



# Amazon SageMaker Introduction and Feature Engineering

Michael Lin

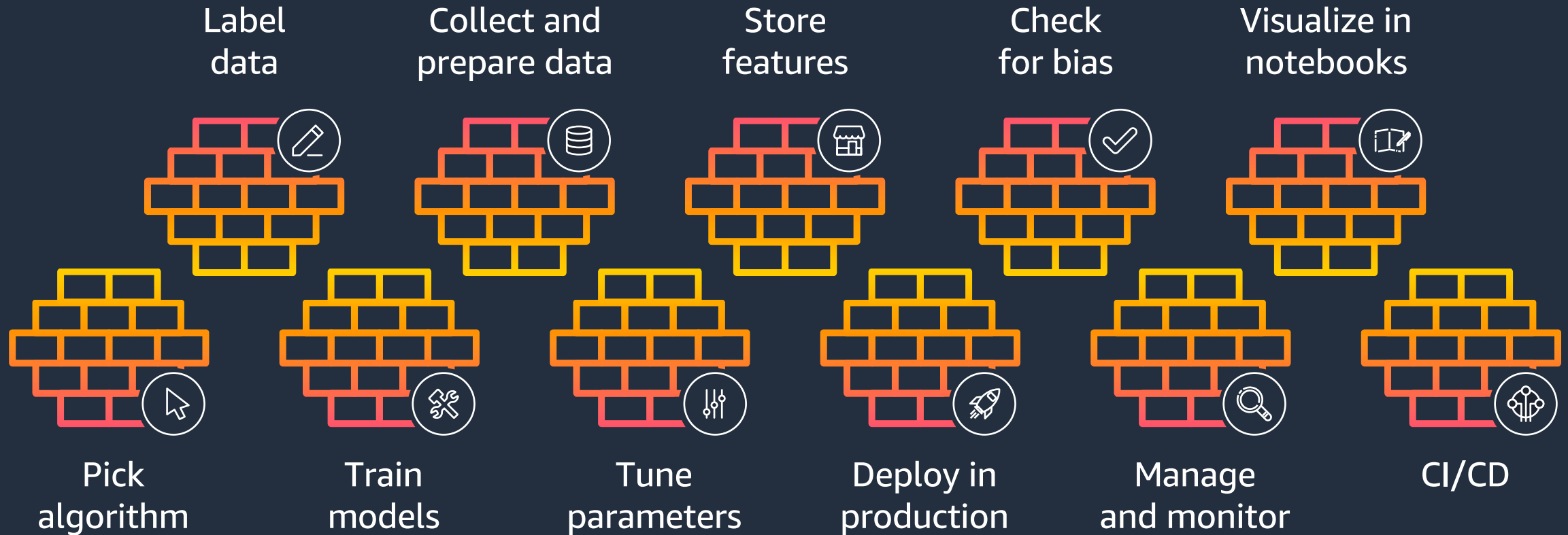
Sr. Solutions Architect  
Amazon Web Services



## Immersion Day

---

# Machine learning development is complex and costly





# What is SageMaker?

# Amazon SageMaker feature tour

PREPARE DATA AND BUILD, TRAIN, AND DEPLOY ML MODEL FOR ANY USE CASE

## PREPARE

- Geospatial**  
Visualize geospatial data
- Ground Truth**  
Create high quality datasets for ML
- Data Wrangler**  
Aggregate and prepare data for ML
- Processing**  
Built-in Python, BYO R/Spark
- Feature Store**  
Store, catalog, search, and reuse features
- Clarify**  
Detect bias and understand model predictions

## BUILD

- Studio Notebooks & Notebook Instances**  
Fully managed Jupyter notebooks with elastic compute
- Studio Lab**  
Free ML development environment
- Built-in Algorithms**  
Integrated tabular, NLP, and vision algorithms
- JumpStart**  
UI based discovery, training, and deployment of models, solutions, and examples
- Autopilot**  
Automatically create ML models with full visibility
- Bring Your Own**  
Bring your own container and algorithms
- Local Mode**  
Test and prototype on your local machine

