Maartens Lourens
Machine Learning Engineer - Automation Logic

Twitter: @thundercomb
Github: @thundercomb
A Pragmatic Introduction to Machine Learning for Engineers
Overview:

1. Introduction
2. The Problem
3. The Solution
Talk takeaways:

1. Anyone can take ML for a spin
2. Deep math knowledge not required
Introduction
RNNs and Me
Andrej Karpathy:

The Unreasonable Effectiveness of Recurrent Neural Networks

http://karpathy.github.io/2015/05/21/rnn-effectiveness/
AlphaGo & DeepMind
My morning coffee turned out to be the difference between going and not going to NIPS 2018 this year. Apparently sold out in <15 minutes. I laughed at this diagram a year ago, but today it is too real.

Soumith Chintala @soumithchintala
[2018] War erupts for tickets
[2019] AI researchers discover time travel
Classical Machine Learning vs Deep Learning
So... What is Machine Learning?
Learning from Data
But is it Machine Learning?
Data science produces insights
Machine learning produces predictions
Artificial intelligence produces actions

*Credit: David Robinson at http://varianceexplained.org/r/ds-ml-ai/*
The Problem
System.log

Jul 3 03:42:41 Maartenss-MacBook-Pro parsecd[307]: BUG in libdispatch client: dispatch_mig_server: mach_msg() failed (ipc/send) msg too small - 0x10000008
Jul 3 03:42:41 Maartenss-MacBook-Pro systemstats[50]: assertion failed: 17E199: systemstats + 689866
[D1E75C38-62CE-3D77-9ED3-5F6D38EF0676]: 0x5
Jul 3 03:42:41 Maartenss-MacBook-Pro systemstats[50]: assertion failed: 17E199: systemstats + 914800
[D1E75C38-62CE-3D77-9ED3-5F6D38EF0676]: 0x40

Wifi.log

Tue Jul 3 03:42:41.149 <kernel> Creating all peerManager reporters
Tue Jul 3 03:42:41.184 <airportd[153]> _initLocaleManager: Started locale manager
Tue Jul 3 03:42:41.194 <airportd[153]> airportdProcessDLILEvent: en0 attached (down)
Tue Jul 3 03:42:41.219 <kernel> wl0: setAWDL_PEER_TRAFFIC_REGISTRATION: active 0, roam_off: 0, err 0 roam_start_set 0 forced_roam_set 0
Tue Jul 3 03:42:41.269 <kernel> AirPort_Brcm43xx::syncPowerState: WWEN[disabled]
Jul 3 05:11:21 --- last message repeated 1 time ---
$ grep -l "\-\-\- last message repeated 1 time \-\-\-" /var/log/*.log

/var/log/system.log
Jul 3 05:11:21 --- last message repeated 13 times ---
Classification is a Supervised Machine Learning problem.
Training
Predicting
The Solution
Python & Scikit-Learn
import os
import glob
import shutil
import numpy as np
import pandas as pd

from sklearn import preprocessing
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer

from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import SGDClassifier
from sklearn import svm, naive_bayes, linear_model, tree, ensemble, neighbors,
                 semi_supervised, neural_network, discriminant_analysis
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
def copy_data(src_file_path, dst_file_path):
    if not os.path.exists(dst_file_path):
        os.mkdir(dst_file_path)
    for logfile in glob.glob(src_file_path + "/*.log"):
        if os.stat(logfile)[6] > 10000:
            logfile_name = logfile.split('/')[-1]
            shutil.copyfile(logfile, dst_file_path + "/" + logfile_name)
\[ y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it} \]
def read_data(logfile_path):
    log_collection = pd.DataFrame()
    logs = pd.DataFrame()
    logfiles = glob.glob(logfile_path + '/*.log') # Get list of log files
    for logfile in logfiles:
        logs = pd.read_csv(logfile, sep="\n", header=None, names=['data'])
        logs['type'] = logfile.split('/')[-1]
        # Add log file data and type to log collection
        log_collection = log_collection.append(logs)

