Copyright Notice

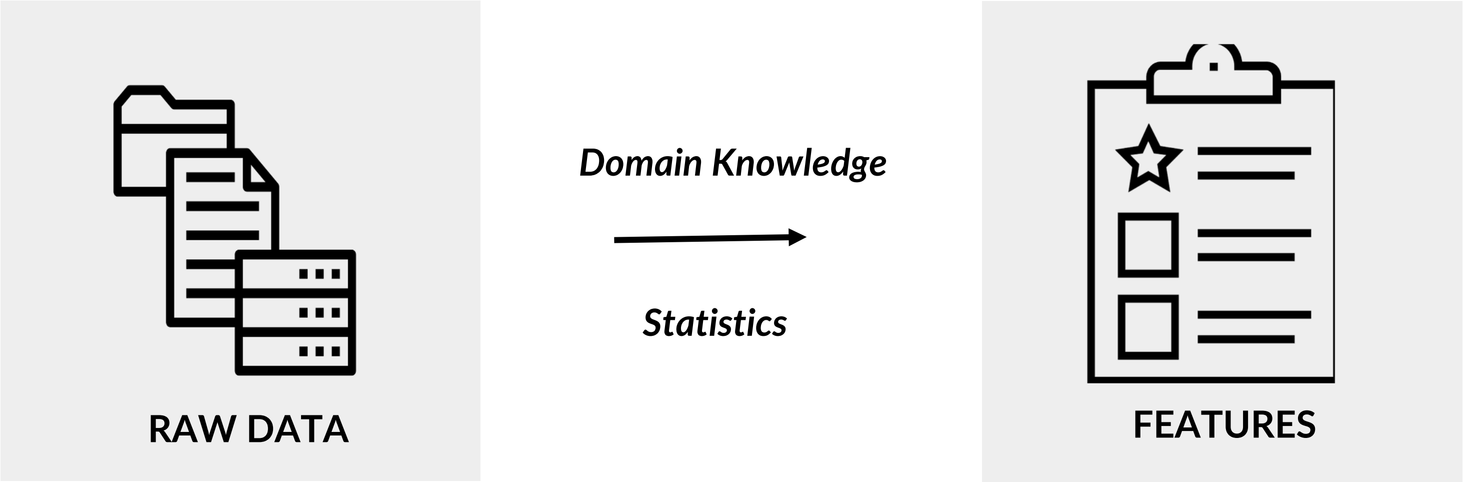
Transform Raw Data into Features for Model Training

Machine Learning Workflow

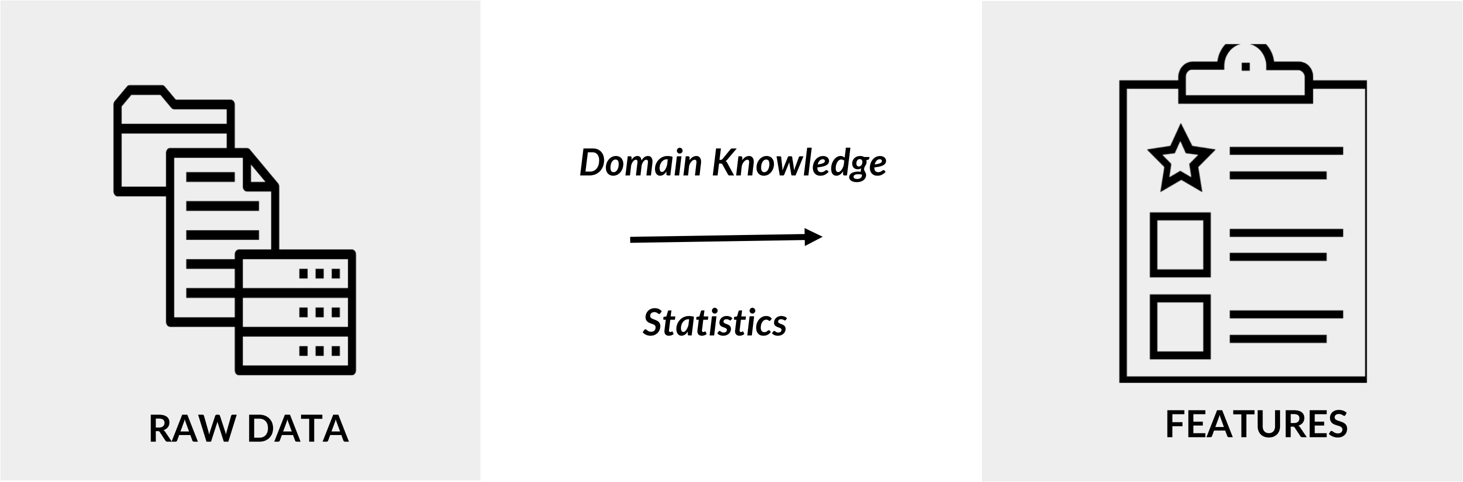
Ingest & Analyze	Prepare & Transform	Train & Tune	Deploy & Manage
Data exploration	Feature engineering	Automated ML	Model deployment
Bias detection	Feature store	Model train and tune	Automated pipelines
Amazon S3 & Amazon Athena	Amazon SageMaker Data Wrangler	Amazon SageMaker Autopilot	Amazon SageMaker Endpoints
AWS Glue	Amazon SageMaker Processing Jobs	Amazon SageMaker Training & Debugger	Amazon SageMaker Batch Transform
Amazon SageMaker Data Wrangler & Clarify	Amazon SageMaker Feature Store	Amazon SageMaker Hyperparameter Tuning	Amazon SageMaker Pipelines