## TRAIN & TUNE

- Fully Managed Training**  
Broad hardware options, easy to setup and scale
- Distributed Training Libraries**  
High performance training for large datasets and models
- Training Compiler**  
Faster deep learning model training
- Automatic Model Tuning**  
Hyperparameter optimization
- Managed Spot Training**  
Reduce training cost by up to 90%
- Debugger and Profiler**  
Debug and profile training runs
- Experiments**  
Track, visualize, and share model artifacts across teams
- Customization Support**  
Integrate with popular open source frameworks and libraries

## DEPLOY & MANAGE

- Fully Managed Deployment**  
Ultra low latency, high throughput inference
- Real-Time Inference**  
For steady traffic patterns
- Serverless Inference**  
For intermittent traffic patterns
- Asynchronous Inference**  
For large payloads or long processing times
- Batch Transform**  
For offline inference on batches of large datasets
- Multi-Model Endpoints**  
Reduce cost by hosting multiple models per instance
- Multi-Container Endpoints**  
Reduce cost by hosting multiple containers per instance
- Shadow Testing**  
Validate model performance in production
- Inference Recommender**  
Automatically select compute instance and configuration
- Model Monitor**  
Maintain accuracy of deployed models
- Kubernetes Operators & Components**  
Manage and monitor models on edge devices
- Edge Manager**  
Manage and monitor models on edge devices

**MLOps: Pipelines | Projects | Model Registry**  
Workflow automation, CI/CD for ML, central model catalog

**Canvas**  
Generate accurate machine learning predictions—no code required

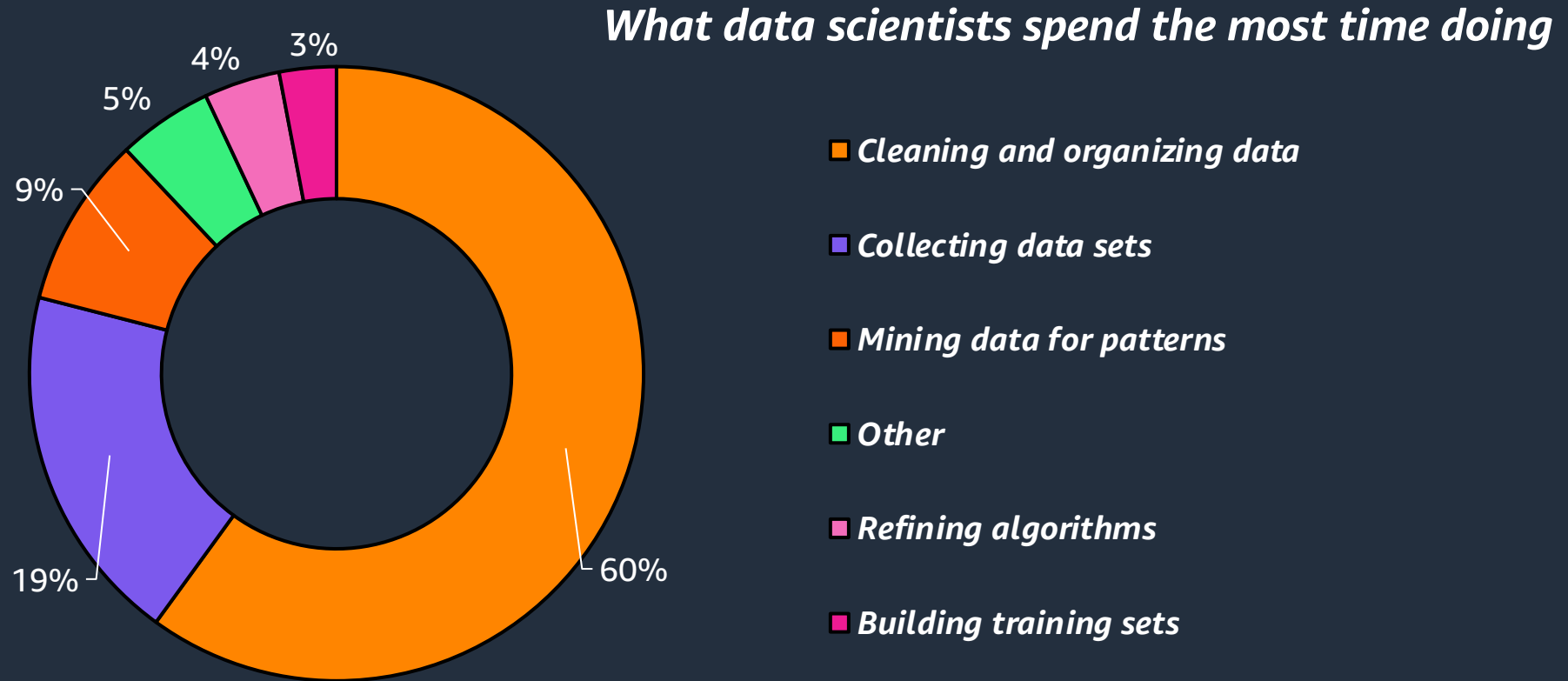
**Studio | RStudio**  
Integrated development environment (IDE) for ML

