CSCI E-103

Data Engineering for Analytics to Solve Business Challenges

Data Concerns leading to model concerns

Lecture 10

Anindita Mahapatra & Eric Gieseke

Harvard Extension, Fall 2025

Agenda

- Model Bias
- Class Imbalance
- Anonymity Concerns
- Feature Computation for fraud detection
  - Credit card transactions
  - Anti Money Laundering (AML)

- Lab
  - Handle data imbalance

AutoML

Glassbox ML

EDA -> model selection with interoperability

Classification models	Regression models	Forecasting models	Forecasting models (serverless)
Decision trees	Decision trees	Prophet	Prophet
Random forests	Random forests	Auto-ARIMA (Available in Databricks Runtime 10.3 ML and above.)	Auto-ARIMA
Logistic regression	Linear regression with stochastic gradient descent		DeepAR
XGBoost	XGBoost		
LightGBM	LightGBM		

Review Previous Material

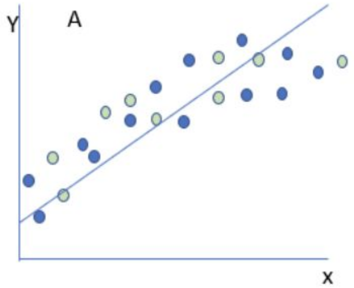
From Big Data to ...	Good Data
Data centric approach to ML better than	Model Centric
No DS/ML project can start without a	Business problem at hand
3 types of model inferencing include	Batch, Streaming, REST EP
SHAP is for	Model explainability
Ensemble is a combination of models	To improve model robustness & accuracy
Improvement to grid search is	Bayesian search
ML Pipeline consists of stages of	Transformers & Estimators
Transformers call transform() & estimators call	fit()
How can you do data versioning	Delta time travel/versioning feature

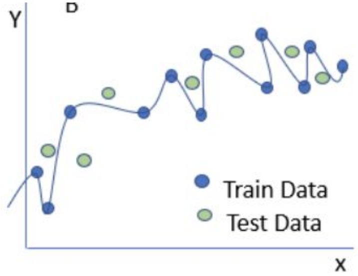
Errors in Machine Learning

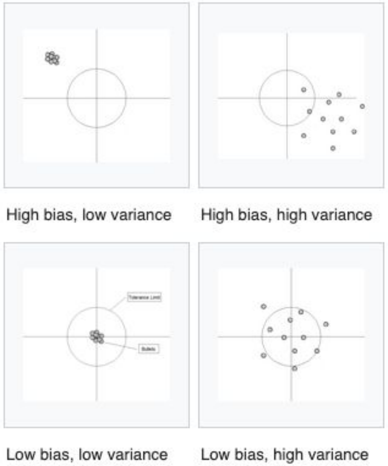
- Bias:
  - Defined as the difference between the **Predicted** and **Expected** values
  - ML is unable to capture the true relationship between the features and target
  - Ex. underfitting
- Variance
  - Result of the model making too complex assumptions
  - Ex. overfitting
- Irreducible
  - Random in nature and not directly controlled by the model

Prediction error = Bias error + Variance error + Irreducible error

Increasing Bias reduces Variance & vice-versa



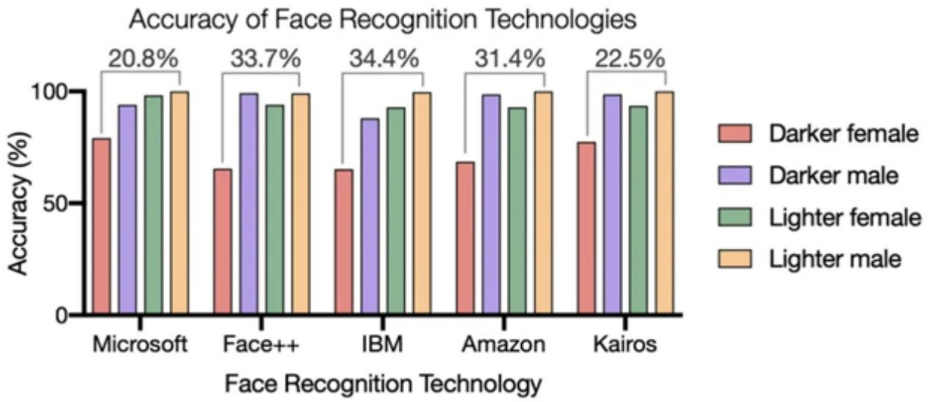




Model Bias

How does it happen?

