Saul Lynn
Ryan Maxwell
CS 5402 HW7 Final Report

Pokemon Dataset Analysis

Cleaning

This report is to analyze a dataset of Pokemon which includes a variety of information including their game statistics as well as traits more unique to the individual pokemon such as personality, likes, and dislikes. In our analysis, we hope to uncover interesting and notable correlations between attributes as well as determine which classification method predicts the particular type of the data set most accurately. Our initial cleaning was completed in python, while our Algorithmic analysis was completed using Weka.

In order to clean the dataset, we first identified attributes we determined were not worth keeping when performing our analysis. The attributes 'ID' and 'Name' were both dropped since these attributes are simple identifiers and are unique to each row, giving us no extra information. We chose to drop the attribute 'EyeColor' due to the fact that the entire attribute set was missing its data and could not be accurately replaced. Finally, we chose to drop the attribute 'Total' which is an attribute based on six other attributes: HP, Attack, Defense, SpAtk (Special Attack), SpDef (Special Defense), and Speed (of which we will refer to as **core attributes** from here on). It is worth noting that the Total attribute was not removed until missing values in the core attributes were replaced.

Our next step was to process noise. Rows with mostly missing values were removed because replacing the values would have a higher potential to skew data than simply removing it. Rows with a missing value in any of the core attributes had the missing value replaced by subtracting the existing values in the other attributes from Total. Missing values in PersonalityTrait, Likes, or DisLikes were replaced with the most common value for each column, but Likes and DisLikes were filtered to ensure the values were different between columns in the same row.

In this dataset, two decision attributes were present: Type1 and Type2. Pokemon have the potential of having two different types, so in such cases the decision attribute Type2 was populated. We chose to reduce the decision attributes to a single attribute: Type. Any instance in our dataset that had a value for Type2 had their row duplicated and re-added to the dataset, with the duplicate row having their Type1 attribute replaced with their Type2 attribute. This increased our row count from 800 to approximately 1214 rows. The reason we went with this method is because we thought this would enhance our dataset by giving more potential data points to use to train. We also believed that handling the two attributes in this way would make the data easier to handle down the line. An alternate method considered was to put both types in a singular string, essentially creating new possible decision values. The downside to this approach is that it is now impossible to predict pokemon with two different types.

Once we finished cleaning our data, we began binning by using Equal Frequency Binning using 30 bins. We decided to bin our dataset to reduce the number of unique values for our core attributes. We chose to use Equal Frequency since we believed our data was diverse enough to warrant having bins with approximately the same number of instances. Before we did so, however, we made sure to normalize our data so that distortion due to the range of our values was minimized during the binning process. Using this binning process, we were also able to detect any notable outliers in our dataset by using the graphs generated by our code. We chose 30 bins to ensure that by using Equal Frequency that each bin was guaranteed to have some values in them, considering the dataset is relatively large with 1200 data points.

In our principal components analysis of the dataset, we found certain correlations in our dataset. PersonalityTrait and Likes correlated positively, while both correlated negatively with DisLikes. We found that Generation and Bonus both did not correlate with or against any other attributes, indicating that these two attributes may have no significance on the data set. The Legendary Attribute correlated positively with HP and SpAtk which may indicate that legendary pokemon have higher values for these particular attributes. We also see that Speed and Attack correlate positively which may indicate that the higher speed a pokemon has, the more attack power it possesses. For the variance in our PCA calculation, we saw that the first 4 attributes accounted for approximately 60% of the variance in the dataset.

Method Analysis

In our analysis, we have applied select data mining algorithms on the dataset in Weka in order to determine which classification model produced the best results for predicting the class of the dataset. Our consideration for the best model for the dataset primarily focuses on the accuracy of the model.

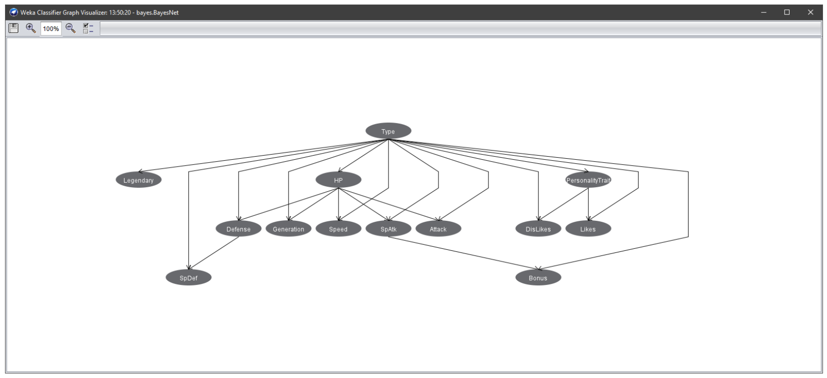
We have found that there are three methods among the data mining algorithms that performed nearly identically in terms of accuracy, the SVM using the Puk kernel, KD-Tree algorithm with accuracies being 65.897% each. However the difference in performance of these algorithms is their relative absolute error, as well as the root relative squared error. The SVM had a 95% error rate for both of these two statistics, however the KD-Tree algorithm had a 37% relative absolute error and 60% error rate for the root relative squared error. The ROC of the KD-Tree algorithm was approximately 0.981 as a weighted average indicating that the test was particularly useful. PRISM had a 36.1% relative absolute error and a ROC weighted average of 0.817. If one had to choose between using KD-Tree and SVM with kernel = Puk for classifying instances in this dataset, KD-Tree would be the most apt decision to go with, with PRISM and SVM being close runner ups.

Another method of note that approaches the accuracy rate of the KD-Tree is the K2 Bayesian Network, with max parents being 2. This method approached 64.827% accuracy with a ROC area weighted average of 0.981. With the relative absolute error being only 40% and the root relative squared error being 68.34%, this choice of method for classifying this dataset is also an incredibly respectable choice. Following this method in accuracy was the SVM with the PolyKernel kernel. This method approached 63.34% accuracy but had a terrible relative absolute error of 95.15%.

In general, the rest of the methods tested (including CART, Naive Bayesian Network, Gradient Boosting Machine, and Stacking) did not approach the aforementioned methods in accuracy. The highest accuracy of this list was the Gradient Boosting Machine with Shrinkage set to 1.75, which produced an accuracy of 42.75%. The lowest accuracy produced was the Stacking method. No matter which classifier we used for Stacking, we only managed to obtain an accuracy of 10.71%. Because of this, these algorithms are more difficult to confidently say they predict the dataset well. DBSCAN's clustering results were not particularly enlightening, as most clusters contained less than 10% of the instances and left 186 instances as unclustered noise.

