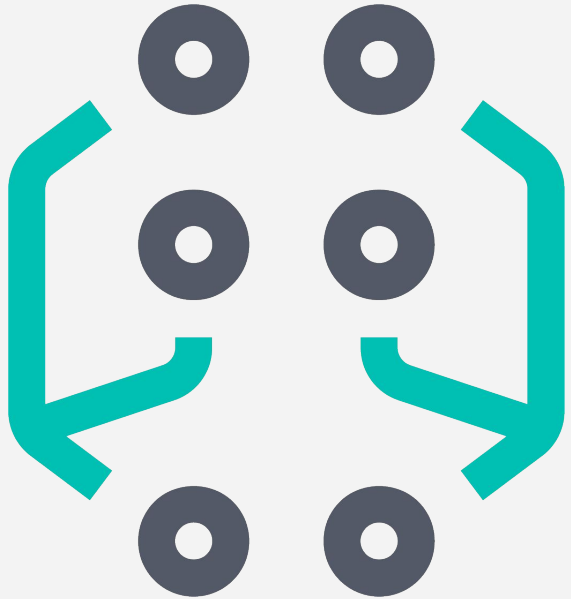
机器学习在 Elasticsearch 中的应用

刘晓国
Elastic 社区布道师
2020年9月5日

Elastic Machine Learning

使数据科学可操作化及简单化



elasticstack.blog.csdn.net

Elastic 产品生态

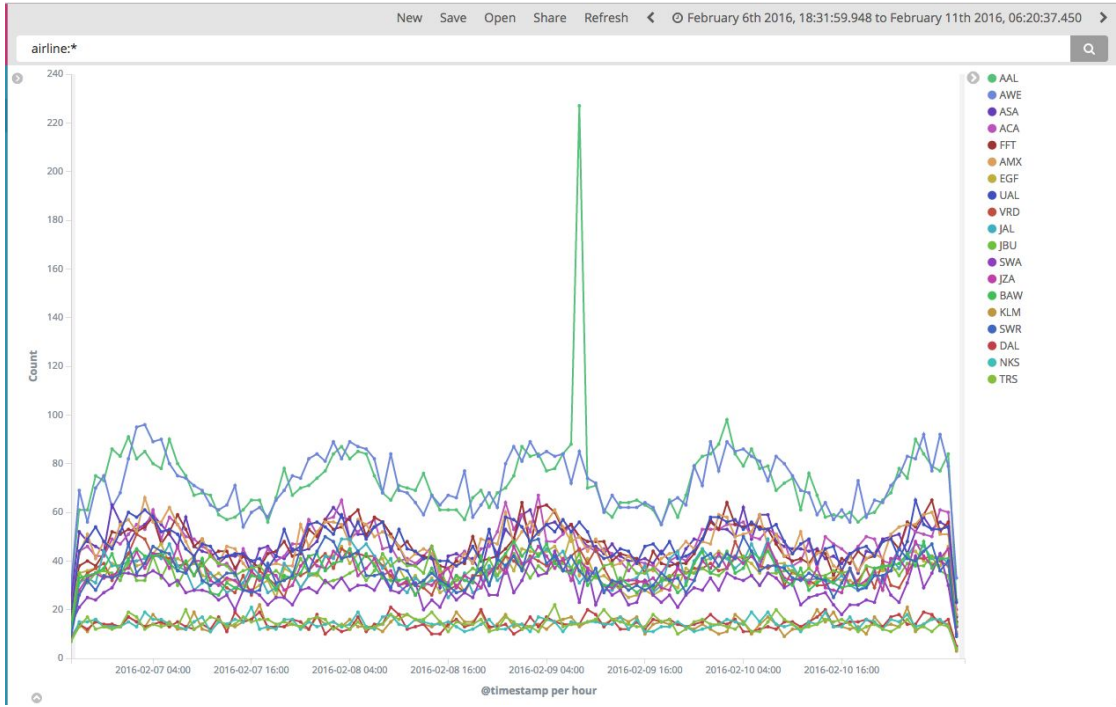
术语

- ***Machine Learning/机器学习***
  - 是一个很广泛的词，但是 Elastic Machine Learning 是针对时序数据的自动异常检测和预测
- ***Anomaly Detection/异常检测***
  - 发现什么是“怪异”或“不同”，而不是什么是“不良”
- ***Unsupervised Learning/非监督学习***
  - 在没有人为标记的例子的情况下学习（无需“教”）。仅依靠数据
- ***Bayesian***
  - 一种基于概率的方法，在该方法中，先前的结果用于计算某些当前或将来事件的概率

什么是“异常”?

在右边的图什么地方是异常?

为什么?



什么是 “异常”?

下面的图有啥异常之处?

为什么?

