Machine Learning and Data Science for Performance and Quality Engineering

Gopal Brugalette
Principal Software Engineer
SAP Concur
A busy day @ SAP Concur
183,000 trips booked
409,000 expense reports
1 million mobile logins
760,000 mobile receipts uploaded

32,000 clients, 100 countries
Gopal Brugalette
Principal Engineer, Performance
Performance Engineering is a Data Science
What is Machine Learning?

Math enabling computers to do a what a human can-

Derive insights from data in a specific situation

1. $\nabla \cdot D = \rho_v$
2. $\nabla \cdot B = 0$
3. $\nabla \times E = -\frac{\partial B}{\partial t}$
4. $\nabla \times H = \frac{\partial D}{\partial t} + J$
Understand the problem, pick the algorithm

- What is the question?
- Machine learning algorithms
  - Supervised
    - Build a model using past data to make future predictions
  - Unsupervised
    - Understand the structure of the data, with no past data to compare
Regression analysis for prediction

Goal: Build a Model \( y=f(x) \) to understand customer experience

Feature Engineering:
1. Understand your data
2. Look for dimensions or features
3. How are they related?
Predict Response Time based on Load

![Graph showing the relationship between load count and response time.](image)

- **Response Time (ms)**
- **Load (Count)**
A little math

- Response time = \( f(\text{Load}) \)
- A linear model fits the data
  - \( y = a + bx \)
  - Examples

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y=x )</th>
<th>( y=2x )</th>
<th>( y=3+2x )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>10</td>
<td>13</td>
</tr>
</tbody>
</table>
A little more math

- **Response time = f(Load)**
- A linear model fits the data
  - $y = a + bx$
- **Count is x, Response time (pnn) is y**
- Solve for $a$ and $b$
  - $a = \frac{\sum y \sum x^2 - \sum x \sum xy}{n \sum x^2 - (\sum x)(\sum xy)^2}$
  - $b = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2}$
- **Response time = constant + factor * Load**
A little code

Train the models

Make a prediction

```r
for(model in rt_model_names) {
  f <- paste(model, "~", "Count")
  modelpnn <- paste(endpoint, model, sep = '')
  modelset[[modelpnn]] <- lm(f, data=model_numbers)
}

lastmonitorprediction <- t(data.frame(predict(modelset[[e_m]],
  data.frame(Count=monitor_numbers$Count), interval="prediction"))[2:3]))
```
Predictive Modeling of Response Times

Model predicts response times (red) based on measured load and compares it to monitored [actual] response times (blue)
Regression Model Use Cases

- Evaluate Changes in Production before peak load
- Less than optimal performance
- Normalize Performance for Load
- Detecting Outages
The hard parts

- Pick an algorithm & model
- Feature engineering
- Data wrangling
- Training set
- Productionizing
Clustering

- **Kmeans Clustering**
  - Unsupervised
  - Segments data by similarity of features into $K$ number of groups

- **Algorithm**
  - Select center for each of $K$ groups (centroid)
  - Assign each point to nearest centroid
  - Calculate new center as mean of points in the centroid
  - Iterate
A little code

17 serverinfo <- read.csv("submitreport.csv")
18 serverinfo <- serverinfo[, -3]
19 server.cluster <- kmeans(serverinfo[, 2], 2, nstart=20)
20 server.cluster

• 17 Read the data
• 18 Clean it up
• 19 Execute Kmeans for K=2
• 20 Print it out
KMeans Clustering of server performance

Lines represent a server

Automatically finds different groups
K = 3

K =3 identified unique clusters

- Red Cluster is Data Center Server Group A
- Blue Cluster is Data Center Server Group B
- Purple is Servers with Power Saving On
Clustering to look for multi-modal distributions
How many clusters?

hierarchical clustering

Elbow Method

Various Calculated Methods

<table>
<thead>
<tr>
<th>Number_clusters</th>
<th>KL</th>
<th>CH</th>
<th>Hartigan</th>
<th>CCC</th>
<th>Scott Marriot</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0000</td>
<td>5.0000</td>
<td>3.0000</td>
<td>2.0000</td>
<td>3.0000</td>
<td>5.0000</td>
</tr>
</tbody>
</table>

Libraries do it for you

```r
74 servers.nb <- NbClust(serverinfo$P50, distance = "euclidean", min.nc = 2, max.nc = 8, method = "ward.D2", index = "all")
75 num.clusters <- length(unique(servers.nb$Best.partition))
```
Big Data & Data Science

- Large data sets needed
- Visualization needed
- Where does it all go?
- Get it to people?
- More data is better?
The Use Case

**Question**
- How can we understand the Customer Experience (performance) over time?

**Requirements**
- Leadership reporting
- Long-term trending
- Agile/Dev-Op team accountability

**Approach**
- KPI’s and derived metrics
- Long Term Storage
- Easy Visualization
Approach Iterations

Dashboard Dump
- Too much data
- Too many questions

Apdex Overload
- What does it mean
- Where’s the insight

A simple metric, trended over time
- Peak hour performance, week over week
- Easy to get
- Easy to understand

Data
Information
Insight
Action
Long term trending of customer experience through key endpoint performance

- Response time Distributions (25%, 50%, 95%)
- Peak Hour – Monday Morning 7-8 AM PST
Solution Architecture

APM Monitors → R Analytics → Hadoop API → Hadoop Hive table → Tableau

Jenkins Job
Machine Learning for Outage RCA

• Question
  • Can we use ML for Root Cause Analysis and Prediction of major system outages?

• Premise
  • Application error logs contain sufficient information to detect an issue
  • Application error logs contain sufficient details to identify and distinguish between system failure modes

• But is this true?
Error Log Counts and Correlations
Form a Vector

• Count messages in small time slices
• Each time slice forms a message count vector
  • 3:00 am (0, 2, 2, 1, 3, 6, 6, 1, 260, ...)
  • 3:10 am (0, 0, 1, 0, 45, 3, 5, 11, 5, 0, 249, ...)
• Normalize for load
Message count vectors define events in a message space
Train the model

- Classify these points as different events
  - “Normal”, “DB Issue”, “App Server”, etc.
- Train the analysis engine to recognize these events
- Use Knn or other classifier to identify what type of event is occurring in real time
- Improve Root Cause Analysis
Classify and predict events

- Identify RCA
- Predict/Prevent Issues

classify/predict events based on trajectory
Can we group messages based on similarity?

Method:
- Clean messages
- Create a DTM (document term matrix)
- Kmeans- Clustering to group messages
- While this works, there is very little semantic similarity between messages.

Clustering them in this way was not valuable.
Regression Analysis for Response Time Prediction
Classification for RCA
Clustering for problem detection
Hadoop/R/Tableau for Deep Analytics

Gopal Brugalette
Principal Software Engineer
SAP Concur