A Model for Energy-Saving in an IoT Smarthome accounting for End-User Convenience

Alistair Francis Bowman Grevis-James

November 2019

Contents

Test for in **??** we see XYZ

1 Introduction

- Using only accuracy as a metric, the original, top 5 and electrical only were compared.

A decision tree classifier

The reported averages include macro average (averaging the unweighted mean per label), weighted average (averaging the support-weighted mean per label) https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

The last line gives a weighted average of precision, recall and f1-score where the weights are the support values. so for precision the avg is (0.50*1 + 0.0*1 + 1.0*3)/5 = 0.70. The total is just for total support which is 5 here.

Compute the F1 score, also known as balanced F-score or F-measure

The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

F1 = 2 * (precision * recall) / (precision + recall) https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

1.0.1 Bedroom Lightswitch

- Confusion matrix
- KPIs

best params: {'criterion': 'entropy', 'max_depth': 4} best score: 0.9795956924239562 best features: ['bathroom_lightswitch' 'kitchen_cabinet' 'kitchen_dishwasher' 'kitchen_laundrydryer' 'livingroom_lightswitch'] feature importances: [0.02321617 0.00763424 0.00526102 0.00518979 0.00453024]

```
          precision    recall  f1-score   support

     0.0       0.98      1.00      0.99      1543
     1.0       0.94      0.36      0.52        45

accuracy                           0.98      1588
```

macro avg 0.96 0.68 0.75 1588 weighted avg 0.98 0.98 0.98 1588

[[1542 1] [29 16]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.3s finished

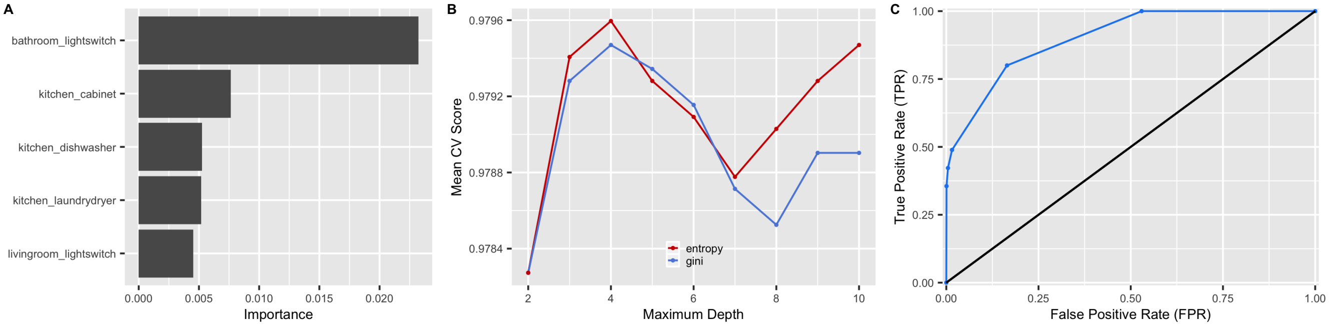

Figure 1: ADD TEXT

1.0.2 Kitchen Laundrydryer

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.1s finished best params: {'criterion': 'gini', 'max_depth': 5} best score: 0.988034510989357 precision recall f1-score support

```
         0.0       0.99      1.00      0.99      1565
         1.0       0.70      0.30      0.42        23

    accuracy                           0.99      1588
```

macro avg 0.84 0.65 0.71 1588 weighted avg 0.99 0.99 0.99 1588

[[1562 3] [16 7]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 1.9s finished

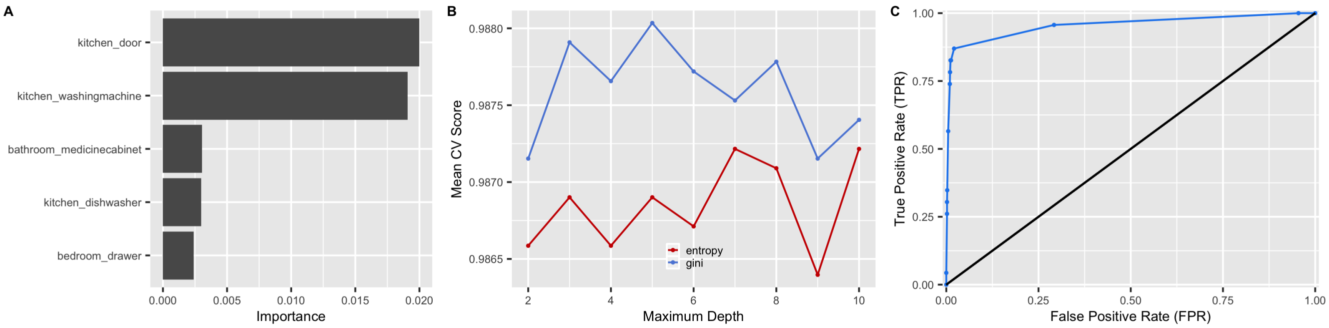

Figure 2: ADD TEXT

1.0.3 Kitchen Freezer

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.5s finished best params: {'criterion': 'gini', 'max_depth': 9} best score: 0.9164305056993514

```
       precision    recall  f1-score   support

     0.0       0.94      0.98      0.96      1423
     1.0       0.76      0.44      0.56       165

accuracy                           0.93      1588
```

macro avg 0.85 0.71 0.76 1588 weighted avg 0.92 0.93 0.92 1588

[[1400 23] [92 73]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.1s finished