- Human factor
    - m/c mimic human cognitive bias
- Poor quality training data
    - Deep Learning Neural Nets learn from input data
- Model performance mismatch
    - Low resolution data



Source: [NIST study findings published by Harvard University](#)

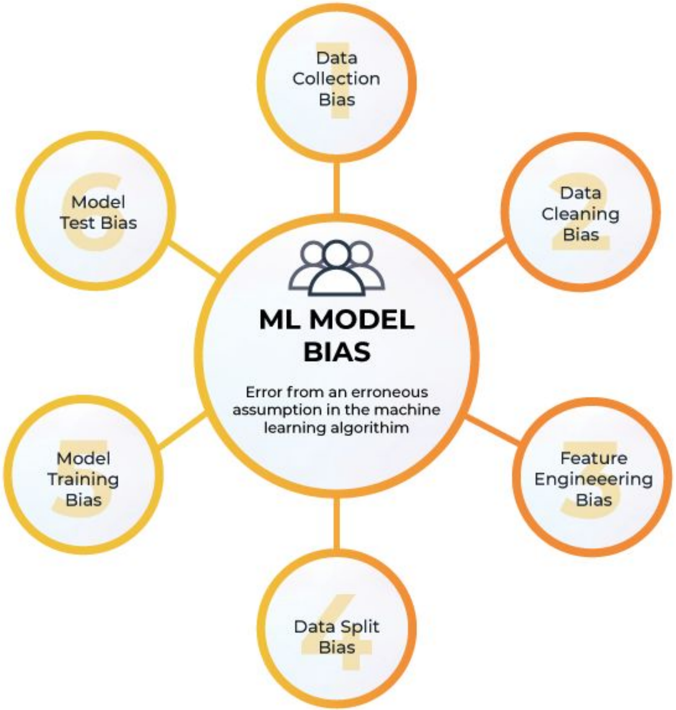
eft: Dog (Volpino Italiano breed), Right: Arctic fox.

Source: [freeCodeCamp](#)

Source: [ML study by Carnegie Mellon University](#)

Ways to Reduce Bias Errors

- Change the model
  - Use different models to vet the outcome
    - Tree algorithm better at handling bias
  - Use appropriate models
    - Do not use a Linear model if features and target of your data do not in fact have a linear relationship
- Ensure the data is truly representative
  - Ensure training data is diverse and represents all possible groups or outcomes
  - For an imbalanced dataset, use weighting or penalized models
- Extensive hyper parameter tuning



Ways to Reduce Variance Errors

- Ensemble learning
  - Train with multiple models
  - Leverage both weak and strong learners in order to improve model prediction
- Train with larger data sets
  - More data increases the data to noise ratio which reduces the variance of the model
  - With more data, model is better able to come up with a general rule which will also apply to new data

Ways to reduce model errors

- Choose the correct learning model
  - Supervised: controlled entirely by the stakeholders who prepare the dataset
  - Unsupervised: depends on the neural network itself
- Use the right training data set
  - Do not reuse datasets – for example, data from an area with an ethnically diverse population cannot be applied to a region with predominantly a single race.
- Perform data processing mindfully
  - Not just training, but pre-processing, in-processing(weights), post-processing (interpretation)
- Monitor real world performance
  - Use of real-world data for testing ML wherever possible
  - Frequent training
- Address infrastructural issues
  - Data collection process scrutiny