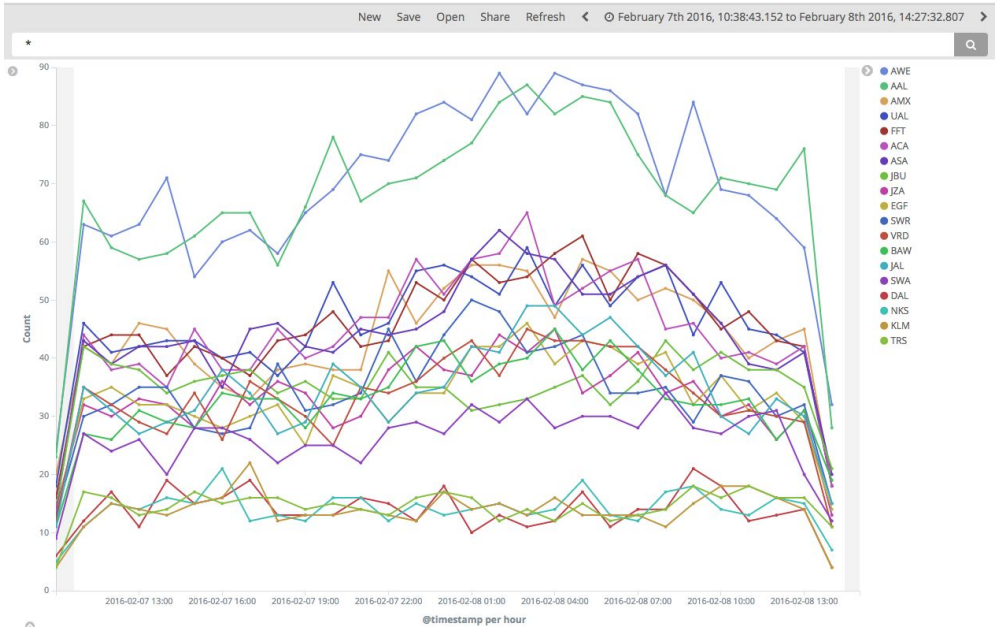
elastic

什么是异常？

通常，可以通过两种方式回答此问题：

1) 随着时间的流逝，某些事情会以自己一致的方式表现

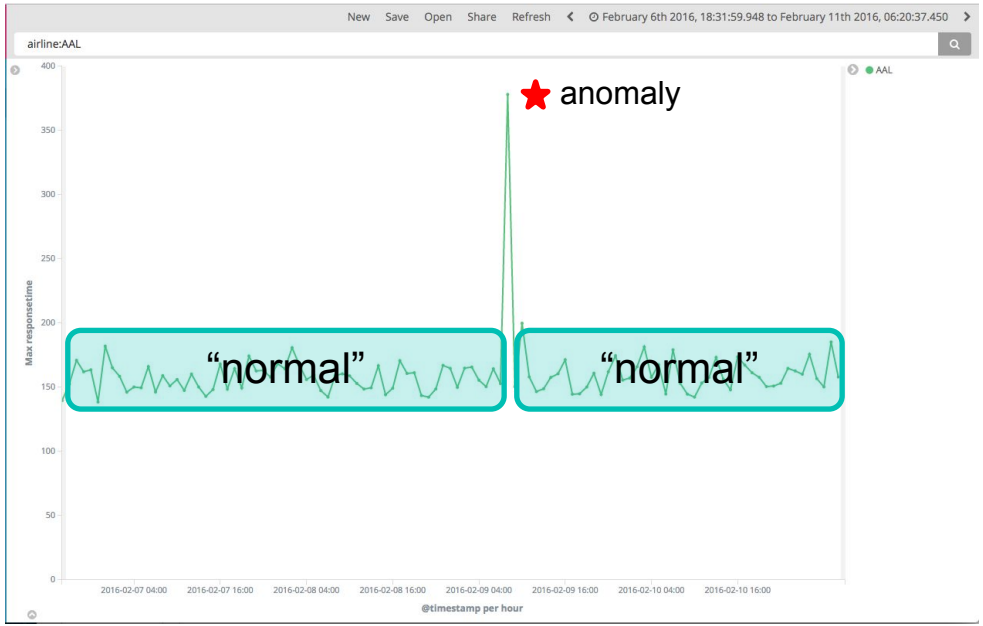
2) 与类似实体相比，某些事物的行为方式一致



elastic

什么是异常？

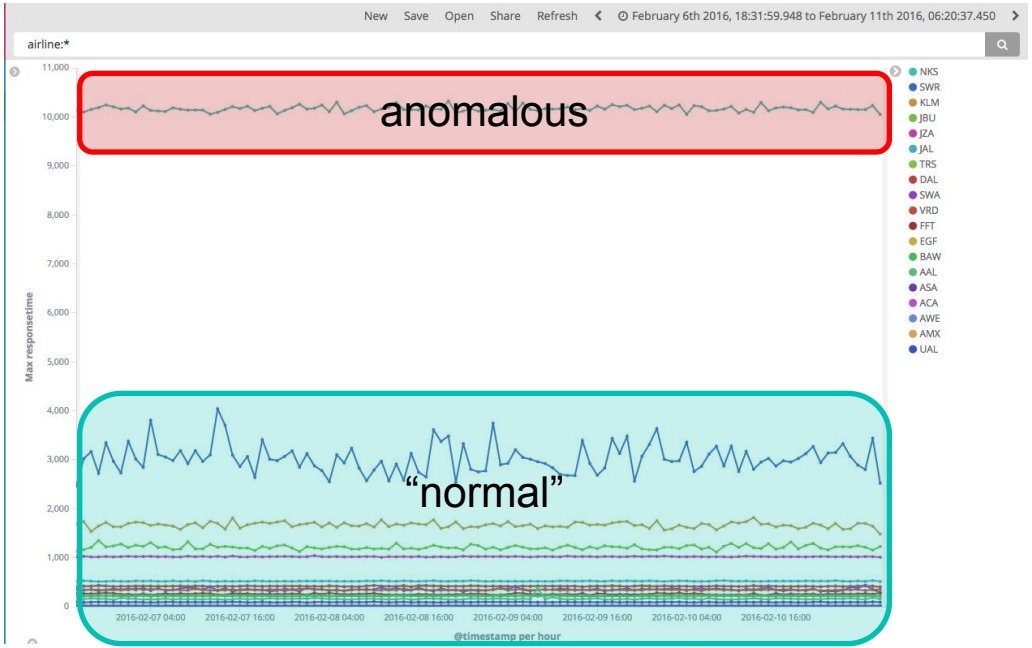
