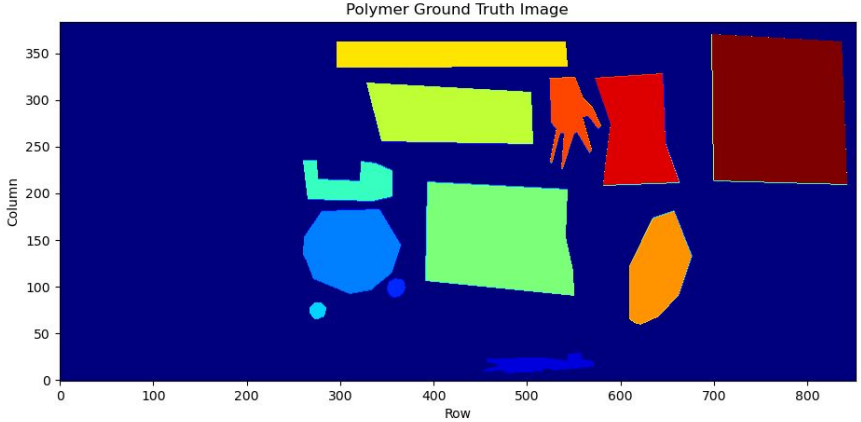


60/40 Split



Polymers Dataset





Polymers Dataset - Model Comparison

80/20 Split

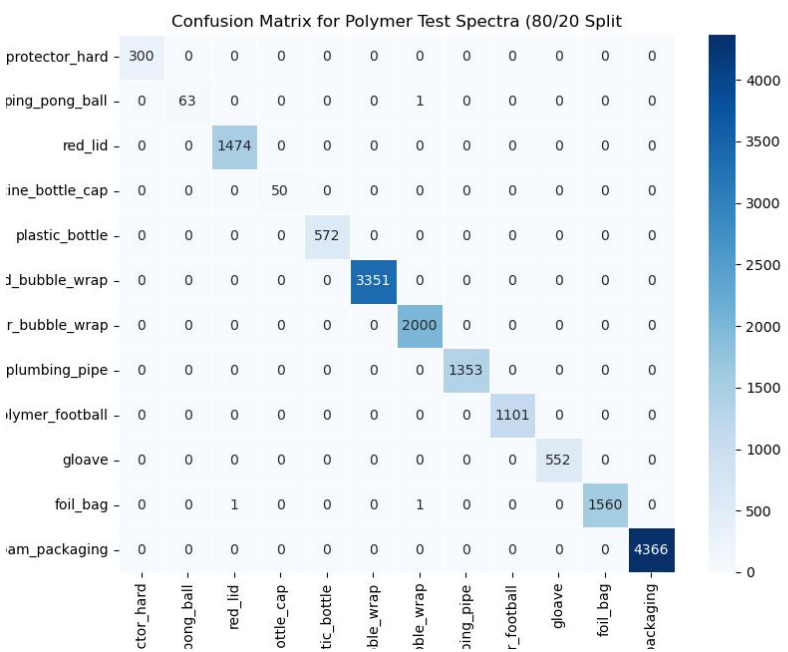
Model	Accuracy	Balanced Accuracy	F1 Score
Sklearn LDA	0.9941	0.9931	0.9969
Our Model	0.9998	0.9990	0.9998