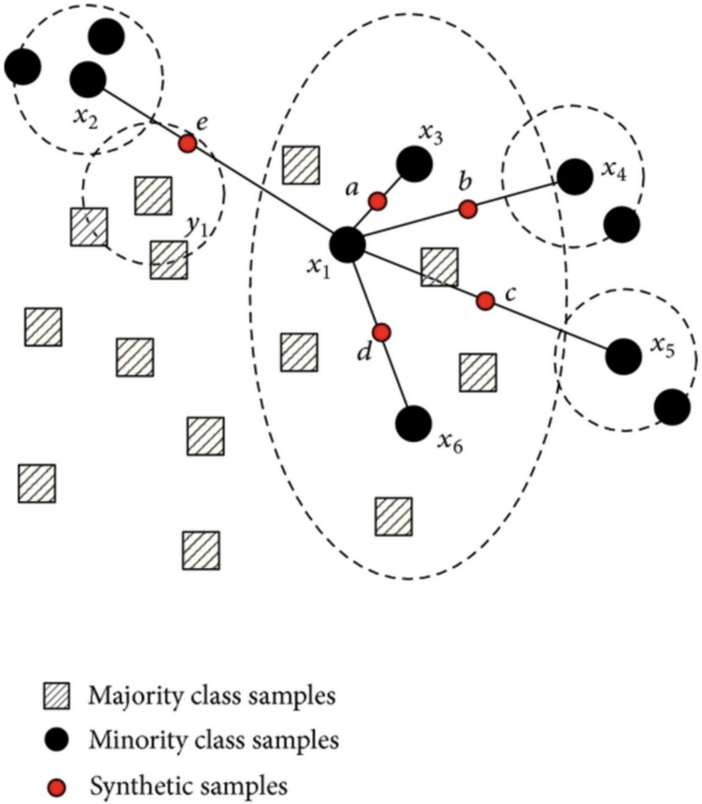
Imbalanced data

- An **imbalance** occurs when one/more classes have low proportions in the training data as compared to other classes.
- Most machine learning algorithms for classification predictive models are designed and demonstrated on problems that **assume** an **equal distribution** of classes - designed to maximize accuracy and reduce error.
- Consequences
  - If the data set is imbalanced the model will be Biased
- How do you detect imbalance
  - Check count of the dependent categorical values
- Imbalance introduces Majority & Minority Class
  - Some modest, others severe cases
  - Applies to multi-class as well
- Classification problems that can have a severe imbalance in the class distribution across industry verticals include:
  - Fraud, Claim, Anomaly, Intrusion Detection

Ways of dealing with Imbalance

- Augment performance metric (beyond just accuracy)
  - Confusion Metric (not just correct results but incorrect ones as well)
  - Precision (Positive Predictive Value) => TP/(TP+FP)
  - Recall (True Positive Rate) signifies completeness/sensitivity => TP/(TP+FN)
  - F1 Score (weighted average of Precision and Recall) => 2*P*R/(P+R)
- Different Algorithm
  - Tree based help in imbalance scenarios
- Resampling techniques (SMOTE, ADASYN)
  - Oversampling of minority class & undersampling of majority class
  - Generate synthetic samples
  - Always split into test and train sets BEFORE trying oversampling techniques!
    - SMOTE or Synthetic Minority Oversampling Technique

Synthetic Data Generation



SMOTE Technique

Package: imbalanced-learn

First it finds the n-nearest neighbors in the minority class for each of the samples in the class . Then it draws a line between the the neighbors an generates random points on the lines.

Anonymity Concerns

- PII data
- Wrong correlations
  - Remove identifier fields before feeding it to the model
  - User id, POS id and similar fields
- Data Classification