Feature Engineering

Feature Engineering

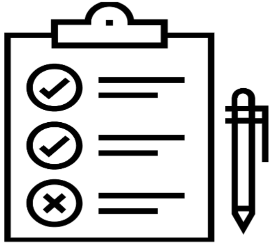


Feature Engineering

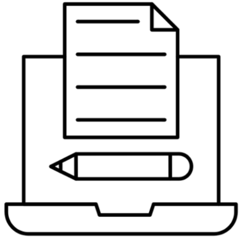


- Dataset best fits the algorithm
- Improve ML model performance

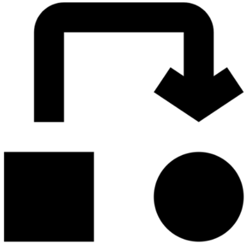
Feature Engineering - Components



Selection

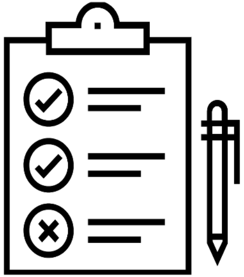


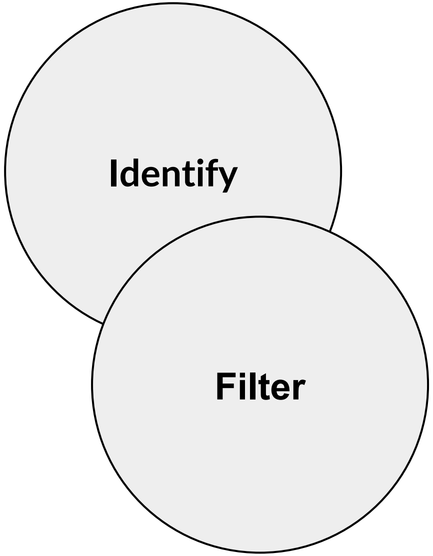
Creation



Transformation

Feature Engineering - Selection

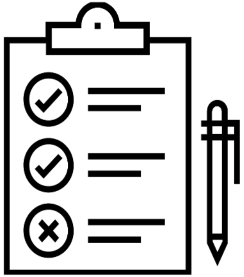


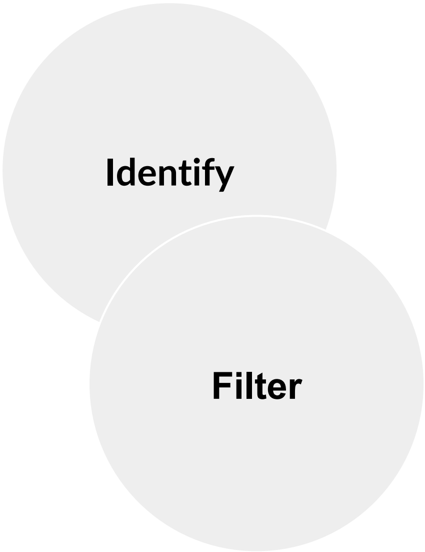


Data attributes

Irrelevant and redundant attributes