1) 如果某件事改变了其行为，那么与它自己的历史相比，那就是**异常**



elastic

什么是异常？

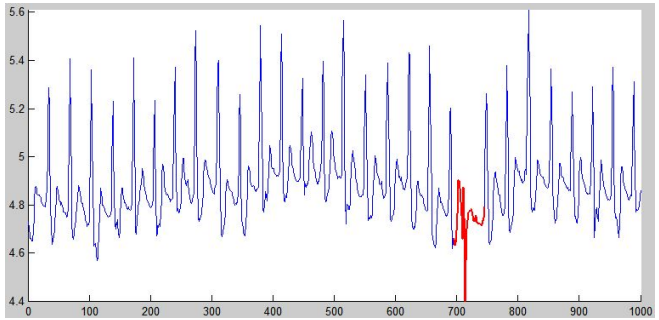
2) 如果某个事物与总体中的其他事物完全不同，则该实体是**异常的**。



总结来说，异常就是

1) 当实体的行为突然发生重大变化时

2) 当实体与总体中的其他实体完全不同时

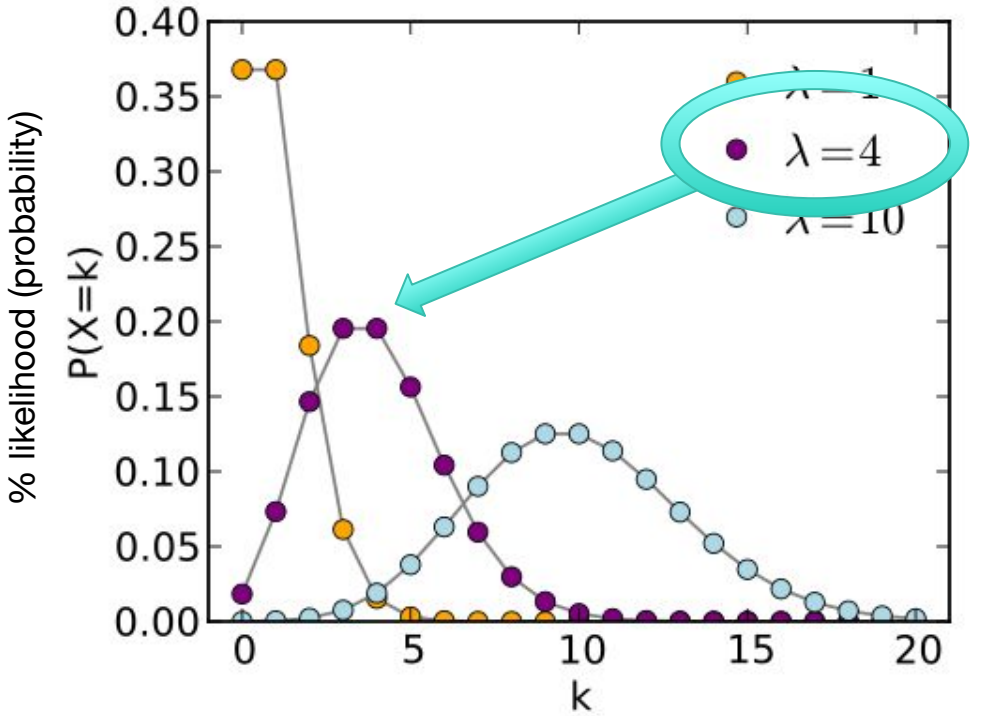


