Quarter 1 Project FINAL REPORT: Predicting the Walkability Index of Regions in America

By Arnav Gupta and Raghav Kamineni (Dr. Yilmaz, Period 5)

September 23, 2024 - October 22, 2024

Part 1: Statement/Project Goal

The U.S.Environmental Protection Agency rates every Census block group (region of land) based on a number of factors, culminating in its Walkability Index (WI), or likelyhood of walking being used as a significantly used mode of transportation in that region. With growing socioeconomic inequality, walking has become one of the major modes of transportation in the modern world, and the U.S. government collected data from the Smart Location Database (a national geographic census-like database consisting of almost 100 factors) to determine the WI of regions across America. The various factors span across 8 broad measures of quality for each region, and they are listed below.

1. Housing Prices
2. Population and Land Density
3. Diversity of Land Utilization
4. Neighborhood Layouts and Designs
5. Location Accessibility
6. Transit Service
7. Employment
8. Individual Demographics

Based on these listed measures (and dozens of sub measures for each of these broader measures), a region's WI is calculated from a range of 1-20 and utilized as a function of the given region's walkability as a usable mode of transportation. However, given the classification nature of the project, we will discretize these values into classifications that we'll cover below.

The WI helps people make better decisions on where to live, and given that almost 30% of individuals walk to work (and 50% for leisure), it's important for the population to know which region they should live in if they are in the walking population. In this project, we will look at all the attributes available in the WI dataset and use them to train a classification model where we predict the region's WI classification. After selecting the most informational attributes and testing which of our models attained the highest accuracy, we'll be able to identify which attributes were most significant to a region's WI and simplify the results for the population so they can make more informed decisions on the region to live in.

Part 2: Description of the Dataset

We found our dataset named 'Walkability Index' (found here) from the U.S. government's public data catalog. After downloading the dataset as a CSV file and converting it into an ARFF file to manipulate on WEKA, we analyzed the characteristics of this dataset.

The complete dataset had 220,740 instances and 117 attributes including the class attribute, thus the dimension of the dataset is 116D. While the class attribute didn't have any missing values, there were eleven other attributes with missing values, and the number of missing attributes ranged from 1-53,031 values. However, we noticed that some attributes had an abnormally high number of seemingly default values (such as zero), and we'll make sure to sort that issue out during the preprocessing stage. As for the class attribute (which had no missing values), the original class attribute values are quantitative continuous values in the range [1-20); however we plan on binning these values into four equal-width classes for our classification models, and these classes are given below.

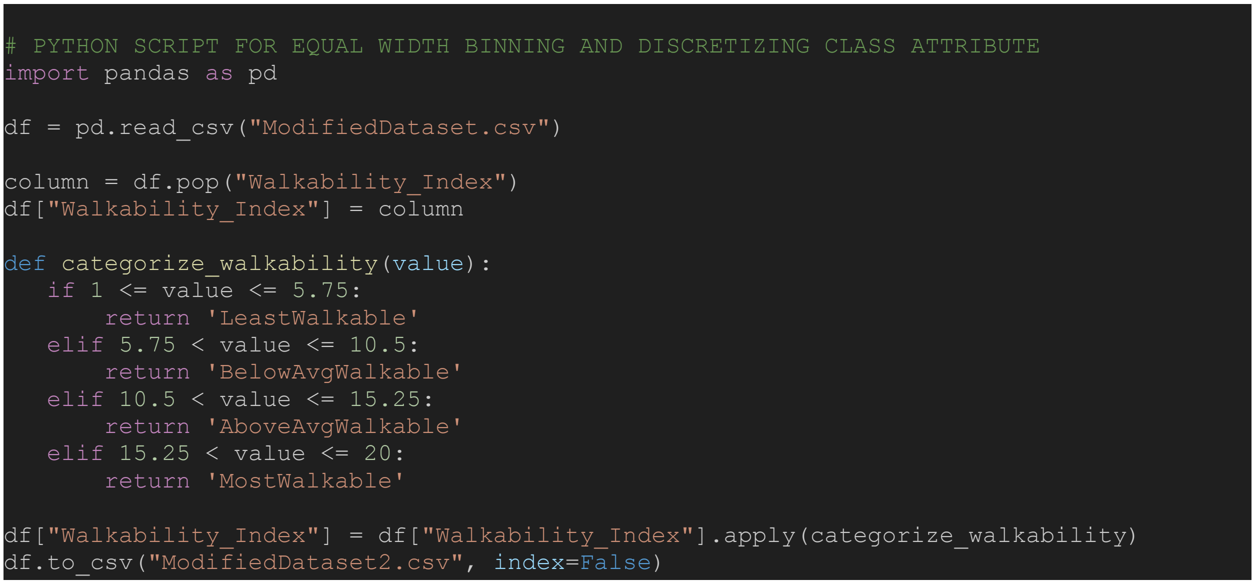
1. Least Walkable, [1,5.75)
2. Below Average Walkable, [5.75,10.5)
3. Above Average Walkable, [10.5,15.25)
4. Most Walkable, [15.25-20)

As for the class attribute distribution, the 220,740 values had a mean of 9.542 and a standard deviation of 4.374, which, given the range of [1,20), tells us that the distribution is skewed to the right.

Part 3: Preprocessing

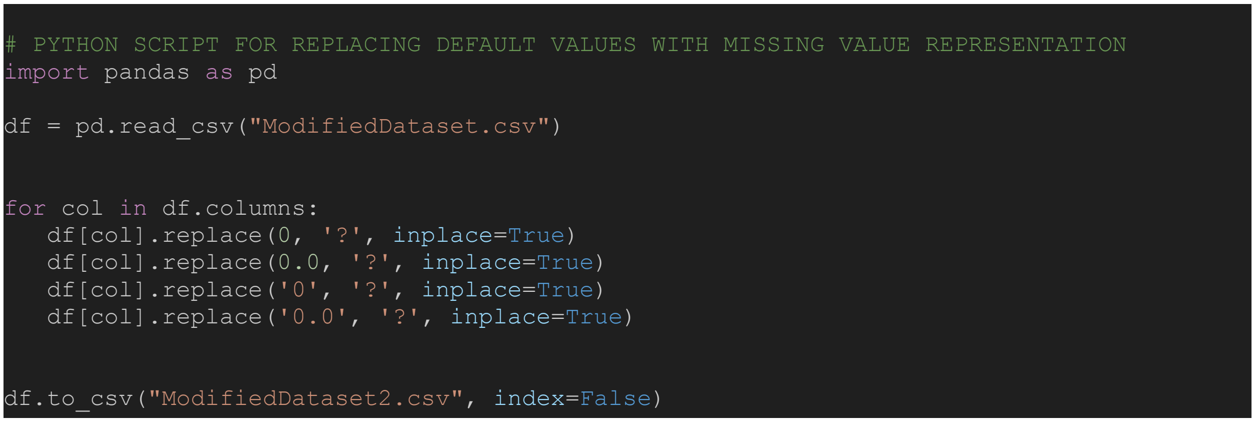
On federal agency websites, the datasets are frequently just raw data, and the WI dataset was no exception. There are numerous steps we took in order to preprocess the dataset, thus making it more consistent, reliable, and even potentially increasing the long-term accuracy of our models. Below is a list of steps we took to preprocess this WI dataset in both Python and WEKA.

1. Binning, Discretizing, and Renaming the Class Attribute: using the raw dataset, we declared that the class attribute should be attribute index 115 ('NatWalkInd' or national WI) so that we could predict the WI of a given region. First, we opened the CSV file in Excel and changed the name of this attribute to a more user-friendly name like ('Walkability_Index'), and then manually shifted the new WI column to the end using Excel's features so WEKA would also correctly declare it the class attribute. With this attribute being a quantitative continuous variable in the range [1, 20], we decided to bin this attribute into four equal-with bins and then discretize said bins into qualitative data labels (mentioned in dataset description) using the Python in the script below. Our class distribution after binning was 66989 instances for AboveAvgWalkable, 74795 instances for BelowAvgWalkable, 25630 instances for MostWalkable, and 53326 instances for LeastWalkable.



```python
# PYTHON SCRIPT FOR EQUAL WIDTH BINNING AND DISCRETIZING CLASS ATTRIBUTE
import pandas as pd

df = pd.read_csv("ModifiedDataset.csv")

column = df.pop("Walkability_Index")
df["Walkability_Index"] = column

def categorize_walkability(value):
   if 1 <= value <= 5.75:
       return 'LeastWalkable'
   elif 5.75 < value <= 10.5:
       return 'BelowAvgWalkable'
   elif 10.5 < value <= 15.25:
       return 'AboveAvgWalkable'
   elif 15.25 < value <= 20:
       return 'MostWalkable'

df["Walkability_Index"] = df["Walkability_Index"].apply(categorize_walkability)
df.to_csv("ModifiedDataset2.csv", index=False)
```

2. Removing Default Values: While going through the dataset, we realized that numerous columns (relating to region size, location, and several other variables) had quite a number of default values, specifically the number zero. Going through the dataset, it seemed impossible for instances to have no area, location, employment, and various other factors, cementing our belief that these zeroes were incorrect default values. With these numbers being much less than 70% of the total number of values in a particular attribute, these incorrect default values needed to be treated like missing values, and thus, the process of removing default values and replacing them with WEKA's placeholder for missing values (the question mark symbol) was carried out by Python in the script below.



```python
# PYTHON SCRIPT FOR REPLACING DEFAULT VALUES WITH MISSING VALUE REPRESENTATION
import pandas as pd

df = pd.read_csv("ModifiedDataset.csv")


for col in df.columns:
   df[col].replace(0, '?', inplace=True)
   df[col].replace(0.0, '?', inplace=True)
   df[col].replace('0', '?', inplace=True)
   df[col].replace('0.0', '?', inplace=True)


df.to_csv("ModifiedDataset2.csv", index=False)
```

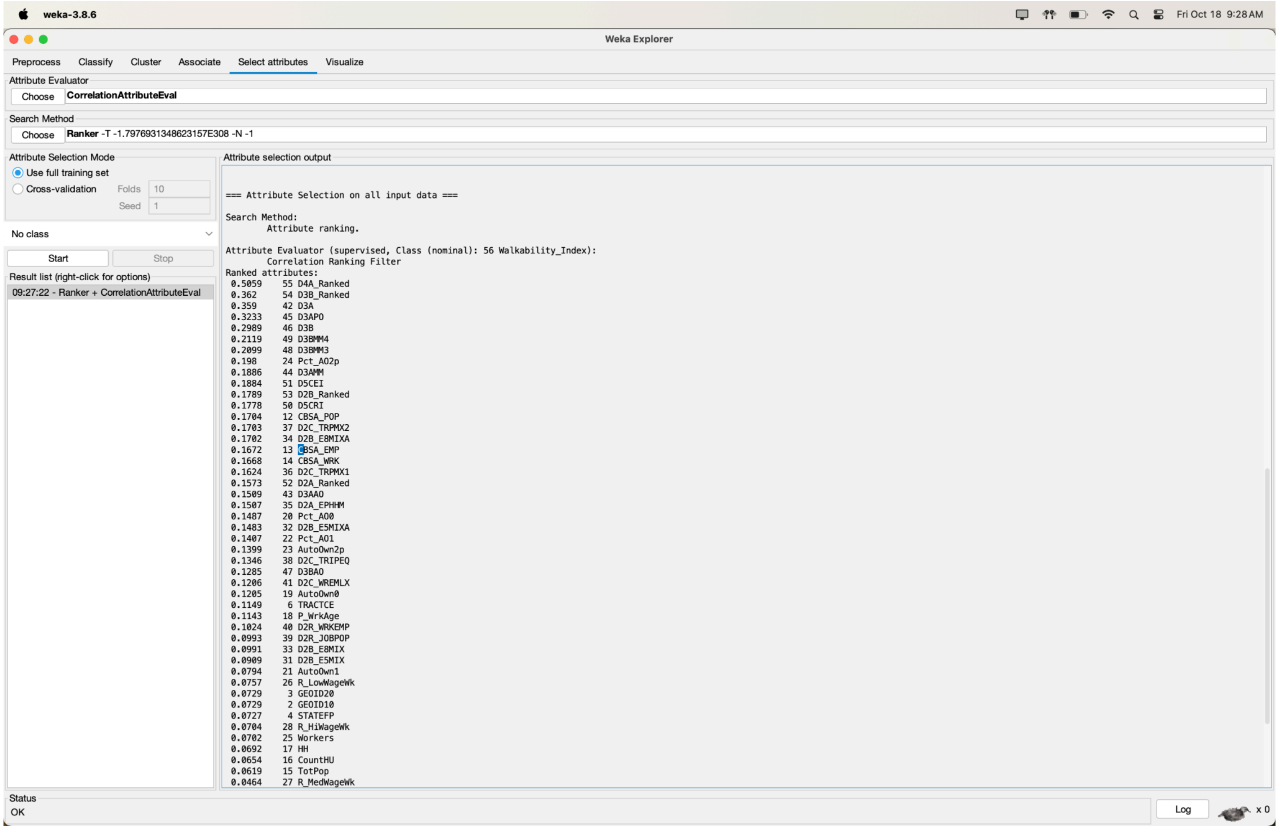
3. Removing Attributes With 70%+ Missing Values: After replacing the default values with missing values, we opened the CSV-file dataset in WEKA and manually went through the dataset to check for missing values within each attribute, and concluded that 60(!) columns (AC_Total, AC_Water, AC_Land, AC_Unpr, TotEmp, E5_ Ret, E5_Off, E5_Ind, E5_ Svc, E5_Ent, E8_Ret, E8_off, E8_ Ind, E8_Svc, E8_Ent, E8_Ed, E8_HIth, E8_Pub, E_LowWageWk, E_MedWageWk, E_HiWageWk, D1A, D1B, D1C, D1C5_RET, D1C5_OFF, D1C5_IND, D1C5_SVC, D1C5_ENT, D1C8_RET, D1C8_OFF, D1C8_IND, D1C8_SVC, D1C8_ENT, D1C8_ED, D1C8_HLTH, D1C8_PUB, D1D, D1_FLAG, D2A_JPHH, DA_WRKEMP, D3BPO4, D4A, D4B025, D4B050, D4C, D4D, D4E, D5AR, D5AE, D5BR, D5BE, D5CR, D5CE, D5DR, D5DRI, D5DE, D5DEI, Shape_Length, Shape_Area) needed to be removed due to their high number of now high number of missing values, specifically 70%+. Using WEKA's remove attribute feature, we simply manipulated the dataset to remove the several above-mentioned attributes and saved our new dataset in WEKA right after.
4. Filling in Missing Values with Mean/Mode: With the dataset rid of attributes with numerous missing values and significantly skewed data, we changed all the missing values to the attribute's mean or mode using WEKA's useful ReplaceMissingValues feature in every single attribute index (note that the class attribute had no missing values in the raw dataset, so accidently replacing class attributes was not a concern).

After removing attributes that had 70% or more missing we had 55 attributes to use for the attribute selection algorithm.

Part 4: Attribute Selection Algorithms and Building Train/Test/Split Sets
For this project, we will use four attribute selection algorithms. We'll declare our thresholds and remaining attributes for each of the five attribute selection algorithms we've chosen below.

1. CorrelationAttributeEval

After running the correlation-based attribute selection algorithm, we decided to keep attributes with a correlation coefficient greater or equal to 0.1. The attributes remaining were the following: D4A_Ranked, D3B_Ranked, D3A, D3APO, D3B, D3BMM4, D3BMM3, Pct_AO2p, D3AMM, D5CEI, D2B_Ranked, D5CRI, CBSA_POP, D2C_TRPMX2, D2B_E8MIXA, CBSA_EMP, CBSA_WRK, D2C_TRPMX1, D2A_Ranked, D3AAO, D2A_EPHHM, Pct_AO0, D2B_E5MIXA, Pct_AO1, AutoOwn2p, D2C_TRIPEQ, D3BAO, D2C_WREMLX, AutoOwn0, TRACTCE, P_WrkAge, and D2R_WRKEMP.



