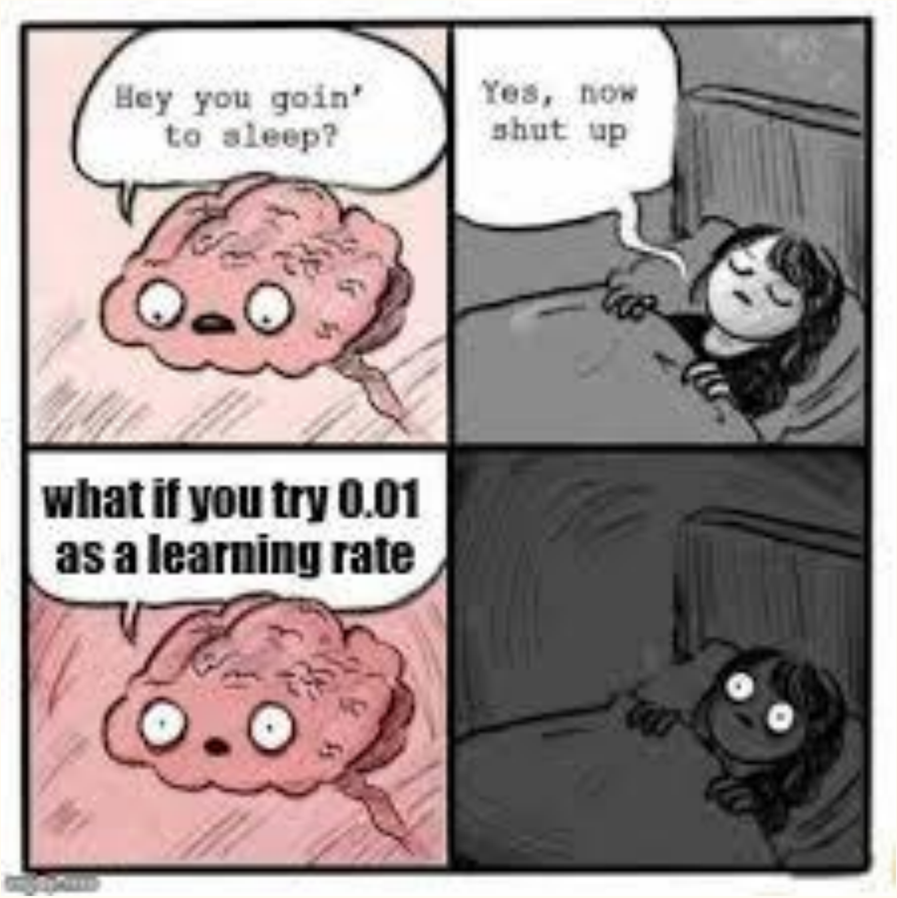
A Comparison of Land Classification Algorithms

CSC 497 - Interdisciplinary Project 2019

By Andrea Nesdoly
Supervisor: Dr. Neil Ernst

Outline

1. Goal
2. Background
3. Study Area
4. Data
5. Methods
6. Results
    a. Accuracy Assessment
    b. Statistical Comparison
7. Conclusion



Goal

To perform an in depth assessment between maximum likelihood estimation, support vector machine, and neural networks machine learning algorithms for land cover classification models.

Background

- Land development, natural resource management, or impact assessment studies
- Tangible product
- Satellites technology
- Advancements in machine learning
- PCI Geomatica, Jupyter Notebook, Python

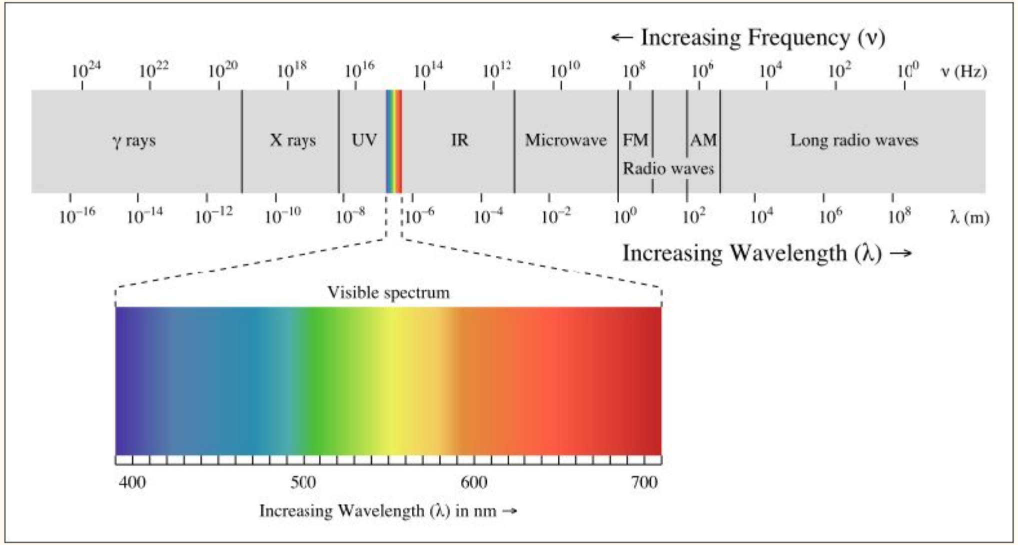


Figure 2: Electromagnetic Spectrum

Study Area

Area 2,890 km^2
28,896,685 pixels
11 bands

Land Cover Classes [1]

1. Urban/Barren
2. Forest
3. Shrub
4. Agriculture
5. Grassland
6. Wetland
7. Water

Data

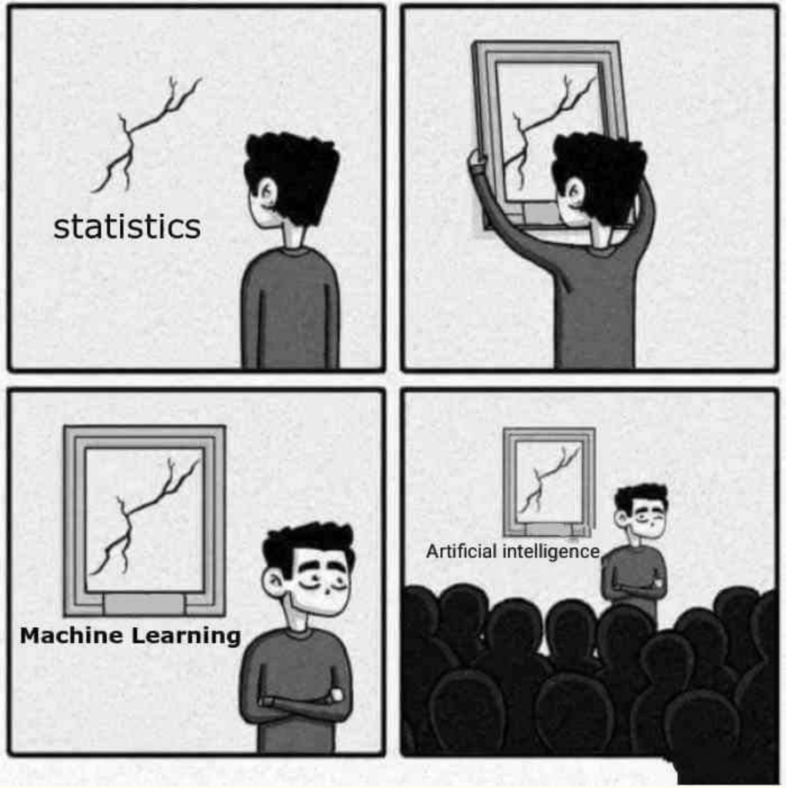
Sentinel 2A

- European Space Agency
- Level 1C product
- 13 spectral bands
- 10, 20, and 60 metre spatial resolution [2]