    # Remove empty lines
    log_collection = log_collection.dropna()

    return log_collection
source_data_dir = "/var/log"
data_dir = "data"

copy_data(source_data_dir, data_dir)
log_collection = read_data(data_dir)
Training
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(log_collection['data'],
    log_collection['type'],
    test_size=0.2,
    random_state=42)
def train(algorithm, X_train, y_train):
    model = Pipeline([('vect', CountVectorizer()), ('tfidf', TfidfTransformer()), ('clf', algorithm)])
Training
def train(algorithm, X_train, y_train):
    model = Pipeline([('vect', CountVectorizer()), ('tfidf', TfidfTransformer()), ('clf', algorithm)])
    model.fit(X_train, y_train)
    return model
algorithms = [
    linear_model.SGDClassifier(loss='hinge', penalty='l2', alpha=1e-3, random_state=42, max_iter=5, tol=None),
    naive_bayes.MultinomialNB(),
    naive_bayes.BernoulliNB(),
    tree.DecisionTreeClassifier(max_depth=1000),
    tree.ExtraTreeClassifier(),
    ensemble.ExtraTreesClassifier(),
    svm.LinearSVC(),
    neighbors.NearestCentroid(),
    ensemble.RandomForestClassifier(),
    linear_model.RidgeClassifier(),
]
for algorithm in algorithms:
    model = train(algorithm, X_train, y_train)
Predicting
for algorithm in algorithms:
    model = train(algorithm, X_train, y_train)
    predictions = model.predict(X_test)
True & False Positives
True & False Negatives
The Boy Who Cried Wolf
True Positive:  Reality:  A wolf threatens  
Shepherd:  “Wolf!”

True Negative:  Reality:  No wolf threatens  
Shepherd:  Quiet.

False Positive:  Reality:  No wolf threatens  
Shepherd:  “Wolf!”

False Negative:  Reality:  A wolf threatens  
Shepherd:  Quiet.

Credit:
https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative
\begin{align*}
\text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \\
\text{Recall} &= \frac{TP}{TP + FN} \\
\text{Precision} &= \frac{TP}{TP + FP} \\
\text{F1 Score} &= 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}
\end{align*}
def report(classifier, actual, predictions):
    print("\033[1m" + classifier + "\033[0m\033[50m\n")

    actual = np.array(actual)

    print(confusion_matrix(actual, predictions))
    print
    print(classification_report(actual, predictions))
    print("Accuracy: " + str(round(accuracy_score(actual, predictions),2)))
    print
for algorithm in algorithms:
    model = train(algorithm, X_train, y_train)
    predictions = model.predict(X_test)
    report((str(algorithm).split('(')[0]), y_test, predictions)
SGDClassifier

```
[[2536  0  0  0  0  0  0  0  0  0]
 [ 0 131  0  0  0  0  0  0  0  0]
 [ 0  0 1569  172  0  0  0  0  0  0]
 [ 0  0  39 1277  0  0  0  3  0  0]
 [ 0  0  0  0  1091  0  0  0  0  0]
 [ 0  0  0  0  0  0 116  0  0  0]
 [ 0  0  0  0  0  0  948  0  0  1]
 [ 0  0  0  0  0  0  919  0  1  1]
 [ 0  0  0  0  0  0  0 761  1  1]
 [ 0  0  0  0  3  0  1  0  4 1714]]
```

<table>
<thead>
<tr>
<th>File</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>corecaptured.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2536</td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>131</td>
</tr>
<tr>
<td>install.log</td>
<td>0.98</td>
<td>0.90</td>
<td>0.94</td>
<td>1741</td>
</tr>
<tr>
<td>system.log</td>
<td>0.88</td>
<td>0.97</td>
<td>0.92</td>
<td>1319</td>
</tr>
<tr>
<td>wifi-08-23-2018__12:54:38.121.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1091</td>
</tr>
<tr>
<td>wifi-08-24-2018__09:09:14.458.log</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>116</td>
</tr>
<tr>
<td>wifi-08-24-2018__12:47:32.191.log</td>
<td>0.89</td>
<td>1.00</td>
<td>0.94</td>
<td>949</td>
</tr>
<tr>
<td>wifi-08-28-2018__14:27:47.184.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>920</td>
</tr>
<tr>
<td>wifi-09-03-2018__12:45:21.309.log</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>762</td>
</tr>
<tr>
<td>wifi.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1722</td>
</tr>
<tr>
<td><strong>avg / total</strong></td>
<td><strong>0.96</strong></td>
<td><strong>0.97</strong></td>
<td><strong>0.97</strong></td>
<td><strong>11287</strong></td>
</tr>
</tbody>
</table>