60/40 Split

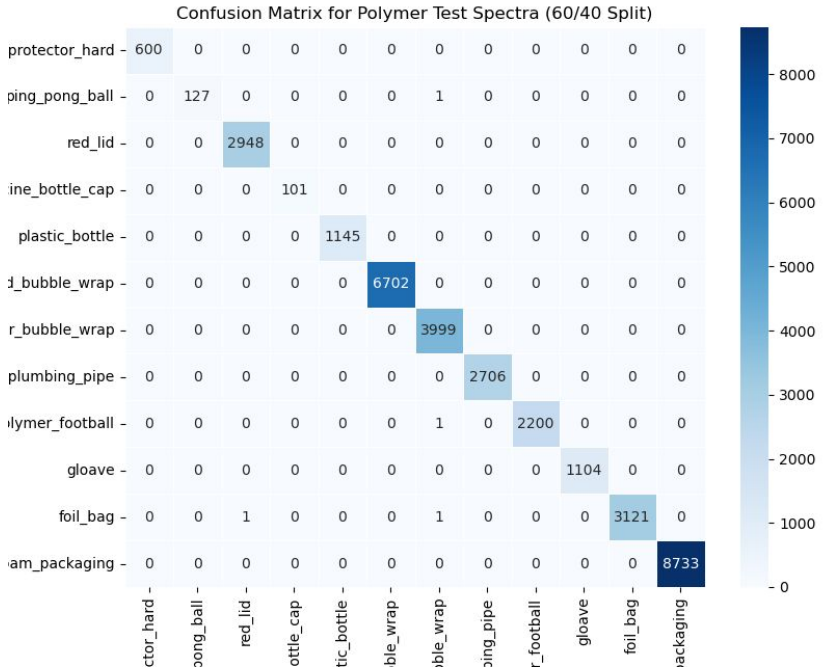
Model	Accuracy	Balanced Accuracy	F1 Score
Sklearn LDA	0.9945	0.9943	0.997
Our Model	0.9999	0.9993	0.9999