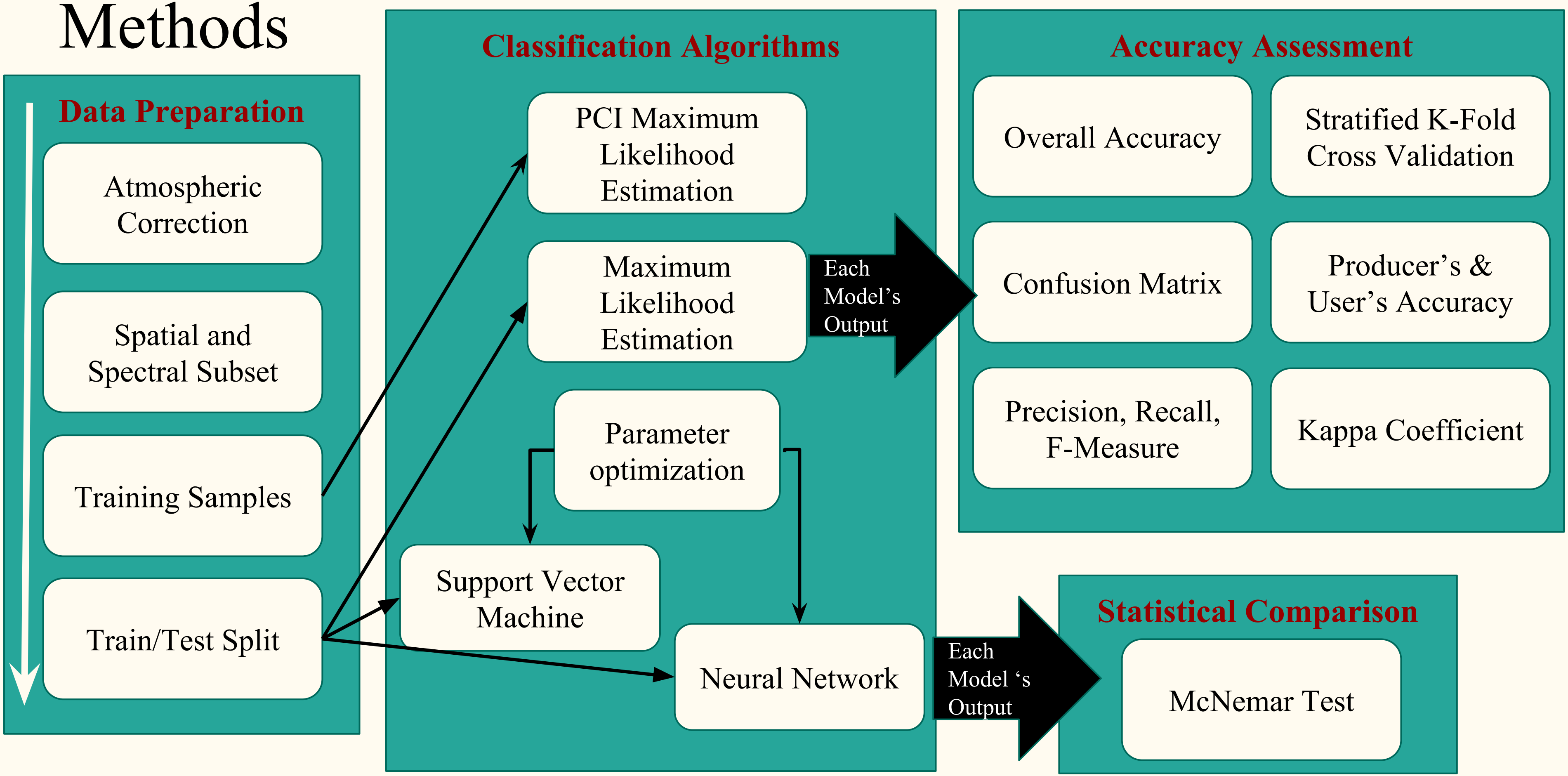
Band Number	Type	Central Wavelength (nm)	Bandwidth (nm)	Resolution (m)
1	Violet	442.7	27	60
2	Blue	492.4	98	10
3	Green	599.8	45	10
4	Red	664.6	38	10
5	VNIR	704.5	19	20
6	VNIR	740.5	18	20
7	VNIR	782.8	28	20
8	NIR	832.8	145	10
8a	VNIR	864.7	33	20
9	MIR	945.1	26	60
10	MIR	1373.5	75	60
11	SWIR	1613.7	143	20
12	SWIR	2202.4	242	20

Methods

1. Data Preprocessing
2. Training Sample Selection
3. Data Processing
4. Land Classification
    a. PCI Maximum Likelihood Estimation
    b. Maximum Likelihood Estimation
    c. Support Vector Machine
    d. Neural Network
5. Algorithm Accuracy Assessment
6. Statistical Comparison Between Algorithms



Methods


Data Preparation
Atmospheric Correction
Spatial and Spectral Subset
Training Samples
Train/Test Split
Classification Algorithms
PCI Maximum Likelihood Estimation
Maximum Likelihood Estimation
Parameter optimization
Support Vector Machine
Neural Network
Each Model's Output
Accuracy Assessment
Overall Accuracy
Stratified K-Fold Cross Validation
Confusion Matrix
Producer's & User's Accuracy
Precision, Recall, F-Measure
Kappa Coefficient
Each Model 's Output
Statistical Comparison
McNemar Test

Data Processing

1. Preprocessing

- ❑ Atmospheric Correction
- ❑ Spatial and Spectral Subset

2. Training Sample Selection

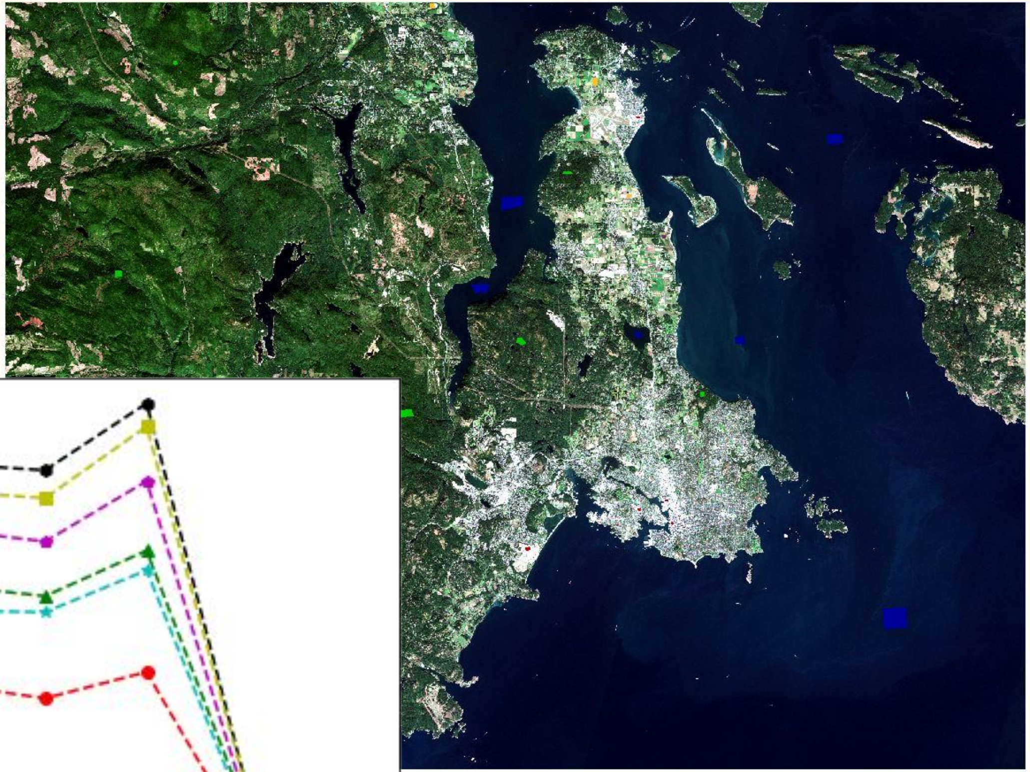
- ❑ Homogenous sites
- ❑ Spectral Signature Separability