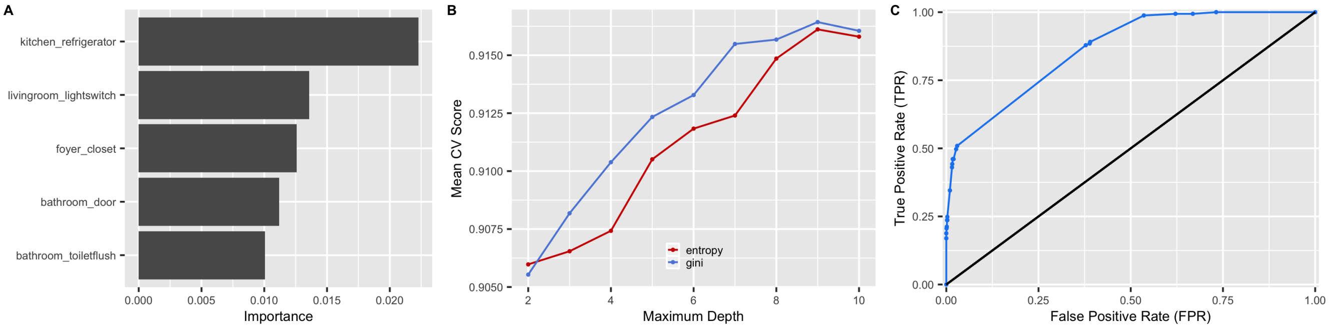

Figure 3: ADD TEXT

1.0.4 Kitchen Toaster

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.5s finished best params: {'criterion': 'gini', 'max_depth': 3} best score: 0.9886642735688645

```
  precision    recall  f1-score   support

      0.0       0.99      1.00      0.99      1566
      1.0       0.83      0.23      0.36        22

accuracy                            0.99      1588
```

macro avg 0.91 0.61 0.68 1588 weighted avg 0.99 0.99 0.99 1588

[[1565 1] [17 5]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.7s finished

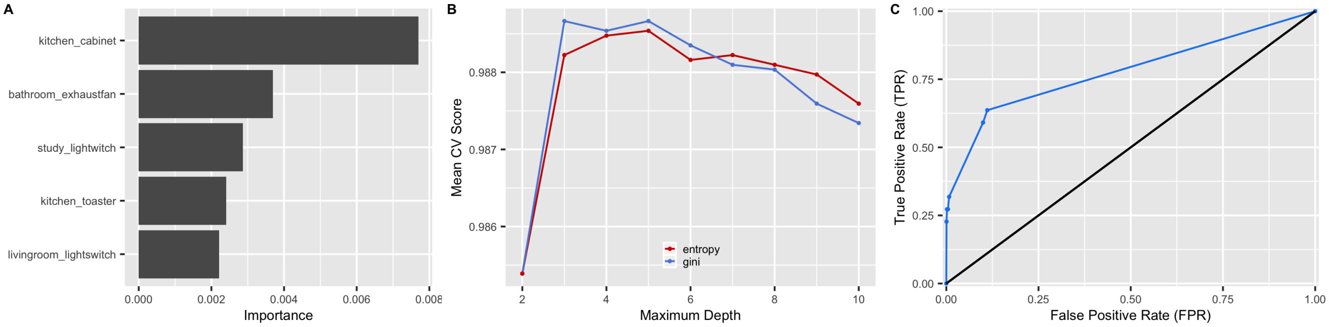

Figure 4: ADD TEXT

1.0.5 Bathroom Exhaustfan

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.3s finished best params: {'criterion': 'gini', 'max_depth': 8} best score: 0.8798412998299641

```
  precision    recall  f1-score   support

     0.0       0.90      0.99      0.94      1384
     1.0       0.77      0.25      0.38       204

accuracy                           0.89      1588
```

macro avg 0.84 0.62 0.66 1588 weighted avg 0.88 0.89 0.87 1588

[[1369 15] [153 51]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.9s finished

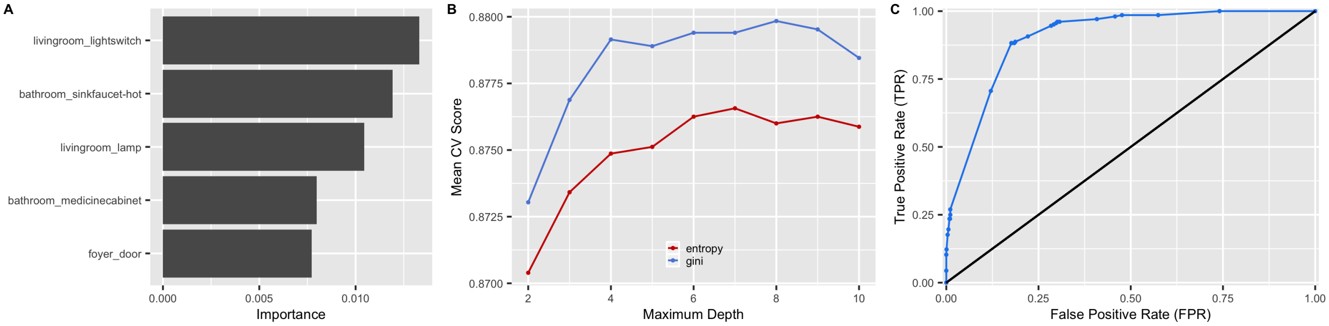

Figure 5: ADD TEXT

1.0.6 Bathroom Showerfaucet

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.5s finished best params: {'criterion': 'entropy', 'max_depth': 8} best score: 0.9669374645758549

```
      precision    recall  f1-score   support

     0.0       0.97      1.00      0.98      1520
     1.0       0.82      0.40      0.53        68

accuracy                           0.97      1588
```

macro avg 0.90 0.70 0.76 1588 weighted avg 0.97 0.97 0.97 1588

[[1514 6] [41 27]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.7s finished