As for information gained from these algorithms, we can determine a few interesting facts about which types of pokemon have certain traits in their statistics. CART seems to indicate that normal type pokemon have weak statistics compared to every other type of pokemon, with bug type pokemon following close behind. Water and grass type pokemon tend to have weaker statistics except for their special attack attribute. Rock and Ghost type pokemon tend to have better Defense than most other pokemon and are only differentiated by which generation they are from. Psychic type pokemon tend to have very high special attack but incredibly weak regular attack. Fire and water type pokemon tend to have moderate attack power and moderate speed and are only differentiated by their Defense statistic. Psychic and electric type pokemon are also very similar with slightly lower HP than the average and only vary by generation. Finally, Dark, Flying, and Dragon type pokemon tend to have higher attack scores, with high speed, with Dark type pokemon having slightly less HP than Dragon and Flying pokemon. Dragon and Flying type Pokemon seem to be differentiated by their defense scores, with Dragon type having higher defense.

Bayesian Network, K2 Algorithm Tree



DOCUMENTATION
Gothrand/Final-repo (github.com) **Repository Link**
Code 1: cleaning.py
Purpose: This file cleans the dataset, producing a CSV titled ‘one_type.csv’.

```
import pandas as pd

FILENAME = "Pokemon.csv"

df = pd.read_csv(FILENAME)

'''
ID, Name: Every instance is different
TotaL:  Calculated from other attributes
EyeColor:  Most of the attribute is missing data, not worth keeping
'''
df = df.drop(columns=['ID', 'Name', 'Total', 'EyeColor'])

'''Make sure all types are lowercase'''
df['Type1'] = df['Type1'].apply(lambda x: "".join([i.lower() for i in x]))
df['Type2'] = df['Type2'].dropna().apply(lambda x: "".join([i.lower() for
i in x]))

'''Any row that does not have a value for their base type is considered
noise'''
df = df[df.Type1 != '?']

'''Change Legendary attribute to use a system where 1 and -1 are treated
as True and False, respectively'''
df['Legendary'] = df['Legendary'].apply(lambda x: 'FALSE' if x == '?' else
x.upper())
df['Legendary'] = df['Legendary'].apply(lambda x: 'TRUE' if x == 'T' else
x)
df['Legendary'] = df['Legendary'].apply(lambda x: 'FALSE' if x == 'F' else
x)
df['Legendary'] = df['Legendary'].apply(lambda x: 1 if x == 'TRUE' else
-1)

'''
Any pokemon with  a missing value for Generation will use the previous
row's generation due to
the fact that the pokemon are already grouped by Generation.
```

```
'''
df.loc[df.Generation == '?', 'Generation'] = df['Generation'].shift(1)

'''Manually modify Sunkern's HP since by inspection he was the only one
with an HP error'''
df['HP']=df['HP'].apply(lambda x: 30 if x == '?' else x)

'''Manually modify Loudred's Defense since he is the only one with a
missing defense value'''
df['Defense']=df['Defense'].apply(lambda x: 43 if x == '?' else x)

'''Make all PersonalityTrait values lowercase and replace missing ones
with the most common PersonalityTrait'''
df['PersonalityTrait']=df['PersonalityTrait'].apply(lambda x: x.lower())
df['PersonalityTrait'] = df['PersonalityTrait'].apply(lambda x:
df['PersonalityTrait'].mode()[0] if x == '?' else x)

'''Make all Likes values lowercase and replace them with the most common
Likes value'''
df['Likes'] = df['Likes'].apply(lambda x: x.lower())
df['Likes'] = df['Likes'].apply(lambda x: df['Likes'].mode()[0] if x ==
'?' else x)

'''Fix spelling mistake for "spicy"'''
df.loc[df.Likes == 'spicey', 'Likes'] = 'spicy'
df.loc[df.DisLikes == 'spicey', 'DisLikes'] = 'spicy'

'''Manually fix Oddish's "Likes" value to be not the same as his DisLikes
value'''
for i, row in df.iterrows():
    if i == 48:
        row['Likes'] = 'sour'
        df.loc[i] = row

# Replace '?' values, ensure all lower case
df['DisLikes']=df['DisLikes'].apply(lambda x: x.lower())
df['DisLikes']=df['DisLikes'].apply(lambda x: df['DisLikes'].mode()[0] if
x == '?' else x)

'''Ensure all values in this attribute are floats'''
```

```
df['Bonus']=df['Bonus'].apply(lambda x: float(x))

'''Export cleaned csv for further manipulation'''
df.to_csv('cleaned2.csv', index=False)

###

'''
The below code fixes the issue of having two decision attributes:
We are choosing to duplicate each row that has a value filled in for Type2
and making the duplicate's Type1 value its Type2 value, and then finally
dropping the Type2 column altogether.  This increased our total row count
from about 800 rows to approximately 1200.
'''
newDF = pd.DataFrame()
for i, row in df[~df.Type2.isnull()].iterrows():
    row['Type1'] = row['Type2']
    newDF = newDF.append(row)

# concatenate old DF with the new one
df2 = pd.concat([df, newDF], ignore_index=True)

# drop type 2 since we don't need it now
df2 = df2.drop(columns=['Type2'])
df2.to_csv('one_type.csv', index=False)
```

Code 2: PrincipalComponents.py
Purpose: This file is used to calculate PCA for our dataset.

```
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

df = pd.read_csv('disc.csv')
del df['Type']

attrList = list(df.columns)
df[attrList] = StandardScaler().fit_transform(df[attrList])

#Compute PCA
```