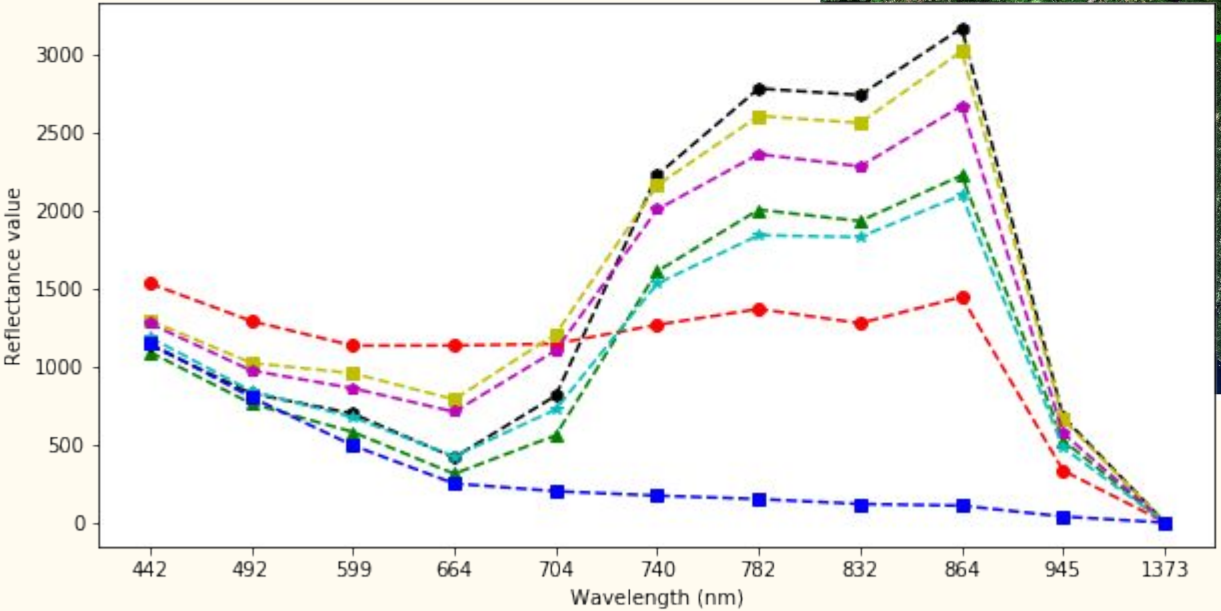
3. Processing

- ❑ Train/Test Split of 70%/30%

Training Samples

- Bhattacharyya Distance→1.86
- Transverse Divergence→1.95





PCI Maximum Likelihood Estimation

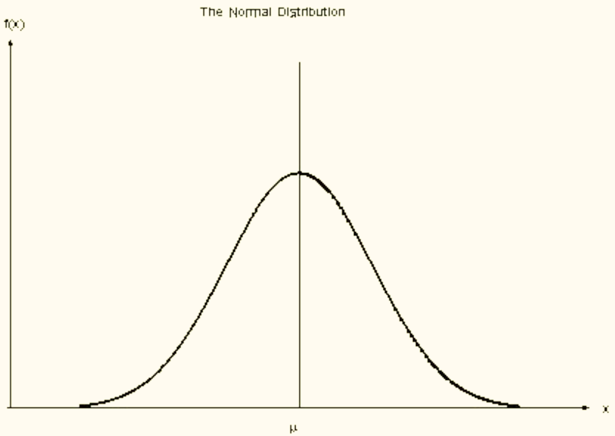
Black box

PCI Maximum Likelihood Estimation

1. Determine if the pixel is within the hyperellipsoid for the class
2. If not assign a NULL class, otherwise calculate the *Mahalanobis minimum distance classifier*
3. Calculate the "a posteriori" with Bayes Rule
    a. Assumes Gaussian distribution for all feature classes [3]

Maximum Likelihood Estimation

- ❑ Very common in remote sensing
- ❑ Class Mean and Covariance Matrix
- ❑ Gaussian Probability Distribution
  - ❑ Assumed Independent and Identical [4,5]



$$P(x_i|y) = \frac{1}{\sqrt{2\pi C_y^2}} \exp\left(-\frac{(x_i - M_y)^2}{2C_y^2}\right)$$

Support Vector Machines

- **Kernels:**
  - Radial Basis Function → attempted
  - Linear
  - Polynomial → 3rd order

- Cost = 1.0
- Gamma = 1/Number of Features

```
[[ 0.01        0.01        0.99083487]
 [ 0.01        0.1         0.99083487]
 [ 0.01        0.5         0.99083487]
 [ 0.01        1.          0.99083487]
 [ 0.01        1.5         0.99083487]
 [ 0.01        5.          0.99083487]
 [ 0.01       10.          0.99083487]
 [ 0.1         0.01        0.99157229]
 [ 0.1         0.1         0.99157229]
 [ 0.1         0.5         0.99157229]
 [ 0.1         1.          0.99157229]
 [ 0.1         1.5         0.99157229]
 [ 0.1         5.          0.99157229]
 [ 0.1        10.          0.99157229]
 [ 0.3         0.01        0.99141427]
 [ 0.3         0.1         0.99141427]
 [ 0.3         0.5         0.99141427]
 [ 0.3         1.          0.99141427]
 [ 0.3         1.5         0.99141427]
 [ 0.3         5.          0.99141427]
 [ 0.3        10.          0.99141427]
 [ 0.8         0.01        0.99094022]
 [ 0.8         0.1         0.99094022]
 [ 0.8         0.5         0.99094022]
 [ 0.8         1.          0.99094022]
 [ 0.8         1.5         0.99094022]
 [ 0.8         5.          0.99094022]
 [ 0.8        10.          0.99094022]
 [ 1.          0.01        0.99130893]
```