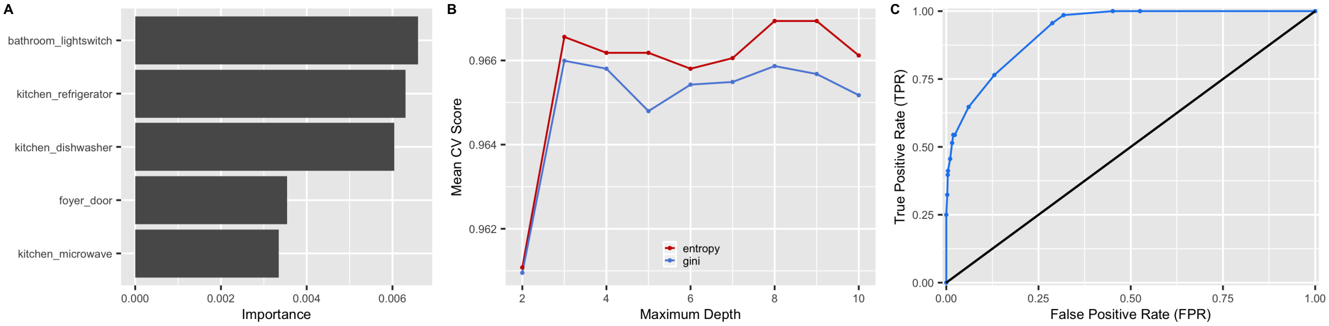

Figure 6: ADD TEXT

1.0.7 bathroom_lightswitch

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.5s finished best params: {‘criterion’: ‘gini’, ‘max_depth’: 10} best score: 0.8392216134517287

```
    precision    recall  f1-score   support

     0.0       0.85      0.99      0.91      1262
     1.0       0.85      0.30      0.45       326

accuracy                           0.85      1588
```

macro avg 0.85 0.64 0.68 1588 weighted avg 0.85 0.85 0.82 1588

[[1244 18] [227 99]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.0s finished

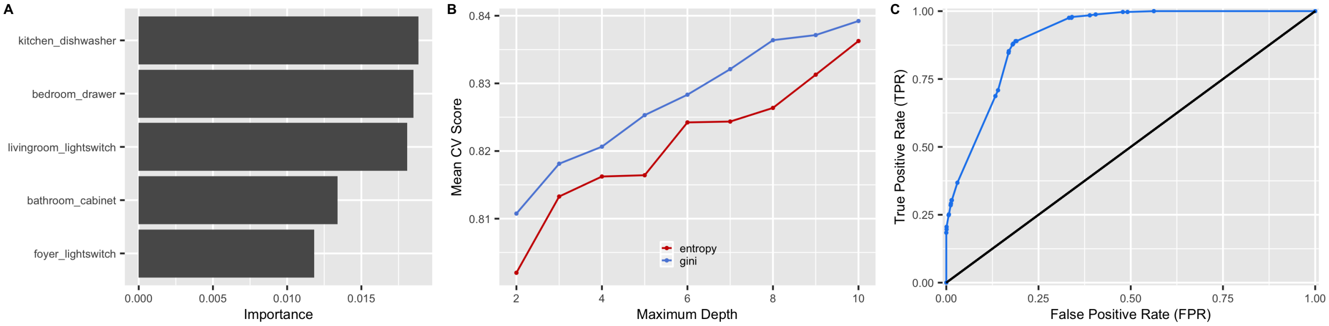

Figure 7: ADD TEXT

1.0.8 kitchen__refrigerator

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n__jobs=1): Done 270 out of 270 | elapsed: 3.0s finished best params: {'criterion': 'gini', 'max__depth': 10} best score: 0.9646073430316771

```
    precision    recall  f1-score   support

       0.0       0.97      1.00      0.99      1506
       1.0       0.95      0.51      0.67        82

accuracy                           0.97      1588
```

macro avg 0.96 0.76 0.83 1588 weighted avg 0.97 0.97 0.97 1588

[[1504 2] [40 42]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n__jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n__jobs=1): Done 270 out of 270 | elapsed: 2.4s finished