Accuracy: 0.97
### MultinomialNB

```
[[2536  0  0  0  0  0  0  0  0  0]
 [  0 131  0  0  0  0  0  0  0  0]
 [ 161 1577  1  0  0  0  0  0  2  0]
 [ 111  90 1111  0  0  0  3  0  4  0]
 [  0  0  0 1088  0  0  2  0  1  0]
 [  0  0  0  0 114  1  0  1  0  1]
 [  0  0  0  0  9  0 938  1  0  1]
 [  0  0  0  23  0  0 895  0  2  0]
 [  0  0  0  0 14  0  0  1 746  1]
 [  0  0  0  0  2  0  0  0 1719 1]]
```

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>corecaptured.log</td>
<td>0.90</td>
<td>1.00</td>
<td>0.95</td>
<td>2536</td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>131</td>
</tr>
<tr>
<td>install.log</td>
<td>0.95</td>
<td>0.91</td>
<td>0.93</td>
<td>1741</td>
</tr>
<tr>
<td>system.log</td>
<td>1.00</td>
<td>0.84</td>
<td>0.91</td>
<td>1319</td>
</tr>
<tr>
<td>wifi-08-23-2018__12:54:38.121.log</td>
<td>0.96</td>
<td>1.00</td>
<td>0.98</td>
<td>1091</td>
</tr>
<tr>
<td>wifi-08-24-2018__09:09:14.458.log</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>116</td>
</tr>
<tr>
<td>wifi-08-24-2018__12:47:32.191.log</td>
<td>0.89</td>
<td>0.99</td>
<td>0.94</td>
<td>949</td>
</tr>
<tr>
<td>wifi-08-28-2018__14:27:47.184.log</td>
<td>0.99</td>
<td>0.97</td>
<td>0.98</td>
<td>920</td>
</tr>
<tr>
<td>wifi-09-03-2018__12:45:21.309.log</td>
<td>1.00</td>
<td>0.98</td>
<td>0.99</td>
<td>762</td>
</tr>
<tr>
<td>wifi.log</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1722</td>
</tr>
</tbody>
</table>

**Accuracy:** 0.95

BernoulliNB

```

[[2536  0  0  0  0  0  0  0  0  0]
 [  0  131  0  0  0  0  0  0  0  0]
 [  293  1429  18  0  0  0  0  0  0  0]
 [  227  0  26 1066  0  0  0  0  0  0]
 [  0  0  0  0 1091  0  0  0  0  0]
 [  0  0  0  0  0  3 113  0  0  0]
 [  0  0  0  0  0 10 938  0  0  1]
 [  0  0  0  0  0  0  0  918  0  2]
 [  0  0  0  0  0  0  0  0  761  1]
 [  0  0  0  0  2  0  0  0  1 1719]]
```

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>corecaptured.log</td>
<td>0.83</td>
<td>1.00</td>
<td>0.91</td>
<td>2536</td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>131</td>
</tr>
<tr>
<td>install.log</td>
<td>0.98</td>
<td>0.82</td>
<td>0.89</td>
<td>1741</td>
</tr>
<tr>
<td>system.log</td>
<td>0.98</td>
<td>0.81</td>
<td>0.89</td>
<td>1319</td>
</tr>
<tr>
<td>wifi-08-23-2018__12:54:38.121.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1091</td>
</tr>
<tr>
<td>wifi-08-24-2018__09:09:14.458.log</td>
<td>0.23</td>
<td>0.03</td>
<td>0.05</td>
<td>116</td>
</tr>
<tr>
<td>wifi-08-24-2018__12:27:42.191.log</td>
<td>0.89</td>
<td>0.99</td>
<td>0.94</td>
<td>949</td>
</tr>
<tr>
<td>wifi-08-28-2018__14:27:47.184.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>920</td>
</tr>
<tr>
<td>wifi-09-03-2018__12:45:21.309.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>762</td>
</tr>
<tr>
<td>wifi.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1722</td>
</tr>
</tbody>
</table>