Attribute Evaluator: CorrelationAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode:
● Use full training set
○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)
09:27:22 - Ranker + CorrelationAttributeEval

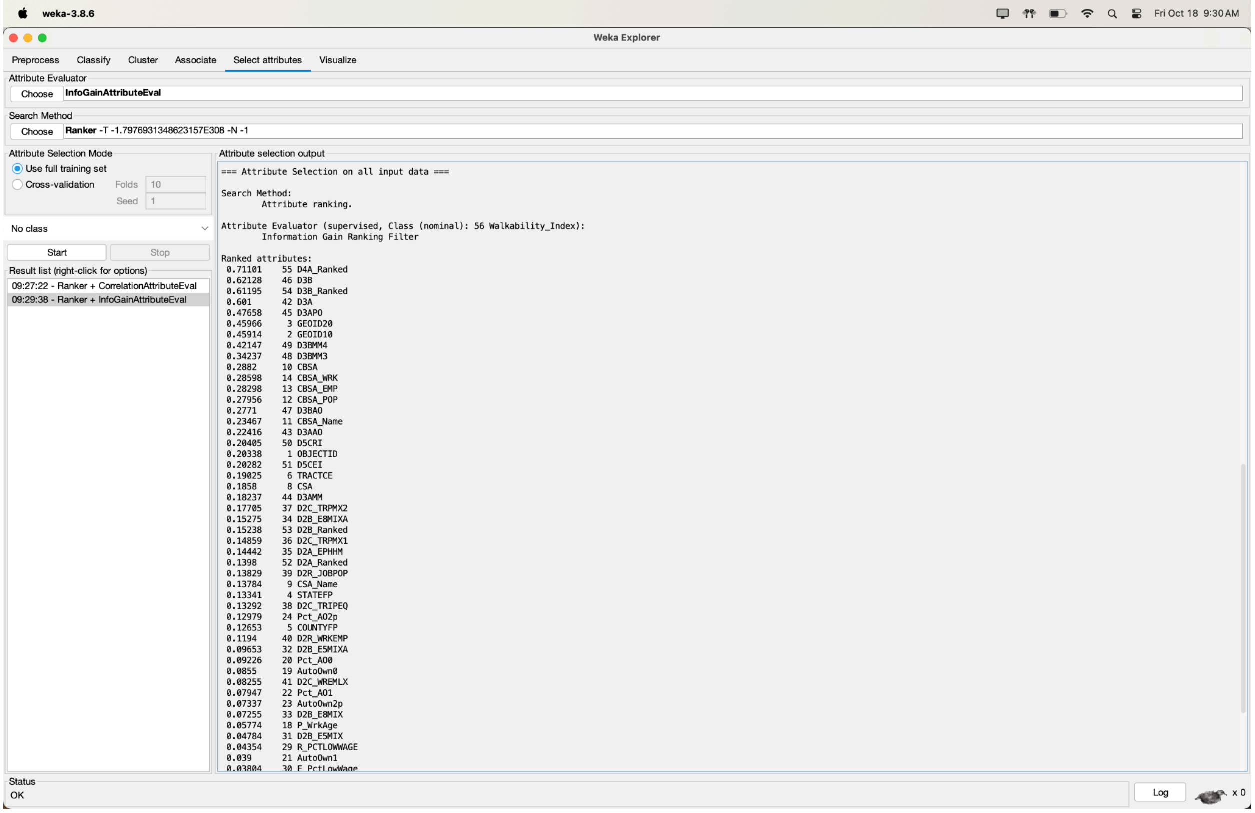
Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
	Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 56 Walkability_Index):
	Correlation Ranking Filter
Ranked attributes:
 0.5059    55 D4A_Ranked
 0.362     54 D3B_Ranked
 0.359     42 D3A
 0.3233    45 D3APO
 0.2989    46 D3B
 0.2119    49 D3BMM4
 0.2099    48 D3BMM3
 0.198     24 Pct_AO2p
 0.1886    44 D3AMM
 0.1884    51 D5CEI
 0.1789    53 D2B_Ranked
 0.1778    50 D5CRI
 0.1704    12 CBSA_POP
 0.1703    37 D2C_TRPMX2
 0.1702    34 D2B_E8MIXA
 0.1672    13 CBSA_EMP
 0.1668    14 CBSA_WRK
 0.1624    36 D2C_TRPMX1
 0.1573    52 D2A_Ranked
 0.1509    43 D3AAO
 0.1507    35 D2A_EPHHM
 0.1487    20 Pct_AO0
 0.1483    32 D2B_E5MIXA
 0.1407    22 Pct_AO1
 0.1399    23 AutoOwn2p
 0.1346    38 D2C_TRIPEQ
 0.1285    47 D3BAO
 0.1206    41 D2C_WREMLX
 0.1205    19 AutoOwn0
 0.1149     6 TRACTCE
 0.1143    18 P_WrkAge
 0.1024    40 D2R_WRKEMP
 0.0993    39 D2R_JOBPOP
 0.0991    33 D2B_E8MIX
 0.0909    31 D2B_E5MIX
 0.0794    21 AutoOwn1
 0.0757    26 R_LowWageWk
 0.0729     3 GEOID20
 0.0729     2 GEOID10
 0.0727     4 STATEFP
 0.0704    28 R_HiWageWk
 0.0702    25 Workers
 0.0692    17 HH
 0.0654    16 CountHU
 0.0619    15 TotPop
 0.0464    27 R_MedWageWk
```

Status
OK

2. InfoGainAttributeEval

After running the InfoGain-based attribute selection algorithm, we decided to keep attributes with a coefficient of greater or equal to 0.15. The attributes remaining were the following: D4A_Ranked, D3B, D3B_Ranked, D3A, D3APO, GEOID20, GEOID10, D3BMM4, D3BMM3, CBSA, CBSA_WRK, CBSA_EMP, CBSA_POP, D3BAO, CBSA_Name, D3AAO, D5CRI, OBJECTID, D5CEI, TRACTCE, CSA, D3AMM, D2C_TRPMX2, D2B_E8MIXA, and D2B_Ranked.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)
09:27:22 - Ranker + CorrelationAttributeEval
09:29:38 - Ranker + InfoGainAttributeEval

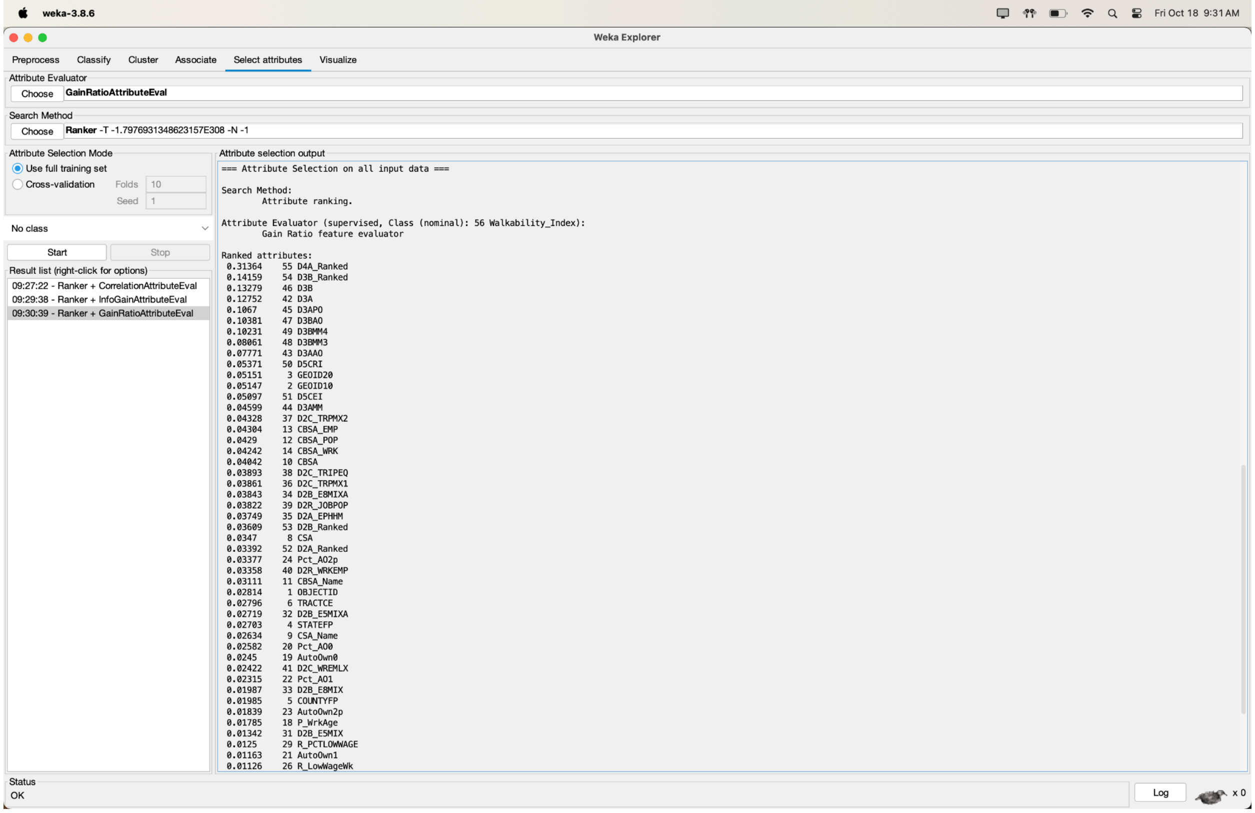
Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 56 Walkability_Index):
        Information Gain Ranking Filter

Ranked attributes:
 0.71101    55 D4A_Ranked
 0.62128    46 D3B
 0.61195    54 D3B_Ranked
 0.601      42 D3A
 0.47658    45 D3APO
 0.45966     3 GEOID20
 0.45914     2 GEOID10
 0.42147    49 D3BMM4
 0.34237    48 D3BMM3
 0.2882     10 CBSA
 0.28598    14 CBSA_WRK
 0.28298    13 CBSA_EMP
 0.27956    12 CBSA_POP
 0.2771     47 D3BAO
 0.23467    11 CBSA_Name
 0.22416    43 D3AAO
 0.20405    50 D5CRI
 0.20338     1 OBJECTID
 0.20282    51 D5CEI
 0.19025     6 TRACTCE
 0.1858      8 CSA
 0.18237    44 D3AMM
 0.17705    37 D2C_TRPMX2
 0.15275    34 D2B_E8MIXA
 0.15238    53 D2B_Ranked
 0.14859    36 D2C_TRPMX1
 0.14442    35 D2A_EPHHM
 0.1398     52 D2A_Ranked
 0.13829    39 D2R_JOBPOP
 0.13784     9 CSA_Name
 0.13341     4 STATEFP
 0.13292    38 D2C_TRIPEQ
 0.12979    24 Pct_AO2p
 0.12653     5 COUNTYFP
 0.1194     40 D2R_WRKEMP
 0.09653    32 D2B_E5MIXA
 0.09226    20 Pct_AO0
 0.0855     19 AutoOwn0
 0.08255    41 D2C_WREMLX
 0.07947    22 Pct_AO1
 0.07337    23 AutoOwn2p
 0.07255    33 D2B_E8MIX
 0.05774    18 P_WrkAge
 0.04784    31 D2B_E5MIX
 0.04354    29 R_PCTLOWWAGE
 0.039      21 AutoOwn1
 0.03804    30 E_PctLowWage
```

Status
OK

3. GainRatioAttributeEval

After running the GainRatio-based attribute selection algorithm, we decided to keep attributes with a coefficient of greater or equal to 0.03. The attributes remaining were the following: D4A_Ranked, D3B_Ranked, D3B, D3A, D3APO, D3BAO, D3BMM4, D3BMM3, D3AAO, D5CRI, GEOID20, GEOID10, D5CEI, D3AMM, D2C_TRPMX2, CBSA_EMP, CBSA_POP, CBSA_WRK, CBSA, D2C_TRIPEQ, D2C_TRPMX1, D2B_E8MIXA, D2R_JOBPOP, D2A_EPHHM, D2B_Ranked, CSA, D2A_Ranked, Pct_AO2p, D2R_WRKEMP, and CBSA_Name.



Attribute Evaluator: GainRatioAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

09:27:22 - Ranker + CorrelationAttributeEval
09:29:38 - Ranker + InfoGainAttributeEval
09:30:39 - Ranker + GainRatioAttributeEval

Attribute selection output

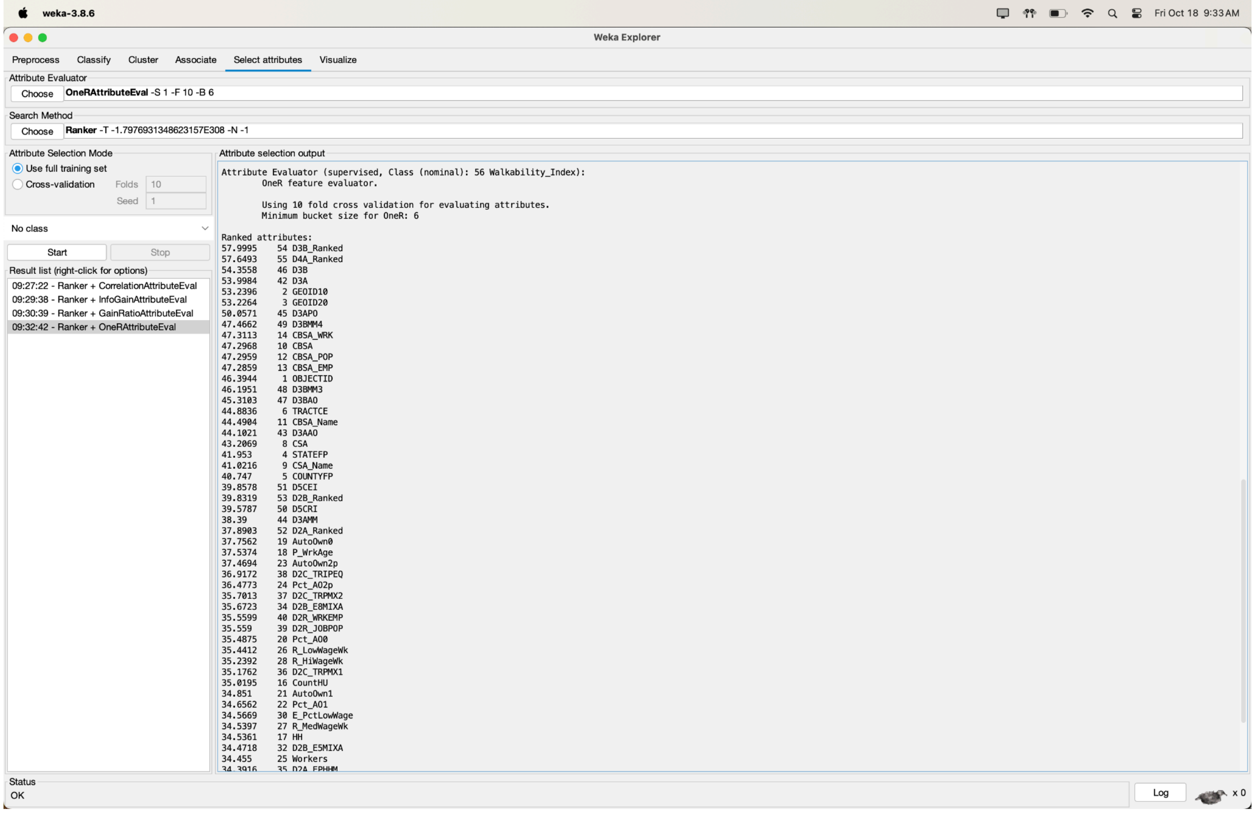
```
=== Attribute Selection on all input data ===

Search Method:
	Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 56 Walkability_Index):
	Gain Ratio feature evaluator

Ranked attributes:
 0.31364    55 D4A_Ranked
 0.14159    54 D3B_Ranked
 0.13279    46 D3B
 0.12752    42 D3A
 0.1067     45 D3APO
 0.10381    47 D3BAO
 0.10231    49 D3BMM4
 0.08061    48 D3BMM3
 0.07771    43 D3AAO
 0.05371    50 D5CRI
 0.05151     3 GEOID20
 0.05147     2 GEOID10
 0.05097    51 D5CEI
 0.04599    44 D3AMM
 0.04328    37 D2C_TRPMX2
 0.04304    13 CBSA_EMP
 0.0429     12 CBSA_POP
 0.04242    14 CBSA_WRK
 0.04042    10 CBSA
 0.03893    38 D2C_TRIPEQ
 0.03861    36 D2C_TRPMX1
 0.03843    34 D2B_E8MIXA
 0.03822    39 D2R_JOBPOP
 0.03749    35 D2A_EPHHM
 0.03609    53 D2B_Ranked
 0.0347      8 CSA
 0.03392    52 D2A_Ranked
 0.03377    24 Pct_AO2p
 0.03358    40 D2R_WRKEMP
 0.03111    11 CBSA_Name
 0.02814     1 OBJECTID
 0.02796     6 TRACTCE
 0.02719    32 D2B_E5MIXA
 0.02703     4 STATEFP
 0.02634     9 CSA_Name
 0.02582    20 Pct_AO0
 0.0245     19 AutoOwn0
 0.02422    41 D2C_WREMLX
 0.02315    22 Pct_AO1
 0.01987    33 D2B_E8MIX
 0.01985     5 COUNTYFP
 0.01839    23 AutoOwn2p
 0.01785    18 P_WrkAge
 0.01342    31 D2B_E5MIX
 0.0125     29 R_PCTLOWWAGE
 0.01163    21 AutoOwn1
 0.01126    26 R_LowWageWk
```

Status

OK

4. OneRAttributeEval

After running the OneR-based attribute selection algorithm, we decided to keep attributes with a coefficient greater than or equal to 36 with a minimum bucket size of 6. The attributes remaining were the following: D3B_Ranked, D4A_Ranked, D3B, D3A, GEOID10, GEOID20, D3APO, D3BMM4, CBSA_WRK, CBSA, CBSA_POP, CBSA_EMP, OBJECTID, D3BMM3, D3BAO, TRACTCE, CBSA_Name, D3AAO, CSA, STATEFP, CSA_Name, COUNTYFP, D5CEI, D2B_Ranked, D5CRI, D3AMM, D2A_Ranked, AutoOwn0, P_WrkAge, AutoOwn2p, D2C_TRIPEQ, and Pct_AO2p.