**Governance**  
Model Cards | Dashboard | Permissions



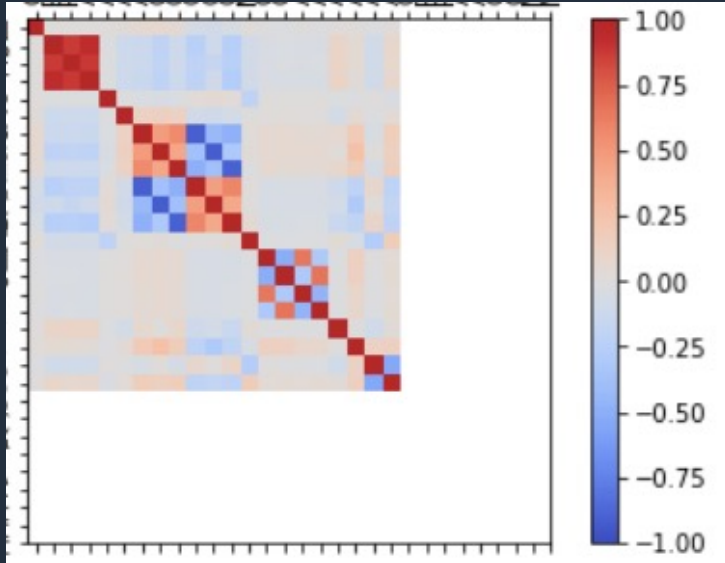
# Data Exploration & Feature Engineering

# 80% of time spent on data prep

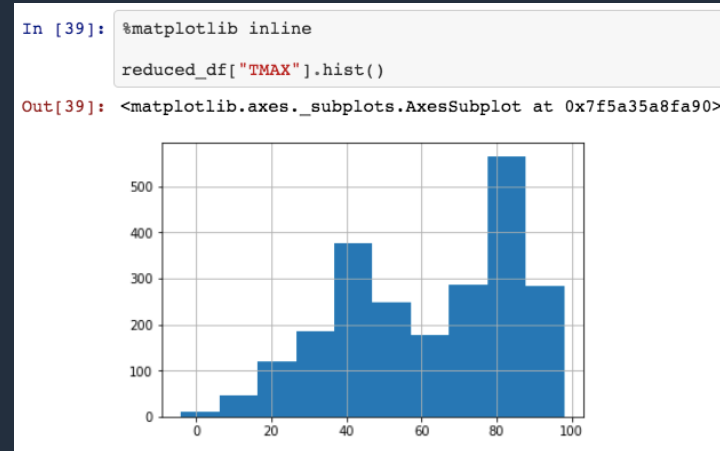


Source: [Forbes survey of 80 data scientists, March 2016](#)

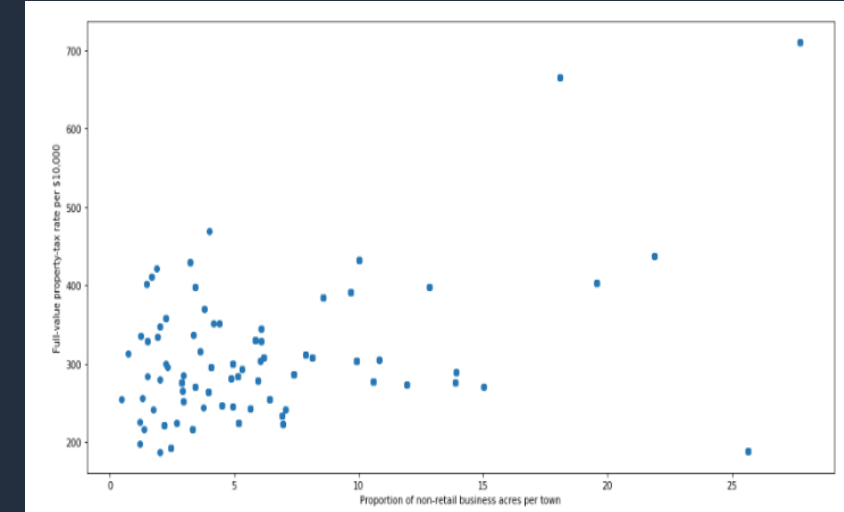
Are My X's correlated with my Y's? With other X's?