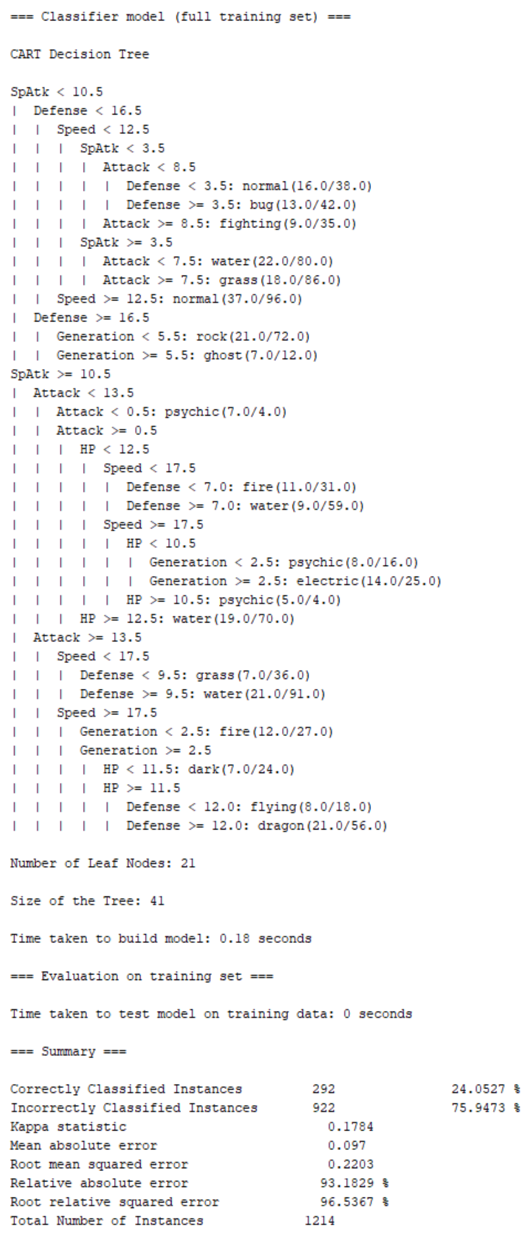
```
n_components = len(attrList)
pca = PCA(n_components=n_components)
reduced = pca.fit_transform(df[attrList])

#Get eigenvalues and sum
eigenvalues = pca.explained_variance_
eigenSum7 = 0
for i in range(7):
    eigenSum7 += eigenvalues[i]

#Determine variance percentages
print(pca.explained_variance_ratio_)

#Add the first 7 because those seven eigen vectors add up to about 0.8
proportionally
varsum = 0
for i in range(7):
    varsum += pca.explained_variance_ratio_[i]
varsum = varsum * 100
print(varsum)
```

Code 3: Code to generate disc.csv and nom.csv

```
import pandas as pd
from feature_engine.discretisers import EqualFrequencyDiscretiser

df = pd.read_csv('binned.csv')
df2 = pd.read_csv('one_type.csv')

english = {0: 'Zero', 1:'One', 2:'Two', 3:'Three', 4: 'Four', 5: 'Five',
6: 'Six', 7: 'Seven', 8: 'Eight', 9: 'Nine', 10: 'Ten', 11: 'Eleven', 12:
'Twelve', 13: 'Thirteen', 14: 'Fourteen', 15: 'Fifteen', 16: 'Sixteen',
17: 'Seventeen', 18: 'Eighteen', 19: 'Nineteen', 20:'Twenty', 21:
'Twenty-one', 22: 'Twenty-two', 23: 'Twenty-three', 24: 'Twenty-four', 25:
'Twenty-five', 26: 'Twenty-six', 27: 'Twenty-seven', 28: 'Twenty-eight',
29: 'Twenty-nine', 30: 'Thirty'}
english[25]

df['Type'] = df2['Type1']
```

```python
#Discretize
discretizerF = EqualFrequencyDiscretiser(q=30, variables = ['HP'])
df = discretizerF.fit_transform(df)
discretizerF = EqualFrequencyDiscretiser(q=30, variables = ['Attack'])
df = discretizerF.fit_transform(df)
discretizerF = EqualFrequencyDiscretiser(q=30, variables = ['Defense'])
df = discretizerF.fit_transform(df)
discretizerF = EqualFrequencyDiscretiser(q=30, variables = ['SpAtk'])
df = discretizerF.fit_transform(df)
discretizerF = EqualFrequencyDiscretiser(q=30, variables = ['SpDef'])
df = discretizerF.fit_transform(df)
discretizerF = EqualFrequencyDiscretiser(q=30, variables = ['Speed'])
df = discretizerF.fit_transform(df)
discretizerF = EqualFrequencyDiscretiser(q=30, variables = ['Bonus'])
df = discretizerF.fit_transform(df)

df.to_csv('disc.csv', index=False)

df['Generation'] = df['Generation'].apply(lambda x: 'One' if x == 1 else
'Two' if x == 2 else 'Three' if x == 3 else 'Four' if x == 4 else 'Five'
if x == 5 else 'Six' if x == 6 else x)

df['Legendary'] = df['Legendary'].apply(lambda x: 'True' if x == 1 else
'False')

#Nominalize
df['HP'] = df['HP'].apply(lambda x: english[x])
df['Attack'] = df['Attack'].apply(lambda x: english[x])
df['Defense'] = df['Defense'].apply(lambda x: english[x])
df['SpAtk'] = df['SpAtk'].apply(lambda x: english[x])
df['SpDef'] = df['SpDef'].apply(lambda x: english[x])
df['Speed'] = df['Speed'].apply(lambda x: english[x])
df['Bonus'] = df['Bonus'].apply(lambda x: english[x])

df['PersonalityTrait'] = df2['PersonalityTrait']
df['Likes'] = df2['Likes']
df['DisLikes'] = df2['DisLikes']

df.to_csv('nom.csv', index=False)
```

CART decision tree + Summary from Weka



```
=== Classifier model (full training set) ===

CART Decision Tree

SpAtk < 10.5
|  Defense < 16.5
|  |  Speed < 12.5
|  |  |  SpAtk < 3.5
|  |  |  |  Attack < 8.5
|  |  |  |  |  Defense < 3.5: normal(16.0/38.0)
|  |  |  |  |  Defense >= 3.5: bug(13.0/42.0)
|  |  |  |  Attack >= 8.5: fighting(9.0/35.0)
|  |  |  SpAtk >= 3.5
|  |  |  |  Attack < 7.5: water(22.0/80.0)
|  |  |  |  Attack >= 7.5: grass(18.0/86.0)
|  |  Speed >= 12.5: normal(37.0/96.0)
|  Defense >= 16.5
|  |  Generation < 5.5: rock(21.0/72.0)
|  |  Generation >= 5.5: ghost(7.0/12.0)
SpAtk >= 10.5
|  Attack < 13.5
|  |  Attack < 0.5: psychic(7.0/4.0)
|  |  Attack >= 0.5
|  |  |  HP < 12.5
|  |  |  |  Speed < 17.5
|  |  |  |  |  Defense < 7.0: fire(11.0/31.0)
|  |  |  |  |  Defense >= 7.0: water(9.0/59.0)
|  |  |  |  Speed >= 17.5
|  |  |  |  |  HP < 10.5
|  |  |  |  |  |  Generation < 2.5: psychic(8.0/16.0)
|  |  |  |  |  |  Generation >= 2.5: electric(14.0/25.0)
|  |  |  |  |  HP >= 10.5: psychic(5.0/4.0)
|  |  |  HP >= 12.5: water(19.0/70.0)
|  Attack >= 13.5
|  |  Speed < 17.5
|  |  |  Defense < 9.5: grass(7.0/36.0)
|  |  |  Defense >= 9.5: water(21.0/91.0)
|  |  Speed >= 17.5
|  |  |  Generation < 2.5: fire(12.0/27.0)
|  |  |  Generation >= 2.5
|  |  |  |  HP < 11.5: dark(7.0/24.0)
|  |  |  |  HP >= 11.5
|  |  |  |  |  Defense < 12.0: flying(8.0/18.0)
|  |  |  |  |  Defense >= 12.0: dragon(21.0/56.0)

Number of Leaf Nodes: 21

Size of the Tree: 41

Time taken to build model: 0.18 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         292               24.0527 %
Incorrectly Classified Instances       922               75.9473 %
Kappa statistic                          0.1784
Mean absolute error                      0.097
Root mean squared error                  0.2203
Relative absolute error                 93.1829 %
Root relative squared error             96.5367 %
Total Number of Instances             1214
```