Polymers Dataset - Confusion Matrices

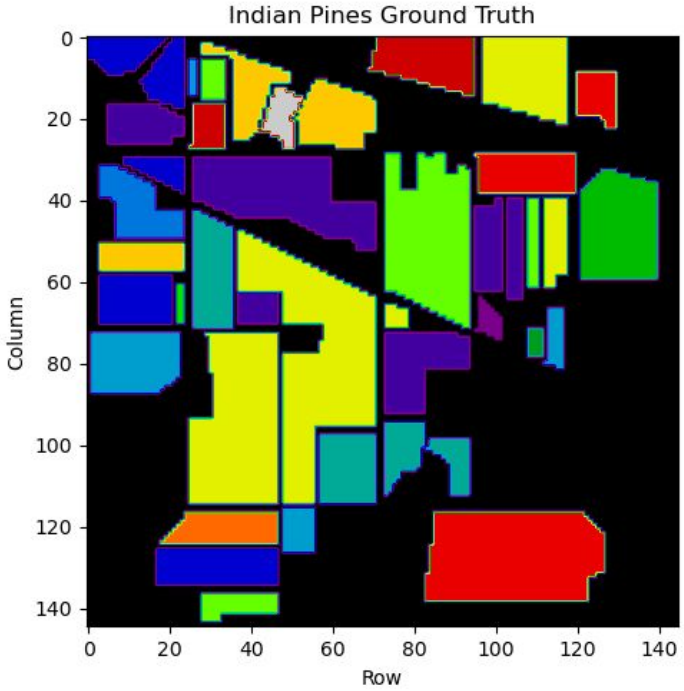
80/20 Split



60/40 Split



Indian Pines Dataset



Indian Pines Dataset - Model Comparison

80/20 Split

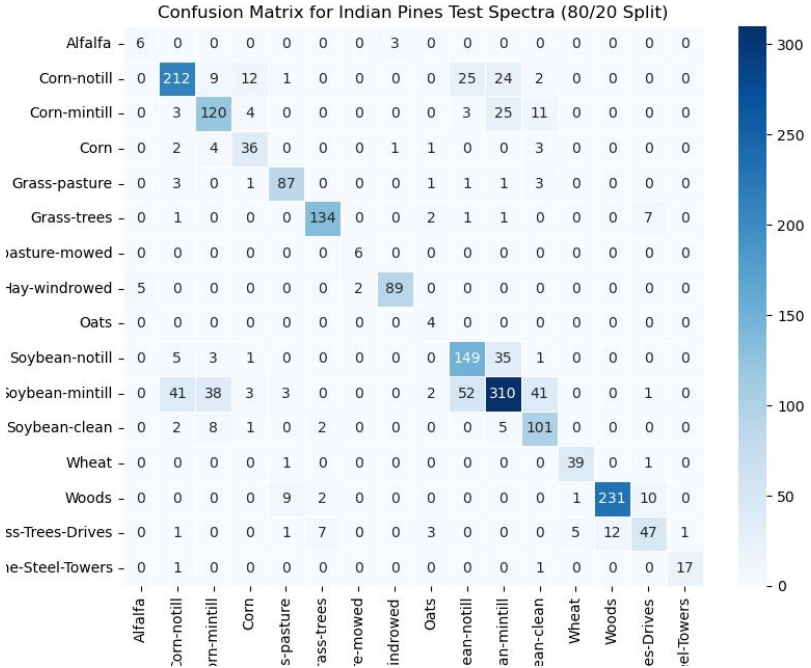
Model	Accuracy	Balanced Accuracy	F1 Score	Time Taken
XGBClassifier	0.8357	0.6719	0.8305	21.5463
LinearDiscriminantAnalysis	0.6675	0.6688	0.6764	0.7551
ExtraTreesClassifier	0.8138	0.6505	0.8011	2.4684
LGBMClassifier	0.8269	0.6356	0.8216	32.0967
LabelPropagation	0.6982	0.6114	0.6986	3.4480
LabelSpreading	0.6980	0.6113	0.6984	9.7008
RandomForestClassifier	0.8100	0.5986	0.7973	11.2852
LinearSVC	0.7715	0.5946	0.7600	9.8594
BaggingClassifier	0.7800	0.5909	0.7664	17.9792
KNeighborsClassifier	0.7337	0.5782	0.7235	0.2966
DecisionTreeClassifier	0.6868	0.5741	0.6875	2.7259
CalibratedClassifierCV	0.7672	0.5326	0.7521	33.6924
ExtraTreeClassifier	0.6509	0.5307	0.6514	0.0783
LogisticRegression	0.7498	0.5265	0.7404	4.8921
GaussianNB	0.2920	0.5127	0.2757	0.0895
BernoulliNB	0.2735	0.4981	0.2286	0.1185
Perceptron	0.6585	0.4744	0.6558	1.5501
NearestCentroid	0.2507	0.4721	0.2415	0.1029
SVC	0.7536	0.4669	0.7288	20.3195
PassiveAggressiveClassifier	0.6637	0.3818	0.6271	1.9815
SGDClassifier	0.6606	0.3785	0.6349	13.0687
QuadraticDiscriminantAnalysis	0.6176	0.3709	0.6096	3.4327
RidgeClassifier	0.6549	0.2481	0.5977	0.1941
RidgeClassifierCV	0.6549	0.2481	0.5977	3.5564
AdaBoostClassifier	0.4744	0.1105	0.4231	7.4715
DummyClassifier	0.5125	0.0588	0.3473	0.0532
Our Model	0.7746	0.8287	0.7754	