Attribute Evaluator: OneRAttributeEval -S 1 -F 10 -B 6

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options)

```
09:27:22 - Ranker + CorrelationAttributeEval
09:29:38 - Ranker + InfoGainAttributeEval
09:30:39 - Ranker + GainRatioAttributeEval
09:32:42 - Ranker + OneRAttributeEval
```

Attribute selection output

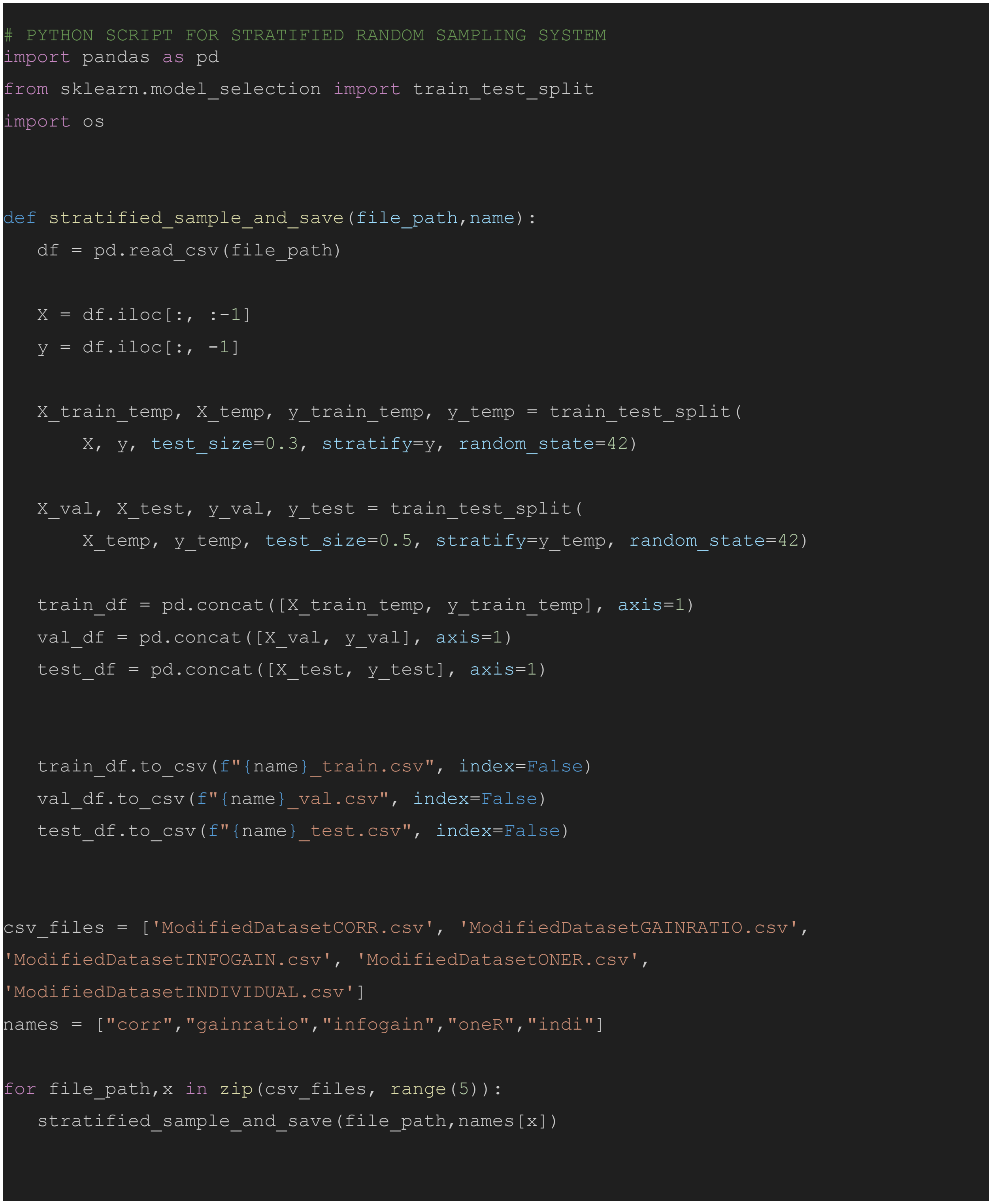
```
Attribute Evaluator (supervised, Class (nominal): 56 Walkability_Index):
        OneR feature evaluator.

        Using 10 fold cross validation for evaluating attributes.
        Minimum bucket size for OneR: 6

Ranked attributes:
57.9995    54 D3B_Ranked
57.6493    55 D4A_Ranked
54.3558    46 D3B
53.9984    42 D3A
53.2396     2 GEOID10
53.2264     3 GEOID20
50.0571    45 D3APO
47.4662    49 D3BMM4
47.3113    14 CBSA_WRK
47.2968    10 CBSA
47.2959    12 CBSA_POP
47.2859    13 CBSA_EMP
46.3944     1 OBJECTID
46.1951    48 D3BMM3
45.3103    47 D3BAO
44.8836     6 TRACTCE
44.4904    11 CBSA_Name
44.1021    43 D3AAO
43.2069     8 CSA
41.953      4 STATEFP
41.0216     9 CSA_Name
40.747      5 COUNTYFP
39.8578    51 D5CEI
39.8319    53 D2B_Ranked
39.5787    50 D5CRI
38.39      44 D3AMM
37.8903    52 D2A_Ranked
37.7562    19 AutoOwn0
37.5374    18 P_WrkAge
37.4694    23 AutoOwn2p
36.9172    38 D2C_TRIPEQ
36.4773    24 Pct_AO2p
35.7013    37 D2C_TRPMX2
35.6723    34 D2B_E8MIXA
35.5599    40 D2R_WRKEMP
35.559     39 D2R_JOBPOP
35.4875    20 Pct_AO0
35.4412    26 R_LowWageWk
35.2392    28 R_HiWageWk
35.1762    36 D2C_TRPMX1
35.0195    16 CountHU
34.851     21 AutoOwn1
34.6562    22 Pct_AO1
34.5669    30 E_PctLowWage
34.5397    27 R_MedWageWk
34.5361    17 HH
34.4718    32 D2B_E5MIXA
34.455     25 Workers
34.3916    35 D2A_EPHHM
```

Status
OK

5. Attribute Selection Algorithm of Our Choice

With the four attribute selection algorithms we've chosen, we believe that a better attribute selection algorithm can be found by combining the best attributes in the previous algorithms. In lieu of that, we've selected (based on our interpretations of their rankings in the last four attribute selection algorithms) a list of the remaining attributes as follows: D4A_Ranked, D3B_Ranked, D3B, D3A, D3APO, D3BMM4, D3BMM3, Pct_AO2p, D5CEI, D2B_Ranked, CBSA_POP, D2C_TRPMX2, D2B_E8MIXA, CBSA_EMP, CBSA_WRK, D3BAO, D3AAO, D5CRI, D3AMM, D2C_TRIPEQ, D2R_WRKEMP, TRACTCE, CBSA_Name, CSA, AutoOwn0, and P_WrkAge.

Train/Validation/Test Sets: After completing each of these attribute selection algorithms, we created a system based on our four classifications to then split the data randomly and effectively into train/test split data of 70% training, 15% validation, and 15% testing. First, we looped through each file post attribute selection, and for each file, we used the Scikit library to first use stratified random sampling on the file to split the file into 70% and 30%. Then we used stratified random sampling again on the 30% dataset to split it into two 15% data sets. After that, we saved the 70% set as the training set, 15% as the validation set, and the last 15% as the test set.



```
# PYTHON SCRIPT FOR STRATIFIED RANDOM SAMPLING SYSTEM
import pandas as pd
from sklearn.model_selection import train_test_split
import os


def stratified_sample_and_save(file_path,name):
   df = pd.read_csv(file_path)

   X = df.iloc[:, :-1]
   y = df.iloc[:, -1]

   X_train_temp, X_temp, y_train_temp, y_temp = train_test_split(
       X, y, test_size=0.3, stratify=y, random_state=42)

   X_val, X_test, y_val, y_test = train_test_split(
       X_temp, y_temp, test_size=0.5, stratify=y_temp, random_state=42)

   train_df = pd.concat([X_train_temp, y_train_temp], axis=1)
   val_df = pd.concat([X_val, y_val], axis=1)
   test_df = pd.concat([X_test, y_test], axis=1)


   train_df.to_csv(f"{name}_train.csv", index=False)
   val_df.to_csv(f"{name}_val.csv", index=False)
   test_df.to_csv(f"{name}_test.csv", index=False)


csv_files = ['ModifiedDatasetCORR.csv', 'ModifiedDatasetGAINRATIO.csv',
'ModifiedDatasetINFOGAIN.csv', 'ModifiedDatasetONER.csv',
'ModifiedDatasetINDIVIDUAL.csv']
names = ["corr","gainratio","infogain","oneR","indi"]

for file_path,x in zip(csv_files, range(5)):
   stratified_sample_and_save(file_path,names[x])
```

When looking at the class distribution for each training set, we see that the distribution was preserved. For instance, when looking at the Correlation Training set, 30.3472% of the set was labeled AboveAvgWalkable, 33.8840% was labeled BelowAvgWalkable, 24.1577% was labeled LeastWalkable, and 11.6109% was labeled MostWalkable. When we now look at the initial distribution of the entire data set we see that they had a 30.3474% of AboveAvgWalkable, 33.8837% of BelowAvgWalkable, 23.1578% of LeastWalkable, and 11.6109% of MostWalkable. Comparing these values we see that they are very similar showing that the distribution was preserved, and although we only showed the values for the Correlation Training Set, the rest of the data sets also had the same proportions

As our attribute selection process comes to an end, we'll have the desired train/validation/test split data in order for the next part of our project, which will be to run the classification models and attempt to attain the highest accuracy for the models based on the given attribute selection algorithms and classification model combinations.

Now that we have the final dataset after preprocessing and applying various attribute selection algorithms, we'll need classification models to train, validate, and test on the given datasets. For this project, we'll use four classification models, namely the models below:

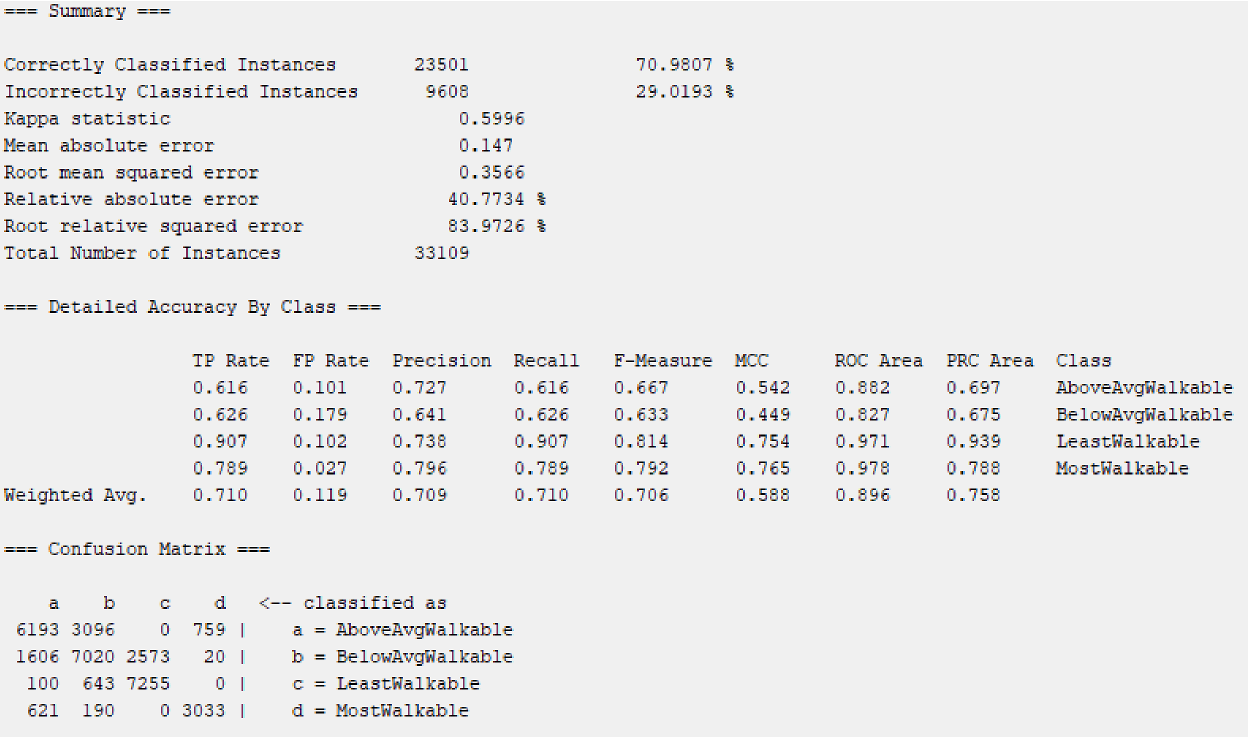
1) NaiveBayes (Bayes)
2) OneR (Rules)
3) J48 (Trees)
4) RandomForest (Trees)

While we have not learned the intricacies of any of these models (with the exception of OneR, which is a one-rule algorithm that independently chooses the best straight forward rule given each instances' chosen attribute option based on each attribute), we have used the four of these models in our ML labs, and given our preprocessing and attribute selection algorithm success, we hope to achieve 95%+ accuracy for the majority of our 20 models, if not at least one model.

Part 6: Results and Analysis

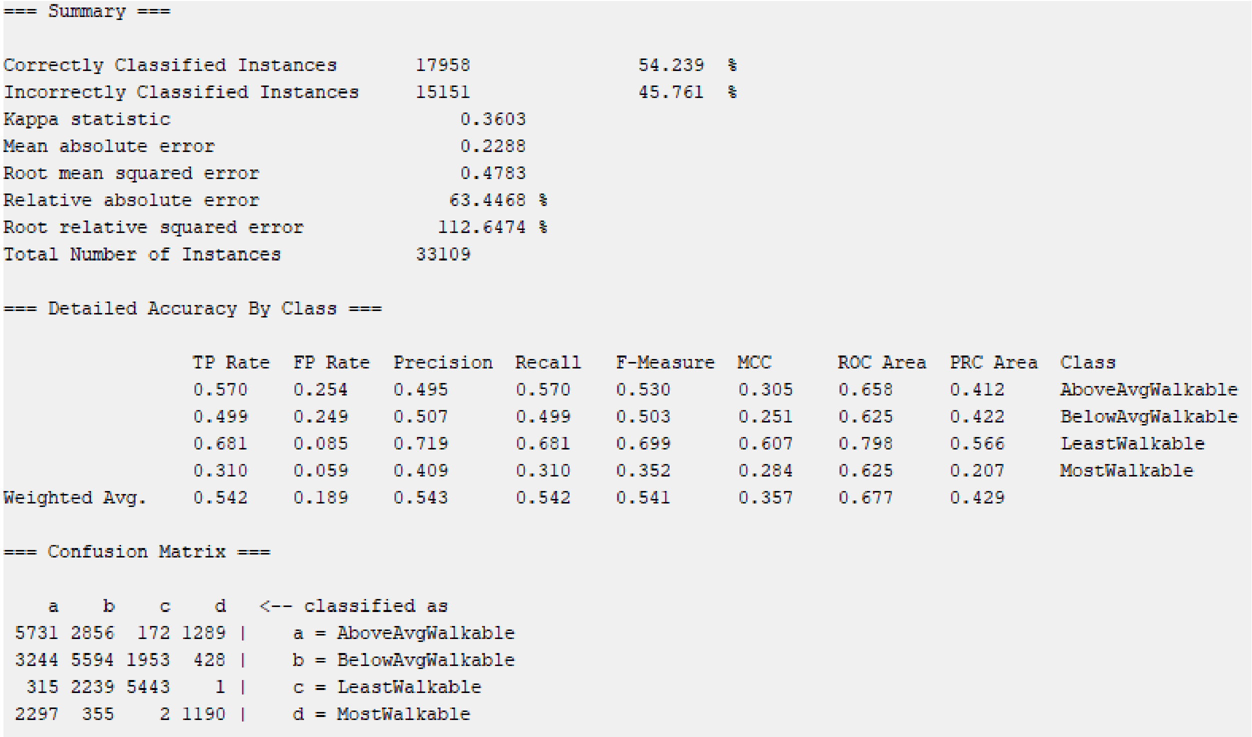
Here are the results of each of our twenty models with each image containing the summary statistics, detailed accuracy by class, and the confusion matrix of the given testing data results.