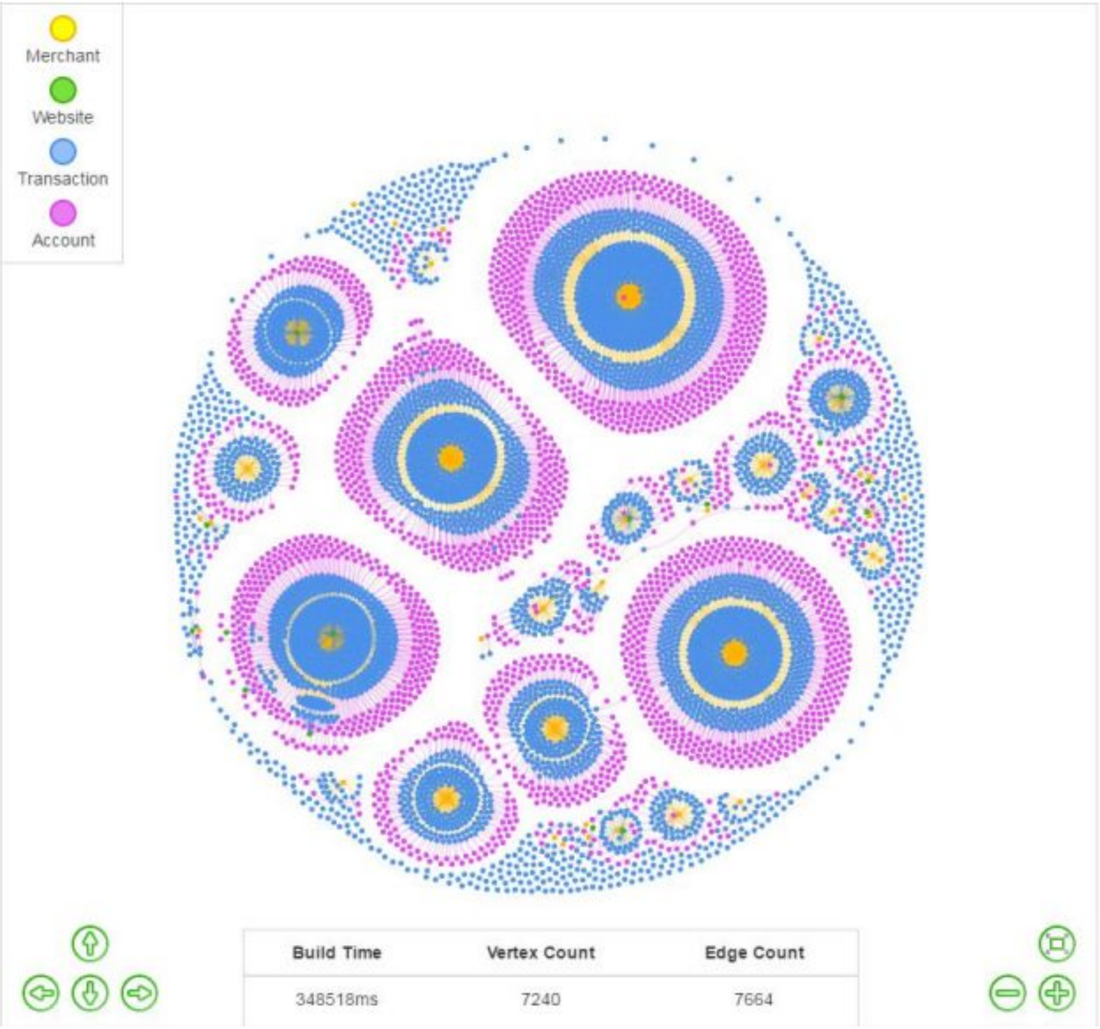
Feature Computation

For Fraud Detection

Fraud Detection Requirements

- Determine if a transaction is fraudulent with a high confidence
- Latency: 15 millisecond processing time per transaction
- Throughput: 50,000 transactions per second
- Configurable Rule and Feature Management
- Support ad hoc deployment and computing of new features, rules, and models