60/40 Split

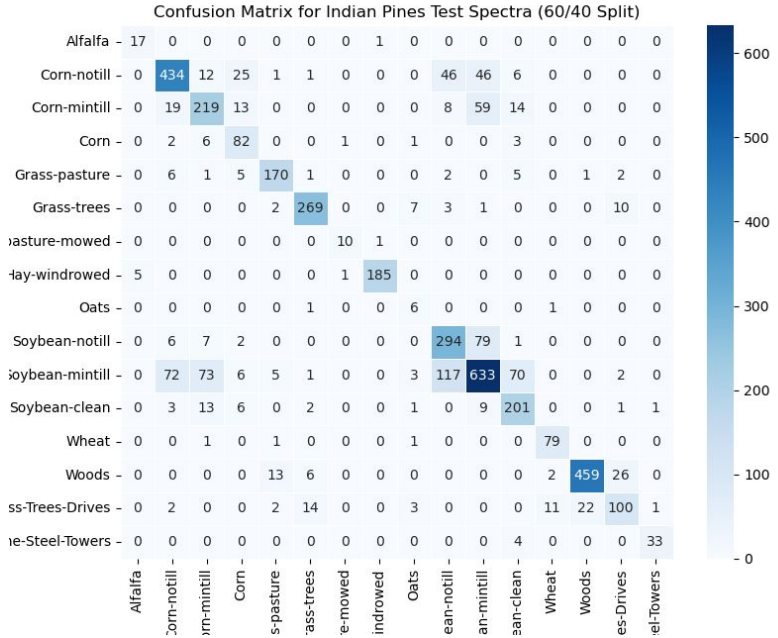
Model	Accuracy	Balanced Accuracy	F1 Score	Time Taken
LinearDiscriminantAnalysis	0.6755	0.6731	0.6837	0.6555
XGBClassifier	0.8237	0.6479	0.8180	17.4769
ExtraTreesClassifier	0.8087	0.6354	0.7949	1.8510
RandomForestClassifier	0.8037	0.6171	0.7903	8.6445
LabelPropagation	0.6904	0.6101	0.6905	3.0177
LabelSpreading	0.6905	0.6089	0.6907	13.7686
LinearSVC	0.7642	0.5995	0.7526	6.6014
KNeighborsClassifier	0.7243	0.5677	0.7142	0.2985
BaggingClassifier	0.7742	0.5607	0.7599	16.4822
DecisionTreeClassifier	0.6775	0.5508	0.6778	2.0997
LogisticRegression	0.7468	0.5444	0.7368	4.3194
CalibratedClassifierCV	0.7590	0.5305	0.7428	23.1915
ExtraTreeClassifier	0.6455	0.5141	0.6466	0.0686
GaussianNB	0.3057	0.5067	0.2874	0.1194
NearestCentroid	0.2562	0.4662	0.2479	0.1716
Perceptron	0.6757	0.4652	0.6702	1.0712
SVC	0.7442	0.4639	0.7143	18.3097
BernoulliNB	0.2644	0.4388	0.2187	0.1263
PassiveAggressiveClassifier	0.6699	0.4251	0.6372	1.6502
LGBMClassifier	0.5195	0.3796	0.5385	21.0704
SGDClassifier	0.6661	0.3330	0.6392	7.7069
QuadraticDiscriminantAnalysis	0.6134	0.2807	0.5874	1.7031
RidgeClassifier	0.6583	0.2487	0.6015	0.1369
RidgeClassifierCV	0.6583	0.2487	0.6015	0.8311
AdaBoostClassifier	0.4778	0.1105	0.4257	8.9255
DummyClassifier	0.5126	0.0588	0.3474	0.0517
Our Model	0.7785	0.8321	0.7794	