avg / total  0.94  0.94   0.93  11287

Accuracy: 0.94
DecisionTreeClassifier

```

corecaptured.log       1.00      1.00      1.00      2536
fsck_apfs.log         1.00      1.00      1.00       131
install.log           0.93      1.00      0.96      1741
system.log            0.99      0.90      0.95      1319
wifi-08-23-2018__12:54:38.121.log       1.00      1.00      1.00      1091
wifi-08-24-2018__09:09:14.458.log       0.28      0.37      0.32       116
wifi-08-24-2018__12:47:32.191.log       0.92      0.88      0.90       949
wifi-08-28-2018__14:27:47.184.log       1.00      1.00      1.00       920
wifi-09-03-2018__12:45:21.309.log       1.00      1.00      1.00       762
wifi.log              1.00      0.99      1.00      1722

precision    recall    f1-score    support

avg / total  0.97      0.97      0.97      11287

Accuracy: 0.97
```
### ExtraTreeClassifier

```
[[ 2531  0  0  0  2  0  0  0  0  3]
 [  0 129  0  0  1  0  0  0  0  1]
 [  1  0 1727 10  0  0  0  1  0  2]
 [  2  0 139 1177  0  0  0  0  0  1]
 [  2  0  0  0 1064  0  10  1  3 11]
 [  0  0  0  0  43  72  1  0  0]
 [  1  0  0  0  15  164  752  4  7  6]
 [  1  0  0  0  0  6 860  48  5]
 [  0  1  1  0  5  0  9  4  733  9]
 [ 11  2  1  0 12  2  5  8  3 1678]]
```

<table>
<thead>
<tr>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>corecaptured.log</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>install.log</td>
<td>0.92</td>
<td>0.99</td>
<td>0.96</td>
</tr>
<tr>
<td>system.log</td>
<td>0.99</td>
<td>0.89</td>
<td>0.94</td>
</tr>
<tr>
<td>wifi-08-23-2018__12:54:38.121.log</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
</tr>
<tr>
<td>wifi-08-24-2018__09:09:14.458.log</td>
<td>0.21</td>
<td>0.37</td>
<td>0.26</td>
</tr>
<tr>
<td>wifi-08-24-2018__12:47:32.191.log</td>
<td>0.88</td>
<td>0.79</td>
<td>0.83</td>
</tr>
<tr>
<td>wifi-08-28-2018__14:27:47.184.log</td>
<td>0.98</td>
<td>0.93</td>
<td>0.96</td>
</tr>
<tr>
<td>wifi-09-03-2018__12:45:21.309.log</td>
<td>0.92</td>
<td>0.96</td>
<td>0.94</td>
</tr>
<tr>
<td>wifi.log</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
</tbody>
</table>

**Accuracy: 0.95**
**ExtraTreesClassifier**

```python
[[2536 0 0 0 0 0 0 0 0 0]
 [ 0 131 0 0 0 0 0 0 0 0]
 [ 0 0 1739 2 0 0 0 0 0 0]
 [ 0 0 126 1192 0 0 0 0 0 1]
 [ 0 0 0 0 1090 0 0 0 0 1]
 [ 0 0 0 0 0 39 77 0 0 0]
 [ 0 0 0 0 0 108 840 0 0 1]
 [ 0 0 0 0 0 0 0 919 0 1]
 [ 0 0 0 0 0 0 0 0 760 2]
 [ 0 0 0 0 0 3 0 0 4 2 1713]]
```