FULL METHOD SUMMARY

Weka, nom.csv -> weka.classifiers.trees.SimpleCart -M 2.0 -N 5 -C 1.0 -S 1

CART

=== Summary ===

```
Correctly Classified Instances         292               24.0527 %
Incorrectly Classified Instances       922               75.9473 %
Kappa statistic                          0.1784
Mean absolute error                      0.097
Root mean squared error                  0.2203
Relative absolute error                 93.1829 %
```

```
Root relative squared error          96.5367 %
Total Number of Instances            1214

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.263    0.109    0.170      0.263    0.207      0.127    0.713     0.147     grass
               0.359    0.050    0.284      0.359    0.317      0.277    0.820     0.204     fire
               0.563    0.276    0.191      0.563    0.286      0.191    0.699     0.183     water
               0.181    0.037    0.236      0.181    0.205      0.163    0.755     0.152     bug
               0.520    0.121    0.283      0.520    0.367      0.307    0.793     0.231     normal
               0.000    0.000    ?          0.000    ?          ?        0.705     0.099     poison
               0.280    0.021    0.359      0.280    0.315      0.291    0.814     0.195     electric
               0.000    0.000    ?          0.000    ?          ?        0.767     0.126     ground
               0.000    0.000    ?          0.000    ?          ?        0.761     0.077     fairy
               0.170    0.030    0.205      0.170    0.186      0.153    0.767     0.121     fighting
               0.222    0.021    0.455      0.222    0.299      0.282    0.783     0.265     psychic
               0.362    0.062    0.226      0.362    0.278      0.240    0.789     0.155     rock
               0.152    0.010    0.368      0.152    0.215      0.218    0.784     0.145     ghost
               0.000    0.000    ?          0.000    ?          ?        0.747     0.068     ice
               0.420    0.048    0.273      0.420    0.331      0.303    0.820     0.180     dragon
               0.137    0.021    0.226      0.137    0.171      0.148    0.764     0.118     dark
               0.000    0.000    ?          0.000    ?          ?        0.839     0.152     steel
               0.079    0.016    0.308      0.079    0.126      0.120    0.734     0.182     flying
Weighted Avg.    0.241    0.063    ?          0.241    ?          ?        0.763     0.166

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r   <-- classified as
 25  6 32  5  8  0  5  0  0  0  1  2  4  0  2  2  0  3 |  a = grass
  8 23 18  0  3  0  1  0  0  1  3  0  0  0  4  0  0  3 |  b = fire
 13  2 71  3 13  0  3  0  0  3  1  5  1  0  4  4  0  3 |  c = water
  7  3 12 13 18  0  1  0  0  2  2 12  0  0  1  1  0  0 |  d = bug
  9  1 24  3 53  0  1  0  0  4  1  2  0  0  4  0  0  0 |  e = normal
 11  5 19  6 13  0  1  0  0  1  4  2  0  0  0  0  0  0 |  f = poison
  1  5 16  0  7  0 14  0  0  0  2  0  0  0  1  0  0  4 |  g = electric
  9  0 16  4  9  0  0  0  0  7  0 14  0  0  8  0  0  0 |  h = ground
  2  1 22  3  2  0  2  0  0  1  2  1  1  0  1  2  0  0 |  i = fairy
 11  2  6  1 11  0  0  0  0  9  0  2  0  0  4  4  0  3 |  j = fighting
  3 10 30  8  4  0  2  0  0  0 20  1  0  0  6  5  0  1 |  k = psychic
  7  1 16  2  2  0  0  0  0  4  0 21  2  0  1  1  0  1 |  l = rock
 12  3 11  1  0  0  3  0  0  2  3  2  7  0  2  0  0  0 |  m = ghost
  3  0 20  0  4  0  3  0  0  1  2  0  1  0  4  0  0  0 |  n = ice
  7  2 12  0  2  0  0  0  0  4  0  1  0  0 21  1  0  0 |  o = dragon
 12  5 10  1  6  0  0  0  0  3  0  5  0  0  2  7  0  0 |  p = dark
  2  0 13  3  1  0  0  0  0  1  0 18  3  0  4  4  0  0 |  q = steel
  5 12 23  2 31  0  3  0  0  1  3  5  0  0  8  0  0  8 |  r = flying
```

```
Weka, nom.csv -> weka.classifiers.rules.Prism
PRISM
=== Summary ===

Correctly Classified Instances         800               65.8979 %
Incorrectly Classified Instances       414               34.1021 %
Kappa statistic                          0.6348
Mean absolute error                      0.0379
Root mean squared error                  0.1947
Relative absolute error                 36.3831 %
Root relative squared error             85.3079 %
Total Number of Instances             1214

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.055    0.605      1.000    0.754      0.756    0.972     0.605     grass
                 1.000    0.031    0.640      1.000    0.780      0.787    0.984     0.640     fire
                 0.968    0.058    0.659      0.968    0.785      0.773    0.955     0.642     water
                 0.875    0.040    0.578      0.875    0.696      0.690    0.917     0.513     bug
                 0.951    0.032    0.729      0.951    0.826      0.816    0.959     0.698     normal
                 0.484    0.013    0.667      0.484    0.561      0.549    0.735     0.349     poison
                 0.820    0.012    0.745      0.820    0.781      0.772    0.904     0.619     electric
                 0.687    0.029    0.582      0.687    0.630      0.609    0.829     0.417     ground
                 0.750    0.013    0.667      0.750    0.706      0.697    0.869     0.508     fairy
                 0.642    0.012    0.708      0.642    0.673      0.660    0.815     0.470     fighting
                 0.700    0.022    0.716      0.700    0.708      0.685    0.839     0.523     psychic
                 0.431    0.014    0.610      0.431    0.505      0.493    0.709     0.290     rock
                 0.478    0.010    0.647      0.478    0.550      0.542    0.734     0.329     ghost
                 0.526    0.006    0.741      0.526    0.615      0.614    0.760     0.405     ice
                 0.460    0.010    0.657      0.460    0.541      0.534    0.725     0.325     dragon
                 0.333    0.006    0.708      0.333    0.453      0.472    0.664     0.264     dark
                 0.122    0.001    0.857      0.122    0.214      0.316    0.561     0.140     steel
                 0.020    0.000    1.000      0.020    0.039      0.135    0.510     0.101     flying
Weighted Avg.    0.659    0.024    0.699      0.659    0.610      0.615    0.817     0.457

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q   r   <-- classified as
  95   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   a = grass
   0  64   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   b = fire
   3   1 122   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   c = water
   6   2   1  63   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   d = bug
   2   2   1   0  97   0   0   0   0   0   0   0   0   0   0   0   0   0 |   e = normal
  15   0   4  13   0  30   0   0   0   0   0   0   0   0   0   0   0   0 |   f = poison
   1   1   3   2   2   0  41   0   0   0   0   0   0   0   0   0   0   0 |   g = electric
   1   4  10   2   1   2   1  46   0   0   0   0   0   0   0   0   0   0 |   h = ground
   2   0   2   0   5   0   1   0  30   0   0   0   0   0   0   0   0   0 |   i = fairy
   3   7   3   2   2   2   0   0   0  34   0   0   0   0   0   0   0   0 |   j = fighting
   3   3   5   0   2   0   0   2   6   6 63   0   0   0   0   0   0   0 |   k = psychic
   2   1  10   5   0   0   0   9   3   1   2  25   0   0   0   0   0   0 |   l = rock
  10   3   2   1   0   4   1   2   0   0   1   0  22   0   0   0   0   0 |   m = ghost
   3   0   6   0   0   0   1   3   0   0   2   2   1  20   0   0   0   0 |   n = ice
   1   2   2   0   0   1   2   7   1   0   4   2   2   3  23   0   0   0 |   o = dragon
   3   3   6   0   0   3   0   3   0   3   3   2   3   2   3  17   0   0 |   p = dark
   2   1   1   7   0   0   3   3   3   3   7   6   4   0   1   2   6   0 |   q = steel
   5   6   7  14  24   3   5   4   2   1   6   4   2   2   8   5   1   2 |   r = flying
```