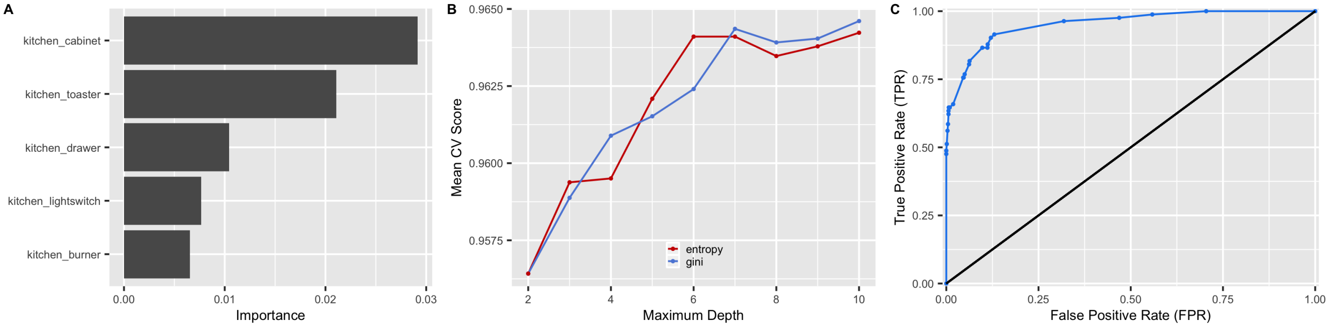

Figure 8: ADD TEXT

1.0.9 foyer__lightswitch

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n__jobs=1): Done 270 out of 270 | elapsed: 2.9s finished best params: {'criterion': 'gini', 'max__depth': 4} best score: 0.9894199886642736

```
     precision    recall  f1-score   support

     0.0       0.99      1.00      1.00      1528
     1.0       0.98      0.77      0.86        60

accuracy                           0.99      1588
```

macro avg 0.98 0.88 0.93 1588 weighted avg 0.99 0.99 0.99 1588

[[1527 1] [14 46]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n__jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n__jobs=1): Done 270 out of 270 | elapsed: 3.0s finished

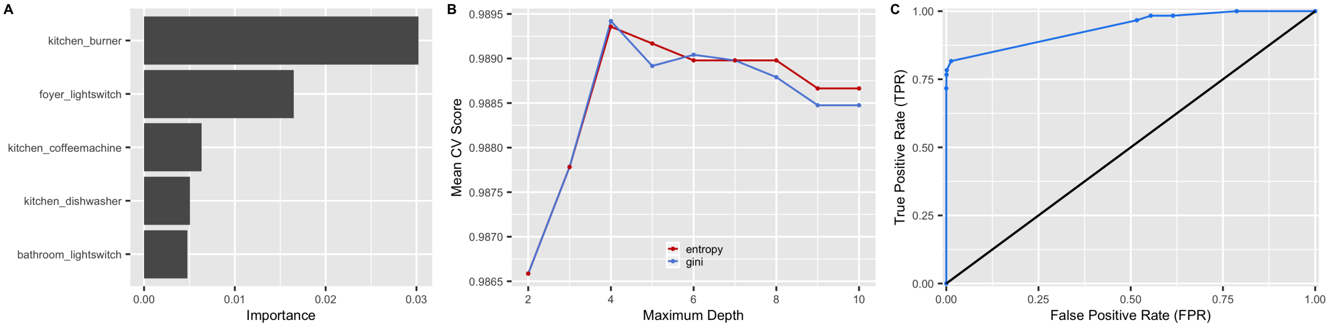

Figure 9: ADD TEXT

1.0.10 kitchen_burner

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.5s finished best params: {'criterion': 'entropy', 'max_depth': 8} best score: 0.9630329365829082

```
 precision    recall  f1-score   support

     0.0       0.97      1.00      0.98      1506
     1.0       0.97      0.34      0.50        82

accuracy                           0.97      1588
```

macro avg 0.97 0.67 0.74 1588 weighted avg 0.97 0.97 0.96 1588

[[1505 1] [54 28]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.2s finished

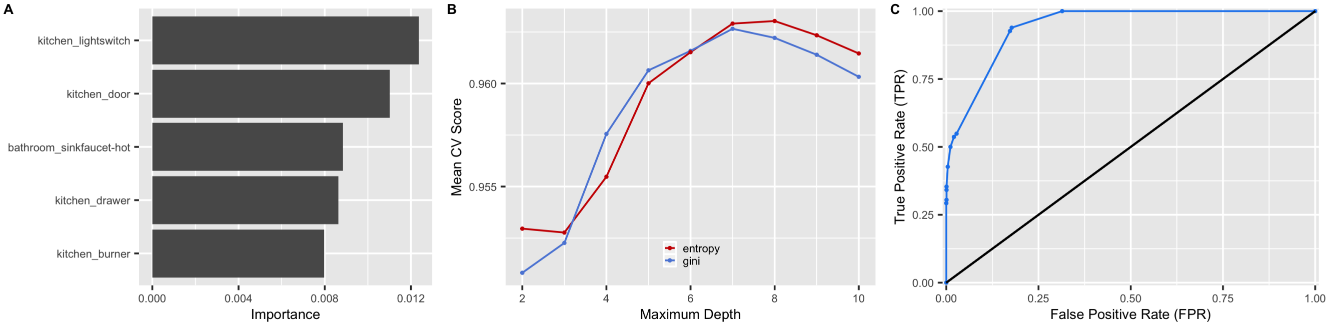

Figure 10: ADD TEXT

1.0.11 study Lightswitch

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.9s finished best params: {'criterion': 'entropy', 'max_depth': 8} best score: 0.8873354745261036

```
     precision    recall  f1-score   support

     0.0        0.89       1.00       0.94       1407
     1.0        0.81       0.07       0.13        181

accuracy                              0.89       1588
```

macro avg 0.85 0.53 0.54 1588 weighted avg 0.88 0.89 0.85 1588

[[1404 3] [168 13]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.3s finished