CorrelationAttributeEval with Naive Bayes:



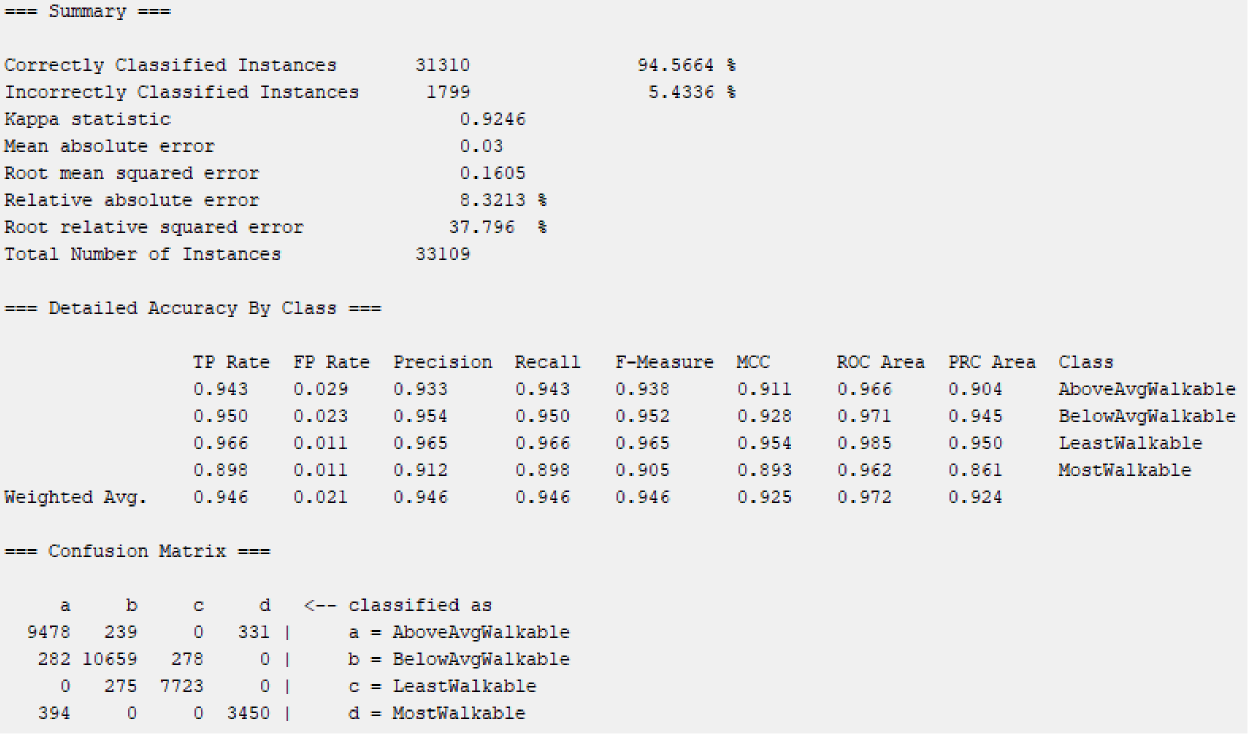
```
=== Summary ===

Correctly Classified Instances       23501               70.9807 %
Incorrectly Classified Instances      9608               29.0193 %
Kappa statistic                          0.5996
Mean absolute error                      0.147
Root mean squared error                  0.3566
Relative absolute error                 40.7734 %
Root relative squared error             83.9726 %
Total Number of Instances            33109

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.616    0.101    0.727      0.616    0.667      0.542    0.882     0.697     AboveAvgWalkable
                 0.626    0.179    0.641      0.626    0.633      0.449    0.827     0.675     BelowAvgWalkable
                 0.907    0.102    0.738      0.907    0.814      0.754    0.971     0.939     LeastWalkable
                 0.789    0.027    0.796      0.789    0.792      0.765    0.978     0.788     MostWalkable
Weighted Avg.    0.710    0.119    0.709      0.710    0.706      0.588    0.896     0.758

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 6193 3096    0  759 |    a = AboveAvgWalkable
 1606 7020 2573   20 |    b = BelowAvgWalkable
  100  643 7255    0 |    c = LeastWalkable
  621  190    0 3033 |    d = MostWalkable
```

CorrelationAttributeEval with OneR:



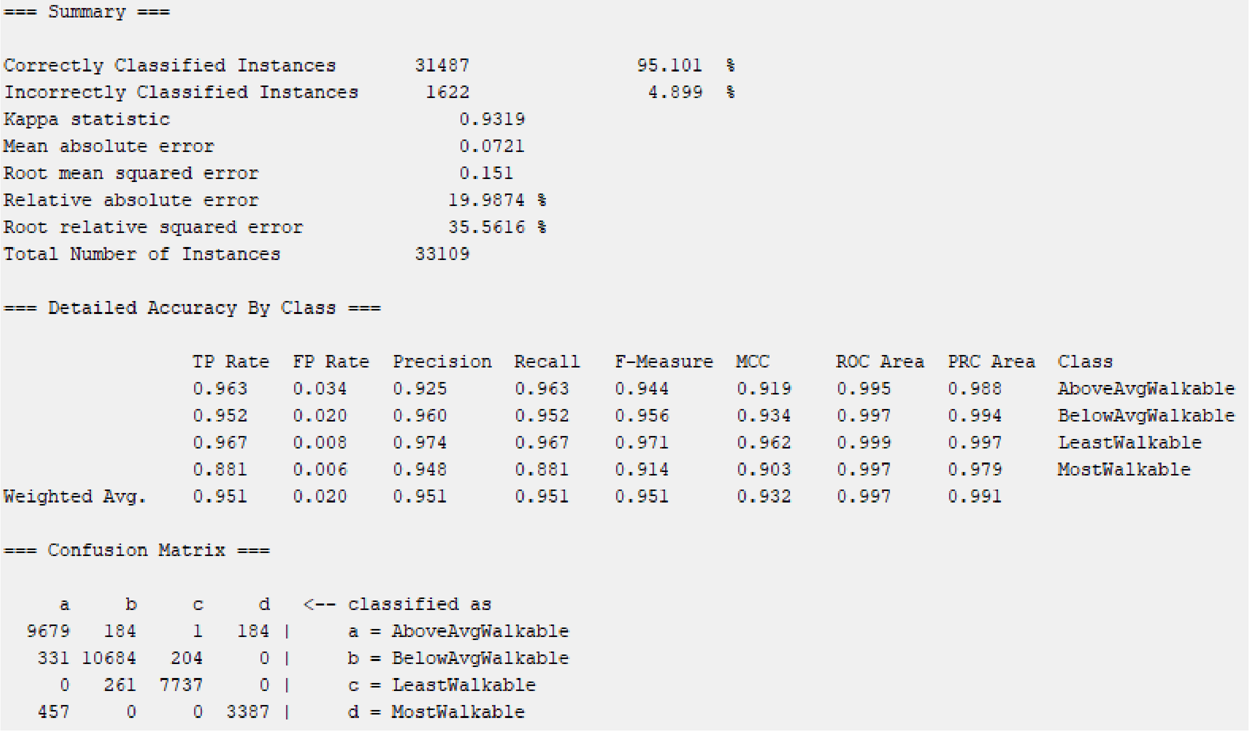
```
=== Summary ===

Correctly Classified Instances       17958               54.239  %
Incorrectly Classified Instances     15151               45.761  %
Kappa statistic                          0.3603
Mean absolute error                      0.2288
Root mean squared error                  0.4783
Relative absolute error                 63.4468 %
Root relative squared error            112.6474 %
Total Number of Instances            33109

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.570    0.254    0.495      0.570    0.530      0.305    0.658     0.412     AboveAvgWalkable
                 0.499    0.249    0.507      0.499    0.503      0.251    0.625     0.422     BelowAvgWalkable
                 0.681    0.085    0.719      0.681    0.699      0.607    0.798     0.566     LeastWalkable
                 0.310    0.059    0.409      0.310    0.352      0.284    0.625     0.207     MostWalkable
Weighted Avg.    0.542    0.189    0.543      0.542    0.541      0.357    0.677     0.429

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 5731 2856  172 1289 |    a = AboveAvgWalkable
 3244 5594 1953  428 |    b = BelowAvgWalkable
  315 2239 5443    1 |    c = LeastWalkable
 2297  355    2 1190 |    d = MostWalkable
```

CorrelationAttributeEval with J48:



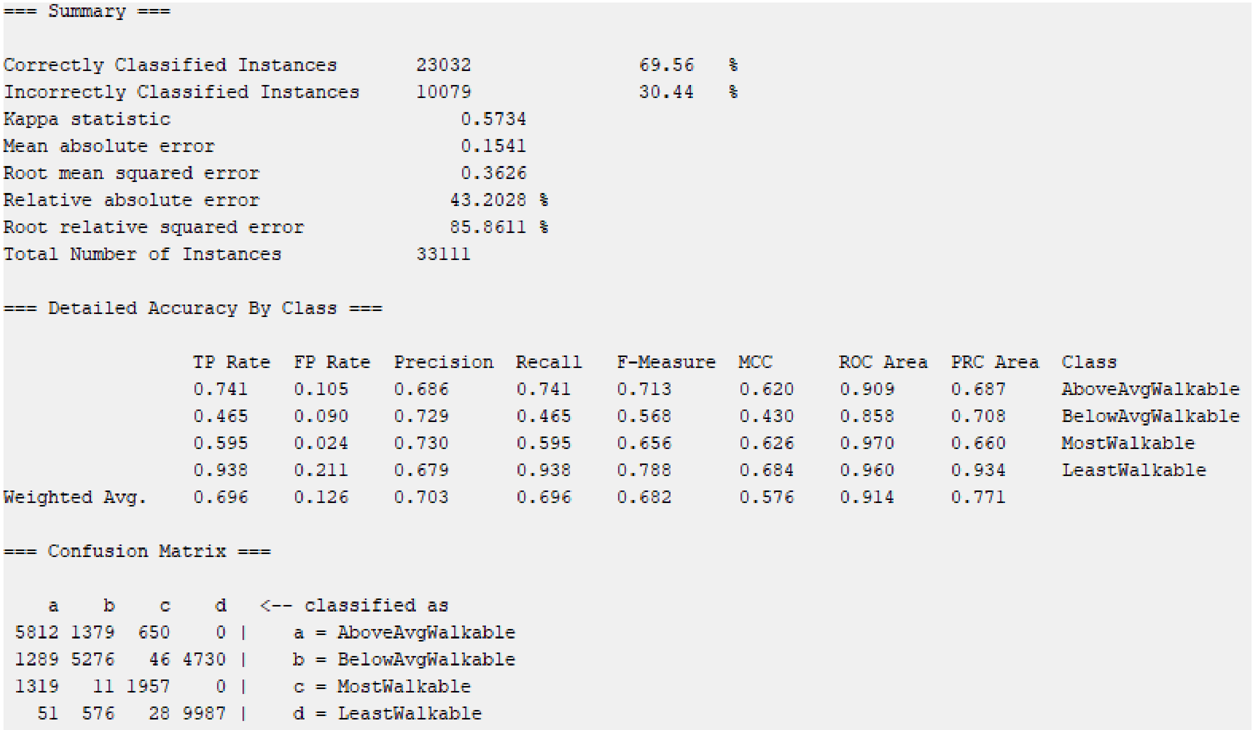
```
=== Summary ===

Correctly Classified Instances       31310               94.5664 %
Incorrectly Classified Instances      1799                5.4336 %
Kappa statistic                          0.9246
Mean absolute error                      0.03
Root mean squared error                  0.1605
Relative absolute error                  8.3213 %
Root relative squared error             37.796  %
Total Number of Instances            33109

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.943    0.029    0.933      0.943    0.938      0.911    0.966     0.904     AboveAvgWalkable
                 0.950    0.023    0.954      0.950    0.952      0.928    0.971     0.945     BelowAvgWalkable
                 0.966    0.011    0.965      0.966    0.965      0.954    0.985     0.950     LeastWalkable
                 0.898    0.011    0.912      0.898    0.905      0.893    0.962     0.861     MostWalkable
Weighted Avg.    0.946    0.021    0.946      0.946    0.946      0.925    0.972     0.924

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9478   239     0   331 |     a = AboveAvgWalkable
   282 10659   278     0 |     b = BelowAvgWalkable
     0   275  7723     0 |     c = LeastWalkable
   394     0     0  3450 |     d = MostWalkable
```

CorrelationAttributeEval with RandomForest:



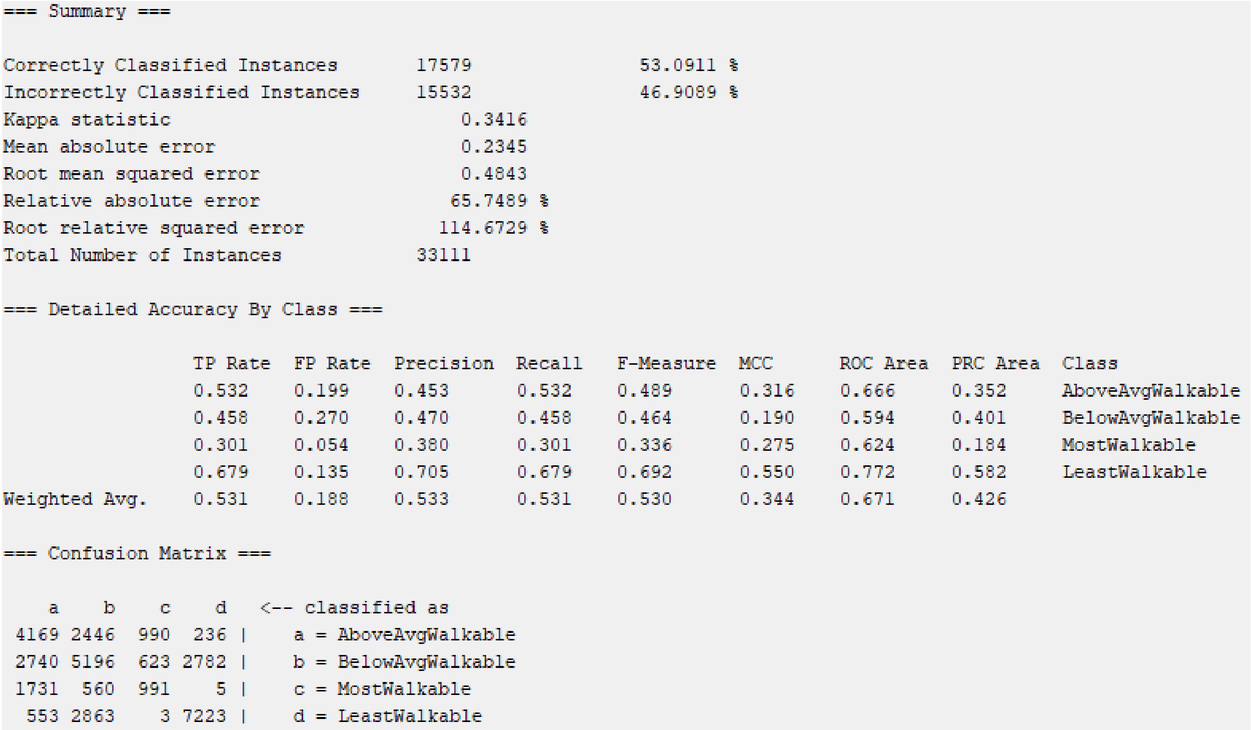
```
=== Summary ===

Correctly Classified Instances       31487               95.101  %
Incorrectly Classified Instances      1622                4.899  %
Kappa statistic                          0.9319
Mean absolute error                      0.0721
Root mean squared error                  0.151
Relative absolute error                 19.9874 %
Root relative squared error             35.5616 %
Total Number of Instances            33109

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.963    0.034    0.925      0.963    0.944      0.919    0.995     0.988     AboveAvgWalkable
                 0.952    0.020    0.960      0.952    0.956      0.934    0.997     0.994     BelowAvgWalkable
                 0.967    0.008    0.974      0.967    0.971      0.962    0.999     0.997     LeastWalkable
                 0.881    0.006    0.948      0.881    0.914      0.903    0.997     0.979     MostWalkable
Weighted Avg.    0.951    0.020    0.951      0.951    0.951      0.932    0.997     0.991

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9679   184     1   184 |     a = AboveAvgWalkable
   331 10684   204     0 |     b = BelowAvgWalkable
     0   261  7737     0 |     c = LeastWalkable
   457     0     0  3387 |     d = MostWalkable
```

InfoGainAttributeEval with Naive Bayes:



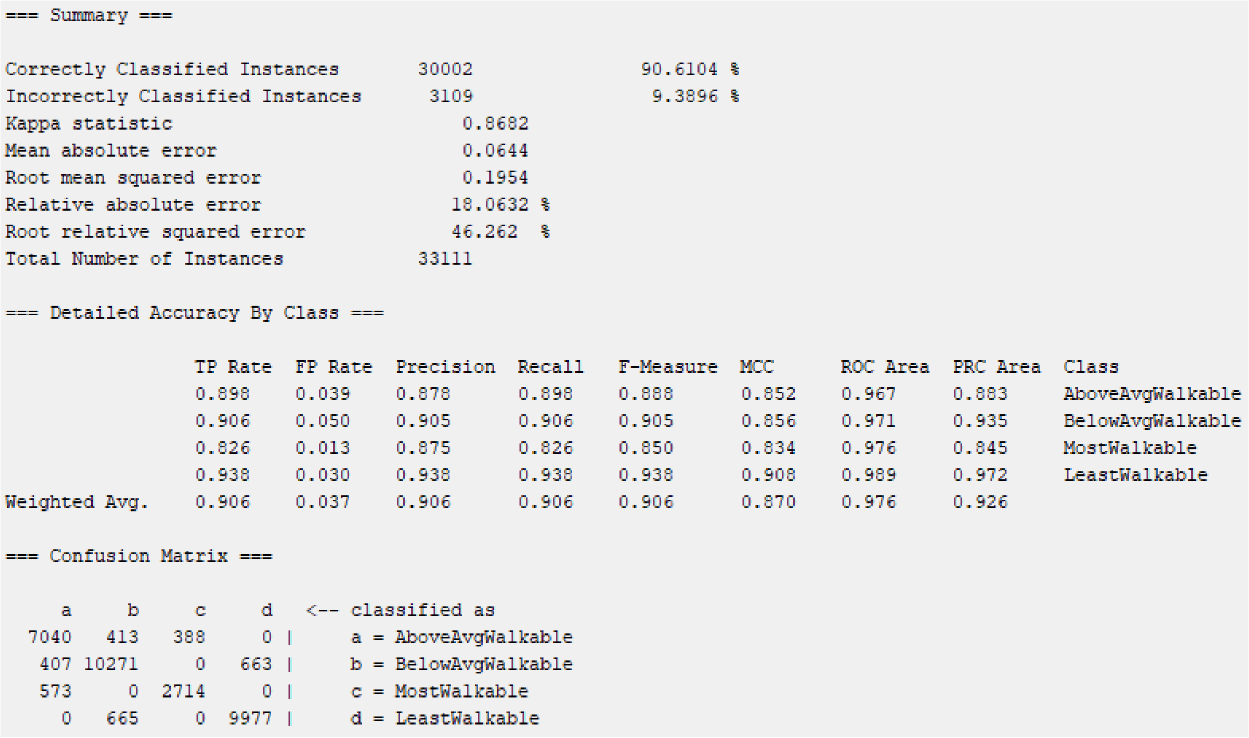
```
=== Summary ===

Correctly Classified Instances       23032               69.56   %
Incorrectly Classified Instances     10079               30.44   %
Kappa statistic                          0.5734
Mean absolute error                      0.1541
Root mean squared error                  0.3626
Relative absolute error                 43.2028 %
Root relative squared error             85.8611 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.741    0.105    0.686      0.741    0.713      0.620    0.909     0.687     AboveAvgWalkable
                 0.465    0.090    0.729      0.465    0.568      0.430    0.858     0.708     BelowAvgWalkable
                 0.595    0.024    0.730      0.595    0.656      0.626    0.970     0.660     MostWalkable
                 0.938    0.211    0.679      0.938    0.788      0.684    0.960     0.934     LeastWalkable
Weighted Avg.    0.696    0.126    0.703      0.696    0.682      0.576    0.914     0.771

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 5812 1379  650    0 |    a = AboveAvgWalkable
 1289 5276   46 4730 |    b = BelowAvgWalkable
 1319   11 1957    0 |    c = MostWalkable
   51  576   28 9987 |    d = LeastWalkable
```

InfoGainAttributeEval with OneR:



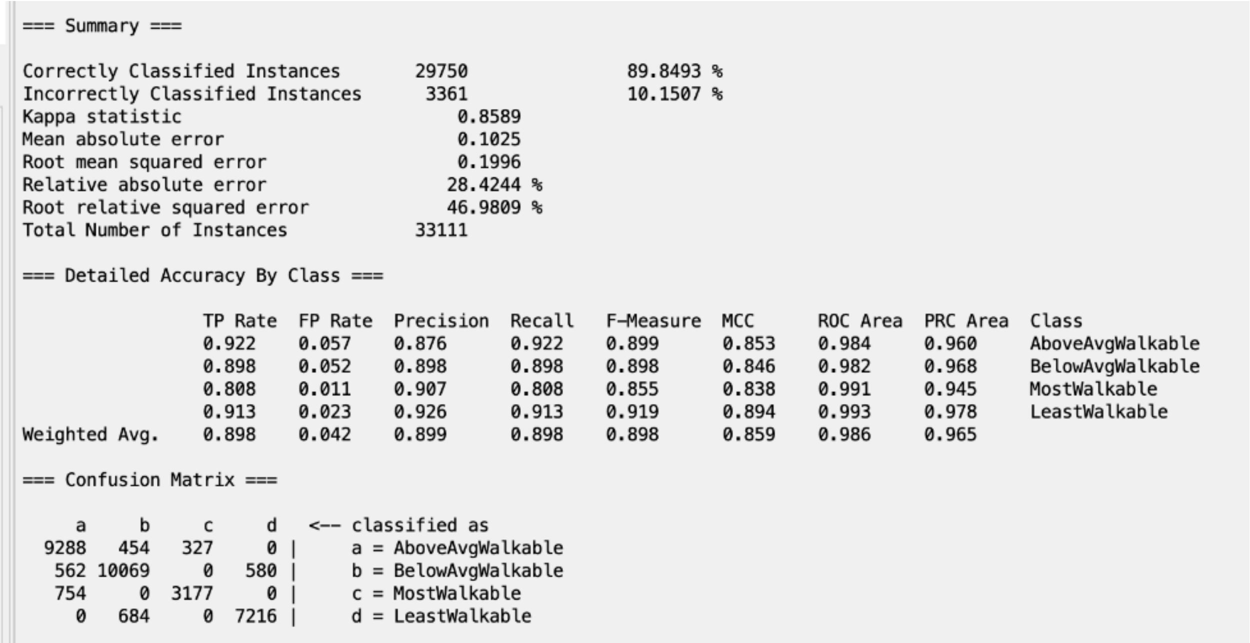
```
=== Summary ===

Correctly Classified Instances       17579               53.0911 %
Incorrectly Classified Instances     15532               46.9089 %
Kappa statistic                          0.3416
Mean absolute error                      0.2345
Root mean squared error                  0.4843
Relative absolute error                 65.7489 %
Root relative squared error            114.6729 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.532    0.199    0.453      0.532    0.489      0.316    0.666     0.352     AboveAvgWalkable
                 0.458    0.270    0.470      0.458    0.464      0.190    0.594     0.401     BelowAvgWalkable
                 0.301    0.054    0.380      0.301    0.336      0.275    0.624     0.184     MostWalkable
                 0.679    0.135    0.705      0.679    0.692      0.550    0.772     0.582     LeastWalkable
Weighted Avg.    0.531    0.188    0.533      0.531    0.530      0.344    0.671     0.426

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 4169 2446  990  236 |    a = AboveAvgWalkable
 2740 5196  623 2782 |    b = BelowAvgWalkable
 1731  560  991    5 |    c = MostWalkable
  553 2863    3 7223 |    d = LeastWalkable
```

InfoGainAttributeEval with J48:



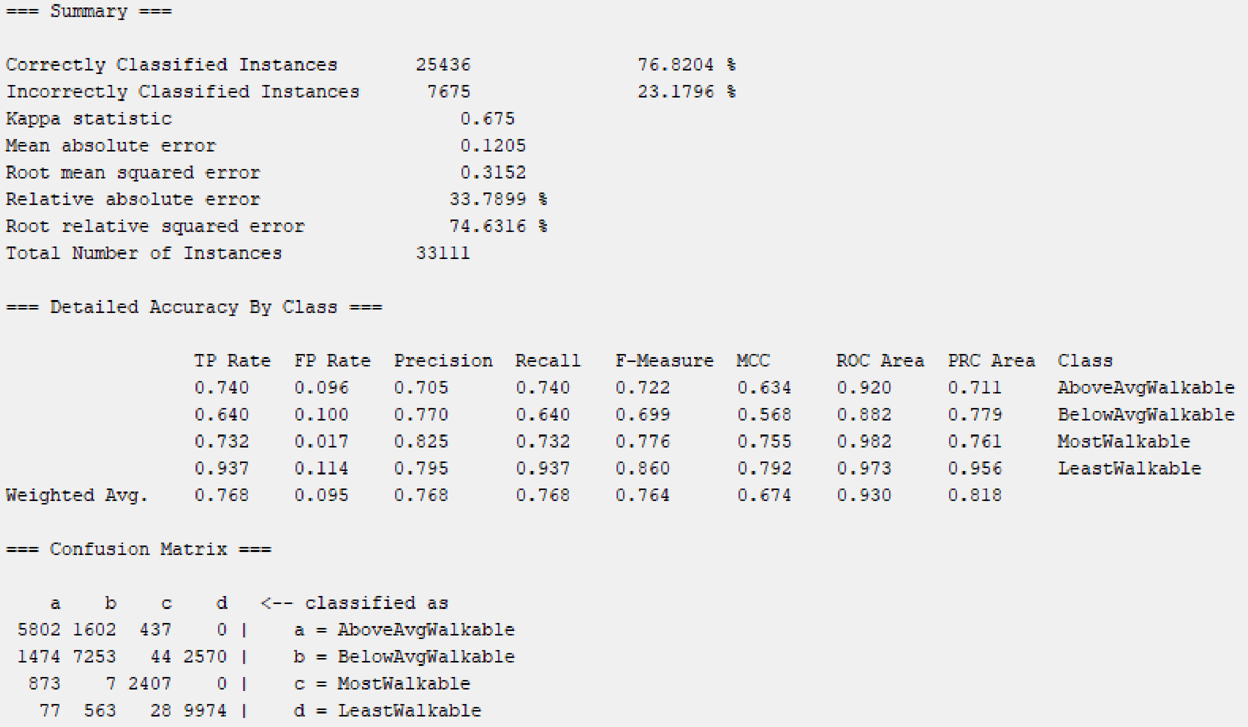
```
=== Summary ===

Correctly Classified Instances       30002               90.6104 %
Incorrectly Classified Instances      3109                9.3896 %
Kappa statistic                          0.8682
Mean absolute error                      0.0644
Root mean squared error                  0.1954
Relative absolute error                 18.0632 %
Root relative squared error             46.262  %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.898    0.039    0.878      0.898    0.888      0.852    0.967     0.883     AboveAvgWalkable
                 0.906    0.050    0.905      0.906    0.905      0.856    0.971     0.935     BelowAvgWalkable
                 0.826    0.013    0.875      0.826    0.850      0.834    0.976     0.845     MostWalkable
                 0.938    0.030    0.938      0.938    0.938      0.908    0.989     0.972     LeastWalkable
Weighted Avg.    0.906    0.037    0.906      0.906    0.906      0.870    0.976     0.926

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  7040   413   388     0 |     a = AboveAvgWalkable
   407 10271     0   663 |     b = BelowAvgWalkable
   573     0  2714     0 |     c = MostWalkable
     0   665     0  9977 |     d = LeastWalkable
```

InfoGainAttributeEval with RandomForest:



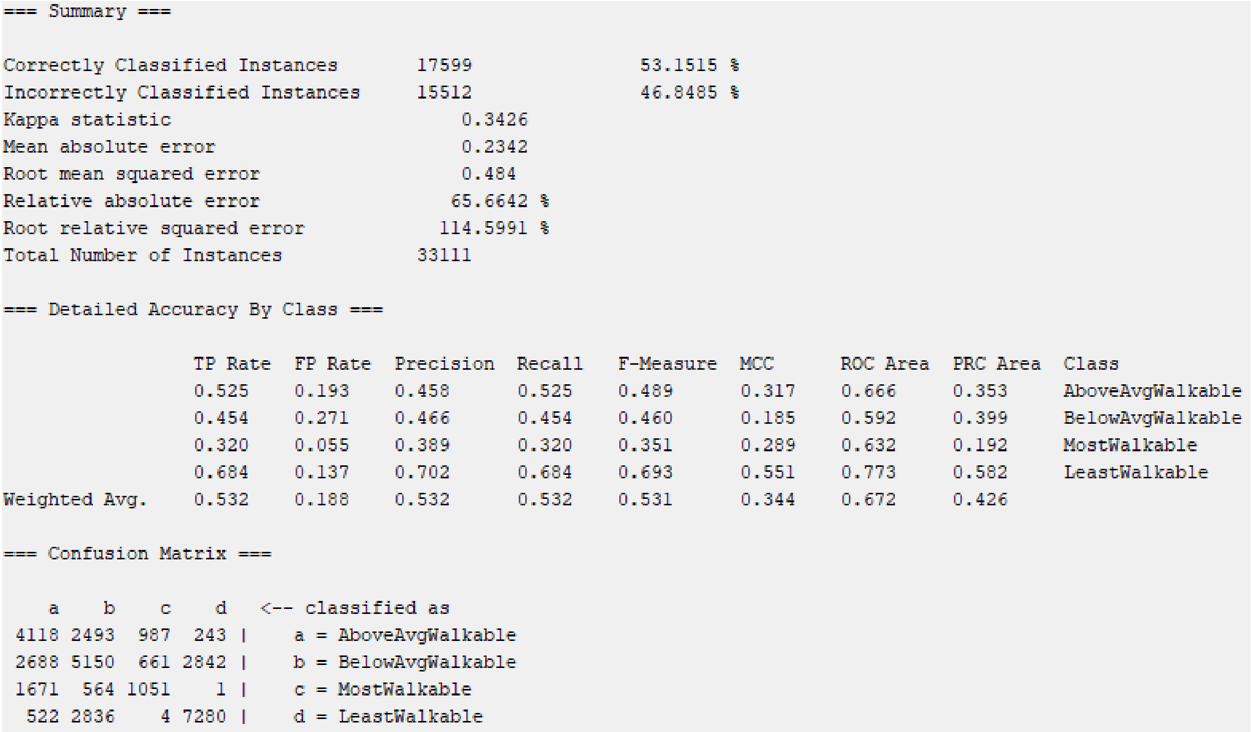
```
=== Summary ===

Correctly Classified Instances       29750               89.8493 %
Incorrectly Classified Instances      3361               10.1507 %
Kappa statistic                          0.8589
Mean absolute error                      0.1025
Root mean squared error                  0.1996
Relative absolute error                 28.4244 %
Root relative squared error             46.9809 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.922    0.057    0.876      0.922    0.899      0.853    0.984     0.960     AboveAvgWalkable
                 0.898    0.052    0.898      0.898    0.898      0.846    0.982     0.968     BelowAvgWalkable
                 0.808    0.011    0.907      0.808    0.855      0.838    0.991     0.945     MostWalkable
                 0.913    0.023    0.926      0.913    0.919      0.894    0.993     0.978     LeastWalkable
Weighted Avg.    0.898    0.042    0.899      0.898    0.898      0.859    0.986     0.965

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9288   454   327     0 |     a = AboveAvgWalkable
   562 10069     0   580 |     b = BelowAvgWalkable
   754     0  3177     0 |     c = MostWalkable
     0   684     0  7216 |     d = LeastWalkable
```

GainRatioAttributeEval with Naive Bayes:



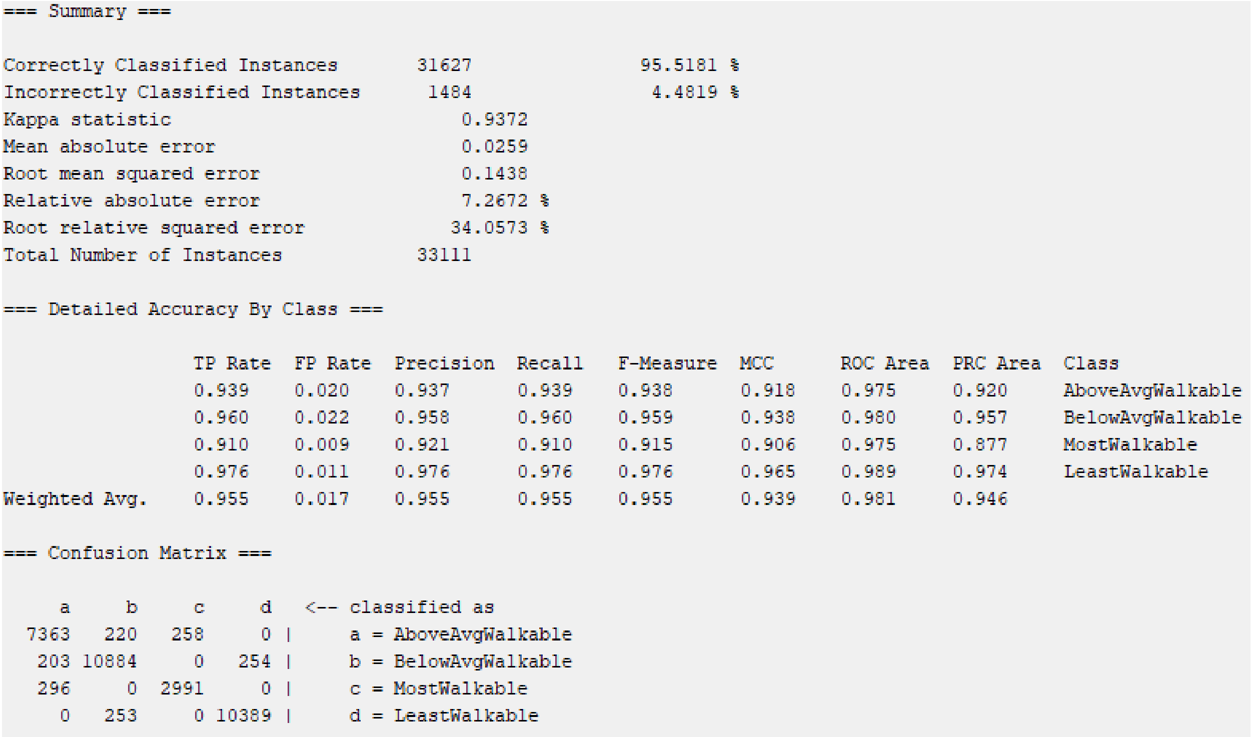
```
=== Summary ===

Correctly Classified Instances       25436               76.8204 %
Incorrectly Classified Instances      7675               23.1796 %
Kappa statistic                          0.675
Mean absolute error                      0.1205
Root mean squared error                  0.3152
Relative absolute error                 33.7899 %
Root relative squared error             74.6316 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.740    0.096    0.705      0.740    0.722      0.634    0.920     0.711     AboveAvgWalkable
                 0.640    0.100    0.770      0.640    0.699      0.568    0.882     0.779     BelowAvgWalkable
                 0.732    0.017    0.825      0.732    0.776      0.755    0.982     0.761     MostWalkable
                 0.937    0.114    0.795      0.937    0.860      0.792    0.973     0.956     LeastWalkable
Weighted Avg.    0.768    0.095    0.768      0.768    0.764      0.674    0.930     0.818

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 5802 1602  437    0 |    a = AboveAvgWalkable
 1474 7253   44 2570 |    b = BelowAvgWalkable
  873    7 2407    0 |    c = MostWalkable
   77  563   28 9974 |    d = LeastWalkable
```

