Using AI to solve performance problems

Salesforce Performance Engineering
Jasmin Nakic | Jackie Chu

June, 2018
Salesforce Performance Engineering

Jasmin Nakic  Lead Performance Engineer
Jackie Chu   Lead Performance Engineer
Forward-Looking Statements

Statement under the Private Securities Litigation Reform Act of 1995:

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could affect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.
Presentation and Tutorial

Welcome

**Audience:** Sysadmins, performance engineers and developers

**Level:** Beginner

Introduction

Introduction to Predictive Performance Analytics

Machine Learning Use Case - Classification

Machine Learning Use Case - Trending Prediction

Hands-on

Prepare Input Data

Build and Compare Predictive Models

Generate Test Result Classification

Generate Dynamic Alerts

Summary

Q&A
Setup

Please follow instructions:

https://github.com/sfperfdemo/vel2018-ai
Machine Learning Use Case

Performance Test Result Classification
**Use Case:** Perf. Test Result Classification

- Hundreds of performance tests executed each day

<table>
<thead>
<tr>
<th>RUN NAME</th>
<th>PAGE TIME (MS)</th>
<th>CLIENT-SIDE TIME</th>
<th>TOTAL SERVER TIME</th>
<th>XHR COUNT</th>
<th>ACTION COUNT</th>
<th>DB TIME (MS)</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page A</td>
<td>2076</td>
<td>1674</td>
<td>2413</td>
<td>5</td>
<td>10</td>
<td>423</td>
<td>Success</td>
</tr>
<tr>
<td>Page A</td>
<td>2576</td>
<td>1862</td>
<td>2593</td>
<td>7</td>
<td>15</td>
<td>856</td>
<td>Regression</td>
</tr>
<tr>
<td>Page A</td>
<td>2013</td>
<td>1620</td>
<td>2324</td>
<td>5</td>
<td>10</td>
<td>455</td>
<td>Success</td>
</tr>
<tr>
<td>Page B</td>
<td>2637</td>
<td>1912</td>
<td>2554</td>
<td>8</td>
<td>19</td>
<td>952</td>
<td>Invalid</td>
</tr>
<tr>
<td>Page B</td>
<td>1896</td>
<td>1490</td>
<td>2295</td>
<td>5</td>
<td>11</td>
<td>587</td>
<td>Error</td>
</tr>
<tr>
<td>Page B</td>
<td>1978</td>
<td>1600</td>
<td>2182</td>
<td>5</td>
<td>10</td>
<td>622</td>
<td>Success</td>
</tr>
<tr>
<td>Page C</td>
<td>2537</td>
<td>1825</td>
<td>2417</td>
<td>15</td>
<td>25</td>
<td>877</td>
<td>Regression</td>
</tr>
<tr>
<td>Page C</td>
<td>1775</td>
<td>1362</td>
<td>2203</td>
<td>11</td>
<td>15</td>
<td>414</td>
<td>Success</td>
</tr>
<tr>
<td>Page C</td>
<td>1971</td>
<td>1594</td>
<td>2267</td>
<td>12</td>
<td>17</td>
<td>397</td>
<td>Success</td>
</tr>
</tbody>
</table>
Use Case: Perf. Test Result Classification

Many potential causes for variances

- Performance bugs in code
- Testing framework problems
- Testing Hardware issues
- Data problems
- New features introduced and causing changes
- ...many more
Use Case: Perf. Test Result Classification

Classification -> Time-consuming work

- Manually validating test result take > 2 minutes
- ~30% tests do not succeed
- Minimum 10 - 15 minutes to identify the issue
- To manually validate all tests executed -> 10 hours each day
  - More features == more test executions
  - Amount of work is constantly increasing!
  - Becoming exponentially difficult over time
Use Case: Perf. Test Result Classification

Sample Key Metrics (Performance for a web page)

- Total Page Time
- Browser Processing Time
- Server Processing Time (for each request)
- Chrome Timelines
- Action Time (for each action in server request)
- # of database operations
- Database process time
- # of api calls
- API process time
- ... many more
Use Case: Perf. Test Result Classification

How can Machine Learning help?