Graphical View of Payment Transactions



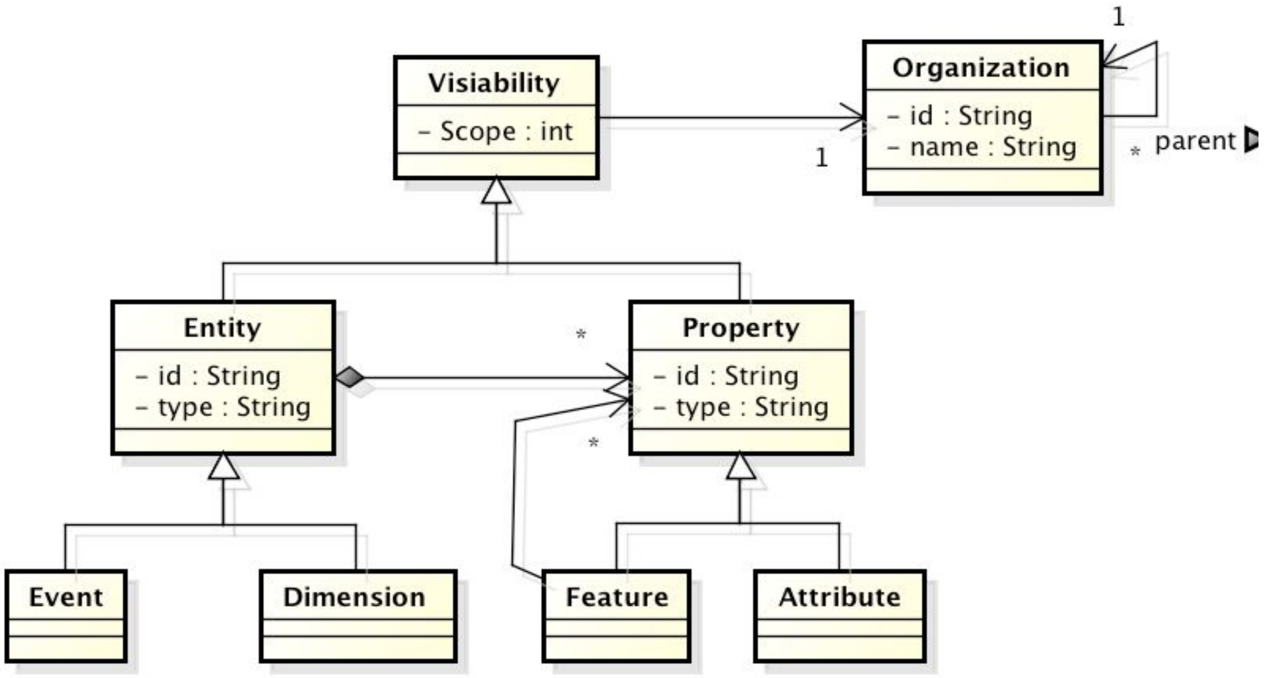
Features

- Derived values based on Transaction and Dimension Attributes and other Features
- Dimensions include:
  - Customers, Accounts, Merchants, Locations, Time of Day, Product, Price, etc.
- Features are computed for individual transactions
  - Distance from home
  - Difference from average, max, min, mean
  - Count of transactions today
- Features are also computed for Dimensions
  - Average, max, min, mean transaction amount for merchant and consumer
  - Frequency of transactions for consumer

Meta Data Model

- Metadata defines transactions, dimensions, features and attributes.
- Attributes are static values, Features are computed
- Visibility controls access for a multi-tenant system

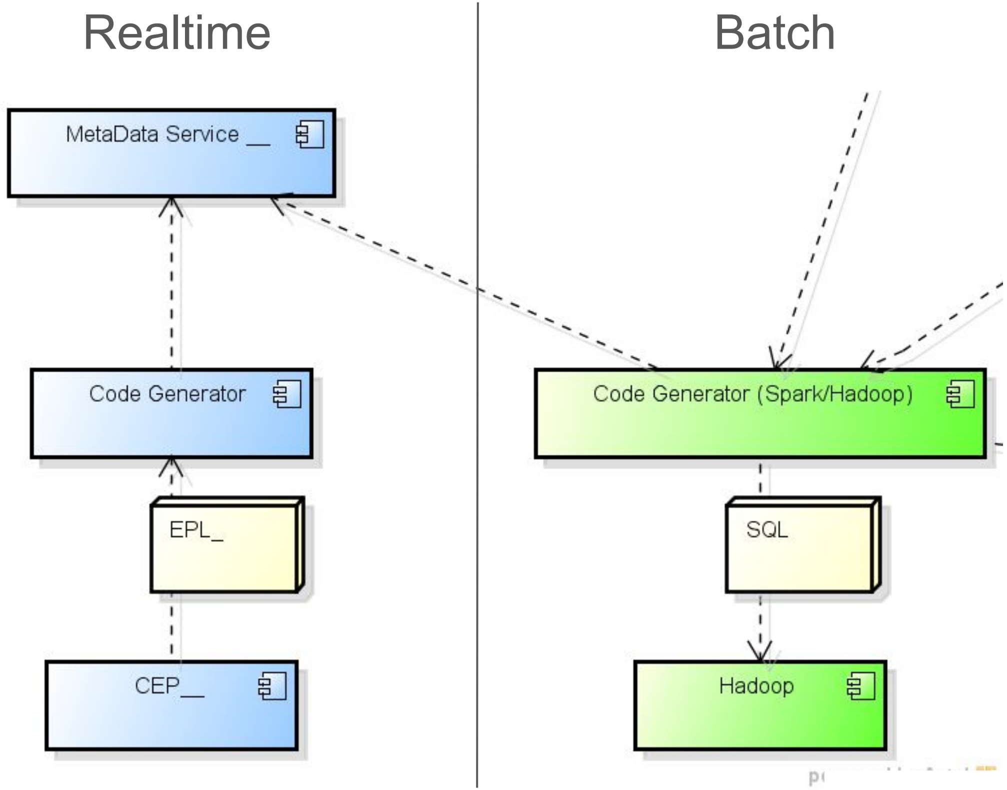
.



