LAB WEEK

R&D Lab Week Project

Farm Job Anomaly detection

November 2020

About the Project

Contributors: Ryan Amundson

Overview:

Attempt to use ML to detect anomalous farm jobs

Motivations

Current farm alerting is a bit hard to explain and understand. It’s also running with old Scala code which is quite scary.

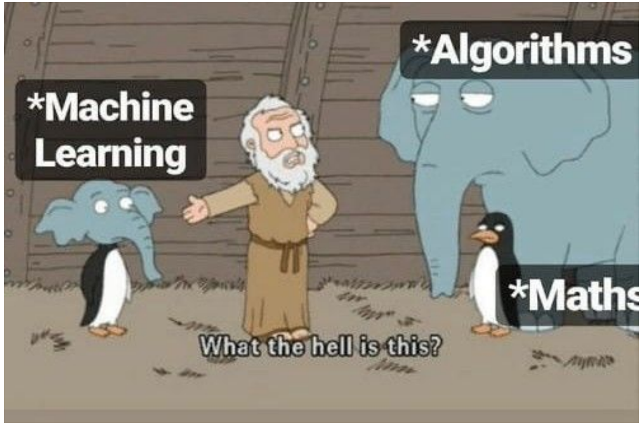
I’ve attempted to use some machine learning tools to find anomalies in farm jobs instead.

This also just gave me the opportunity to learn a bit about some different techniques for finding anomalies in data.

How

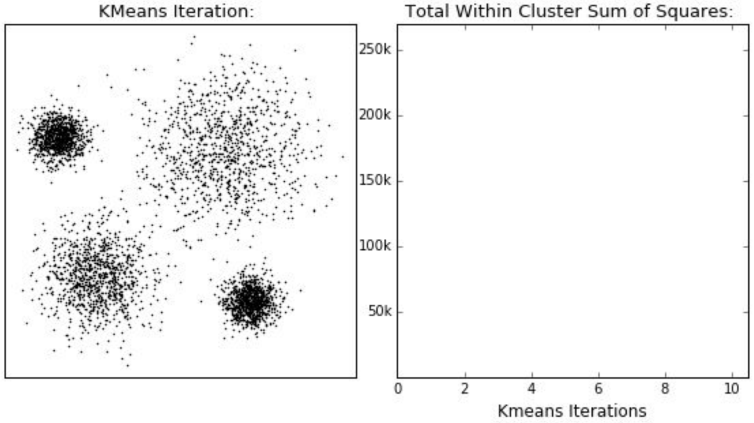
First attempt was to use k-means clustering to auto-categorize farm jobs. Then I looked for jobs that should be similar, but instead are put into separate categories by k-means.

Secondly, I used an isolation forest and looked at low distance values from the results.



K-means Clustering

- Unsupervised learning algorithm: (IE, you don’t have to give it a goal, it just splits data into a specified number of categories)
- Pros: Well known algorithm, so it’s easy to find in many ML toolsets
- Cons: Getting the right number of clusters is hard, feels a bit like guesswork

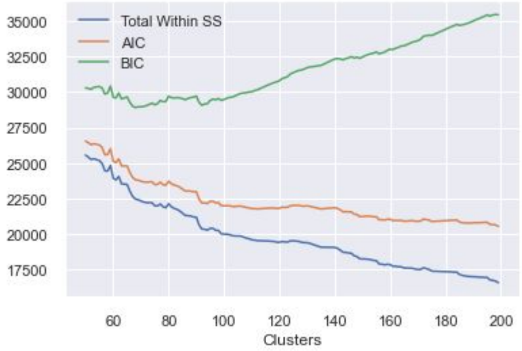


K-means Detection Methodology

- Pull farm job data using SQL query
- Train multiple H2OKMeansEstimators using farm data
- Pick an H2OKMeansEstimator where the total sum

  of squares have flattened and either AIC or BIC has

  started in increase (Seems a little like guessing to me)



- Re-pull farm job data and get a prediction for which category the jobs are assigned
- Check individual nodes for any jobs that contain a frame that has a single, separate category from the majority and then flag those jobs as variant

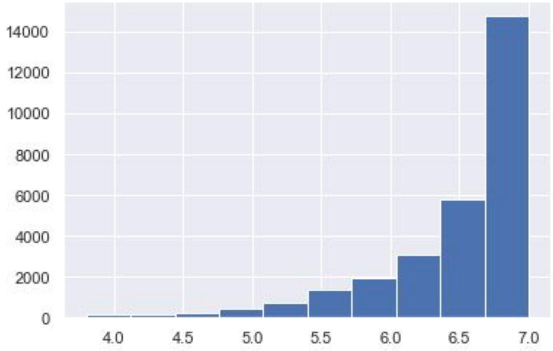
K-means Clustering Results

- Checking what came from k-means clustering had some success, for instance these jobs were flagged by that methodology, but had no alert in our system: http://farm/groups/group/123705034#jobSetId=6&jobId=136&filterName=Nodes&filterUser=ramundson http://farm/groups/group/123685478#jobSetId=13&jobId=207&filterName=Nodes&filterUser=ramundson

- Just spot checking, there were definitely some false positives among some true positives. When checking against our current system, there was only an overlap of 6 jobs flagged by our current system against 122 flagged w/k-means

2nd Methodology: Isolation Forest

- Unsupervised learning algorithm, specifically used for anomaly detection
- Used H2OIsolationForestEstimator to find anomalies
- Very similar to a random forest, it creates an ensemble of decision trees
- With the H2O estimator, anomalies will return a shorter path length
- Much easier to use for anomaly detection just look for lower distance values
- Still requires a little magic for picking a suitable distance value, but I just picked values in the lower 5% for anomalies.