GainRatioAttributeEval with OneR:



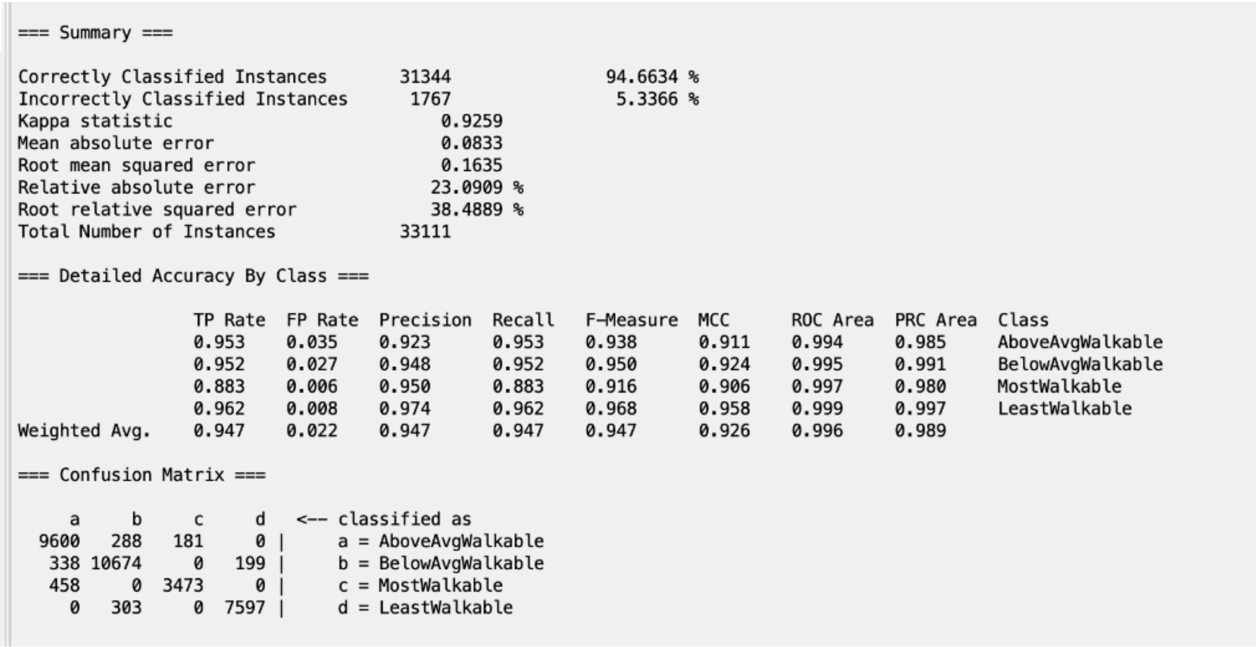
```
=== Summary ===

Correctly Classified Instances       17599               53.1515 %
Incorrectly Classified Instances     15512               46.8485 %
Kappa statistic                          0.3426
Mean absolute error                      0.2342
Root mean squared error                  0.484
Relative absolute error                 65.6642 %
Root relative squared error            114.5991 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.525    0.193    0.458      0.525    0.489      0.317    0.666     0.353     AboveAvgWalkable
                 0.454    0.271    0.466      0.454    0.460      0.185    0.592     0.399     BelowAvgWalkable
                 0.320    0.055    0.389      0.320    0.351      0.289    0.632     0.192     MostWalkable
                 0.684    0.137    0.702      0.684    0.693      0.551    0.773     0.582     LeastWalkable
Weighted Avg.    0.532    0.188    0.532      0.532    0.531      0.344    0.672     0.426

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 4118 2493  987  243 |    a = AboveAvgWalkable
 2688 5150  661 2842 |    b = BelowAvgWalkable
 1671  564 1051    1 |    c = MostWalkable
  522 2836    4 7280 |    d = LeastWalkable
```

GainRatioAttributeEval with J48:



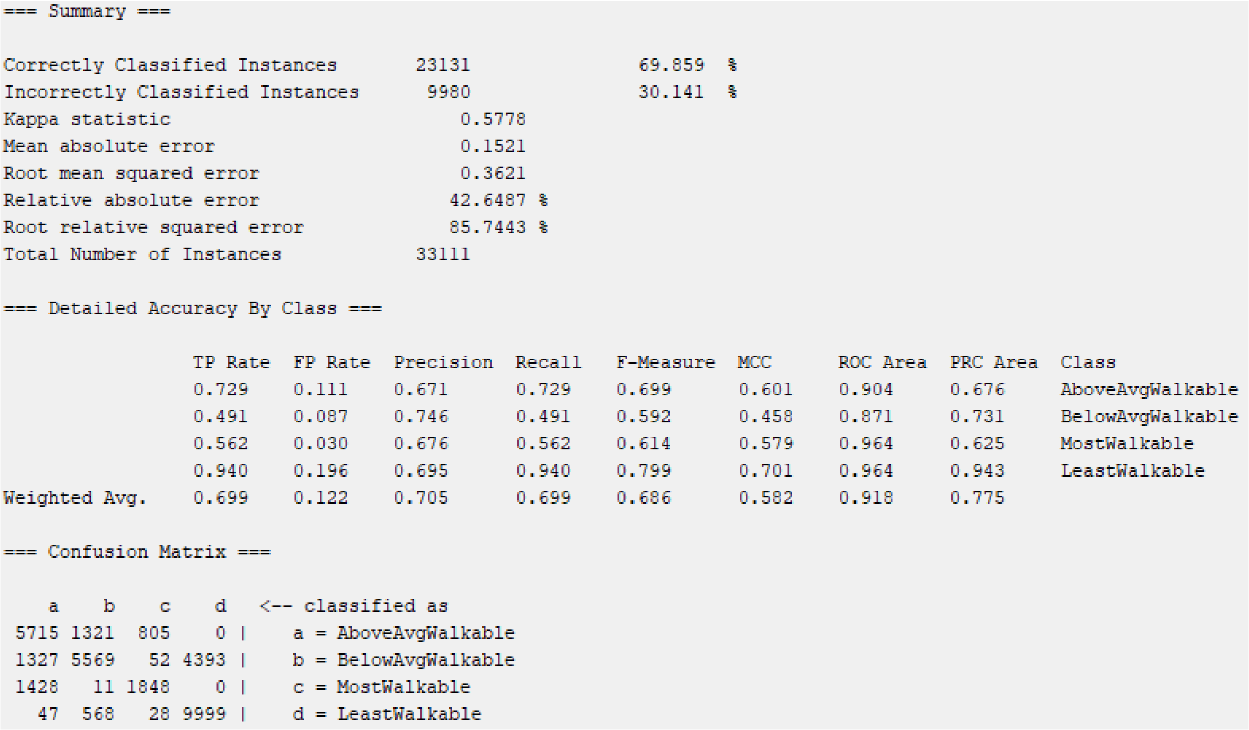
```
=== Summary ===

Correctly Classified Instances       31627               95.5181 %
Incorrectly Classified Instances      1484                4.4819 %
Kappa statistic                          0.9372
Mean absolute error                      0.0259
Root mean squared error                  0.1438
Relative absolute error                  7.2672 %
Root relative squared error             34.0573 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.939    0.020    0.937      0.939    0.938      0.918    0.975     0.920     AboveAvgWalkable
                 0.960    0.022    0.958      0.960    0.959      0.938    0.980     0.957     BelowAvgWalkable
                 0.910    0.009    0.921      0.910    0.915      0.906    0.975     0.877     MostWalkable
                 0.976    0.011    0.976      0.976    0.976      0.965    0.989     0.974     LeastWalkable
Weighted Avg.    0.955    0.017    0.955      0.955    0.955      0.939    0.981     0.946

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  7363   220   258     0 |     a = AboveAvgWalkable
   203 10884     0   254 |     b = BelowAvgWalkable
   296     0  2991     0 |     c = MostWalkable
     0   253     0 10389 |     d = LeastWalkable
```

GainRatioAttributeEval with RandomForest:



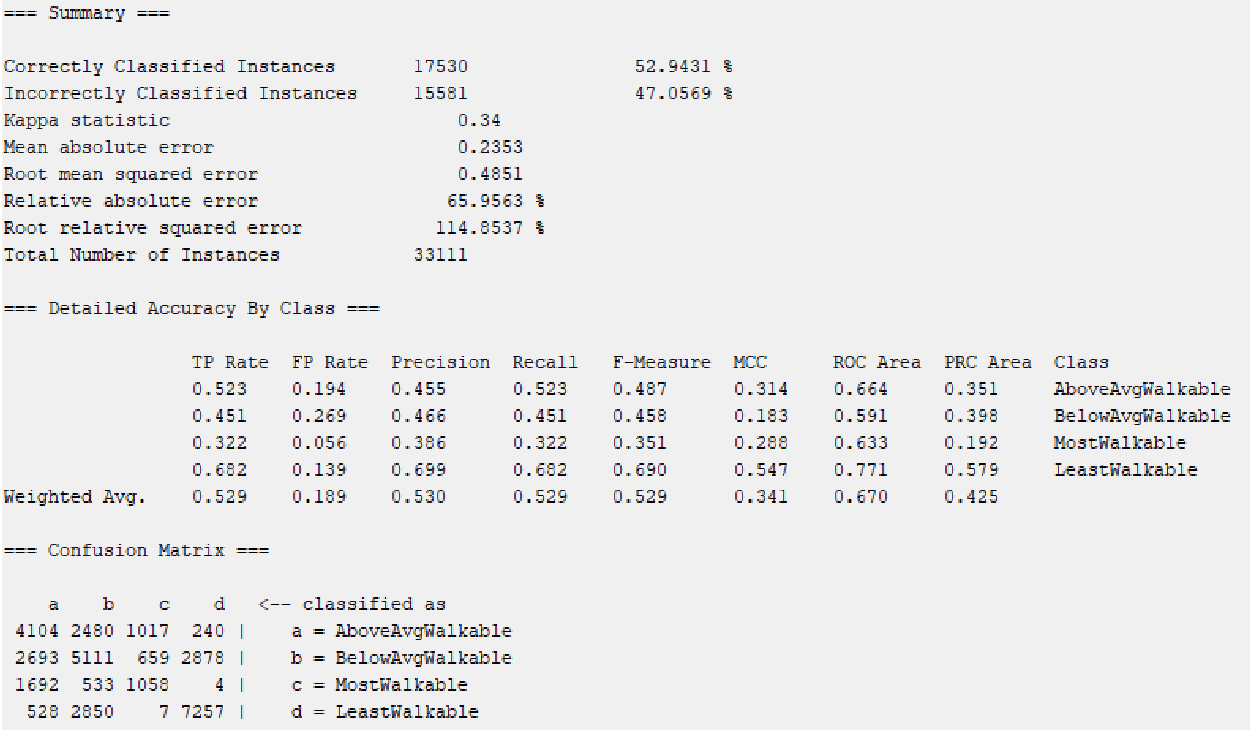
```
=== Summary ===

Correctly Classified Instances       31344               94.6634 %
Incorrectly Classified Instances      1767                5.3366 %
Kappa statistic                          0.9259
Mean absolute error                      0.0833
Root mean squared error                  0.1635
Relative absolute error                 23.0909 %
Root relative squared error             38.4889 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.953    0.035    0.923      0.953    0.938      0.911    0.994     0.985     AboveAvgWalkable
                 0.952    0.027    0.948      0.952    0.950      0.924    0.995     0.991     BelowAvgWalkable
                 0.883    0.006    0.950      0.883    0.916      0.906    0.997     0.980     MostWalkable
                 0.962    0.008    0.974      0.962    0.968      0.958    0.999     0.997     LeastWalkable
Weighted Avg.    0.947    0.022    0.947      0.947    0.947      0.926    0.996     0.989

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9600   288   181     0 |     a = AboveAvgWalkable
   338 10674     0   199 |     b = BelowAvgWalkable
   458     0  3473     0 |     c = MostWalkable
     0   303     0  7597 |     d = LeastWalkable
```

OneRAttributeEval with Naive Bayes:



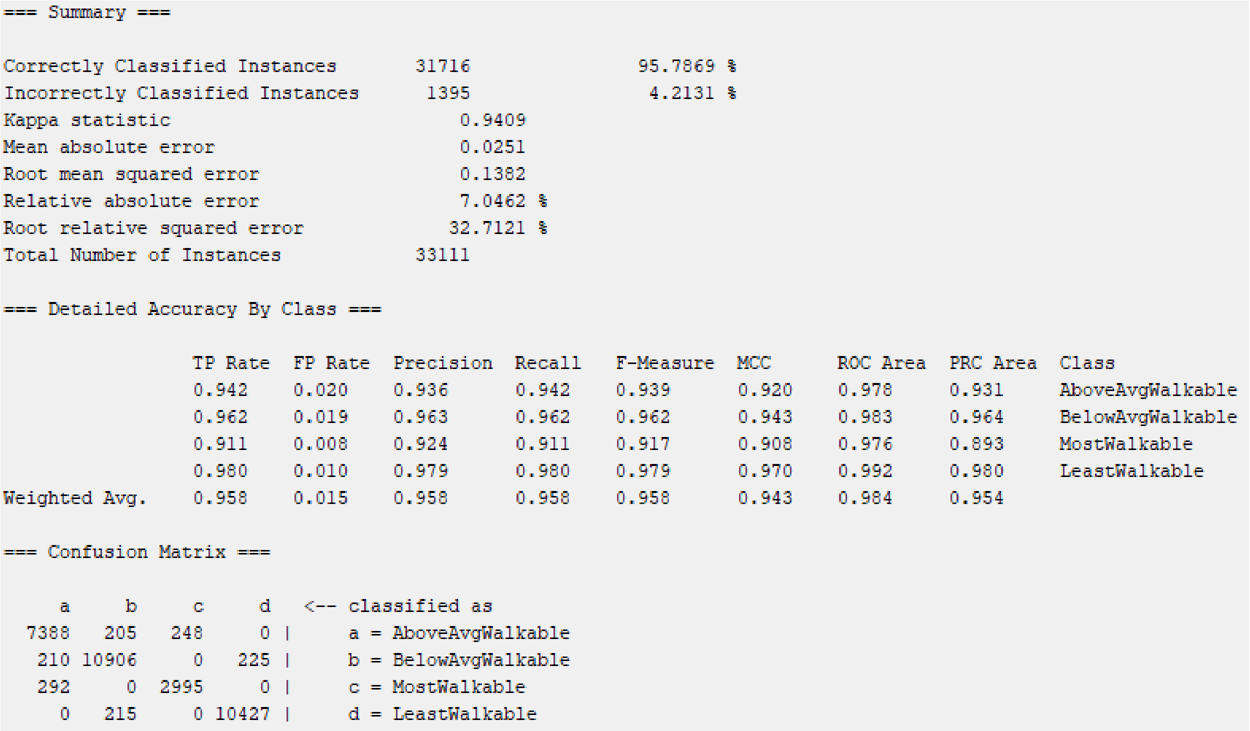
```
=== Summary ===

Correctly Classified Instances       23131               69.859  %
Incorrectly Classified Instances      9980               30.141  %
Kappa statistic                          0.5778
Mean absolute error                      0.1521
Root mean squared error                  0.3621
Relative absolute error                 42.6487 %
Root relative squared error             85.7443 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.729    0.111    0.671      0.729    0.699      0.601    0.904     0.676     AboveAvgWalkable
                 0.491    0.087    0.746      0.491    0.592      0.458    0.871     0.731     BelowAvgWalkable
                 0.562    0.030    0.676      0.562    0.614      0.579    0.964     0.625     MostWalkable
                 0.940    0.196    0.695      0.940    0.799      0.701    0.964     0.943     LeastWalkable
Weighted Avg.    0.699    0.122    0.705      0.699    0.686      0.582    0.918     0.775

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 5715 1321  805    0 |    a = AboveAvgWalkable
 1327 5569   52 4393 |    b = BelowAvgWalkable
 1428   11 1848    0 |    c = MostWalkable
   47  568   28 9999 |    d = LeastWalkable
```