Neural Network: Multi-Layer Perceptron

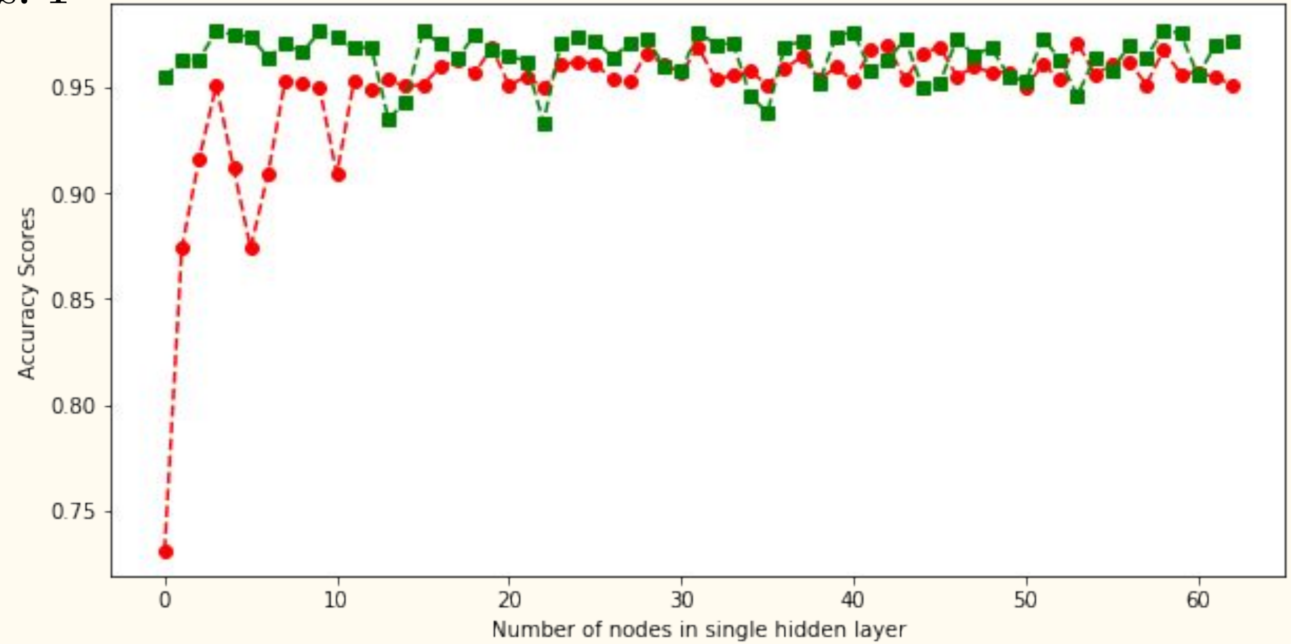
Number of hidden Layers: 1

Activation Functions:
Tahn → 55 nodes
Identity → 30 nodes

Adam Solver [6]

Alpha = 0.0001

Learning Rate = 0.001



Algorithm Accuracy Assessment

- Overall Accuracy
- Confusion Matrix
- Stratified K-fold Cross Validation
  - Average Accuracy
  - Average F-measure
- Precision, Recall, and F-measure [7]
- Producer's and User's Accuracy
  - Probability reference pixel is correctly labelled
  - Probability a pixel's classification represents ground truth [8]
- Kappa Coefficient
  - Different from random [9]

$$k = \frac{p_o - p_c}{1 - p_c} = \frac{1 - p_o}{1 - p_c}$$

Statistical Assessment Between Algorithm

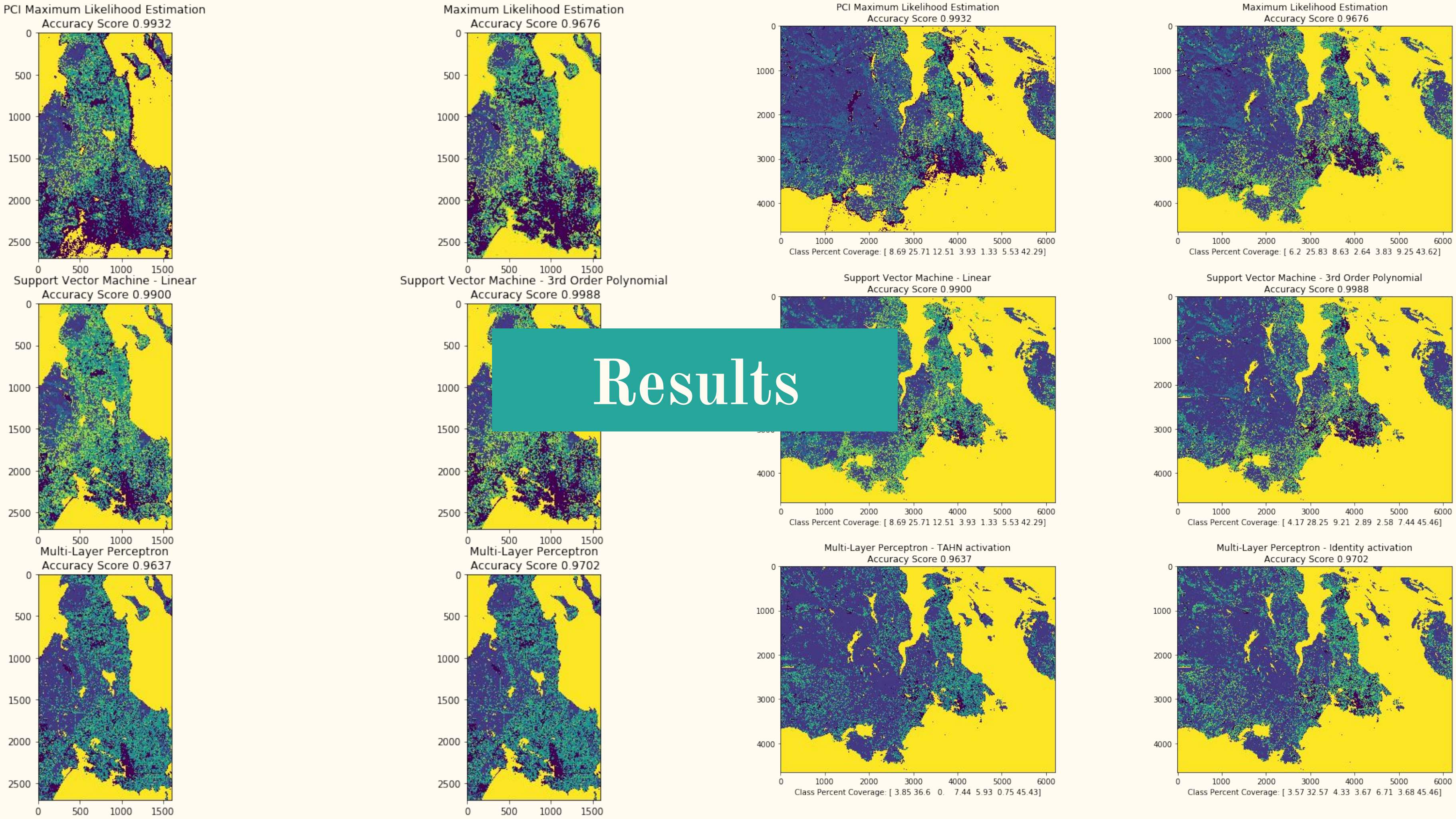
McNemar Test

- Statistically Significant Difference of Errors
- Binary metrics → Contingency Table
- Chi-square Statistic
- p-value [10]