Do they represent the real world?



Do I need to remove outliers?



What rows and columns are in my data set?

```
In [3]: df.head()
```

Out[3]:

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...
0	24170	JB429040	09/09/2018 10:30:00 PM	019XX E 74TH ST	0110	HOMICIDE	FIRST DEGREE MURDER	STREET	False	False	...
1	11447764	JB437679	09/09/2018 12:00:00 PM	085XX W HIGGINS RD	1210	DECEPTIVE PRACTICE	THEFT OF LABOR/SERVICES	HOTEL/MOTEL	False	False	...

Do I need to combine columns?

```
] data['no_previous_contact'] = np.where(data['pdays'] == 999, 1, 0)
# Indicator variable to capture when pdays takes a value of 999
data['not_working'] = np.where(np.in1d(data['job'], ['student', 'retired', 'unemployed']), 1, 0)
```



# How do I handle strings? One Hot Encoding

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

CompanyName	Categoricalvalue	Price
VW	1	20000
Acura	2	10011
Honda	3	50000
Honda	3	10000



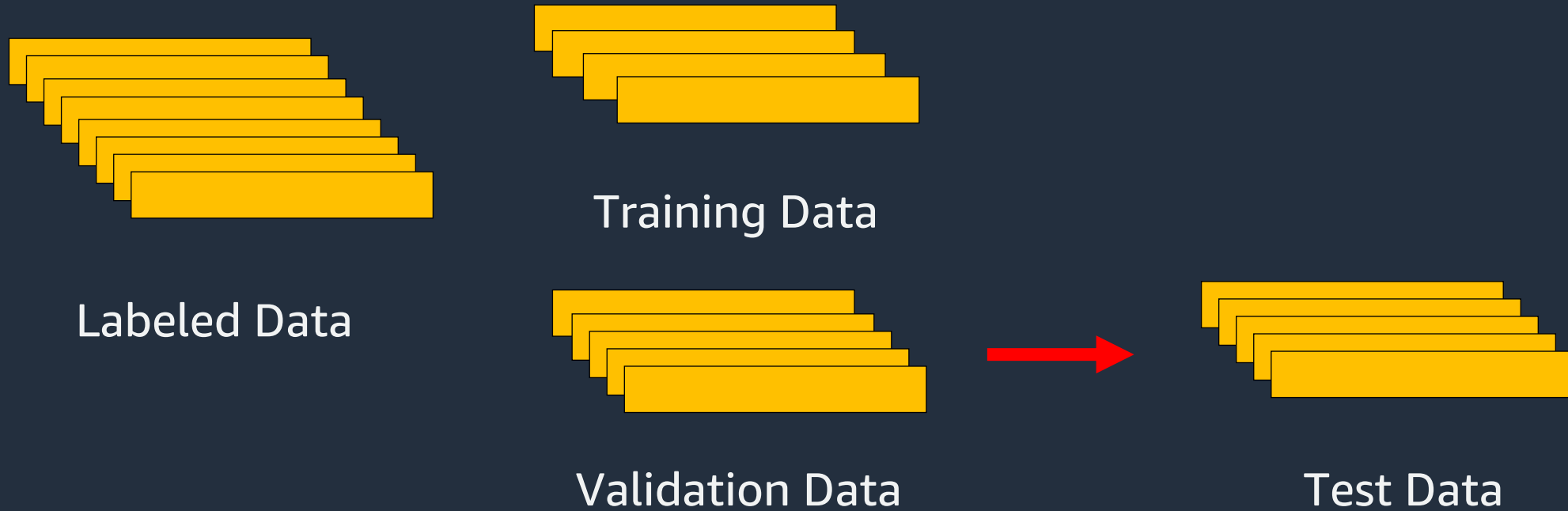
VW	Acura	Honda	Price
1	0	0	20000
0	1	0	10011
0	0	1	50000
0	0	1	10000

```
In [64]: model_df = pd.get_dummies(model_df, columns = ["Block"])
```