Indian Pines Dataset - Confusion Matrices

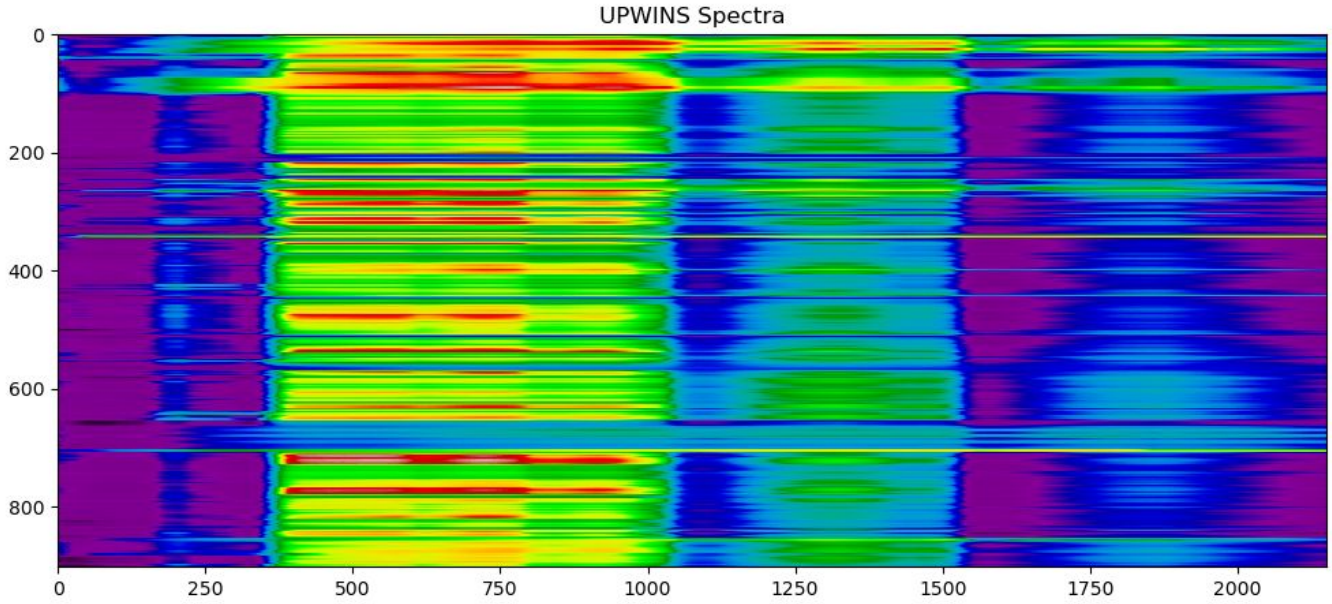
80/20 Split



60/40 Split



UPWINS Dataset



UPWINS Dataset - Model Comparison

80/20 Split

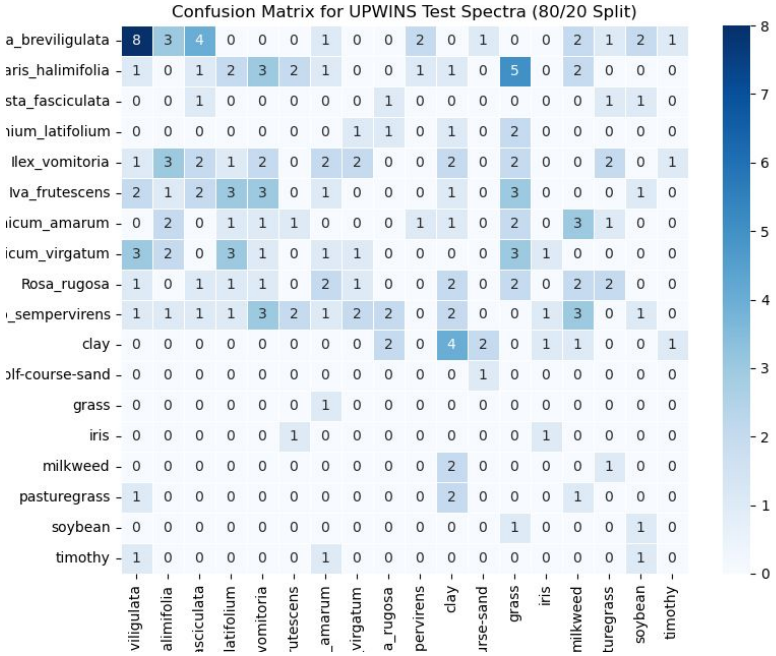
Model	Accuracy	Balanced Accuracy	F1 Score	Time Taken
LinearDiscriminantAnalysis	0.9503	0.9722	0.9500	1.7719
RidgeClassifierCV	0.9006	0.8349	0.8953	0.6287
LinearSVC	0.8232	0.7903	0.8259	17.1745
ExtraTreesClassifier	0.8840	0.7885	0.8796	0.2704
RandomForestClassifier	0.8785	0.7841	0.8731	1.2227
RidgeClassifier	0.8232	0.6980	0.8127	0.1623
CalibratedClassifierCV	0.8232	0.6849	0.8154	79.4940
BaggingClassifier	0.8287	0.6718	0.8212	4.7823
ExtraTreeClassifier	0.8343	0.6584	0.8275	0.0333
DecisionTreeClassifier	0.7624	0.6535	0.7584	1.1096
LogisticRegression	0.7569	0.6328	0.7422	3.1614
LGBMClassifier	0.8398	0.6217	0.8311	33.1873
XGBClassifier	0.8508	0.5938	0.8310	23.7305
PassiveAggressiveClassifier	0.6740	0.5757	0.6888	1.8107
KNeighborsClassifier	0.6630	0.5554	0.6421	0.0643
SGDClassifier	0.5912	0.5239	0.5892	0.7330
GaussianNB	0.4309	0.5206	0.4100	0.0479
SVC	0.6022	0.4416	0.5709	0.7105
LabelSpreading	0.6630	0.4225	0.6861	0.2903
LabelPropagation	0.6630	0.4225	0.6861	0.3152
NearestCentroid	0.3425	0.4108	0.3238	0.0687
Perceptron	0.4807	0.3859	0.4945	0.5754
BernoulliNB	0.3370	0.3254	0.3096	0.0984
QuadraticDiscriminantAnalysis	0.3536	0.2226	0.3226	1.2541
AdaBoostClassifier	0.2376	0.1005	0.1200	6.0161
DummyClassifier	0.1436	0.0556	0.0361	0.0307
Our LDA	0.1050	0.1722	0.0986	