DBSCAN
Weka, disc.csv -> weka.clusterers.DBSCAN -E 0.9 -M 6 -A "weka.core.EuclideanDistance -R first-last"

```
=== Model and evaluation on training set ===

Clustered Instances

 0       91 (  9%)
 1      121 ( 12%)
 2       83 (  8%)
 3       65 (  6%)
 4       15 (  1%)
 5        6 (  1%)
 6       15 (  1%)
 7       49 (  5%)
 8       46 (  4%)
 9       39 (  4%)
10       41 (  4%)
11       67 (  7%)
12       18 (  2%)
13       29 (  3%)
14       16 (  2%)
15       37 (  4%)
16       39 (  4%)
17       53 (  5%)
18       26 (  3%)
19       55 (  5%)
20       96 (  9%)
21        9 (  1%)
22        6 (  1%)
23        6 (  1%)

Unclustered instances : 186
```

SVM
Weka, nom.csv -> weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"
(kernel = PolyKernel)

```
=== Summary ===

Correctly Classified Instances         769               63.3443 %
Incorrectly Classified Instances       445               36.6557 %
Kappa statistic                          0.6085
Mean absolute error                      0.0991
Root mean squared error                  0.2182
Relative absolute error                 95.1532 %
Root relative squared error             95.6113 %
Total Number of Instances             1214

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.653    0.031    0.639      0.653    0.646      0.615    0.967     0.584     grass
                 0.734    0.019    0.681      0.734    0.707      0.690    0.983     0.638     fire
                 0.690    0.047    0.630      0.690    0.659      0.618    0.955     0.597     water
                 0.625    0.025    0.608      0.625    0.616      0.592    0.975     0.570     bug
                 0.814    0.029    0.722      0.814    0.765      0.744    0.979     0.705     normal
                 0.613    0.023    0.585      0.613    0.598      0.576    0.977     0.544     poison
                 0.740    0.012    0.725      0.740    0.733      0.721    0.989     0.674     electric
                 0.552    0.021    0.607      0.552    0.578      0.555    0.975     0.555     ground
                 0.625    0.009    0.714      0.625    0.667      0.658    0.991     0.652     fairy
                 0.623    0.011    0.717      0.623    0.667      0.654    0.985     0.637     fighting
                 0.733    0.032    0.647      0.733    0.688      0.662    0.973     0.609     psychic
                 0.552    0.020    0.582      0.552    0.566      0.545    0.979     0.547     rock
                 0.457    0.010    0.636      0.457    0.532      0.524    0.985     0.575     ghost
                 0.526    0.007    0.714      0.526    0.606      0.603    0.989     0.620     ice
                 0.660    0.019    0.600      0.660    0.629      0.613    0.984     0.576     dragon
                 0.608    0.018    0.596      0.608    0.602      0.584    0.982     0.559     dark
                 0.633    0.022    0.544      0.633    0.585      0.568    0.981     0.524     steel
                 0.406    0.036    0.506      0.406    0.451      0.410    0.945     0.464     flying
Weighted Avg.    0.633    0.025    0.632      0.633    0.630      0.607    0.974     0.589

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r   <-- classified as
 62  0  2  2  2  9  1  0  0  3  4  0  4  1  0  1  2  2 |  a = grass
  0 47  3  0  0  0  1  3  0  2  3  1  1  0  0  0  0  3 |  b = fire
  1  1 87  0  3  4  3  4  1  0  5  3  1  2  2  5  1  3 |  c = water
  4  2  1 45  0  4  0  1  0  1  0  2  0  0  0  0  5  7 |  d = bug
  1  2  3  1 83  0  0  0  3  2  1  0  0  0  0  0  0  6 |  e = normal
  6  1  2  9  1 38  0  0  0  1  0  0  0  0  1  2  0  1 |  f = poison
  0  0  2  2  4  0 37  0  0  0  0  0  0  0  2  0  1  2 |  g = electric
  1  1  7  1  1  2  1 37  0  0  2  5  0  2  1  2  2  2 |  h = ground
  2  0  1  0  2  0  1  0 25  0  4  3  0  0  1  0  1  0 |  i = fairy
  1  5  3  1  0  1  0  0  0 33  3  0  0  0  0  3  2  1 |  j = fighting
  3  0  2  0  2  1  0  0  2  2 66  1  0  0  1  1  5  4 |  k = psychic
  2  0  8  3  0  0  0  3  0  1  1 32  0  1  0  0  4  3 |  l = rock
  6  2  3  1  0  4  1  2  0  0  1  0 21  1  2  0  1  1 |  m = ghost
  2  0  5  0  0  0  1  1  0  0  2  1  0 20  3  1  0  2 |  n = ice
  1  2  0  0  0  0  0  6  0  0  3  2  0  0 33  2  0  1 |  o = dragon
  2  3  2  0  0  1  0  1  0  0  2  2  2  1  1 31  1  2 |  p = dark