Contingency Table

	MLA 2	**MLA 2**
MLA 1	True/True	True/False
MLA 1	False/True	False/False

Results

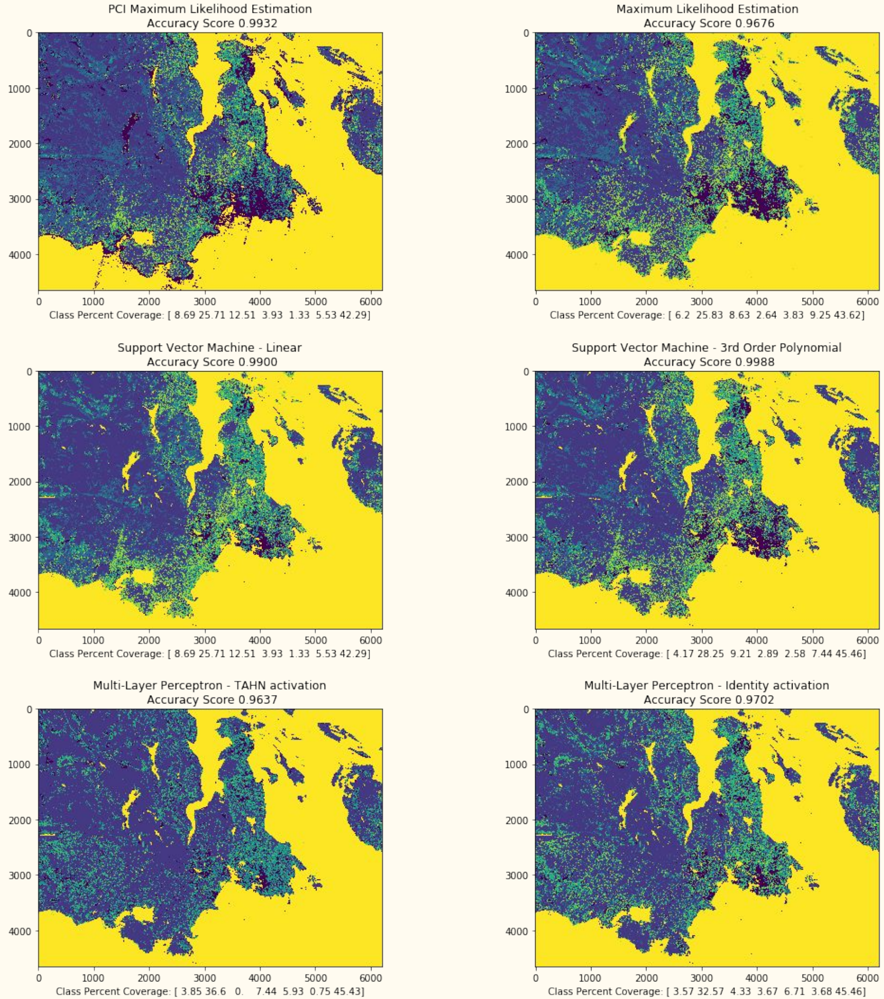

PCI Maximum Likelihood Estimation
Accuracy Score 0.9932
Maximum Likelihood Estimation
Accuracy Score 0.9676
Support Vector Machine - Linear
Accuracy Score 0.9900
Support Vector Machine - 3rd Order Polynomial
Accuracy Score 0.9988
Multi-Layer Perceptron
Accuracy Score 0.9637
Multi-Layer Perceptron
Accuracy Score 0.9702
PCI Maximum Likelihood Estimation
Accuracy Score 0.9932
Class Percent Coverage: [8.69 25.71 12.51 3.93 1.33 5.53 42.29]
Maximum Likelihood Estimation
Accuracy Score 0.9676
Class Percent Coverage: [6.2 25.83 8.63 2.64 3.83 9.25 43.62]
Support Vector Machine - Linear
Accuracy Score 0.9900
Class Percent Coverage: [8.69 25.71 12.51 3.93 1.33 5.53 42.29]
Support Vector Machine - 3rd Order Polynomial
Accuracy Score 0.9988
Class Percent Coverage: [4.17 28.25 9.21 2.89 2.58 7.44 45.46]
Multi-Layer Perceptron - TAHN activation
Accuracy Score 0.9637
Class Percent Coverage: [3.85 36.6 0. 7.44 5.93 0.75 45.43]
Multi-Layer Perceptron - Identity activation
Accuracy Score 0.9702
Class Percent Coverage: [3.57 32.57 4.33 3.67 6.71 3.68 45.46]

Overall Accuracy

- ❑ Visual inspection
- ❑ > 96% overall accuracy score

Ranking:

1. SVM polynomial kernel
2. PCI MLE
3. SVM linear kernel
4. NN identity activation function
5. MLE
6. NN tahn activation function