Source <https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f>



# Splitting Data for Machine Learning





# **SageMaker-Purpose- built data preparation tools**

# Amazon SageMaker Data Wrangler

*EXPLORE, PREPARE, AND PROCESS  
DATA WITH LITTLE TO NO CODE*



**Import data from multiple sources**



**Get insights on data and data quality**



**Visually explore, analyze, and prepare data**



**Quickly perform feature engineering**



**Automate ML data preparation workflows**

# Easily transform data for ML with 300+ built-in transforms

300+ built-in data transformations (no code) for common data prep needs and ML specific needs

Built by data scientists for data scientists

*ML specific transforms such as:*

One hot encoding

Balance data

Time series transforms

<

ADD TRANSFORM

×

**Custom transform**  
Use Pyspark, Pandas, or Pyspark (SQL) to define custom transformations...

**Balance data**  
Balance the data for binary classification problems using random oversampling...

**Custom formula**  
Define a new column using a Spark SQL expression to query data in the dataset...

**Encode categorical**  
Convert categorical variables to numeric or vector representations. [Learn more...](#)

**Featurize date/time**  
Encode date/time values to numeric and vector representations. [Learn more...](#)

**Featurize text**  
Generate vector representations from natural language text. [Learn more...](#)

**Format string**  
Clean and prepare strings using standard string formatting operations. [Learn more...](#)

**Group by**  
Add an aggregated column after group by as a new column.

**Handle missing**  
Replace, drop, or add indicators for missing values. [Learn more.](#) [↗](#)

**Handle outliers**  
Remove or replace outlier numeric and categorical values. [Learn more.](#) [↗](#)

<

ADD TRANSFORM

×

[Replace, drop, or add indicators for missing values. \[Learn more.\]\(#\) \[↗\]\(#\)](#)