  0  1  0  2  0  0  2  1  2  1  2  2  3  0  1  1 31  0 |  q = steel
  3  2  7  7 17  1  3  2  2  0  3  1  1  0  7  3  1 41 |  r = flying
```

```
(kernel = Puk)
Weka, nom.csv -> weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K
"weka.classifiers.functions.supportVector.Puk -O 1.0 -S 1.0 -C 250007" -calibrator
"weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"
=== Summary ===

Correctly Classified Instances         800               65.8979 %
Incorrectly Classified Instances       414               34.1021 %
Kappa statistic                          0.6329
Mean absolute error                      0.099
Root mean squared error                  0.2184
Relative absolute error                 95.0722 %
Root relative squared error             95.7071 %
Total Number of Instances             1214

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.884    0.046    0.622      0.884    0.730      0.716    0.974     0.620     grass
                 0.703    0.015    0.726      0.703    0.714      0.699    0.988     0.700     fire
                 1.000    0.062    0.653      1.000    0.790      0.783    0.969     0.653     water
                 0.708    0.030    0.600      0.708    0.650      0.628    0.978     0.590     bug
                 0.990    0.036    0.716      0.990    0.831      0.826    0.982     0.714     normal
                 0.403    0.009    0.714      0.403    0.515      0.519    0.983     0.627     poison
                 0.680    0.006    0.829      0.680    0.747      0.741    0.994     0.783     electric
                 0.701    0.030    0.580      0.701    0.635      0.615    0.978     0.572     ground
                 0.375    0.000    1.000      0.375    0.545      0.606    0.993     0.760     fairy
                 0.491    0.005    0.813      0.491    0.612      0.619    0.990     0.713     fighting
                 0.856    0.035    0.664      0.856    0.748      0.732    0.979     0.659     psychic
                 0.431    0.014    0.610      0.431    0.505      0.493    0.981     0.571     rock
                 0.239    0.001    0.917      0.239    0.379      0.460    0.988     0.646     ghost
                 0.342    0.000    1.000      0.342    0.510      0.579    0.993     0.739     ice
                 0.420    0.009    0.677      0.420    0.519      0.518    0.986     0.610     dragon
                 0.373    0.008    0.679      0.373    0.481      0.488    0.985     0.601     dark
                 0.245    0.006    0.632      0.245    0.353      0.379    0.983     0.553     steel
                 0.673    0.059    0.507      0.673    0.579      0.541    0.956     0.507     flying
Weighted Avg.    0.659    0.027    0.690      0.659    0.636      0.634    0.980     0.639

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r   <-- classified as
```

```
 84  0   3   0   2   0   0   0   0   0   3   0   0   0   0   0   0   3 |   a = grass
  0 45   1   2   2   0   1   4   0   0   3   0   0   0   0   0   0   6 |   b = fire
  0  0 126   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   c = water
  6  0   1  51   0   0   0   0   0   0   0   0   0   0   0   0   0  14 |   d = bug
  0  0   1   0 101   0   0   0   0   0   0   0   0   0   0   0   0   0 |   e = normal
 15  0   4  13   0  25   0   2   0   0   0   0   0   0   0   0   0   3 |   f = poison
  1  0   3   2   2   0  34   1   0   0   0   0   0   0   2   0   0   5 |   g = electric
  1  0  10   2   1   0   0  47   0   0   2   0   0   0   0   0   0   4 |   h = ground
  2  0   2   0   5   0   1   0  15   0   6   3   0   0   1   0   3   2 |   i = fairy
  3  7   3   2   2   2   0   0   0  26   6   1   0   0   0   0   0   1 |   j = fighting
  0  0   5   0   2   0   0   0   0   0  77   0   0   0   0   0   0   6 |   k = psychic
  2  1  10   5   0   0   0   9   0   0   2  25   0   0   0   0   0   4 |   l = rock
 10  3   2   1   0   4   1   2   0   0   1   0  11   0   2   3   4   2 |   m = ghost
  3  0   6   0   0   0   1   3   0   0   2   2   1  13   3   2   0   2 |   n = ice
  1  2   2   0   0   1   0   7   0   0   4   2   0   0  21   2   0   8 |   o = dragon
  3  3   6   0   0   3   0   3   0   3   3   2   0   0   1  19   0   5 |   p = dark
  2  1   1   7   0   0   3   3   0   3   7   6   0   0   1   2  12   1 |   q = steel
  2  0   7   0  24   0   0   0   0   0   0   0   0   0   0   0   0  68 |   r = flying
```