一个比喻

- 我如何得知你每天收到多少邮政邮件，以及我如何使用这些信息来预测明天你可能收到多少邮件？

elastic

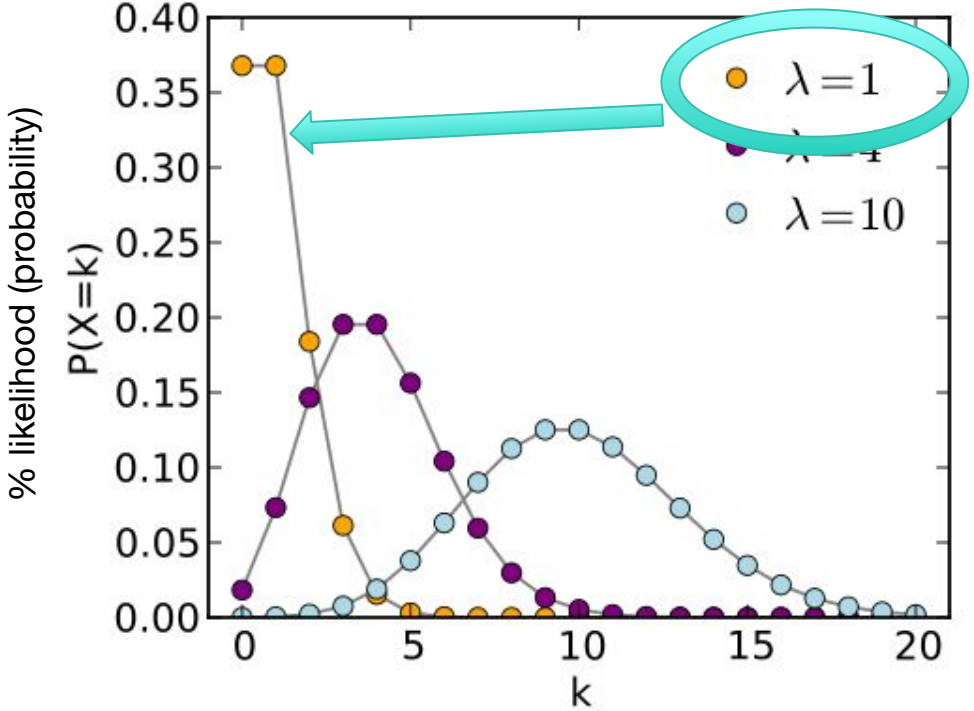
概率分布函数



针对某个人来说

elastic

概率分布函数



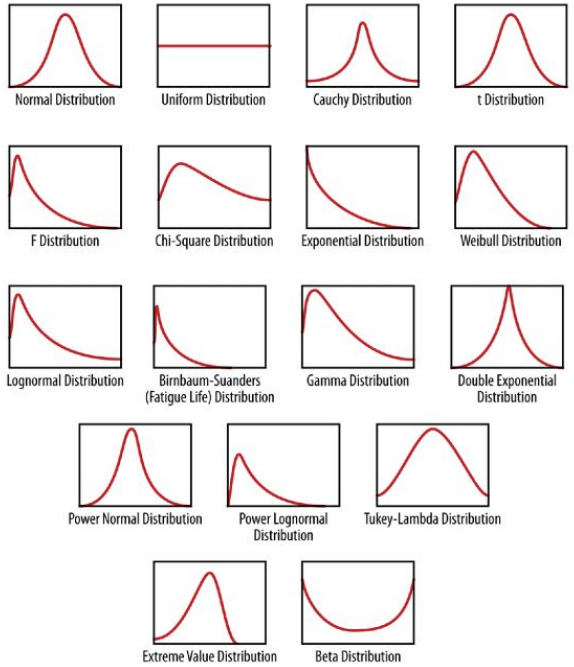
针对大学学生来说?

每天收到的邮件数量

elastic

如何挑选模型?