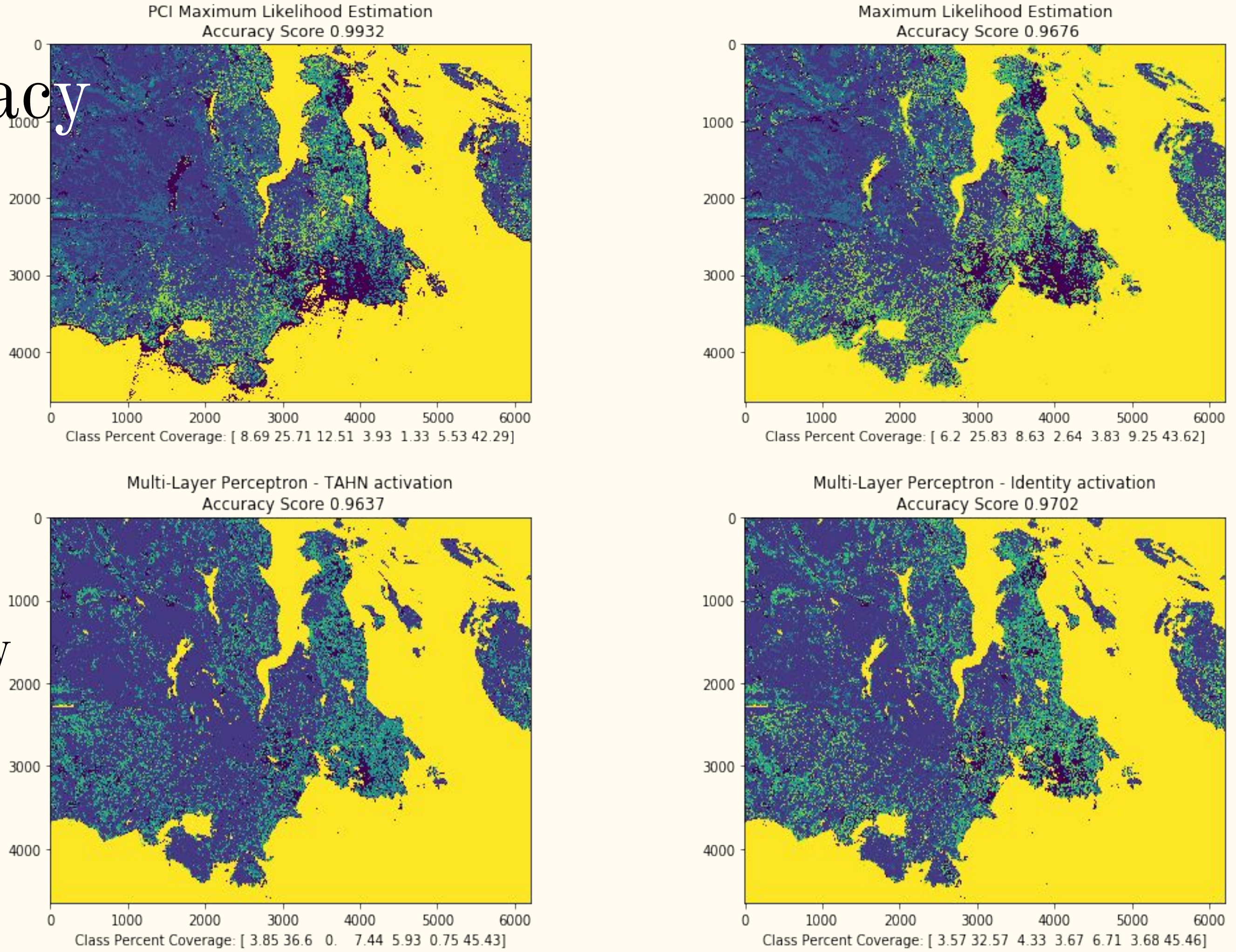
Overall Accuracy

PCI MLE

- ❑ Errors → water

NN

- ❑ Errors → forest
- ❑ Lacking Complexity



Overall Accuracy

MLE

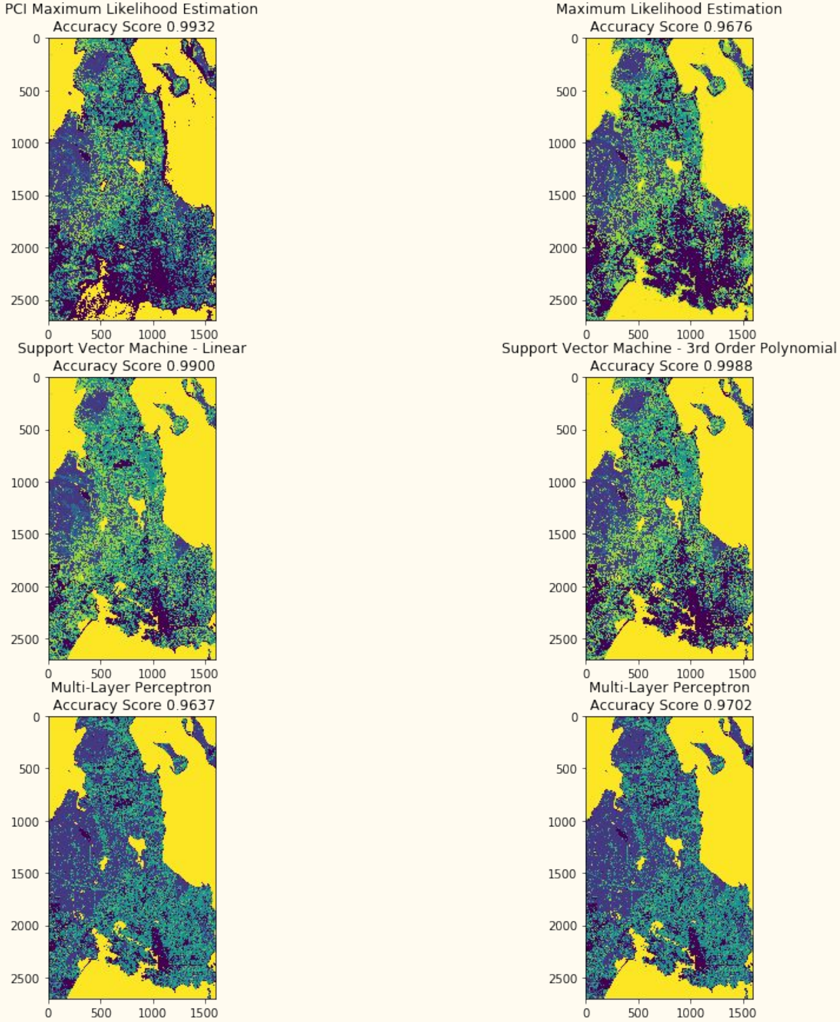
- ❑ Errors → water

SVM

- ❑ Complex

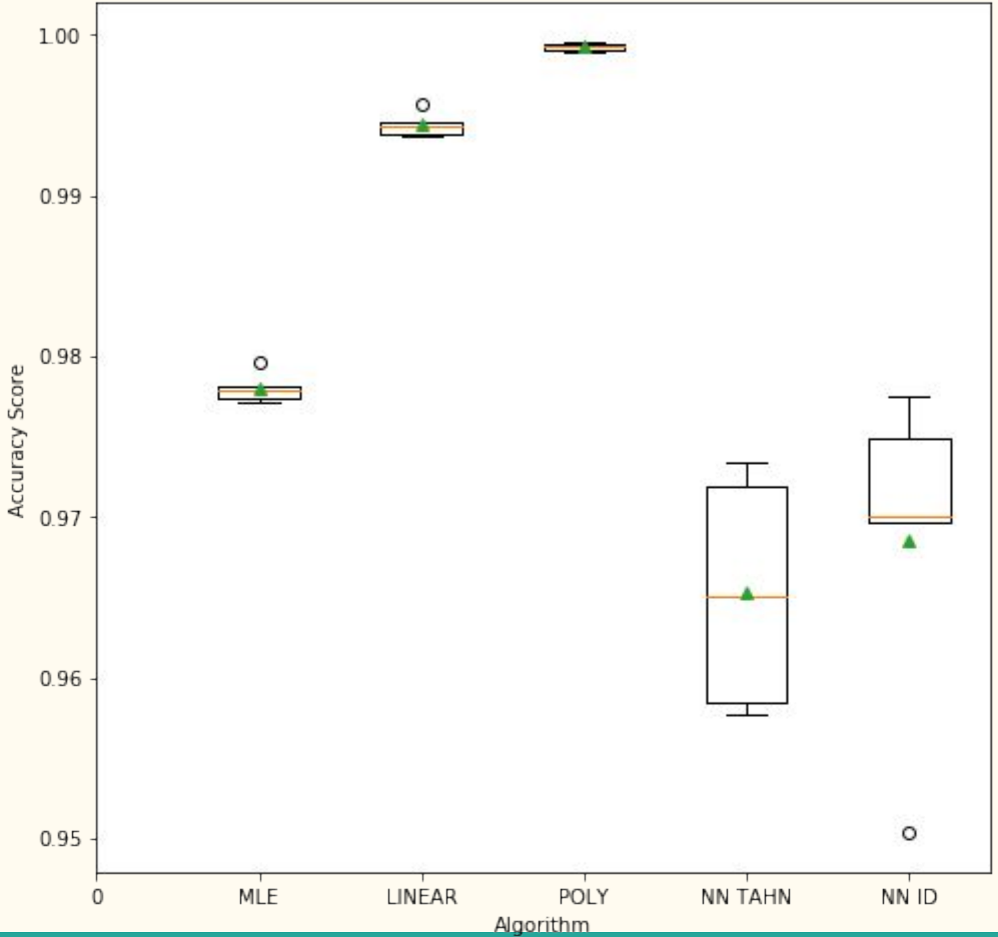
NN

- ❑ Errors → forest
- ❑ Lacking Complexity



Stratified K-fold Cross Validation

MLA	Overall Accuracy	Mean Accuracy
PCI MLE	99.32%	97.38%
MLE	96.67%	97.8%
SVM Linear	99.13%	99.44%
SVM Polynomial	99.84%	99.90%
NN Tahn	96.44%	96.06%
NN Identity	97.16%	97.38%