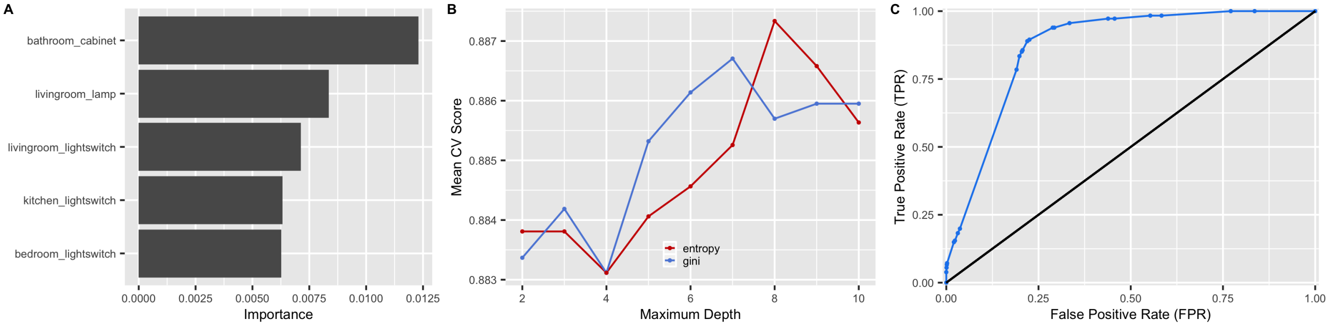

Figure 11: ADD TEXT

1.0.12 kitchen_washingmachine

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.1s finished best params: {'criterion': 'gini', 'max_depth': 3} best score: 0.9892310598904213

```
 precision    recall  f1-score   support

     0.0       0.99      1.00      0.99      1562
     1.0       0.83      0.19      0.31        26

accuracy                           0.99      1588
```

macro avg 0.91 0.60 0.65 1588 weighted avg 0.98 0.99 0.98 1588

[[1561 1] [21 5]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.0s finished

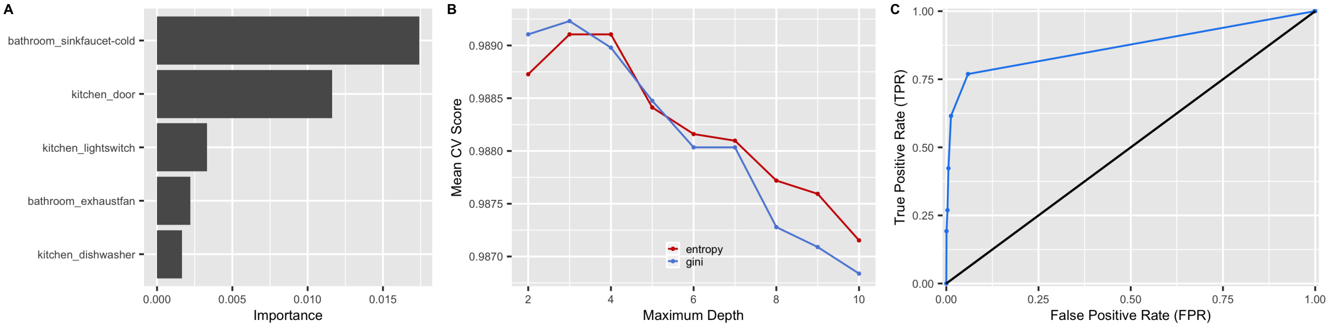

Figure 12: ADD TEXT

1.0.13 Kitchen Lightswitch

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.4s finished best params: {'criterion': 'gini', 'max_depth': 10} best score: 0.797153473140626

```
      precision    recall  f1-score   support

     0.0       0.88      0.87      0.87      1100
     1.0       0.71      0.74      0.72       488

accuracy                           0.83      1588
```

macro avg 0.80 0.80 0.80 1588 weighted avg 0.83 0.83 0.83 1588

[[952 148] [127 361]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.4s finished

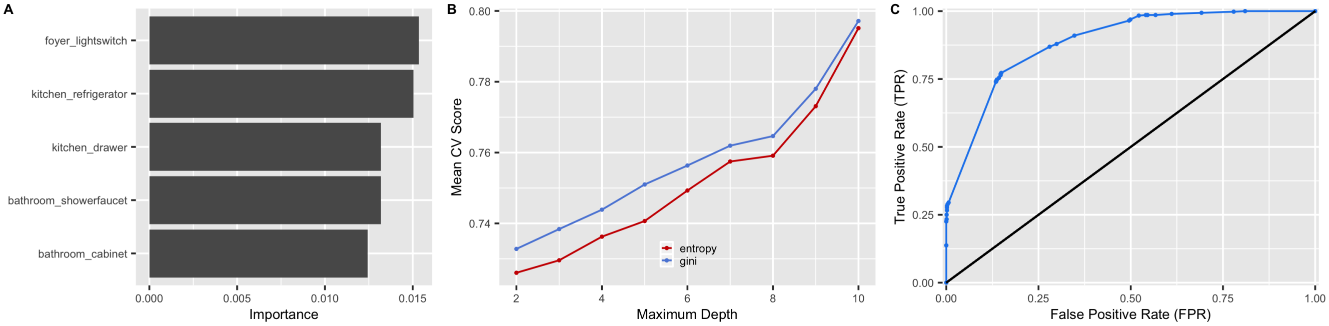

Figure 13: ADD TEXT

1.0.14 Livingroom Lamp

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.4s finished best params: {'criterion': 'gini', 'max_depth': 8} best score: 0.8140311102714277

```
  precision    recall  f1-score   support

      0.0       0.81      0.99      0.89      1252
      1.0       0.83      0.16      0.27       336

accuracy                          0.82      1588
```

macro avg 0.82 0.58 0.58 1588 weighted avg 0.82 0.82 0.76 1588

[[1241 11] [282 54]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.0s finished