到底哪个适合你的
数据呢？

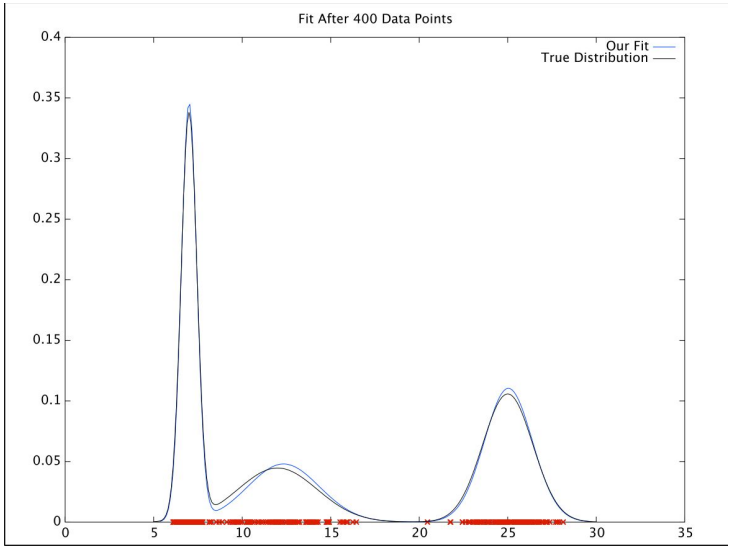


source: “Doing Data Science”
O’Neil & Schutt

elastic

机器学习为您选择

- ML使用复杂的机器学习技术来为您的数据最佳地拟合正确的统计模型。
- 更好的模型=更好的 outlier 检测=更少的误报
- 在低概率区域中观察时发生异常

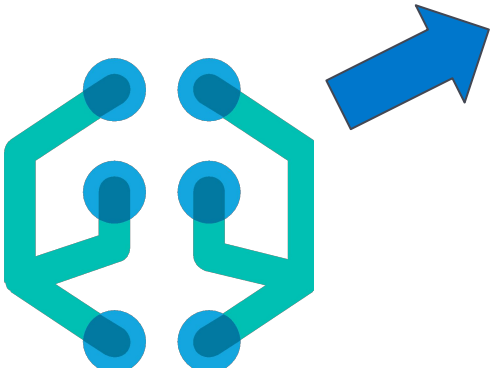


为什么要机器学习？

- 随着数据集的大小和复杂性的增加，检查仪表板或维护发现基础结构问题，网络攻击或业务问题的规则所需的人工工作变得不切实际。诸如异常检测和异常检测之类的 Elastic 机器学习功能使在人为干扰最小的情况下更容易注意到可疑活动
- Elastic机器学习异常检测功能可实时自动建模时间序列数据的正常行为（学习趋势，周期性等），以识别异常，简化根本原因分析并减少误报。异常检测在 Elasticsearch 中运行并随其扩展，并且在 Kibana Machine Learning 页面上包括一个直观的 UI，用于创建异常检测作业并了解结果。

7.6 版本之前的 Machine Learning

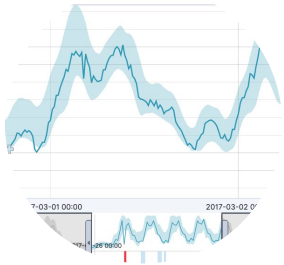
Unsupervised learning



Anomaly Detection
Outlier Detection
Forecasting

Data Driven:
Pattern
Recognition

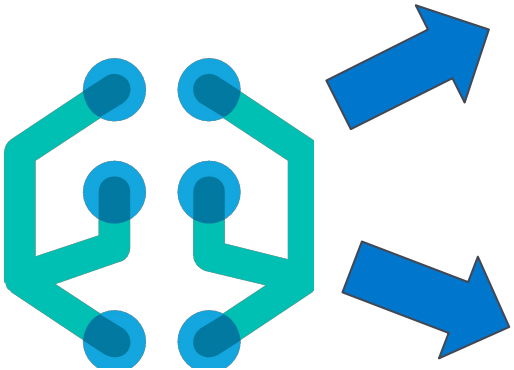
Unsupervised



elastic

机器学习扩展用例

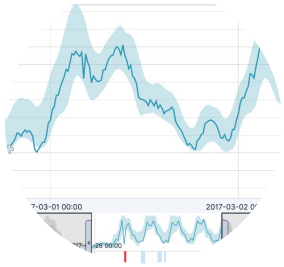
End-to-End Supervised Learning in 7.6



Anomaly Detection
Outlier Detection
Forecasting

Unsupervised

Data Driven: Pattern Recognition



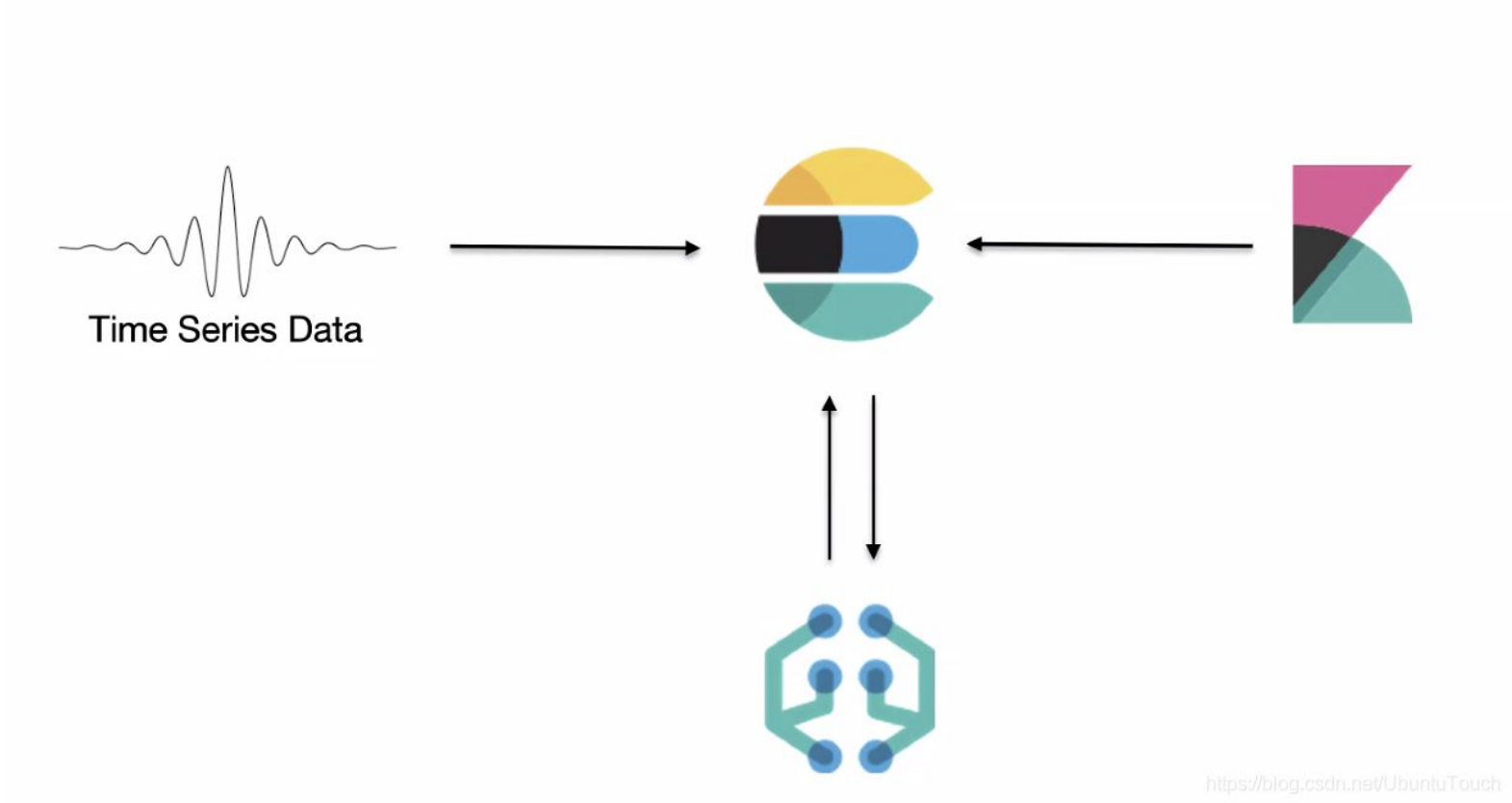
Language ID
Fraud Detection
User classification