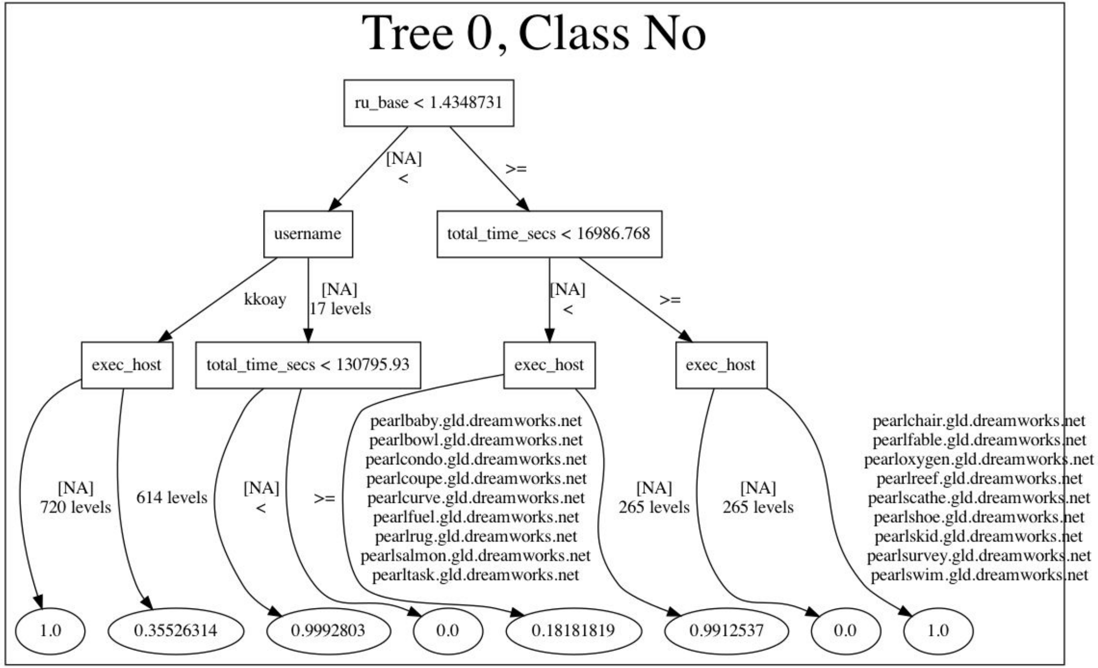


Isolation Forest results

- Overlap with actual alerts by our current system is very high, with 349 anomalies and 308 have had alerts (repeated runs maintain a 90% overlap)
- Most of the overlap with was with assumed stuck, long running alerts (Where job has 55% less CPU efficiency and longer runtime then neighbor jobs)
- With added data and some additional tuning, it seems like a pretty promising method to find potentially problematic jobs

Isolation Forest Results cont.

Looking at the decision tree can give you more insights:


Tree 0, Class No
ru_base < 1.4348731
[NA]
<
>=
username
total_time_secs < 16986.768
kkoay
[NA]
17 levels
[NA]
<
>=
exec_host
total_time_secs < 130795.93
exec_host
exec_host
[NA]
720 levels
614 levels
[NA]
<
>=
pearlbaby.gld.dreamworks.net
pearlbowl.gld.dreamworks.net
pearlcondo.gld.dreamworks.net
pearlcoupe.gld.dreamworks.net
pearlcurve.gld.dreamworks.net
pearlfuel.gld.dreamworks.net
pearlrug.gld.dreamworks.net
pearlsalmon.gld.dreamworks.net
pearltask.gld.dreamworks.net
[NA]
265 levels
[NA]
265 levels
pearlchair.gld.dreamworks.net
pearlfable.gld.dreamworks.net
pearloxygen.gld.dreamworks.net
pearlreef.gld.dreamworks.net
pearlscathe.gld.dreamworks.net
pearlshoe.gld.dreamworks.net
pearlskid.gld.dreamworks.net
pearlsurvey.gld.dreamworks.net
pearlswim.gld.dreamworks.net
1.0
0.35526314
0.9992803
0.0
0.18181819
0.9912537
0.0
1.0

Project Findings

Some of my current takeaways are that using some automated anomaly detection could be useful to augment our current variant job detection methods, but perhaps not entirely replace them.

Isolation Forests also seem to outperform k-means clustering as far as ability to detect legitimate variant farm jobs and we should look into that method further.

For project next steps, I would like to see if I could add the model to ForestFlow and then add an additional alert for these anomalous jobs to the farm UI and see if people find it helpful.

All work was done was done using Juypter notebook, code is available here for the clustering method and here for Isolation forests

Q & A

Thanks so much!!

Questions?