Code Generation for Feature Computation

Code to compute features are automatically generated based on the feature metadata

Supports realtime and batch (lambda architecture)



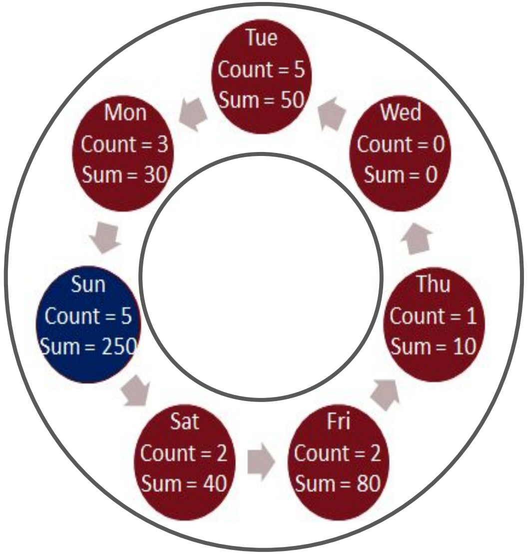
Circular buffer for fast time based feature computation

Compact structure for storing a moving window of transaction data

Each cell represents a unit of time (e.g., min, hour, day, week, month)

Supports aggregate functions: sum, avg, mean, max, min, etc

Apply to dimensions: merchants, consumers, regions, etc



Example: Jim's average transaction amount in last 7 days

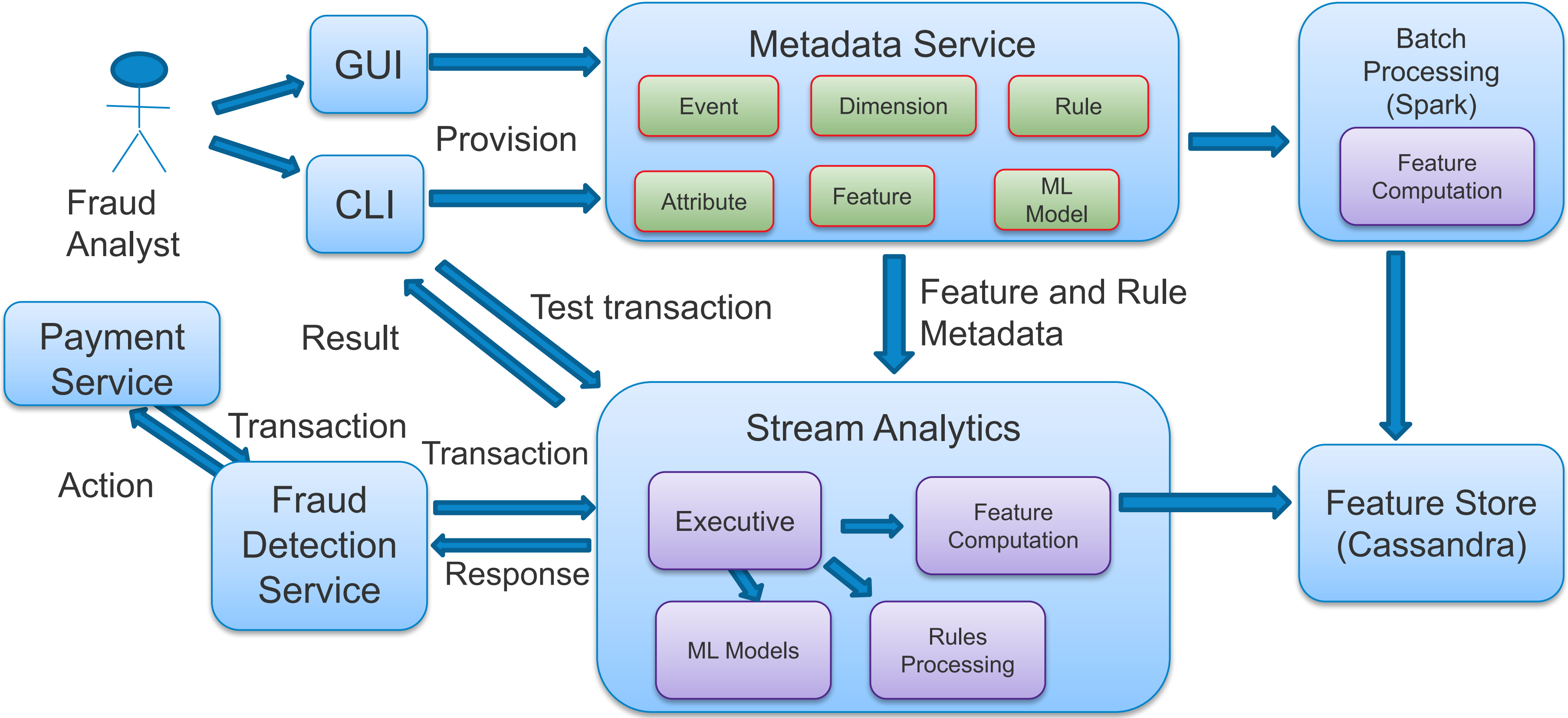
Old average:

$$\frac{20+30+5+0+10+80+40}{4+3+5+0+1+2+2} = 10.9$$

New average:

$$\frac{250+30+5+0+10+80+40}{5+3+5+0+1+2+2} = 23.1$$

Streaming Fraud Detection



Event and Dimension Tables

Event Table:

Org Id	**Event Type**	**Event Id**	**Event Version**	e1 Attr1 (amt)	e1 Attr2 (acct)	e2 Attr1 (amt)	e2 Attr2 (acct)	...	Dim1 (amt)	Dim2 (acct)	...	Feature1	Feature2 (fraud)	...
100	8583	5009	1	200	203				200	203			NO	
100	8583	5010	1	600	204				600	204			NO	
102	ACH	2301	1			2000	203		2000	203			NO	