Supervised

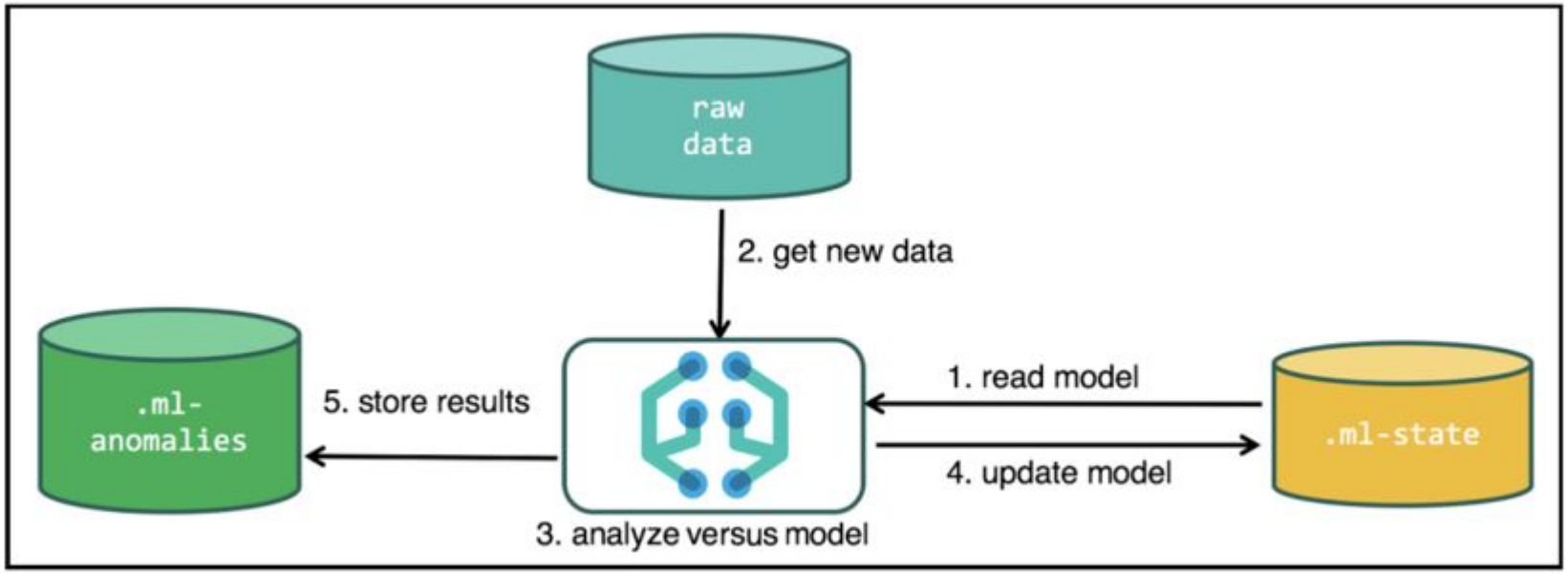
Labelled data for Learning and Predicting

elastic

机器学习是如何操作的？(1/3)