- Classify test-runs based on relevant features
- Help quickly identify potentially issues
- Provide insights on test-runs
  - E.g. “<XHR/API name> had a significant improvement/regression.”
  - E.g. “Database time is too high. There is potential hardware issue”
  - E.g. “Client side processing time is unstable. Potential Test device issue”
Use Case: Perf. Test Result Classification

Benefits

- Reduce time spent on manual process & allows engineers to focus on larger scale projects
- Automatically adapt and handle changes
- Quickly identifies potential cause of variance / issue
Hands-on Exercise
Decision Tree Classifier
Use Case: Perf. Test Result Classification

Training Dataset
- examples used for learning
- fit the parameters

For this exercise: PerfRun_TrainingData.csv

Testing Dataset
- independent of the training dataset
- follows the same probability distribution as the training dataset

For this exercise: PerfRun_TestData.csv
Use Case: Perf. Test Result Classification

Features (Simplified example for this exercise)

- PageTime_ms,
- TotalServerTime_ms,
- TotalBrowserTime_ms,
- Action_count,
- Api_count,
- Db_count,
- DbTime_ms,
- Xhr_count
Use Case: Perf. Test Result Classification

Basic Classifiers

- Exercise #1 – Decision Tree
  - Choose variable at each level that best splits the dataset
    - splits the data

```
  X1 < 25
  /     \\   
true false
/     \\   
X2 < 15 true false
|     |   |
|     |   |
X1 < 10 true false 
|     |   |
|     |   |
A     B     E     F
```
Hands-on Exercise

Decision Tree Classifier Model Fit
Use Case: Perf. Test Result Classification

Overfitting
- model is too flexible - performs well on the training data, but not on the testing data

Underfitting
- model is too simple - performs poorly on the training data
Use Case: Perf. Test Result Classification

Decision Tree Classifier

Tree Depth  
Model Flexibility  
Training Score  

Accuracy - Decision Tree

Decision Tree Train Score
Use Case: Perf. Test Result Classification

Decision Tree Classifier

- Overfitting & underfitting?
Use Case: Perf. Test Result Classification

Decision Tree Classifier

Min Sample / Leaf

Model Flexibility

Training Score

Accuracy - Decision Tree

Train Score

Test Score

Minimum Samples per leaf node
Hands-on Exercise

Random Forest Classifier
Basic Classifiers

- **Exercise #2 - Random Forest**
  - ensemble learning - constructing a multitude of decision trees at training time
  - Avoid overfitting to training set

*Use Case: Perf. Test Result Classification*
Machine Learning Use Case

Trending Prediction
Challenges in Performance Analytics

Understand
- Understand short and long term trends
- Find periodic pattern in performance metrics

Predict
- Make predictions using machine learning
- Detect anomalies and exceptions on current system performance data

Alert
- Generate alerts based on dynamic thresholds
- Estimate impact of difference between predicted and real values

Visualize
- Visualize predictions vs. actual metrics
- Deliver data to other teams and executives
Use Case: Predicting Performance and Detecting Anomalies

- Data show daily and weekly periodic patterns
- Long running trend with gradual increase
- Exceptions from typical daily metric shape
## Selecting the Feature Set

Features can be generated or collected from external systems.

<table>
<thead>
<tr>
<th>Timestamp Formats</th>
<th>Timestamp Components</th>
<th>Binary Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp as a string:</td>
<td>Year, Month, Day</td>
<td>isMonday, isTuesday, ..., isSunday</td>
</tr>
<tr>
<td><em>2016-05-08 19:00:00</em></td>
<td>Hours, Minutes, Seconds</td>
<td>isHour0, isHour1, isHour2, ..., isHour23</td>
</tr>
<tr>
<td>Ordinal date as decimal date</td>
<td>DayInWeek, WeekInYear</td>
<td></td>
</tr>
<tr>
<td>plus time:</td>
<td>Quarter</td>
<td><em>H_d_i_h_j</em> where d_i is for weekday</td>
</tr>
<tr>
<td><em>736092.791667</em></td>
<td></td>
<td>in range 0..6 and h_j is for hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in range 0..23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>isHoliday</td>
</tr>
</tbody>
</table>

Use ordinal time for long term trends

Commonly used in Analytics

Binary features have values 0 or 1

---

**Feature** is an individual measurable property of a phenomenon being observed. (Wikipedia)
Examples of linear regression models that apply to performance data