Bayesian Networks

K2 with 2 parents max

```
Weka, nom.csv -> weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 2 -S
BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5
=== Summary ===

Correctly Classified Instances         787               64.827  %
Incorrectly Classified Instances       427               35.173  %
Kappa statistic                          0.6245
Mean absolute error                      0.0425
Root mean squared error                  0.1559
Relative absolute error                 40.7804 %
Root relative squared error             68.3392 %
Total Number of Instances             1214

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.611    0.028    0.652      0.611    0.630      0.601    0.973     0.759     grass
                 0.672    0.015    0.717      0.672    0.694      0.677    0.984     0.819     fire
                 0.714    0.037    0.692      0.714    0.703      0.668    0.975     0.826     water
                 0.583    0.022    0.627      0.583    0.604      0.581    0.981     0.728     bug
                 0.814    0.022    0.769      0.814    0.790      0.771    0.991     0.907     normal
                 0.597    0.019    0.627      0.597    0.612      0.591    0.984     0.696     poison
                 0.780    0.012    0.736      0.780    0.757      0.747    0.991     0.716     electric
                 0.612    0.024    0.594      0.612    0.603      0.579    0.980     0.697     ground
                 0.575    0.008    0.719      0.575    0.639      0.632    0.993     0.792     fairy
                 0.717    0.016    0.679      0.717    0.697      0.683    0.990     0.769     fighting
                 0.756    0.028    0.687     0.756    0.720     0.697    0.985     0.811     psychic
                 0.552    0.020    0.582     0.552    0.566     0.545    0.982     0.653     rock
                 0.457    0.009    0.656     0.457    0.538     0.533    0.988     0.711     ghost
                 0.526    0.006    0.741     0.526    0.615     0.614    0.986     0.767     ice
                 0.700    0.021    0.583     0.700    0.636     0.622    0.985     0.643     dragon
                 0.706    0.022    0.581     0.706    0.637     0.623    0.985     0.661     dark
                 0.612    0.021    0.545     0.612    0.577     0.559    0.980     0.611     steel
                 0.505    0.045    0.505     0.505    0.505     0.460    0.954     0.524     flying
Weighted Avg.    0.648    0.024    0.650     0.648    0.647     0.625    0.981     0.736

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r   <-- classified as
 58  0  3  2  2  9  1  1  1  2  3  1  6  0  0  3  1  2 |  a = grass
  0 43  2  1  0  0  1  1  0  4  3  1  1  0  1  2  1  3 |  b = fire
  3  1 90  0  2  1  3  6  0  3  4  4  0  1  2  3  0  3 |  c = water
  4  1  2 42  0  6  1  1  0  2  0  1  0  0  0  0  5  7 |  d = bug
  0  2  0  0 83  0  1  1  4  0  1  0  0  0  1  0  0  9 |  e = normal
  6  0  3  7  0 37  0  2  0  1  0  0  1  0  1  2  0  2 |  f = poison
  0  0  2  0  2  0 39  0  0  0  0  0  0  0  1  0  2  4 |  g = electric
  0  3  4  1  0  0  1 41  0  0  1  6  0  2  2  2  3  1 |  h = ground
  1  0  2  0  1  0  1  0 23  0  5  2  0  0  1  0  3  1 |  i = fairy
  2  3  0  0  2  1  0  0  0 38  3  0  0  0  0  2  1  1 |  j = fighting
  2  0  1  0  1  0  0  1  1  2 68  0  0  1  4  2  2  5 |  k = psychic
  1  0  6  4  0  0  0  3  1  1  2 32  0  1  2  1  3  1 |  l = rock
  4  2  2  1  0  3  1  2  0  0  1  0 21  1  2  3  3  0 |  m = ghost
  3  0  5  0  0  0  1  1  0  0  1  1  0 20  3  2  0  1 |  n = ice
  1  1  0  0  0  0  1  5  0  0  0  0  0  0 35  2  0  5 |  o = dragon
  0  1  3  0  0  1  0  1  0  1  1  1  0  0  1 36  1  4 |  p = dark
  1  0  1  2  0  0  1  0  0  2  5  3  1  0  1  1 30  1 |  q = steel
  3  3  4  7 15  1  1  3  2  0  1  3  2  1  3  1  0 51 |  r = flying
```

Naive
Weka, nom.csv -> weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.fixed.NaiveBayes -- -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5

```
=== Summary ===

Correctly Classified Instances         442               36.4086 %
Incorrectly Classified Instances       772               63.5914 %
Kappa statistic                          0.3239
Mean absolute error                      0.0824
Root mean squared error                  0.2116
Relative absolute error                 79.1352 %
Root relative squared error             92.7344 %
Total Number of Instances             1214

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.358    0.048    0.386      0.358    0.372      0.321    0.841     0.326     grass
                 0.422    0.036    0.397      0.422    0.409      0.375    0.888     0.377     fire
                 0.302    0.061    0.365      0.302    0.330      0.263    0.800     0.356     water
                 0.403    0.055    0.315      0.403    0.354      0.310    0.881     0.369     bug
                 0.412    0.050    0.429      0.412    0.420      0.368    0.852     0.388     normal
                 0.355    0.030    0.393      0.355    0.373      0.341    0.901     0.339     poison
                 0.420    0.026    0.412      0.420    0.416      0.391    0.918     0.372     electric
                 0.313    0.033    0.356      0.313    0.333      0.298    0.883     0.297     ground
                 0.450    0.022    0.409      0.450    0.429      0.409    0.932     0.406     fairy
                 0.509    0.051    0.314      0.509    0.388      0.365    0.883     0.345     fighting
                 0.300    0.025    0.491      0.300    0.372      0.347    0.888     0.398     psychic
                 0.293    0.020    0.425      0.293    0.347      0.326    0.919     0.356     rock
                 0.217    0.013    0.400      0.217    0.282      0.275    0.905     0.334     ghost
                 0.263    0.007    0.556      0.263    0.357      0.369    0.907     0.399     ice
                 0.540    0.074    0.239      0.540    0.331      0.319    0.879     0.277     dragon
                 0.490    0.040    0.352      0.490    0.410      0.385    0.907     0.328     dark
                 0.551    0.048    0.325      0.551    0.409      0.392    0.940     0.391     steel
                 0.198    0.039    0.317      0.198    0.244      0.198    0.806     0.280     flying
Weighted Avg.    0.364    0.040    0.381      0.364    0.361      0.326    0.873     0.350

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r   <-- classified as
 34  4  5  7  3  8  3  3  1  5  4  0  5  1  3  4  3  2 |  a = grass
  2 27  3  4  0  2  2  3  3  3  1  0  1  0  8  1  0  4 |  b = fire
  8  5 38  6  6  2  5  7  3  6  5  7  3  2  7  8  4  4 |  c = water
  5  1  3 29  0  4  3  1  2  6  0  0  0  0  1  4  7  6 |  d = bug
  3  5  9  6 42  1  1  3  5  8  3  0  1  0  6  2  2  5 |  e = normal
  3  3  6  9  2 22  0  3  1  1  2  2  1  0  1  2  2  2 |  f = poison
  1  2  3  2  2  2 21  1  1  1  2  0  0  0  3  1  3  5 |  g = electric
  3  1  4  3  4  2  1 21  1  3  1  5  1  1  7  2  5  2 |  h = ground
  1  0  4  2  3  0  0  0 18  1  3  1  0  0  4  0  3  0 |  i = fairy
  2  5  0  4  3  3  0  0  0 27  1  1  0  0  1  4  2  0 |  j = fighting
  5  4  5  4  6  1  4  1  2  2 27  1  0  2 14  2  6  4 |  k = psychic
  3  0  2  1  0  1  1  4  3  3  1 17  0  2  3  5 10  2 |  l = rock
  6  1  3  2  1  2  2  2  0  1  4  0 10  0  3  2  5  2 |  m = ghost
  3  0  4  2  2  1  2  2  0  1  1  1  1 10  5  1  1  1 |  n = ice
  1  2  3  1  3  1  0  2  0  3  0  2  0  0 27  2  1  2 |  o = dragon
  2  1  4  1  1  2  1  3  0  5  0  0  0  0  4 25  0  2 |  p = dark
  1  1  1  2  1  0  1  1  1  3  0  3  1  0  4  2 27  0 |  q = steel
  5  6  7  7 19  2  4  2  3  7  0  0  1  0 12  4  2 20 |  r = flying
```