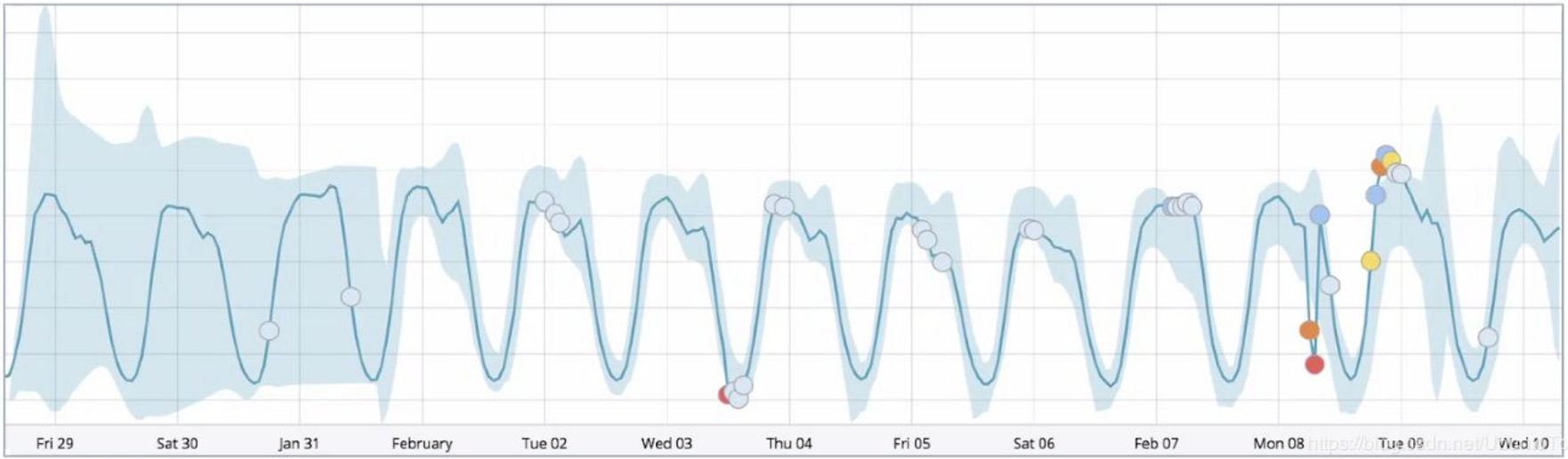
机器学习是如何操作的？(2/3)



Simplified sequence of ML's procedures per bucket_span

elastic

机器学习是如何操作的？(3/3)



Elastic 使它更容易，更有效

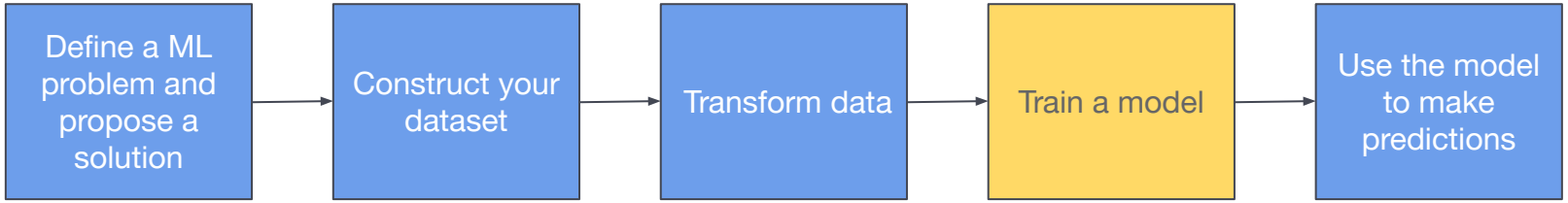
提供可操作性的端到端路径的 ML

1	2	3	4	5
Elastic Ingest & Transforms	Transforms & Kibana	Data Frame Analytics	Inference Ingest Pipeline	Data Frames & Inference

elastic

Machine Learning 端到端技术

Build a model on historical data that has a churn indicator



训练/验证/测试

	customer a	customer b
total duration of customer sessions	80:21:07	1:01:11
tv episodes watched	24	1
films watched in last month	5	0
newness of titles watched in last month	9.8	1.2
Change in duration	6:22:17	16:43:29
subscription plan	gold	platinum
customer tenure	32	26
has churned?	no	yes

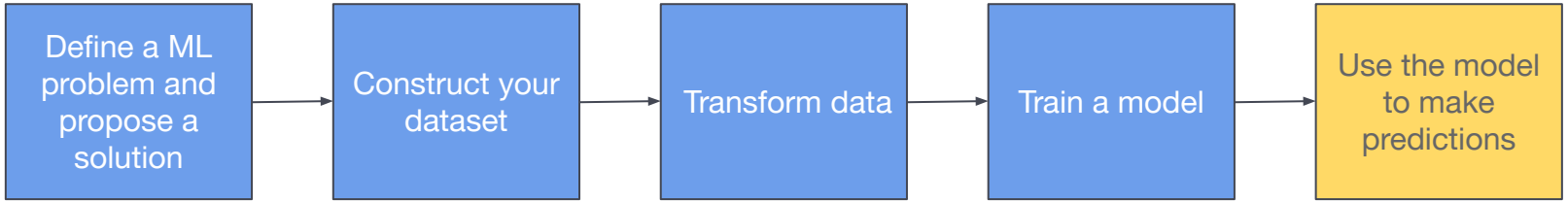
ML
Supervised
Model

```
Model Name: churn_e2r21
Model Precision: 96.3%
Model Recall: 95.7%
Model F1 score: 96.0%
```