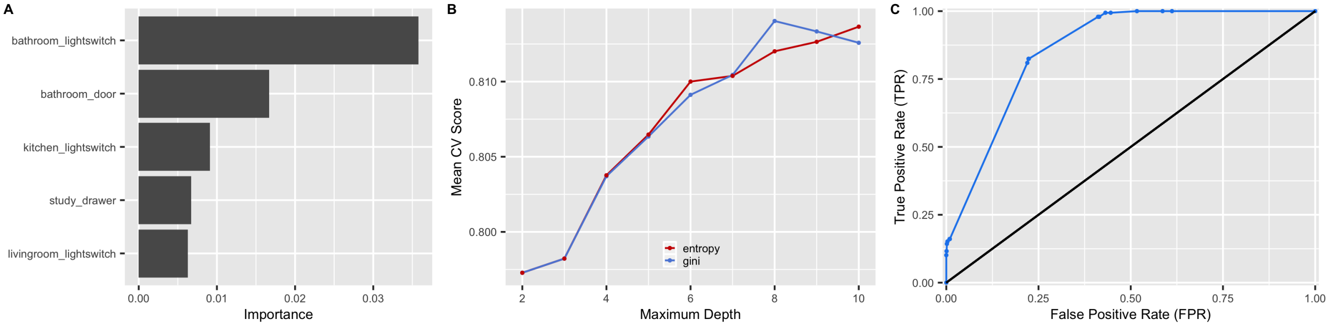

Figure 14: ADD TEXT

1.0.15 kitchen_microwave

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.3s finished best params: {'criterion': 'entropy', 'max_depth': 2} best score: 0.989042131116569

```
 precision    recall  f1-score   support

      0.0       0.99      1.00      1.00      1575
      1.0       0.00      0.00      0.00        13

accuracy                           0.99      1588
```

macro avg 0.50 0.50 0.50 1588 weighted avg 0.98 0.99 0.99 1588

[[1575 0] [13 0]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.5s finished

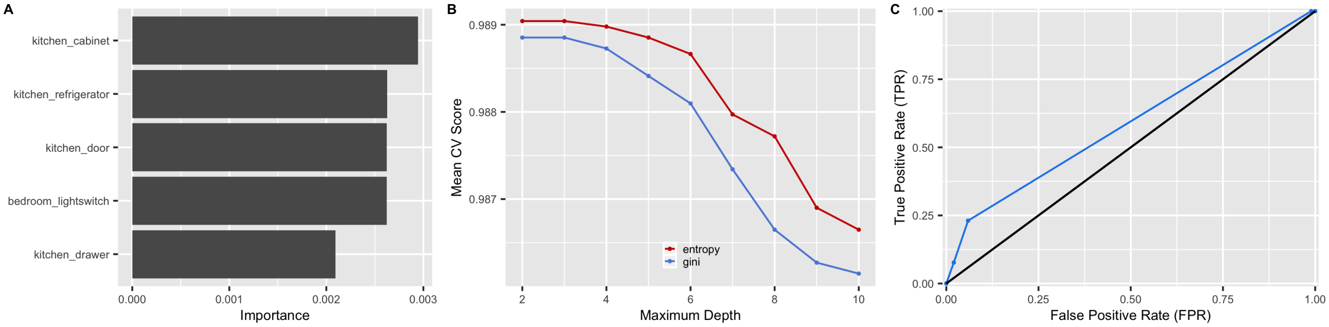

Figure 15: ADD TEXT

1.0.16 kitchen_garbagedisposal

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.4s finished best params: {'criterion': 'gini', 'max_depth': 2} best score: 0.9975439259399207

```
   precision    recall  f1-score   support

     0.0       1.00      1.00      1.00      1586
     1.0       0.00      0.00      0.00         2

accuracy                           1.00      1588
```

macro avg 0.50 0.50 0.50 1588 weighted avg 1.00 1.00 1.00 1588

[[1586 0] [2 0]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 2.9s finished