60/40 Split

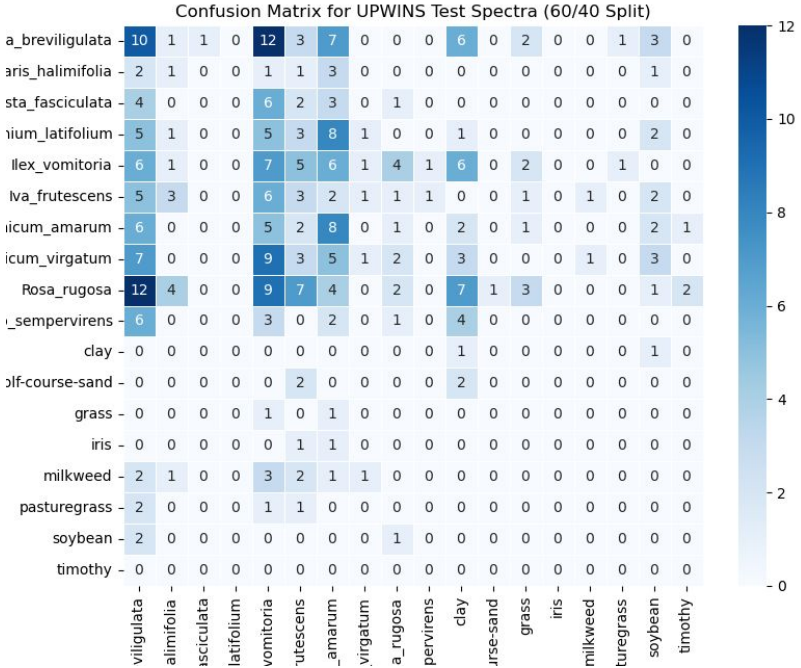
Model	Accuracy	Balanced Accuracy	F1 Score	Time Taken
LinearDiscriminantAnalysis	0.9141	0.9482	0.9160	1.1989
RidgeClassifierCV	0.8615	0.8165	0.8603	0.2661
RandomForestClassifier	0.8449	0.7624	0.8385	0.9254
LogisticRegression	0.7895	0.7523	0.7891	3.4729
LinearSVC	0.8144	0.7334	0.8142	13.6971
RidgeClassifier	0.8116	0.7181	0.8079	0.0575
ExtraTreesClassifier	0.8255	0.7067	0.8135	0.3374
CalibratedClassifierCV	0.8144	0.6870	0.8104	67.5420
LGBMClassifier	0.8366	0.6766	0.8288	21.2609
BaggingClassifier	0.8033	0.6556	0.7920	3.5819
XGBClassifier	0.7784	0.6179	0.7669	19.8486
ExtraTreeClassifier	0.7562	0.6145	0.7470	0.0445
DecisionTreeClassifier	0.7202	0.5825	0.7092	0.8728
SGDClassifier	0.6343	0.5723	0.6136	0.5390
GaussianNB	0.4626	0.5420	0.4353	0.0896
PassiveAggressiveClassifier	0.6260	0.5087	0.6272	1.3856
KNeighborsClassifier	0.6454	0.4668	0.6210	0.1366
SVC	0.5734	0.4183	0.5458	0.6469
Perceptron	0.4875	0.4143	0.5028	0.5239
NearestCentroid	0.3269	0.3949	0.3181	0.0534
LabelPropagation	0.5346	0.3349	0.5757	0.0996
LabelSpreading	0.5346	0.3349	0.5757	0.2436
BernoulliNB	0.3380	0.3177	0.3040	0.1163
QuadraticDiscriminantAnalysis	0.3213	0.1863	0.2815	0.8714
AdaBoostClassifier	0.2327	0.1261	0.1234	4.2808
DummyClassifier	0.1440	0.0556	0.0363	0.0438
Our LDA	0.1634	0.2126	0.1758	