OneRAttributeEval with OneR:



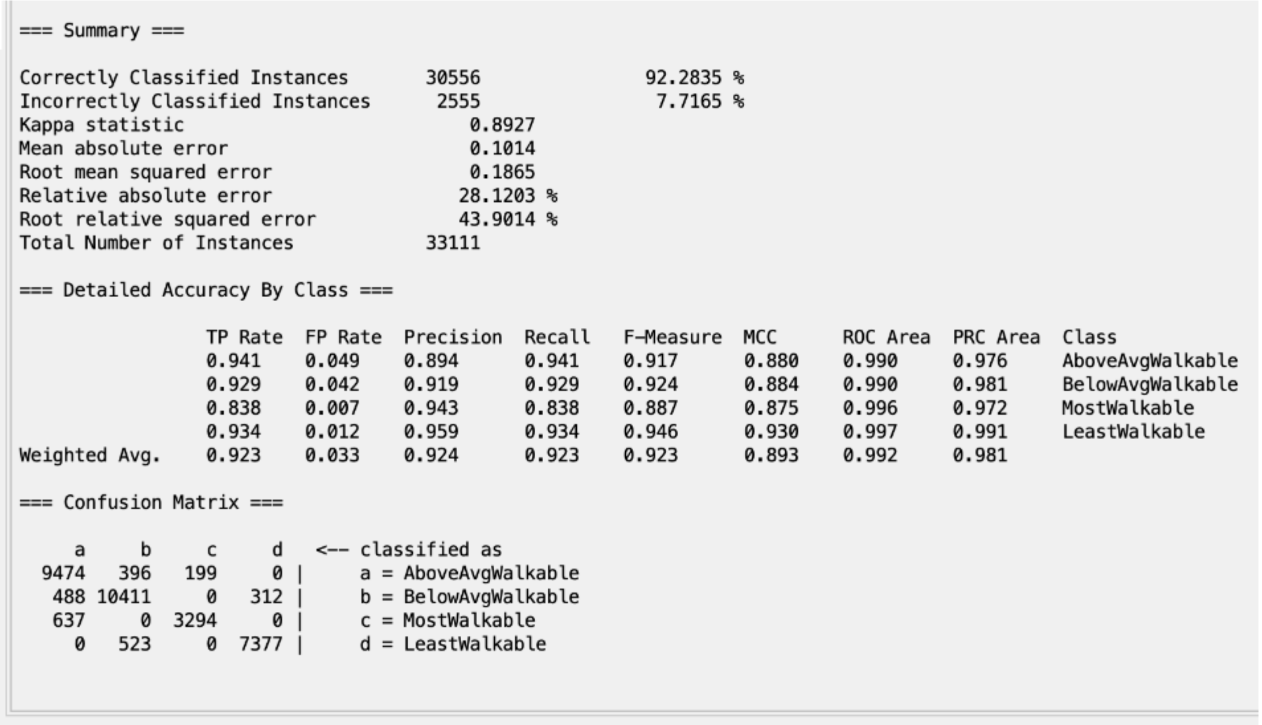
```
=== Summary ===

Correctly Classified Instances       17530               52.9431 %
Incorrectly Classified Instances     15581               47.0569 %
Kappa statistic                          0.34
Mean absolute error                      0.2353
Root mean squared error                  0.4851
Relative absolute error                 65.9563 %
Root relative squared error            114.8537 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.523    0.194    0.455      0.523    0.487      0.314    0.664     0.351     AboveAvgWalkable
                 0.451    0.269    0.466      0.451    0.458      0.183    0.591     0.398     BelowAvgWalkable
                 0.322    0.056    0.386      0.322    0.351      0.288    0.633     0.192     MostWalkable
                 0.682    0.139    0.699      0.682    0.690      0.547    0.771     0.579     LeastWalkable
Weighted Avg.    0.529    0.189    0.530      0.529    0.529      0.341    0.670     0.425

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 4104 2480 1017  240 |    a = AboveAvgWalkable
 2693 5111  659 2878 |    b = BelowAvgWalkable
 1692  533 1058    4 |    c = MostWalkable
  528 2850    7 7257 |    d = LeastWalkable
```

OneRAttributeEval with J48:



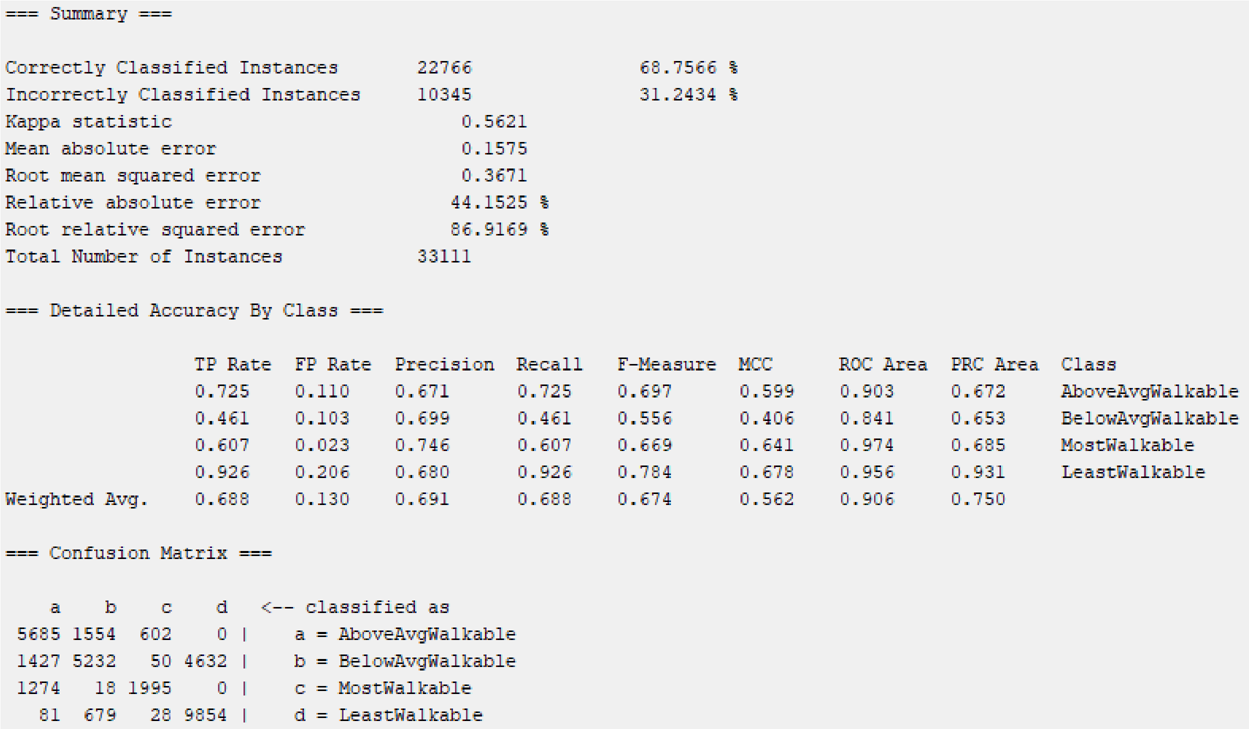
```
=== Summary ===

Correctly Classified Instances       31716               95.7869 %
Incorrectly Classified Instances      1395                4.2131 %
Kappa statistic                          0.9409
Mean absolute error                      0.0251
Root mean squared error                  0.1382
Relative absolute error                  7.0462 %
Root relative squared error             32.7121 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.942    0.020    0.936      0.942    0.939      0.920    0.978     0.931     AboveAvgWalkable
                 0.962    0.019    0.963      0.962    0.962      0.943    0.983     0.964     BelowAvgWalkable
                 0.911    0.008    0.924      0.911    0.917      0.908    0.976     0.893     MostWalkable
                 0.980    0.010    0.979      0.980    0.979      0.970    0.992     0.980     LeastWalkable
Weighted Avg.    0.958    0.015    0.958      0.958    0.958      0.943    0.984     0.954

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  7388   205   248     0 |     a = AboveAvgWalkable
   210 10906     0   225 |     b = BelowAvgWalkable
   292     0  2995     0 |     c = MostWalkable
     0   215     0 10427 |     d = LeastWalkable
```

OneRAttributeEval with RandomForest:



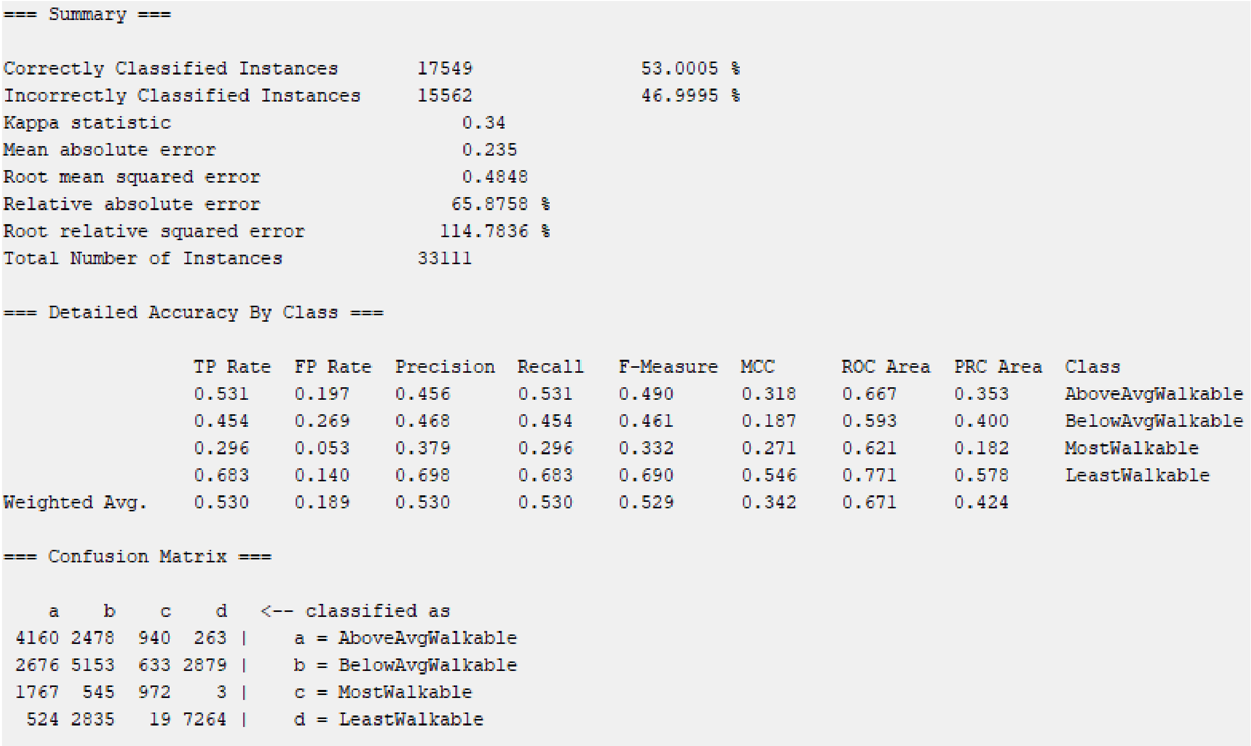
```
=== Summary ===

Correctly Classified Instances       30556               92.2835 %
Incorrectly Classified Instances      2555                7.7165 %
Kappa statistic                          0.8927
Mean absolute error                      0.1014
Root mean squared error                  0.1865
Relative absolute error                 28.1203 %
Root relative squared error             43.9014 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.941    0.049    0.894      0.941    0.917      0.880    0.990     0.976     AboveAvgWalkable
                 0.929    0.042    0.919      0.929    0.924      0.884    0.990     0.981     BelowAvgWalkable
                 0.838    0.007    0.943      0.838    0.887      0.875    0.996     0.972     MostWalkable
                 0.934    0.012    0.959      0.934    0.946      0.930    0.997     0.991     LeastWalkable
Weighted Avg.    0.923    0.033    0.924      0.923    0.923      0.893    0.992     0.981

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9474   396   199     0 |     a = AboveAvgWalkable
   488 10411     0   312 |     b = BelowAvgWalkable
   637     0  3294     0 |     c = MostWalkable
     0   523     0  7377 |     d = LeastWalkable
```

Team Choice “AttributeEval” with Naive Bayes:



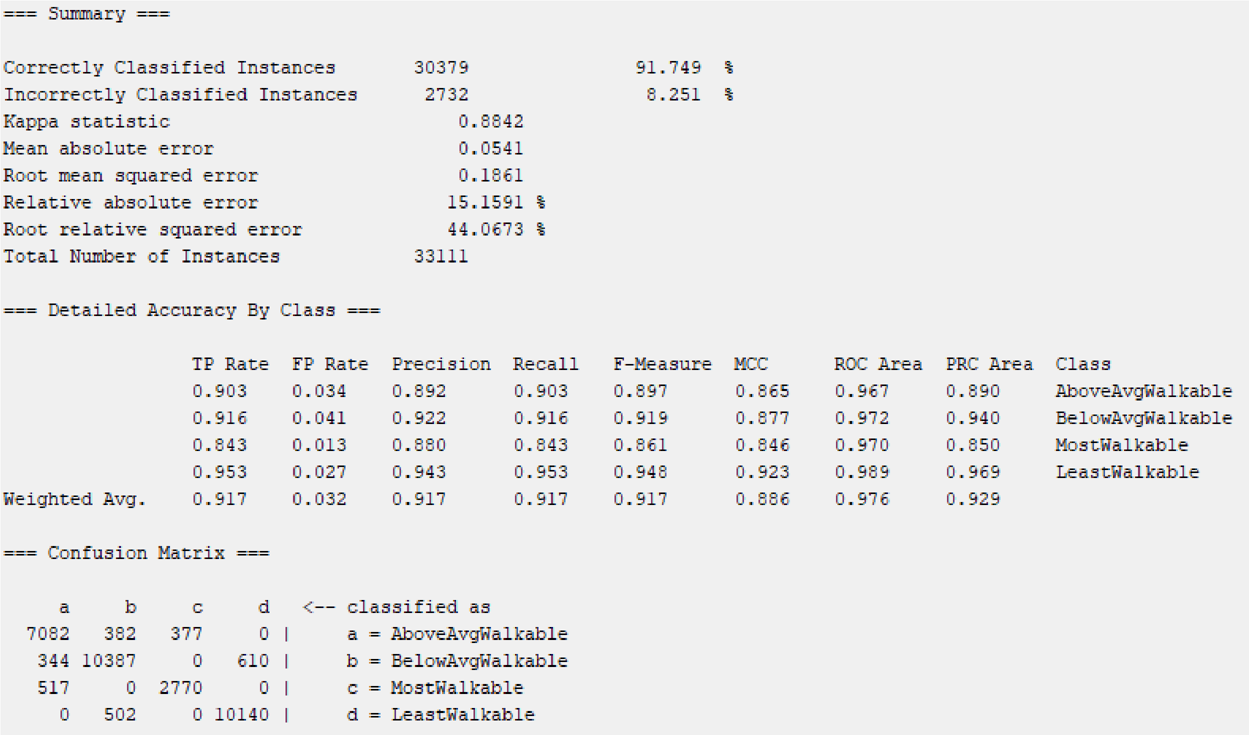
```
=== Summary ===

Correctly Classified Instances       22766               68.7566 %
Incorrectly Classified Instances     10345               31.2434 %
Kappa statistic                          0.5621
Mean absolute error                      0.1575
Root mean squared error                  0.3671
Relative absolute error                 44.1525 %
Root relative squared error             86.9169 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.725    0.110    0.671      0.725    0.697      0.599    0.903     0.672     AboveAvgWalkable
                 0.461    0.103    0.699      0.461    0.556      0.406    0.841     0.653     BelowAvgWalkable
                 0.607    0.023    0.746      0.607    0.669      0.641    0.974     0.685     MostWalkable
                 0.926    0.206    0.680      0.926    0.784      0.678    0.956     0.931     LeastWalkable
Weighted Avg.    0.688    0.130    0.691      0.688    0.674      0.562    0.906     0.750

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 5685 1554  602    0 |    a = AboveAvgWalkable
 1427 5232   50 4632 |    b = BelowAvgWalkable
 1274   18 1995    0 |    c = MostWalkable
   81  679   28 9854 |    d = LeastWalkable
```

Team Choice “AttributeEval” with OneR:



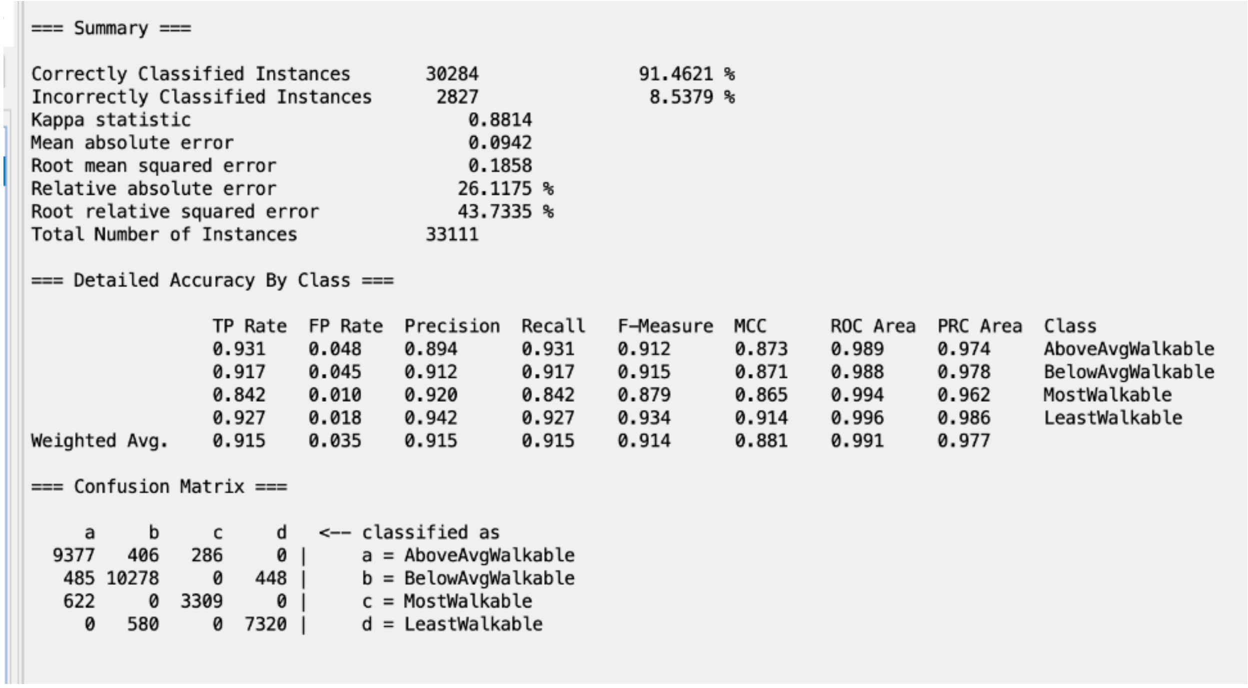
```
=== Summary ===

Correctly Classified Instances       17549               53.0005 %
Incorrectly Classified Instances     15562               46.9995 %
Kappa statistic                          0.34
Mean absolute error                      0.235
Root mean squared error                  0.4848
Relative absolute error                 65.8758 %
Root relative squared error            114.7836 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.531    0.197    0.456      0.531    0.490      0.318    0.667     0.353     AboveAvgWalkable
                 0.454    0.269    0.468      0.454    0.461      0.187    0.593     0.400     BelowAvgWalkable
                 0.296    0.053    0.379      0.296    0.332      0.271    0.621     0.182     MostWalkable
                 0.683    0.140    0.698      0.683    0.690      0.546    0.771     0.578     LeastWalkable
Weighted Avg.    0.530    0.189    0.530      0.530    0.529      0.342    0.671     0.424

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 4160 2478  940  263 |    a = AboveAvgWalkable
 2676 5153  633 2879 |    b = BelowAvgWalkable
 1767  545  972    3 |    c = MostWalkable
  524 2835   19 7264 |    d = LeastWalkable
```

Team Choice "AttributeEval" with J48:



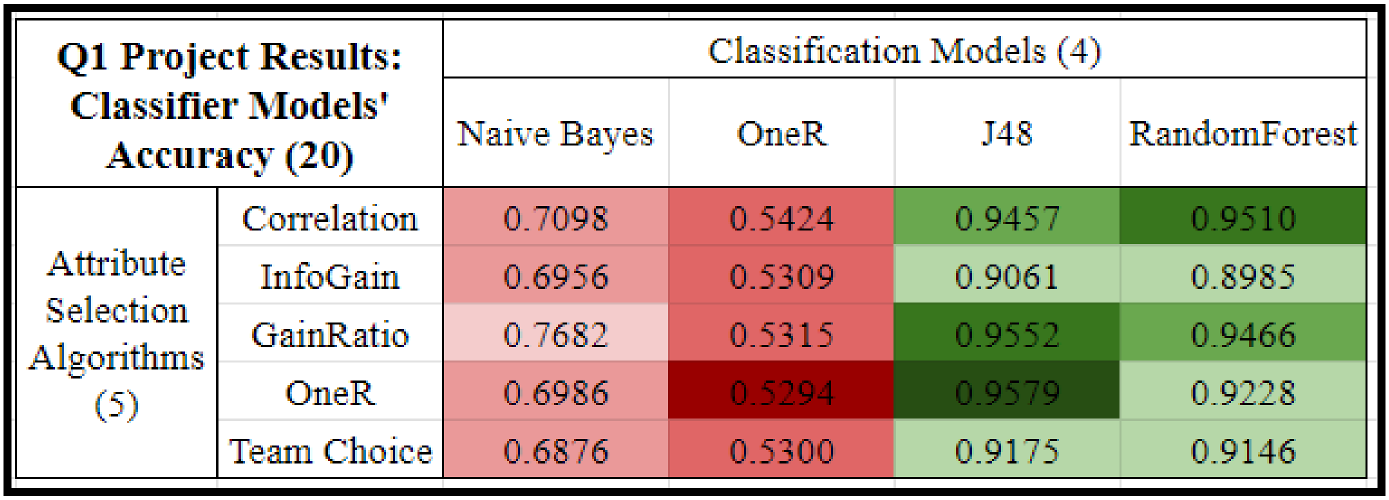
```
=== Summary ===

Correctly Classified Instances       30379               91.749  %
Incorrectly Classified Instances      2732                8.251  %
Kappa statistic                          0.8842
Mean absolute error                      0.0541
Root mean squared error                  0.1861
Relative absolute error                 15.1591 %
Root relative squared error             44.0673 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.903    0.034    0.892      0.903    0.897      0.865    0.967     0.890     AboveAvgWalkable
                 0.916    0.041    0.922      0.916    0.919      0.877    0.972     0.940     BelowAvgWalkable
                 0.843    0.013    0.880      0.843    0.861      0.846    0.970     0.850     MostWalkable
                 0.953    0.027    0.943      0.953    0.948      0.923    0.989     0.969     LeastWalkable
Weighted Avg.    0.917    0.032    0.917      0.917    0.917      0.886    0.976     0.929

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  7082   382   377     0 |     a = AboveAvgWalkable
   344 10387     0   610 |     b = BelowAvgWalkable
   517     0  2770     0 |     c = MostWalkable
     0   502     0 10140 |     d = LeastWalkable
```

Team Choice "AttributeEval" with RandomForest:



```
=== Summary ===

Correctly Classified Instances       30284               91.4621 %
Incorrectly Classified Instances      2827                8.5379 %
Kappa statistic                          0.8814
Mean absolute error                      0.0942
Root mean squared error                  0.1858
Relative absolute error                 26.1175 %
Root relative squared error             43.7335 %
Total Number of Instances            33111

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.931    0.048    0.894      0.931    0.912      0.873    0.989     0.974     AboveAvgWalkable
                 0.917    0.045    0.912      0.917    0.915      0.871    0.988     0.978     BelowAvgWalkable
                 0.842    0.010    0.920      0.842    0.879      0.865    0.994     0.962     MostWalkable
                 0.927    0.018    0.942      0.927    0.934      0.914    0.996     0.986     LeastWalkable
Weighted Avg.    0.915    0.035    0.915      0.915    0.914      0.881    0.991     0.977

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9377   406   286     0 |     a = AboveAvgWalkable
   485 10278     0   448 |     b = BelowAvgWalkable
   622     0  3309     0 |     c = MostWalkable
     0   580     0  7320 |     d = LeastWalkable
```

After running the 20 models (4 different models on the five different attribute selection algorithm-processed datasets) using the train/test/validation sets, we created a table to view each combination of attribute selection algorithm and classifier model result in a clean way below



Q1 Project Results: Classifier Models' Accuracy (20)		Classification Models (4)			
		Naive Bayes	OneR	J48	RandomForest
Attribute Selection Algorithms (5)	Correlation	0.7098	0.5424	0.9457	0.9510
	InfoGain	0.6956	0.5309	0.9061	0.8985
	GainRatio	0.7682	0.5315	0.9552	0.9466
	OneR	0.6986	0.5294	0.9579	0.9228
	Team Choice	0.6876	0.5300	0.9175	0.9146

After obtaining the results of all 20 models' performances, it's clear that our best model is one of three options: Correlation AttributeEval with RandomForest, GainRatio AttributeEval with J48, or OneR AttributeEval with J48 due to their blanket accuracy percentages being so close to each other (making it seem like they could all converge in the long run to very similar results). However, a closer look at the information in WEKA's summary (and detailed summary) shows that the OneR AttributeEval with J48 Classifier Model is the best model. This is due to the fact that all of its error rates (Mean Absolute Error, Root Mean Squared Error, Relative Absolute Error, Root Relative Squared Error) are the lowest amongst the three models, while both its curves (ROC and PRC curves) exhibit the highest values amongst the three models, thus cementing its position as the best model amongst the twenty models.

Part 7: Discussion, Conclusion, and Model Replication

As discovered above, our best model was the OneR AttributeEval with J48 Classifier, which obtained the maximum accuracy of all twenty models: 95.79%. With this accuracy on roughly 33,000 instances of testing data, we believe our project, and especially this model, proved to be highly successful. Despite being successfully able to train, test, and validate the classification model that predicted an American regions' walkability index with strong consistency, we believe there is definitely room for improvement. Namely, we maintained over 20 attributes after each attribute selection process, and if we had assigned higher, more stringent thresholds, we might have been able to potentially increase our overall accuracy by a significant percentage. If we were to replicate this project, we would have made sure to test various thresholds for the attribute selection algorithms (starting with the higher ones) and/or research the several classifier models to determine the optimal model for our type of project (classification).

In conclusion, we learned numerous concepts about machine learning in general, as well as effectively utilized our theoretical knowledge to potentially help others in a real-world setting. First, we learned how to find a raw dataset from reputable sources. Next we learned how to comb through raw files with both Python and WEKA to conduct numerous preprocessing steps and attribute selection algorithms and techniques that we learned in the class labs. Then, after learning how to make train/test/split files in Python, we learned how to run several models in WEKA to predict accuracies of our testing datasets based on manual input, cross-fold validation techniques, or raw train/test split features. Lastly, we learned how to interpret the results with more sophistication than the accuracy, where we effectively analyzed various error rates, area curves, and the popular confusion matrix to determine our best model. We are confident in our new skills learned in this project, and we look forward to applying them to real world applications similar to this project.

To reproduce our datasets, models, and results, there are numerous steps that'll need to be taken in order to achieve the final datasets and models we've created. Below, you'll find the ordered process to recreate our best model (OneR AttributeEval with J48 Classifier) from scratch.

1. Navigate to the publicly available dataset on the U.S. Government's database of datasets called Walkability Index by clicking here, and download the CSV file of this dataset.
2. Open the CSV file in Microsoft Excel and manually change the name of the 115th column from 'NatWalkInd' to 'Walkability_Index' and shift this column two columns over from the 115th column to the 117th (last) column. Save the dataset as ModifiedDataset.csv
3. Open VSCode and run the first Python script on the fifth page of this report. This process will discretize the numeric values of the class attribute Walkability_Index and then bin them into one of four bins depending on their WI.
4. Run the second Python script on the sixth page of this report. This process should remove all default values (namely 0 and 0.0) and replace them with commas (WEKA's default missing value symbol).
5. Open the new CSV file (now named ModifiedDataset2.csv after the last step saving process) and manually remove all the attributes with 70%+ missing values. By this process, sixty attributes were removed, namely AC_Total, AC_Water, AC_Land, AC_Unpr, TotEmp, E5_ Ret, E5_Off, E5_Ind, E5_ Svc, E5_Ent, E8_Ret, E8_off, E8_ Ind, E8_Svc, E8_Ent, E8_Ed, E8_HIth, E8_Pub, E_LowWageWk, E_MedWageWk, E_HiWageWk, D1A, D1B, D1C, D1C5_RET, D1C5_OFF, D1C5_IND, D1C5_SVC, D1C5_ENT, D1C8_RET, D1C8_OFF, D1C8_IND, D1C8_SVC, D1C8_ENT, D1C8_ED, D1C8_HLTH, D1C8_PUB, D1D, D1_FLAG, D2A_JPHH, DA_WRKEMP, D3BPO4, D4A, D4B025, D4B050, D4C, D4D, D4E, D5AR, D5AE, D5BR, D5BE, D5CR, D5CE, D5DR, D5DRI, D5DE, D5DEI, Shape_Length, and Shape_Area.
6. Fill in missing values by finding WEKA's ReplaceMissingValues filter located in filters > Choose > filters > unsupervised > attribute > ReplaceMissingValues. Next, click 'Apply' to apply this filter to the entire dataset.

7. Run the attribute selection on the current dataset and note the threshold; for our best model, these features are OneR. To do this, select the 'Select Attributes' tab. Next, in Attribute Evaluator, click 'Choose' > attributeSelection > OneRAttributeEval, and accept WEKA's pop-up to make the Search Method Ranker. Click Start, and note the attributes and their coefficients. Go back to the Preprocess tab and remove all the attributes with a coefficient less than 36 (with a minimum bucket size of 6). All the attributes should be removed except the following: D3B_Ranked, D4A_Ranked, D3B, D3A, GEOID10, GEOID20, D3APO, D3BMM4, CBSA_WRK, CBSA, CBSA_POP, CBSA_EMP, OBJECTID, D3BMM3, D3BAO, TRACTCE, CBSA_Name, D3AAO, CSA, STATEFP, CSA_Name, COUNTYFP, D5CEI, D2B_Ranked, D5CRI, D3AMM, D2A_Ranked, AutoOwn0, P_WrkAge, AutoOwn2p, D2C_TRIPEQ, and Pct_AO2p.
8. Save the dataset as a CSV file, and run the third Python script on the fifteenth page of this report to complete the stratified random sampling process to maintain class distributions for the next step, building train/validation/testing datasets.
9. Run the fourth Python script on the sixteenth page of this report to make the three train, validation, and testing datasets on 70/15/15 splits. Note that this will create three files.
10. Open the largest file (training dataset) in WEKA, and head to the Classify tab for the last parts of the project: uploading the testing dataset, building the model, and obtaining the final results.
11. Change the classifier by clicking Choose > trees > J48 (for our best model) and in the test options, click 'Supplied test set' and open then upload the testing dataset to this location.
12. Click the 'Start' button and wait a couple minutes for the model to build and results to be calculated and cleanly outputted to the Classifier Output space.

The above process should result in the Classifier Output showing the same results as the first image on the twenty-fifth page of this report. This concludes the replication of our best model.

Part 8: Team Members and Task Distribution

The entirety of this project was completed by only two students, Arnav Gupta and Raghav Kamineni, who both are in Dr. Yilmaz's fifth period Machine Learning 1 class. Below is a list of the parts Arnav Gupta, Raghav Kamineni, and both teammates completed together (with the first name contributing significantly more for that specific task at hand than the second name).

Finding Dataset and Building Proposal: Arnav Gupta
Preprocessing Initial Attempt: Raghav Kamineni
Preprocessing & Project Update: Raghav Kamineni and Arnav Gupta
Non-WEKA Attribute Selection Algorithm: Raghav Kamineni
Attribute Selection Algorithms and Classifiers: Arnav Gupta
Results Output: Raghav Kamineni
Results Analysis: Arnav Gupta
Building Final Report: Arnav Gupta and Raghav Kamineni

Part 9: Sources and Acknowledgements

For this project, we used the U.S. Government's Data Catalog to find our dataset. Out of the 300,000+ datasets, we chose the Walkability Index dataset from the U.S. Environmental Protection Agency, which can be found here: https://catalog.data.gov/dataset/walkability-index7.

In addition to the dataset, we also used Visual Studio Code to run our several Python scripts, and lastly, we used WEKA to complete the entirety of our attribute selection algorithm and classifier model processes to obtain the final results and analysis.

Lastly, we'd like to thank the U.S. Environment Protection Agency for uploading their comprehensive Walkability Index dataset to a publicly available dataset database, and more importantly, we'd like to thank Dr. Yilmaz for supporting our team throughout the whole process: from teaching us to use WEKA through labs to reviewing our dataset and deliverables with frequently valuable feedback for us. Thank you.

END OF QUARTER ONE PROJECT BY ARNAV GUPTA AND RAGHAV KAMINENI