elastic

Machine Learning 端到端技术

Use model inference to make predictions on streaming data



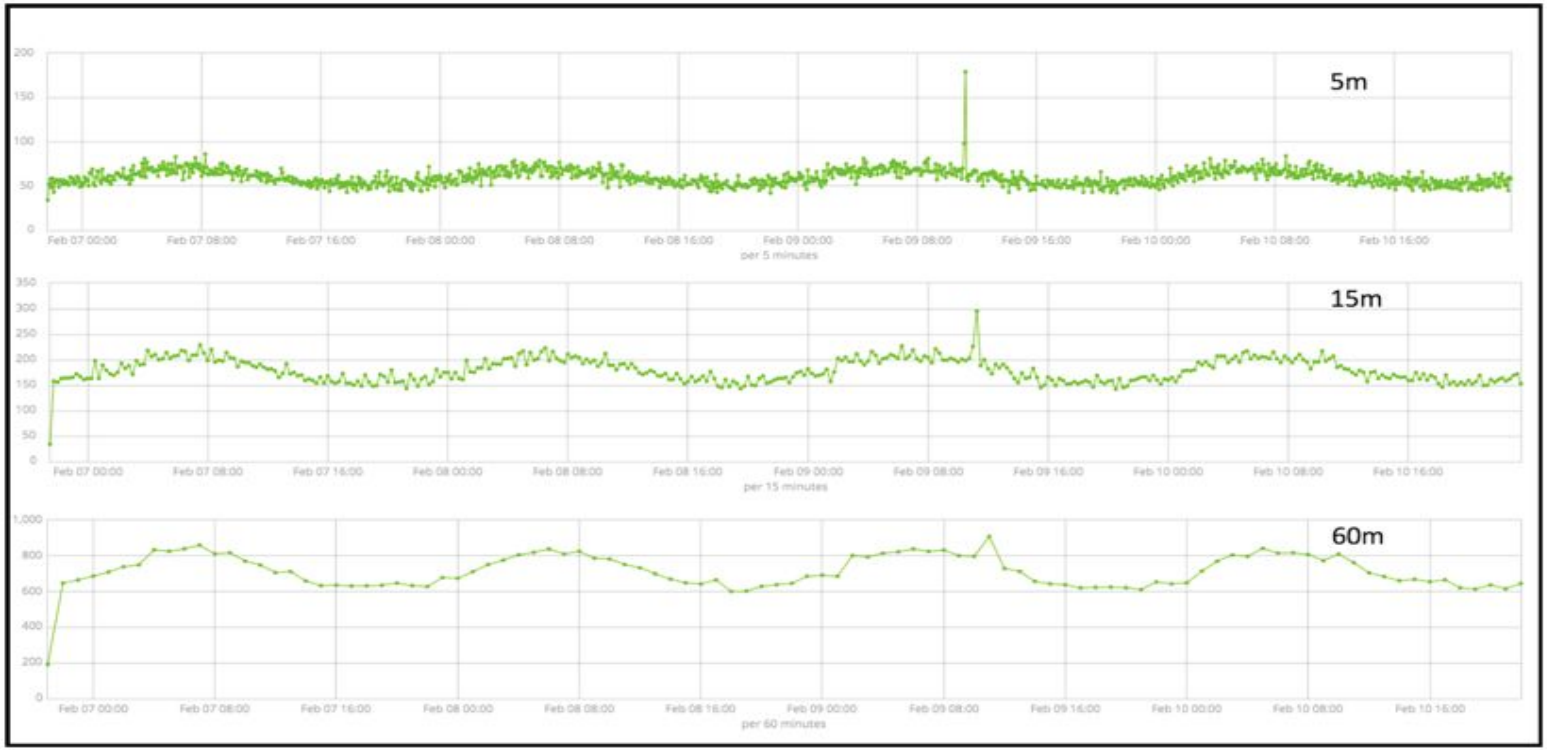
预测

	customer c
total duration of customer sessions	10:10:06
tv episodes watched	2
films watched in last month	1
newness of titles watched in last month	1.6
change in duration this month	17:22:17
customer plan	gold
customer tenure	5

	customer c	Feature Influence
total duration of customer sessions	10:10:06	0.1
tv episodes watched	2	0.8
films watched in last month	1	0.8
newness of titles watched in last month	1.6	0.01
change in duration of this month	17:22:17	0.6
customer plan	gold	0.01
customer tenure	5	0.1
will churn?	p(churn) = 97%	

elastic

Bucket_span 的影响



Aggregations of the same data over three different time intervals

elastic

学习资源

- Elastic：机器学习的实践 - single metric job
  - https://elasticstack.blog.csdn.net/article/details/102788922
- Elastic：机器学习的实践 - multi metric job
  - https://elasticstack.blog.csdn.net/article/details/106941847
- Elastic：机器学习的实践 - population job
  - https://elasticstack.blog.csdn.net/article/details/106950196
- Elastic：机器学习的实践 - categorization
  - https://elasticstack.blog.csdn.net/article/details/106984151
- Elastic：使用 Elastic 有监督的机器学习进行二进制分类
  - https://elasticstack.blog.csdn.net/article/details/107759860

THANK YOU