**Handle outliers**  
Remove or replace outlier numeric and categorical values. [Learn more.](#) [↗](#)

**Handle structured column**  
Flatten JSON and perform other operations on structured data

**Manage columns**  
Move, drop, duplicate or rename columns in the dataset. [Learn more.](#) [↗](#)

**Manage rows**  
Sort, shuffle or drop duplicate rows.

**Manage vectors**  
Expand or create vector columns. [Learn more.](#) [↗](#)


**Parse column as type**  
Cast a column to a new data type. [Learn more.](#) [↗](#)

**Process numeric**  
Transform numeric values to improve machine learning model performance...

**Search and edit**  
Find, replace, split, and otherwise transform input string values using search and replace...

**Time Series**  
Transformers to preprocess and manipulate time series. [Learn more.](#) [↗](#)

**Validate string**  
Validate the format of string values using standard string functions. [Learn more.](#) [↗](#)

 machine learning

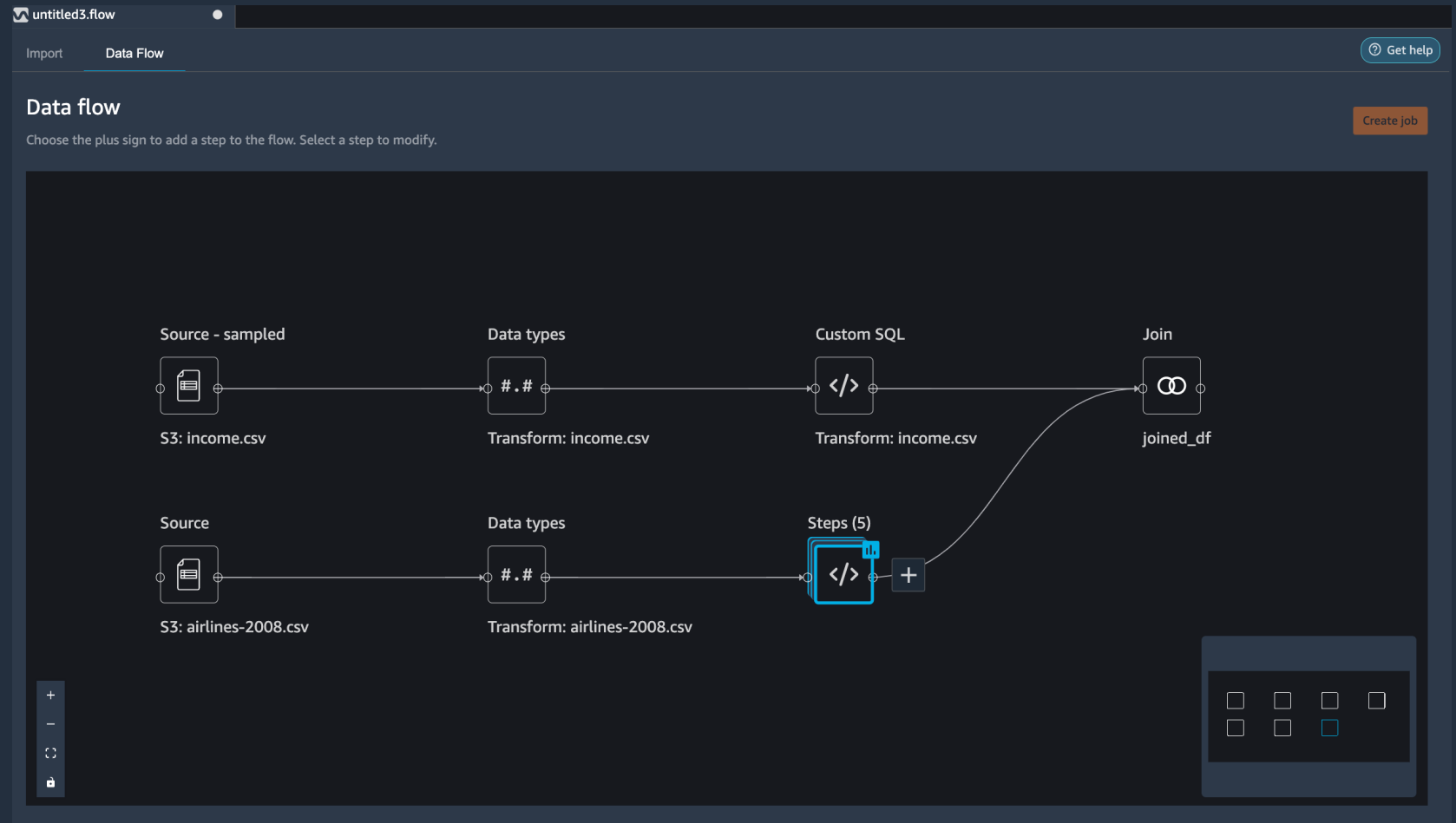
© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Easily visualize the steps of your data processing pipeline

Data Wrangler records all the steps of data prep workflow in a data flow graph

Visualize the order of transformations, join and concatenate operators

Easily navigate data transformation flow, and modify and delete steps iteratively



# What is a feature and why is it important?

Raw data

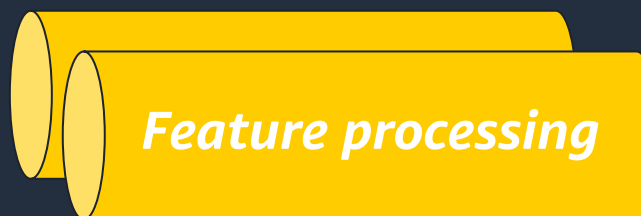
Gender	Male, female
Driver rating	Poor, Fair, Good, Excellent
Vehicle color	Red, blue, black, gold, silver, white

Feature vector

Gender	[0,1]
Driver rating	[0,1,2,3]
Vehicle color	[1,0,0,0,0,0]