Confusion Matrices

```
Support Vector Machine - Linear
[[  801     0     0    28     0     0     0]
 [     0  2697     8     0     0     0     0]
 [     0     5   179     1     1     0     0]
 [    44     0     4   836    23     0     0]
 [     1     0     3    37   341     0     0]
 [     0     0    10     0     0    74     0]
 [     0     0     0     0     0     0 13892]]
```

```
PCI Maximum Likelihood Estimation
 40 Random Samples
[[ 3  0  0  1  0  0  1]
 [ 0  9  1  0  0  0  0]
 [ 0  3  2  0  0  0  0]
 [ 1  0  0  1  1  0  0]
 [ 0  0  0  0  0  0  0]
 [ 0  1  1  0  0  0  0]
 [ 0  0  0  0  0  0 15]]
```

```
Support Vector Machine - 3rd Order Polynomial
[[  824     0     0     0     5     0     0]
 [     0  2700     5     0     0     0     0]
 [     0     4   178     0     1     3     0]
 [     1     0     0   904     2     0     0]
 [     0     0     4     2   376     0     0]
 [     0     0     4     0     0    80     0]
 [     0     0     0     0     0     0 13892]]
```

```
Neural Network - Tahn Activation
[[  747     0     0    80     1     1     0]
 [     0  2506    55    47    97     0     0]
 [     0    19    62    57    48     0     0]
 [     6     0    25   743   133     0     0]
 [     5     0     0    17   360     0     0]
 [    16    18     0     9    41     0     0]
 [     0     0     0     0     0     0 13892]]
```

```
Maximum Likelihood Estimation
[[  690     0     0   114     0    25     0]
 [     0  2554   140     0     0    11     0]
 [     0     0   160     1     5    20     0]
 [   187     0     0   687    33     0     0]
 [     9     0    27    30   316     0     0]
 [     0     7    22     0     2    53     0]
 [     0     0     0     0     0     0 13892]]
```

```
Neural Network - Identity Acitvation
[[  742     1     1    56    15    14     0]
 [     0  2655     2     0     0    48     0]
 [     0    67    66     8     1    44     0]
 [   105     0     3   758    28    13     0]
 [     0     0     3    97   270    12     0]
 [     0    14     3     1     4    62     0]
 [     0     0     0     0     0     0 13892]]
```

Kappa Coefficient - Producer’s & User’s Accuracy

Kappa Coefficient

PCI MLE → 98.47
MLE → 92.43
SVM Linear → 98.02
SVM Polynomial → 99.63
NN Tahn → 91.93
NN Identity → 93.53

Producer’s & User’s Accuracy

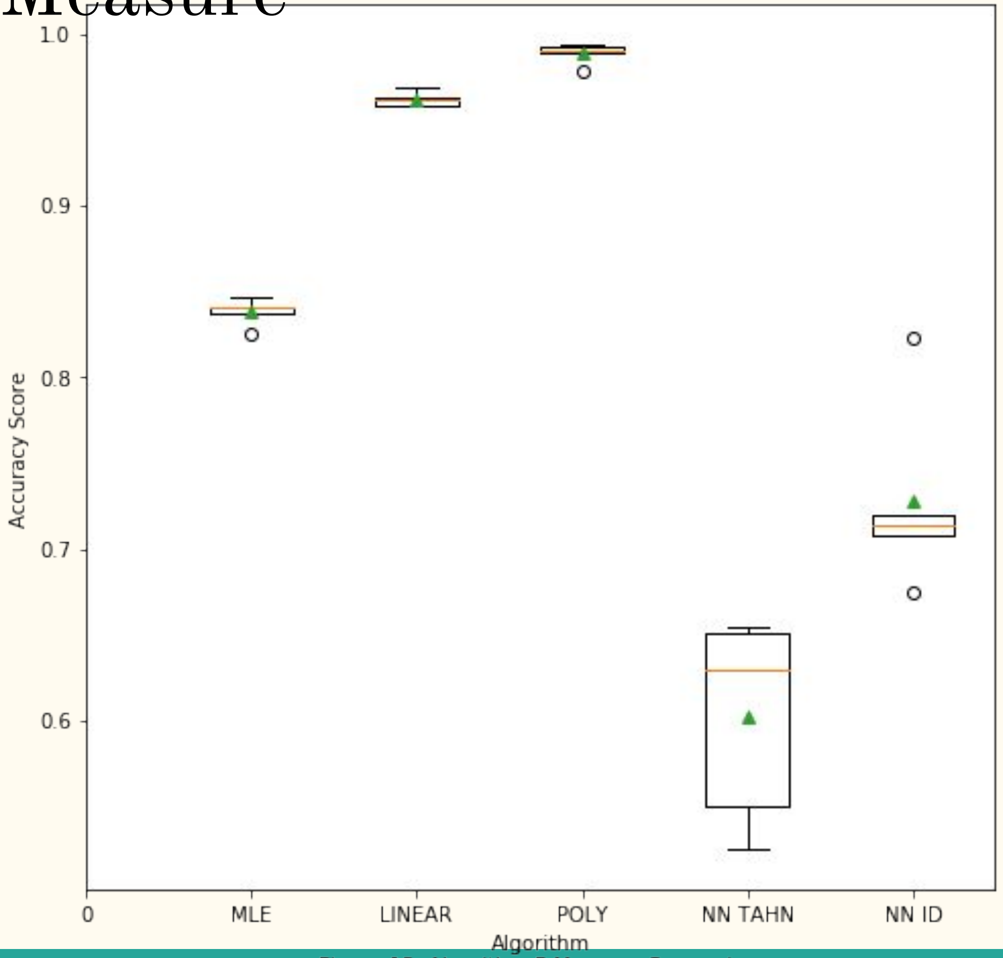
- ❑ Water → 100%
- ❑ Shrub & Wetland → lowest
- ❑ SVM Polynomial best overall
- ❑ NN Tahn worst overall

```
Support Vector Machine - Polynomial
Producer's Accuracy is:
[ 99.8788  99.8521  93.1937  99.7792  97.9167  96.3855 100.    ]
 User's Accuracy is:
[ 99.3969  99.8152  95.6989  99.6692  98.4293  95.2381 100.    ]

Neural Network - Tahn Activation
Producer's Accuracy is:
[ 96.5116  98.545   43.662   77.9643  52.9412   0.     100.    ]
 User's Accuracy is:
[ 90.1086  92.6433  33.3333  81.9184  94.2408   0.     100.    ]
```

Precision, Recall, & F-Measure

MLA	Overall Accuracy	Mean Accuracy	F-Measure
PCI MLE	99.32%	97.38%	N/A
MLE	96.67%	97.8%	83.79%
SVM Linear	99.13%	99.44%	96.18%
SVM Polynomial	99.84%	99.90%	98.86%
NN Tahn	96.44%	96.06%	57.58%
NN Identity	97.16%	97.38%	76.96%