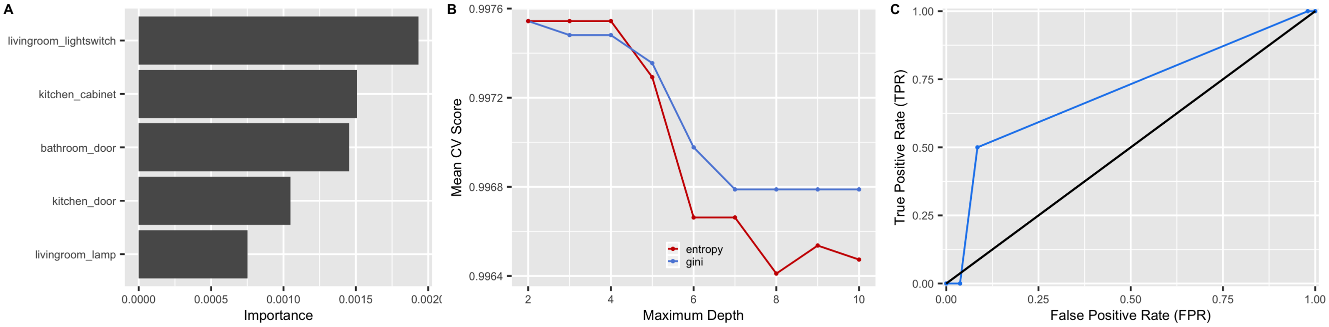

Figure 16: ADD TEXT

1.0.17 kitchen_coffeemachine

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.2s finished best params: {'criterion': 'gini', 'max_depth': 2} best score: 0.9986774985830342

```
   precision    recall  f1-score   support

     0.0       1.00      1.00      1.00      1587
     1.0       0.00      0.00      0.00         1

accuracy                           1.00      1588
```

macro avg 0.50 0.50 0.50 1588 weighted avg 1.00 1.00 1.00 1588

[[1587 0] [1 0]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.1s finished

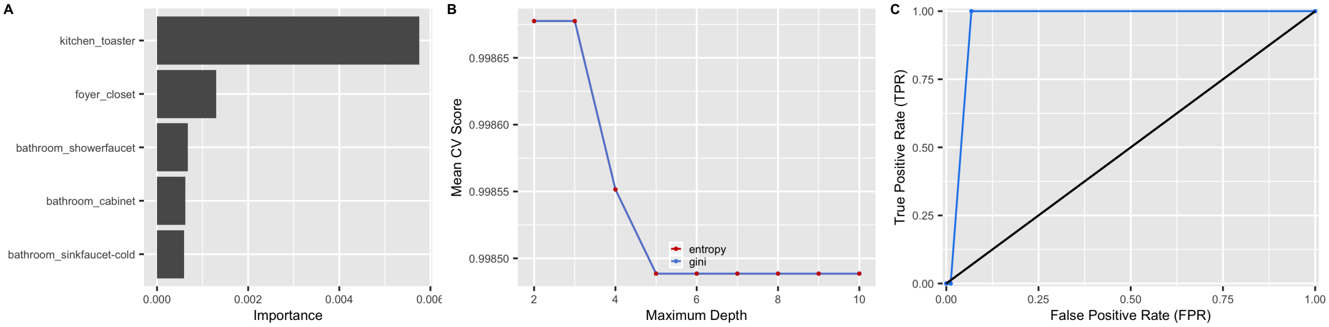

Figure 17: ADD TEXT

1.0.18 livingroom_lightswitch

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.6s finished best params: {'criterion': 'entropy', 'max_depth': 7} best score: 0.9080546633919012

```
     precision    recall  f1-score   support

     0.0       0.91      1.00      0.95      1386
     1.0       1.00      0.28      0.44       202

accuracy                           0.91      1588
```

macro avg 0.95 0.64 0.70 1588 weighted avg 0.92 0.91 0.89 1588

[[1386 0] [145 57]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 4.0s finished

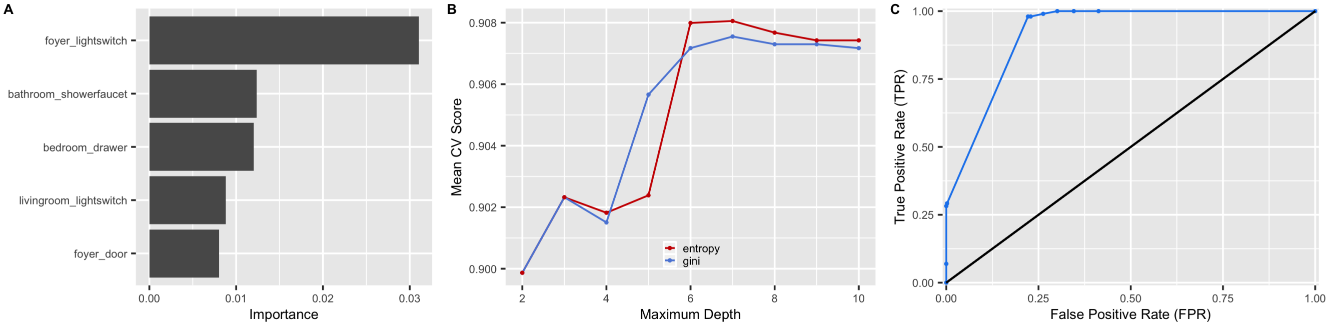

Figure 18: ADD TEXT

1.0.19 kitchen_oven

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.3s finished best params: {'criterion': 'entropy', 'max_depth': 2} best score: 0.9951508281377921

```
   precision    recall  f1-score   support

     0.0       0.99      1.00      1.00      1579
     1.0       0.00      0.00      0.00         9

accuracy                           0.99      1588
```