*Feature engineering*



# Challenges of separate feature stores



Feature drift



Feature duplication



Slow model  
development/deployment



# Amazon SageMaker Feature Store

*SECURELY STORE, DISCOVER,  
AND SHARE FEATURES FOR ML*



**Online and off-line**



**Millisecond latency**



**Consistent features**



**Visual search**



**Sharing and collaboration**

# Support for separate feature stores



## Online feature store

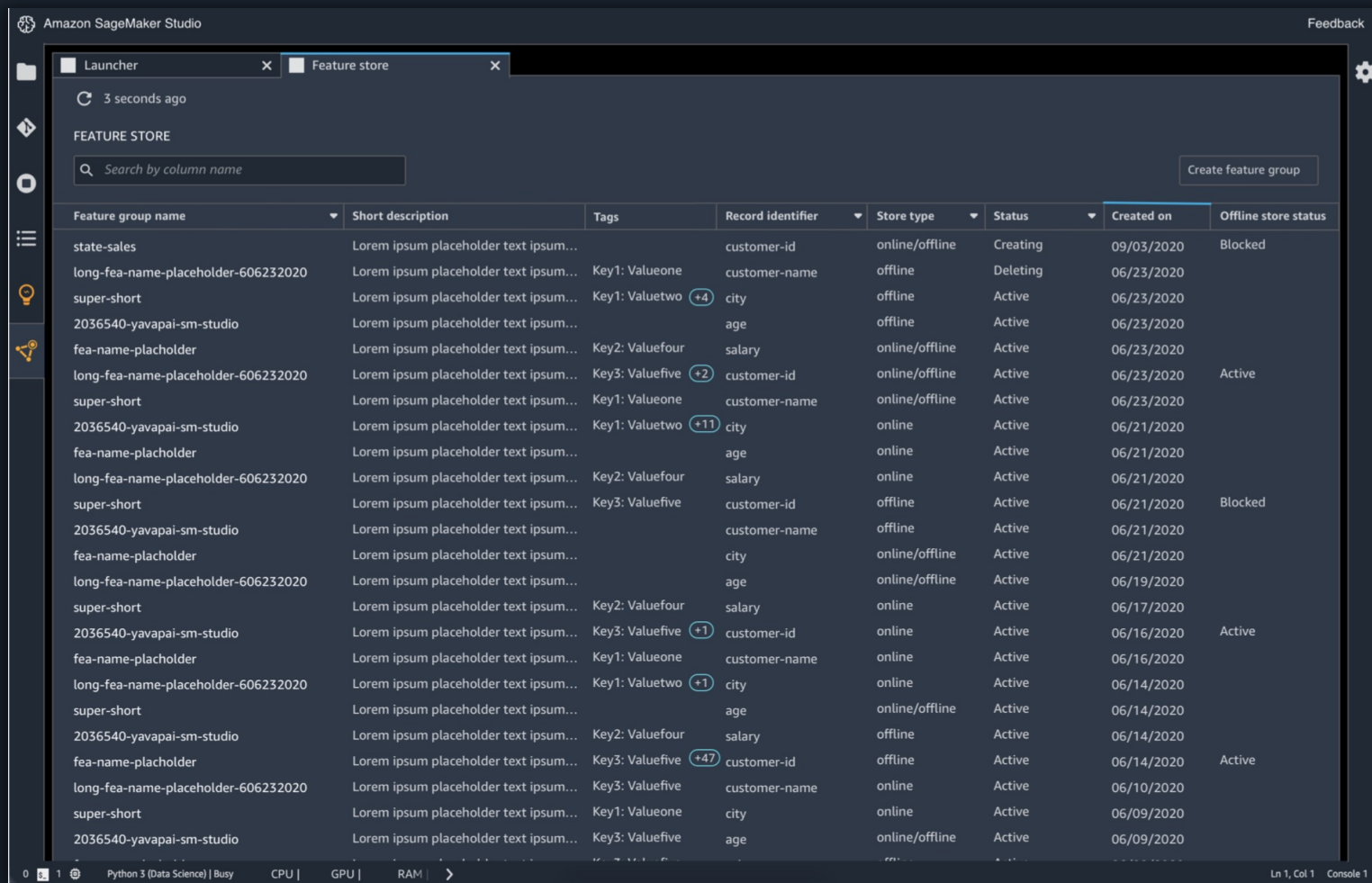
- Primarily used for real time predictions
- Use cases such as real-time fraud detection
- Latest copy of feature data
- High throughput writes
- Low millisecond latency reads



## Offline feature store

- Primarily used for batch predictions and model training
- Historical record of feature data
- High throughput writes
- <15 minutes read after write consistency

# Search and discover features using Feature Store



The screenshot shows the Amazon SageMaker Studio interface with the Feature Store tab active. A search bar at the top allows searching by column name. Below the search bar is a table listing feature groups. The table has columns for Feature group name, Short description, Tags, Record identifier, Store type, Status, Created on, and Offline store status. The table contains 20 rows of data, including feature groups like 'state-sales', 'long-fea-name-placeholder-606232020', 'super-short', '2036540-yavapai-sm-studio', and 'fea-name-placholder'. Some rows have a small blue circle with a number next to the Record identifier column, indicating a count of features.