McNemar Test

- Statistically Significantly Different
  - All p-values < 0.001
- Pixels classified differently for each MLA
  - Largest Percent Difference → 26.6%
  - Smallest Percent Difference → 17.9%

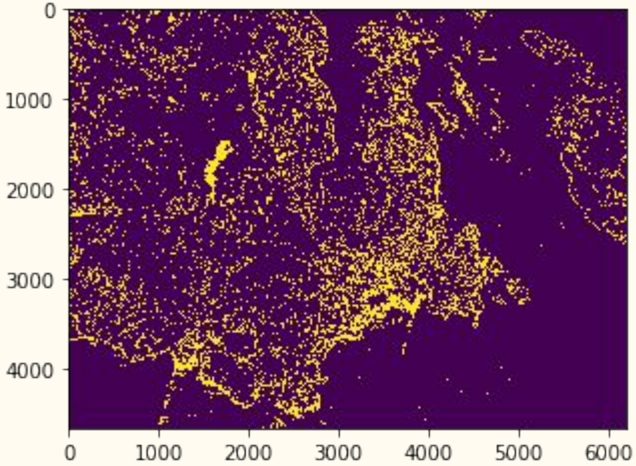


Figure 14: Comparison of PCI and SVM

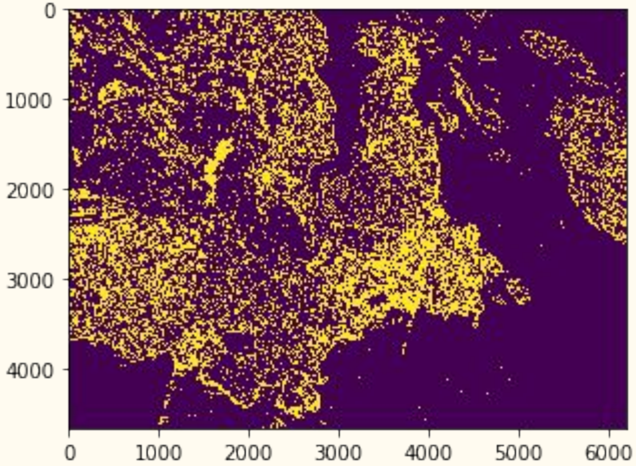


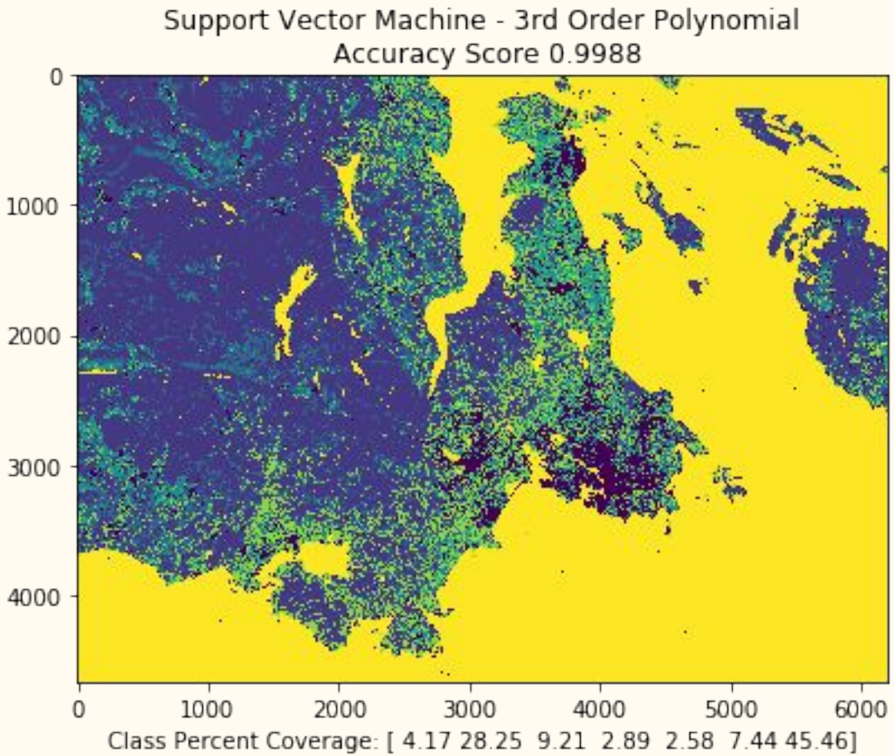
Figure 15: Comparison of PCI and NN

Conclusion

1st: SVM 3rd Order Polynomial was the best across all statistics → *Slow Speed*

2nd: MLE performed well overall → *Medium Speed*

3rd: NN was the worst across all statistics → *Fast Speed*



Conclusion

- Limitations:
  - Uneven sample sizes for each feature class
  - Difficulty finding optimal parameters for SVM and hyperparameters for Neural Network
  - Computing power/time limited for RBF kernel
- Future Studies:
  - More even distribution of pixels between feature classes
  - Parameters/hyperparameters taylored to this study area
  - Exploration of different spectral bands and higher levels products (eg. NDVI) []

Acknowledgements

Thank you to Dr. Neil Ernst for supervising this project, and to David Johnson for his advice during this process.

Special thanks to Dr. Randy Scharien and Terry Evans for giving me a broad understanding of remote sensing throughout my degree; and to Dr. George Tzanetakis and Dr. Alex Thomo for providing a solid base to build my machine learning abilities on.

References

[1] Frisk, J. 2011. Guidance for the preparation of ESTR products – land classification scheme. Canadian Biodiversity: Ecosystem Status and Trends 2010, Technical Thematic Report No. 3. Canadian Councils of Resource Ministers. Ottawa, ON. iv + 34 p. http://www.biodivcanada.ca/default.asp?lang=En&n=137E1147-1

[2] ESA (2019) Sentinel Online User Guides: https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric

[3] PCI Geomatic's (2018) Maximum Likelihood Classifier: http://www.pcigeomatics.com/geomatica-help/references/pciFunction_r/modeler/M_mlc.html

[4] Pedregosa et al., (2011) Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830.

[5] Tzanetakis, G. (2017) Course Slides from SENG 474 Introduction to Machine Learning. University of Victoria.

[6] Kingma, D. & Ba, J. (2015) Adam: A Method for Stochastic Optimization. ICLR conference paper

[7] Di Martino, M., Hernadez, G., Fiori, M., & Fernandex, A.(2013) New framework for optimal classifier design. Pattern Recognition, 46, 2249-2255; doi:https://doi.org/10.1016/j.patcog.2013.01.006

[8] Comber, A.J. (2013) Geographically weighted methods for estimating local surfaces of overall, user and producer accuracies. Remote Sensing Letters, 4:4, 373-380; doi:10.1080/2150704X.2012.736694

[9] Ganbold, G. & Chasia, S. (2017) Comparison between Possibilistic c_means (PCM) and Artificial Neural Network (ANN) Classification Algorithms in Land use/Land cover Classification. International Journal of Knowledge Content Development & Technology, Vol. 7, No.1, 57-78; doi:https://doi.org/10.5865/IJKCT.2017.7.1.057

[10] Li, X., Chen, W., Cheng, X., & Wang, L. (2016) A Comparison of Machine Learning Algorithms for Mapping of Complex Surface-Minded and Agricultural Landscapes Using ZiYuan-3 Stero Satellite Imagery. Remote Sensing, 8, 514; doi:10.3390/rs8060514