Simple linear regression model:
\[ y = c_0 + c_1 x \]

Model with periodic trigonometric features:
\[ y = c_0 + c_1 x + c_2 \sin(x \cdot 2\pi / f + t) \]
Where \( t \) is time offset and \( f \) is frequency

Model with binary features:
\[ y = c_0 + c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \]
Where \( x_n \) can be 0 or 1

Linear regression helps you find coefficient values \((c_0, c_1, \ldots, c_n)\)
Machine Learning Development Process

From raw data to predictions

- **Training Data**
- **Select Feature Set**
- **Build the Model**
- **Validate the Model**
- **Is Model OK?**
  - **Yes**
  - **No**
    - **Iterative Process**
    - Convert input data to a format accepted by ML tools
    - Measure the model score until it stops improving

- **Extract Feature Set**
- **Apply the Model**
- **Predictions**
- **Load Model**
How Good is the Model?

Generate the “score” for each model to see how well it fits input data

<table>
<thead>
<tr>
<th>Model</th>
<th>Training Score</th>
<th>Test Score</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Linear</td>
<td>0.0014</td>
<td>-0.0052</td>
<td>Not useful</td>
</tr>
<tr>
<td>Trigonometric</td>
<td>0.7284</td>
<td>0.7148</td>
<td>Periodic, but functions are complicated</td>
</tr>
<tr>
<td>Prediction Hour/Day</td>
<td>0.8280</td>
<td>0.8048</td>
<td>Each day has the same pattern</td>
</tr>
<tr>
<td>Prediction Hour/Week</td>
<td>0.9240</td>
<td>0.8918</td>
<td>Does not make good prediction on holidays</td>
</tr>
<tr>
<td>Prediction incl. Holidays</td>
<td>0.9520</td>
<td>0.9444</td>
<td>Does not make good prediction on Sundays</td>
</tr>
<tr>
<td>Prediction incl. Sundays</td>
<td>0.9522</td>
<td>0.9504</td>
<td>Optimal</td>
</tr>
</tbody>
</table>

- Models with more relevant features tend to have higher score
- Test data score is the measure of model quality
Visualization using Salesforce Analytics

Effective visualization of predictive data

➢ Compare actual metrics to prediction
➢ Display alerts for detected anomalies
➢ Correlate performance metrics with business data
➢ Build the dashboard to share with other teams and executives
Dynamic Alerts

Anomaly detection for near real time notification

- Typically alerts are based on static threshold values, for example higher than 85%.
- If the value is higher (or lower) then prediction based threshold, generate the alert.
- We want to separate data noise from anomalies using prediction interval.
- Cover false positives and false negatives if possible.
- Define the impact, for example: if a metric is for three hours more than $X$ in absolute value and more than $Y$ in percentage.
Anomaly Detection

Using predictive models

- Define “moving window”
  - Horizontal size – number of measures (for example: last 3 hours)
  - Vertical size – Allowed difference between the actual value and the prediction

- Anomaly Impact
  - High value – The metric value is higher than the prediction
  - Low value – The metric value is lower than the prediction

- Impact Thresholds – define if the anomaly needs immediate action
  - “Alert” – send notification and require immediate attention from operations
  - “Warning” – create a follow up action for technical or business experts
Implementation Summary

What did we cover today?

**Develop** predictive system applying **machine learning** methods
- Prepare the training data set,
- Build the predictive model,
- Predict the metric value for the test data set
- Measure the quality of the predictive model

**Analyze** input data and predictions to solve business needs
- Find short and long term performance trends
- Detect Anomalies in near real time
- Implement a simple alert system
- Visualize data using Salesforce Wave
Next Steps and Resources

Get Demo Scripts from GitHub
https://github.com/sfperfdemo/vel2018-ai

Get Salesforce Wave Utilities
https://github.com/forcedotcom/Analytics-Cloud-Dataset-Utils

Visit Salesforce Wave Tutorial
http://www.salesforce.com/analytics-cloud/overview/

Explore Machine Learning in Python (scikit)
http://scikit-learn.org
Q & A
Questions and Answers
THANK YOU