102	ACH	2302	1			600	345		600	345			YES	
	...													

Dimension Table:

Org Id	**Dimension**	**Element Id**	**d1_f1**	**d1_f2**	**d2_f1**	**d2_f2**	**...**	**c_f1**	**c_f2**	**c_f3**	**...**
			Avg Trx Amt (1 day)	*Trx per hour*	*Avg Trx Amt 90 day*	*Max Trx Amt (1 year)*	...	*Fraud count*	*Non fraud count*	*Fraud Probability (fraud/nonfraud)*	**...**
100	Terminal (d1)	ATM102	100	23				1	800	0.00125	
100	Terminal (d1)	POS12	200	54				5	5000	0.001	
102	Account (d2)	200023			100	2000		0	80	0.0	
102	Account (d2)	192321			500	10000		5	20	0.25	
	...										

Cassandra DB

- Fast read/write (2.3 ms)
- Horizontally scalable to petabytes
- Multi tenant
- Flexible schema and columns

Architecture

- Disaster Recovery with 2 data centers in US and Europe
- Complex Event processing Framework
  - Compute features real time
  - Apply models and rules
  - Return result
- Cassandra database for fast retrieval and storage
  - Proprietary data structure for storing/computing features
- Hadoop cluster for batch computation of features
  - Add new features
  - Add or update data
  - Correct for outages
  - Build and test models

.