macro avg 0.50 0.50 0.50 1588 weighted avg 0.99 0.99 0.99 1588

[[1579 0] [9 0]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.4s finished

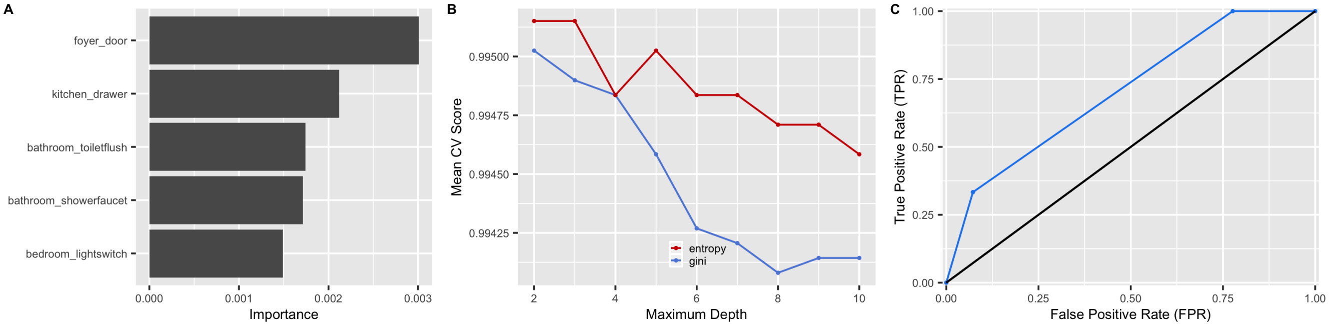

Figure 19: ADD TEXT

1.0.20 kitchen_dishwasher

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.8s finished best params: {'criterion': 'entropy', 'max_depth': 8} best score: 0.949870898671201

```
          precision    recall  f1-score   support

     0.0       0.96      0.99      0.98      1496
     1.0       0.75      0.26      0.39        92

accuracy                           0.95      1588
```

macro avg 0.85 0.63 0.68 1588 weighted avg 0.94 0.95 0.94 1588

[[1488 8] [68 24]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 4.0s finished

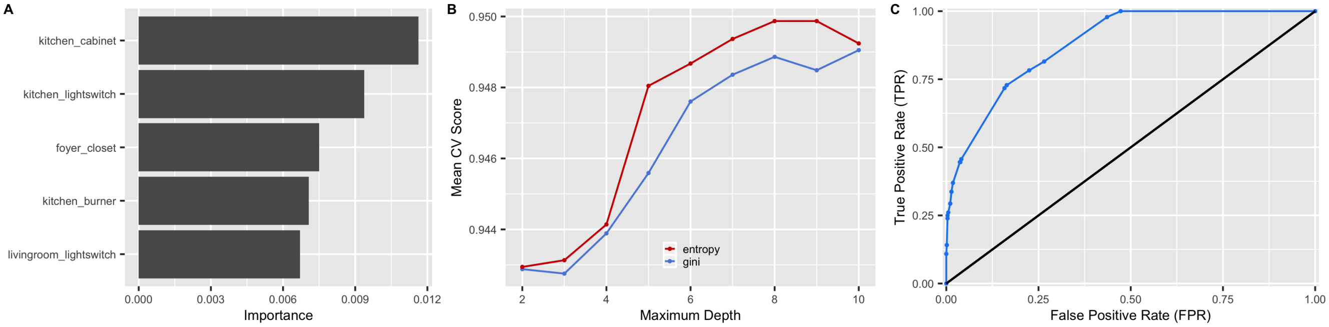

Figure 20: ADD TEXT

1.0.21 bathroom_sinkfaucet-hot

- Confusion matrix
- KPIs

Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 5.6s finished best params: {'criterion': 'entropy', 'max_depth': 6} best score: 0.9702752062472448

```
  precision    recall  f1-score   support

      0.0       0.97      1.00      0.98      1514
      1.0       0.86      0.42      0.56        74

accuracy                          0.97      1588
```

macro avg 0.92 0.71 0.77 1588 weighted avg 0.97 0.97 0.96 1588

[[1509 5] [43 31]] Fitting 15 folds for each of 18 candidates, totalling 270 fits Parallel(n_jobs=1): Using backend SequentialBackend with 1 concurrent workers. Parallel(n_jobs=1): Done 270 out of 270 | elapsed: 3.5s finished

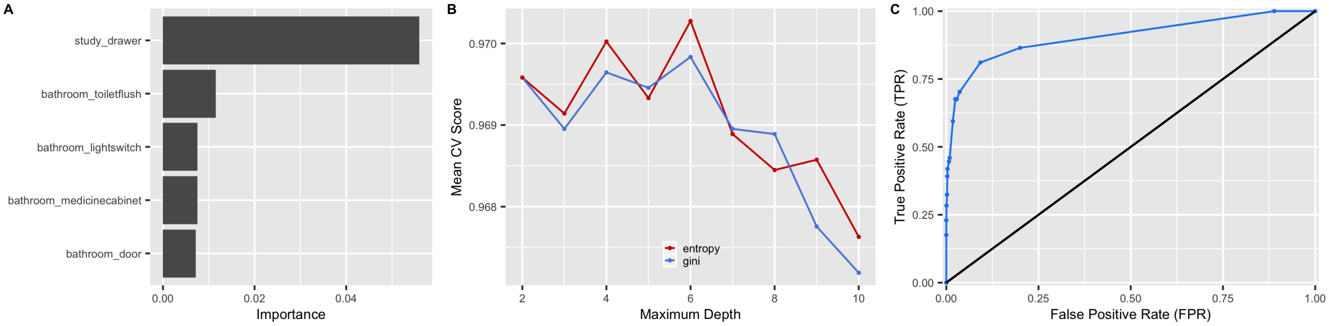

Figure 21: ADD TEXT

2 References