UPWINS Dataset - Confusion Matrices

80/20 Split



60/40 Split



Conclusions

General Trends in performance metrics for hsi classification

- From the lazy classifier results that ran, generally the tree-based methods performed best but it was variable across datasets

- This shows how hard it is to create high performing generalized models for hsi classification

Compare results (specifically evaluation metrics) to Future Prospects paper. What methods does that paper describe that use PCA (whitening?) along with neural networks? Why might these be good, and why might they not be good?

- The paper focused on accuracy (average, overall, and kappa) for their metric - we used this metric as well as the balanced accuracy and f1-score to account for class imbalance
- They described their use of PCA as an unsupervised learning technique for spectral representation that becomes less useful when the spectral mixing effect is present - therefore they use spectral-spatial representation for improved classification accuracy (Section III and IV)
- Spectral-Spatial Representation (p.972)

What challenges does that paper describe with respect to acquiring labeled hyperspectral image data? How do they propose overcoming these challenges?

Challenges:

- It's time consuming and expensive to label data- requires human domain experts or investigation of real-time scenarios (p.971)
- High intraclass variability makes it difficult to label correctly (p.971)

Proposed Solutions:

- Data Augmentation
- Unsupervised and semi-supervised learning
- GANS
- Transfer Learning and Active Learning (Section X)
- Their results showed that GRU and MorphCNN can overcome limited availability of training samples to some extent (p. 989)

The conclusion states, "Although the current HSIC techniques reflect a rapid, remarkable, and sophistication of the task, further developments are stillrequired to improve the generalization capabilities." What do you think the authors mean by 'generalization capabilities'?

- Generalization is the ability of the model to perform well on unseen samples. Many times this means reducing overfitting.

- They might be referring to generalizing to other datasets as well which is a known issue.

- They also mention how many studies use all samples in their training/testing which skews results and presumably leads to low generalization capabilities.

How many times do the words 'normal', 'Gaussian', and 'probability distribution' each appear?

- normal: 1 (normal* : 3)
- Gaussian: 2
- probability distribution: 0