Feature group name	Short description	Tags	Record identifier	Store type	Status	Created on	Offline store status
state-sales	Lorem ipsum placeholder text ipsum...		customer-id	online/offline	Creating	09/03/2020	Blocked
long-fea-name-placeholder-606232020	Lorem ipsum placeholder text ipsum...	Key1: Valueone	customer-name	offline	Deleting	06/23/2020	
super-short	Lorem ipsum placeholder text ipsum...	Key1: Valuetwo +4	city	offline	Active	06/23/2020	
2036540-yavapai-sm-studio	Lorem ipsum placeholder text ipsum...		age	offline	Active	06/23/2020	
fea-name-placholder	Lorem ipsum placeholder text ipsum...	Key2: Valuefour	salary	online/offline	Active	06/23/2020	
long-fea-name-placeholder-606232020	Lorem ipsum placeholder text ipsum...	Key3: Valuefive +2	customer-id	online/offline	Active	06/23/2020	Active
super-short	Lorem ipsum placeholder text ipsum...	Key1: Valueone	customer-name	online/offline	Active	06/23/2020	
2036540-yavapai-sm-studio	Lorem ipsum placeholder text ipsum...	Key1: Valuetwo +11	city	online	Active	06/21/2020	
fea-name-placholder	Lorem ipsum placeholder text ipsum...		age	online	Active	06/21/2020	
long-fea-name-placeholder-606232020	Lorem ipsum placeholder text ipsum...	Key2: Valuefour	salary	online	Active	06/21/2020	
super-short	Lorem ipsum placeholder text ipsum...	Key3: Valuefive	customer-id	offline	Active	06/21/2020	Blocked
2036540-yavapai-sm-studio	Lorem ipsum placeholder text ipsum...		customer-name	offline	Active	06/21/2020	
fea-name-placholder	Lorem ipsum placeholder text ipsum...		city	online/offline	Active	06/21/2020	
long-fea-name-placeholder-606232020	Lorem ipsum placeholder text ipsum...		age	online/offline	Active	06/19/2020	
super-short	Lorem ipsum placeholder text ipsum...	Key2: Valuefour	salary	online	Active	06/17/2020	
2036540-yavapai-sm-studio	Lorem ipsum placeholder text ipsum...	Key3: Valuefive +1	customer-id	online	Active	06/16/2020	Active
fea-name-placholder	Lorem ipsum placeholder text ipsum...	Key1: Valueone	customer-name	online	Active	06/16/2020	
long-fea-name-placeholder-606232020	Lorem ipsum placeholder text ipsum...	Key1: Valuetwo +1	city	online	Active	06/14/2020	
super-short	Lorem ipsum placeholder text ipsum...		age	online/offline	Active	06/14/2020	
2036540-yavapai-sm-studio	Lorem ipsum placeholder text ipsum...	Key2: Valuefour	salary	offline	Active	06/14/2020	
fea-name-placholder	Lorem ipsum placeholder text ipsum...	Key3: Valuefive +47	customer-id	offline	Active	06/14/2020	Active
long-fea-name-placeholder-606232020	Lorem ipsum placeholder text ipsum...	Key3: Valuefive	customer-name	online	Active	06/10/2020	
super-short	Lorem ipsum placeholder text ipsum...	Key1: Valueone	city	online	Active	06/09/2020	
2036540-yavapai-sm-studio	Lorem ipsum placeholder text ipsum...	Key3: Valuefive	age	online/offline	Active	06/09/2020	

- Search features individually or by groups visually with SageMaker Studio
- Discover features by name, description, tags, and other metadata
- Understand how features are grouped relevant to ML applications



# Thank you!

Michael Lin

[linmicht@amazon.com](mailto:linmicht@amazon.com)