Gradient Boosting
REPTree default parameters
Weka, nom.csv ->

```
=== Summary ===
Correctly Classified Instances         440               36.2438 %
Incorrectly Classified Instances       774               63.7562 %
Kappa statistic                          0.3156
Mean absolute error                      0.0901
Root mean squared error                  0.2092
Relative absolute error                 86.5414 %
Root relative squared error             91.6609 %
Total Number of Instances             1214
```

Changing max depth did not change the result at all

Shrinkage reduced to 0.1

```
=== Summary ===

Correctly Classified Instances         346               28.5008 %
Incorrectly Classified Instances       868               71.4992 %
Kappa statistic                          0.2347
Mean absolute error                      0.1045
Root mean squared error                  0.2281
Relative absolute error                100.3353 %
Root relative squared error             99.971  %
Total Number of Instances             1214
```

Shrinkage increased to 3.0

```
=== Summary ===

Correctly Classified Instances         442               36.4086 %
Incorrectly Classified Instances       772               63.5914 %
Kappa statistic                          0.3203
Mean absolute error                      0.0741
Root mean squared error                  0.2292
Relative absolute error                 71.113  %
Root relative squared error            100.4435 %
Total Number of Instances             1214
```

Shrinkage increased to 10

```
=== Summary ===

Correctly Classified Instances         369               30.3954 %
Incorrectly Classified Instances       845               69.6046 %
Kappa statistic                          0.2595
Mean absolute error                      0.0774
Root mean squared error                  0.2698
Relative absolute error                 74.2843 %
Root relative squared error            118.2271 %
Total Number of Instances             1214
```

Shrinkage set at 1.75 ← this seems to be the best we can get it

```
=== Summary ===
Correctly Classified Instances         519               42.7512 %
Incorrectly Classified Instances       695               57.2488 %
Kappa statistic                          0.388
Mean absolute error                      0.0759
Root mean squared error                  0.1992
Relative absolute error                 72.8868 %
Root relative squared error             87.2958 %
Total Number of Instances             1214
```

Increasing numIterations to 20, 25, and 100 had no effect

Decrease numIterations to 5

```
=== Summary ===

Correctly Classified Instances         437               35.9967 %
Incorrectly Classified Instances       777               64.0033 %
Kappa statistic                          0.3129
Mean absolute error                      0.0901
Root mean squared error                  0.2092
Relative absolute error                 86.5116 %
Root relative squared error             91.6656 %
Total Number of Instances             1214
```

Stacking

Weka, disc.csv -> Using PRISM, Bayesian Network with K2, and SVM with kernel Puk and a Bayesian Network with K2 as the meta classifier, since those produced the highest correctly classified instances thus far.

```
=== Summary ===

Correctly Classified Instances         130               10.7084 %
Incorrectly Classified Instances      1084               89.2916 %
Kappa statistic                          0.0067
Mean absolute error                      0.1046
Root mean squared error                  0.2309
Relative absolute error                100.4735 %
Root relative squared error            101.1978 %
Total Number of Instances             1214
```

Supervised

Weka, nom.csv -> weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.KDTree -A \"weka.core.EuclideanDistance -R first-last\" -S weka.core.neighboursearch.kdtrees.SlidingMidPointOfWidestSide -W 0.01 -L 40 -N"

KDTree

```
=== Summary ===

Correctly Classified Instances         800               65.8979 %
```

```
Incorrectly Classified Instances       414               34.1021 %
Kappa statistic                          0.6348
Mean absolute error                      0.0386
Root mean squared error                  0.1377
Relative absolute error                 37.0892 %
Root relative squared error             60.3289 %
Total Number of Instances             1214

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q   r   <-- classified as
  95   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   a = grass
   0  64   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   b = fire
   3   1 122   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   c = water
   6   2   1  63   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   d = bug
   2   2   1   0  97   0   0   0   0   0   0   0   0   0   0   0   0   0 |   e = normal
  15   0   4  13   0  30   0   0   0   0   0   0   0   0   0   0   0   0 |   f = poison
   1   1   3   2   2   0  41   0   0   0   0   0   0   0   0   0   0   0 |   g = electric
   1   4  10   2   1   2   1  46   0   0   0   0   0   0   0   0   0   0 |   h = ground
   2   0   2   0   5   0   1   0  30   0   0   0   0   0   0   0   0   0 |   i = fairy
   3   7   3   2   2   2   0   0   0  34   0   0   0   0   0   0   0   0 |   j = fighting
   3   3   5   0   2   0   0   2   6   6  63   0   0   0   0   0   0   0 |   k = psychic
   2   1  10   5   0   0   0   9   3   1   2  25   0   0   0   0   0   0 |   l = rock
  10   3   2   1   0   4   1   2   0   0   1   0  22   0   0   0   0   0 |   m = ghost
   3   0   6   0   0   0   1   3   0   0   2   2   1  20   0   0   0   0 |   n = ice
   1   2   2   0   0   1   2   7   1   0   4   2   2   3  23   0   0   0 |   o = dragon
   3   3   6   0   0   3   0   3   0   3   3   2   3   2   3  17   0   0 |   p = dark
   2   1   1   7   0   0   3   3   3   3   7   6   4   0   1   2   6   0 |   q = steel
   5   6   7  14  24   3   5   4   2   1   6   4   2   2   8   5   1   2 |   r = flying
```