<table>
<thead>
<tr>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>corecaptured.log</td>
<td>1.00</td>
<td>1.00</td>
<td>2536</td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>1.00</td>
<td>1.00</td>
<td>131</td>
</tr>
<tr>
<td>install.log</td>
<td>0.93</td>
<td>1.00</td>
<td>1741</td>
</tr>
<tr>
<td>system.log</td>
<td>1.00</td>
<td>0.90</td>
<td>1319</td>
</tr>
<tr>
<td>wifi-08-23-2018__12:54:38.121.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1091</td>
</tr>
<tr>
<td>wifi-08-24-2018__09:09:14.458.log</td>
<td>0.27</td>
<td>0.34</td>
<td>116</td>
</tr>
<tr>
<td>wifi-08-24-2018__12:47:32.191.log</td>
<td>0.92</td>
<td>0.89</td>
<td>949</td>
</tr>
<tr>
<td>wifi-08-28-2018__14:27:47.184.log</td>
<td>1.00</td>
<td>1.00</td>
<td>920</td>
</tr>
<tr>
<td>wifi-09-03-2018__12:45:21.309.log</td>
<td>1.00</td>
<td>1.00</td>
<td>762</td>
</tr>
<tr>
<td>wifi.log</td>
<td>1.00</td>
<td>0.99</td>
<td>1722</td>
</tr>
</tbody>
</table>

**Accuracy:** 0.97
### LinearSVC

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>corecaptured.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2536</td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>131</td>
</tr>
<tr>
<td>install.log</td>
<td>0.93</td>
<td>1.00</td>
<td>0.96</td>
<td>1741</td>
</tr>
<tr>
<td>system.log</td>
<td>1.00</td>
<td>0.91</td>
<td>0.95</td>
<td>1319</td>
</tr>
<tr>
<td>wifi-08-23-2018__12:54:38.121.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1091</td>
</tr>
<tr>
<td>wifi-08-24-2018__09:09:14.458.log</td>
<td>0.28</td>
<td>0.28</td>
<td>0.28</td>
<td>116</td>
</tr>
<tr>
<td>wifi-08-24-2018__12:47:32.191.log</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td>949</td>
</tr>
<tr>
<td>wifi-08-28-2018__14:27:47.184.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>920</td>
</tr>
<tr>
<td>wifi-09-03-2018__12:45:21.309.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>762</td>
</tr>
<tr>
<td>wifi.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1722</td>
</tr>
<tr>
<td></td>
<td><strong>avg / total</strong></td>
<td><strong>0.97</strong></td>
<td><strong>0.97</strong></td>
<td><strong>11287</strong></td>
</tr>
</tbody>
</table>

**Accuracy: 0.97**
### NearestCentroid

```python
[[2004 0 16 312 13 3 34 30 15 109]
 [ 0 131 0 0 0 0 0 0 0 0]
 [ 0 0 1387 354 0 0 0 0 0 0]
 [ 0 0 100 1216 0 0 0 3 0 0]
 [ 0 0 0 0 1091 0 0 0 0 0]
 [ 0 0 0 0 0 75 41 0 0 0]
 [ 0 0 0 0 0 105 844 0 0 0]
 [ 0 0 0 0 0 0 0 920 0 0]
 [ 0 0 0 2 0 0 0 0 760 0]
 [ 0 0 0 9 12 2 16 14 10 1659]]
```

<table>
<thead>
<tr>
<th>File</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>corecaptured.log</td>
<td>1.00</td>
<td>0.79</td>
<td>0.88</td>
<td>2536</td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>131</td>
</tr>
<tr>
<td>install.log</td>
<td>0.92</td>
<td>0.80</td>
<td>0.86</td>
<td>1741</td>
</tr>
<tr>
<td>system.log</td>
<td>0.64</td>
<td>0.92</td>
<td>0.76</td>
<td>1319</td>
</tr>
<tr>
<td>wifi-08-23-2018__12:54:38.121.log</td>
<td>0.98</td>
<td>1.00</td>
<td>0.99</td>
<td>1091</td>
</tr>
<tr>
<td>wifi-08-24-2018__09:09:14.458.log</td>
<td>0.41</td>
<td>0.65</td>
<td>0.50</td>
<td>116</td>
</tr>
<tr>
<td>wifi-08-24-2018__12:47:32.191.log</td>
<td>0.90</td>
<td>0.89</td>
<td>0.90</td>
<td>949</td>
</tr>
<tr>
<td>wifi-08-28-2018__14:27:47.184.log</td>
<td>0.95</td>
<td>1.00</td>
<td>0.98</td>
<td>920</td>
</tr>
<tr>
<td>wifi-09-03-2018__12:45:21.309.log</td>
<td>0.97</td>
<td>1.00</td>
<td>0.98</td>
<td>762</td>
</tr>
<tr>
<td>wifi.log</td>
<td>0.94</td>
<td>0.96</td>
<td>0.95</td>
<td>1722</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.91</td>
<td>0.89</td>
<td>0.90</td>
<td>11287</td>
</tr>
</tbody>
</table>

Accuracy: 0.89
RandomForestClassifier

[[2536 0 0 0 0 0 0 0 0 0]
 [ 0 131 0 0 0 0 0 0 0 0]
 [ 0 0 1738 3 0 0 0 0 0 0]
 [ 0 0 125 1194 0 0 0 0 0 0]
 [ 0 0 0 1090 0 0 0 0 0 1]
 [ 0 0 0 0 0 0 30 86 0 0 0]
 [ 0 0 0 0 0 0 95 854 0 0 0]
 [ 0 0 0 0 0 0 0 0 0 0 918 0 2]
 [ 0 0 0 0 0 0 0 0 0 0 0 761 1]
 [ 0 0 0 0 0 0 4 0 0 0 3 2 1713]]

<table>
<thead>
<tr>
<th>corecaptured.log</th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2536</td>
<td></td>
</tr>
<tr>
<td>fsck_apfs.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>131</td>
</tr>
<tr>
<td>install.log</td>
<td>0.93</td>
<td>1.00</td>
<td>0.96</td>
<td>1741</td>
</tr>
<tr>
<td>system.log</td>
<td>1.00</td>
<td>0.91</td>
<td>0.95</td>
<td>1319</td>
</tr>
<tr>
<td>wifi-08-23-2018_12:54:38.121.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1091</td>
</tr>
<tr>
<td>wifi-08-24-2018_09:09:14.458.log</td>
<td>0.24</td>
<td>0.26</td>
<td>0.25</td>
<td>116</td>
</tr>
<tr>
<td>wifi-08-24-2018_12:47:32.191.log</td>
<td>0.91</td>
<td>0.90</td>
<td>0.90</td>
<td>949</td>
</tr>
<tr>
<td>wifi-08-28-2018_14:27:47.184.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>920</td>
</tr>
<tr>
<td>wifi-09-03-2018_12:45:21.309.log</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>762</td>
</tr>
<tr>
<td>wifi.log</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
<td>1722</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>11287</td>
</tr>
</tbody>
</table>

Accuracy: 0.97
RidgeClassifier

[[2536 0 0 0 0 0 0 0 0 0]
 [ 0 131 0 0 0 0 0 0 0 0]
 [ 0 0 1739 1 0 0 0 0 0 0 1]
 [ 0 0 126 1190 0 0 0 3 0 0]
 [ 0 0 0 0 1091 0 0 0 0 0 0]
 [ 0 0 0 0 1 9 105 0 0 0 1]
 [ 0 0 0 0 1 24 923 0 0 0 1]
 [ 0 0 0 0 1 0 0 918 0 0 1]
 [ 0 0 0 0 0 0 0 0 761 0 1]
 [ 0 0 0 0 0 2 0 0 0 0 1 1719]]

precision  recall  f1-score  support

corecaptured.log  1.00  1.00  1.00  2536
fsck_apfs.log  1.00  1.00  1.00  131
install.log  0.93  1.00  0.96  1741
system.log  1.00  0.90  0.95  1319
wifi-08-23-2018__12:54:38.121.log  1.00  1.00  1.00  1091
wifi-08-24-2018__09:09:14.458.log  0.27  0.08  0.12  116
wifi-08-24-2018__12:47:32.191.log  0.90  0.97  0.93  949
wifi-08-28-2018__14:27:47.184.log  1.00  1.00  1.00  920
wifi-09-03-2018__12:45:21.309.log  1.00  1.00  1.00  762
wifi.log  1.00  1.00  1.00  1722

avg / total  0.97  0.98  0.97  11287

Accuracy: 0.98
What have I learned?

1. Get to know your data
2. Understand the algorithm
Code:

https://github.com/automationlogic/log-analysis
Thank You!

Questions?

Twitter: @thundercomb

Code: https://github.com/automationlogic/log-analysis