Feature Engineering - Selection



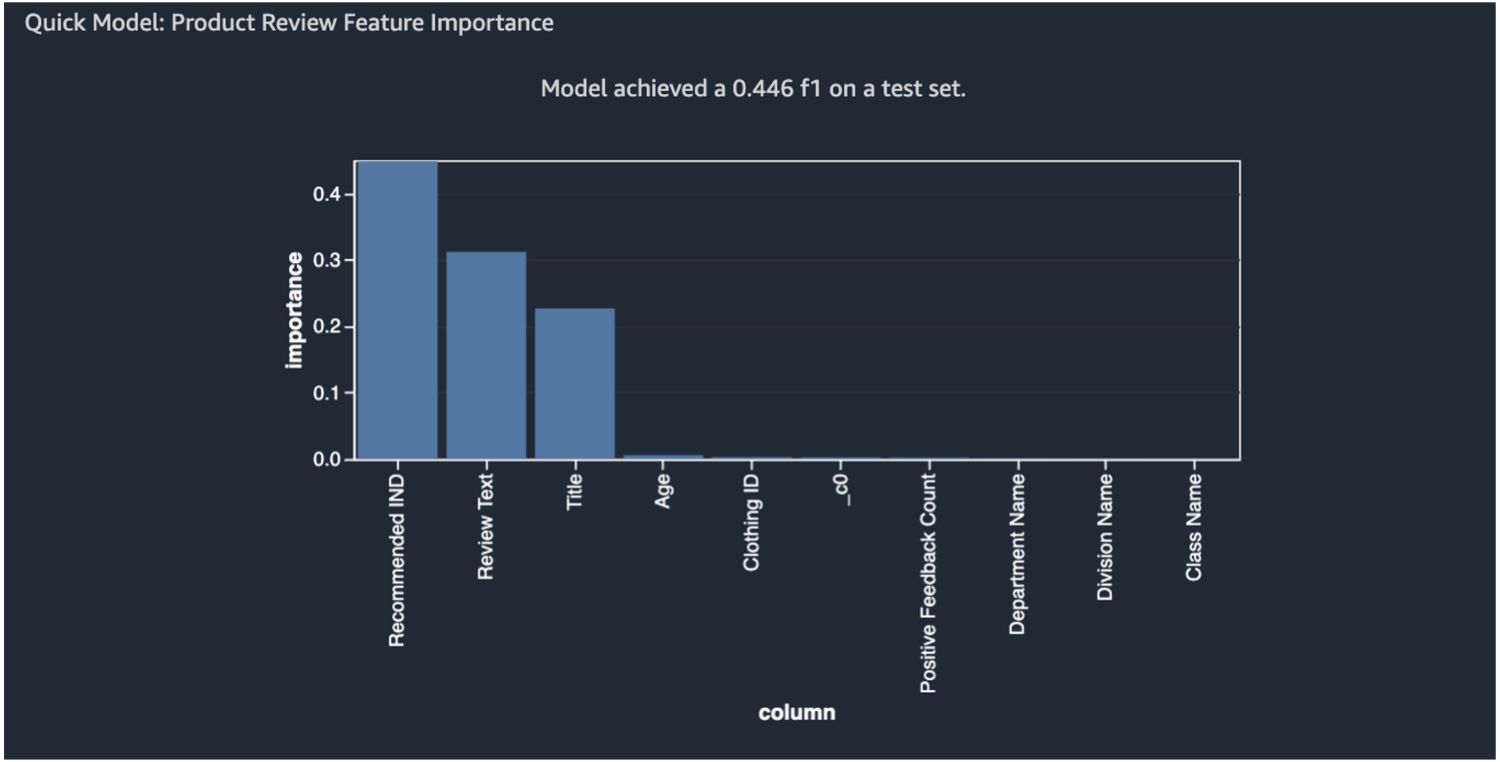


Data attributes

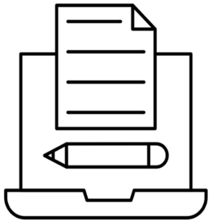
Irrelevant and redundant attributes

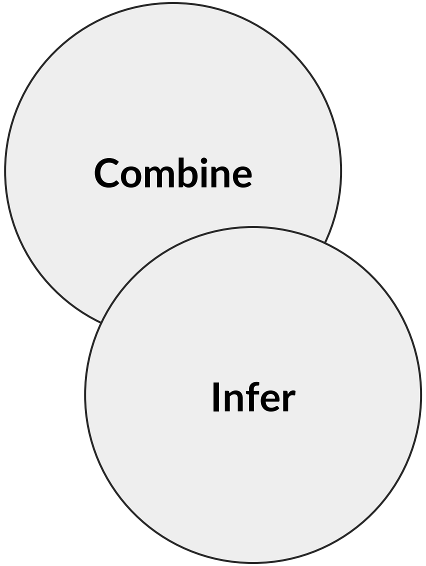
- ✓ Reduce feature dimensionality
- ✓ Train models faster

Feature Importance Report



Feature Engineering - Creation



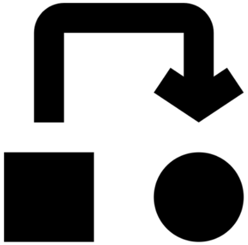


Existing data points into new features

New attributes

✓ Lead to more accurate predictions

Feature Engineering - Transformation

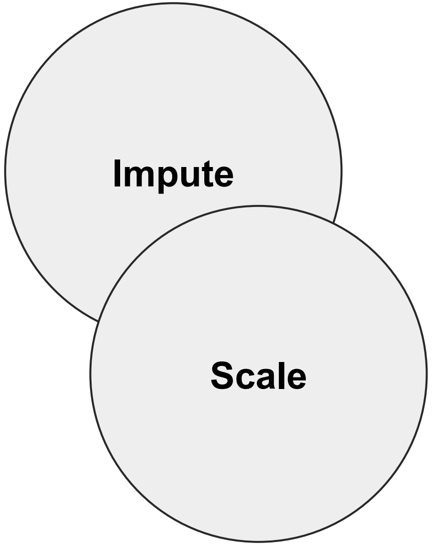


Feature Engineering - Transformation

Impute

Missing feature values

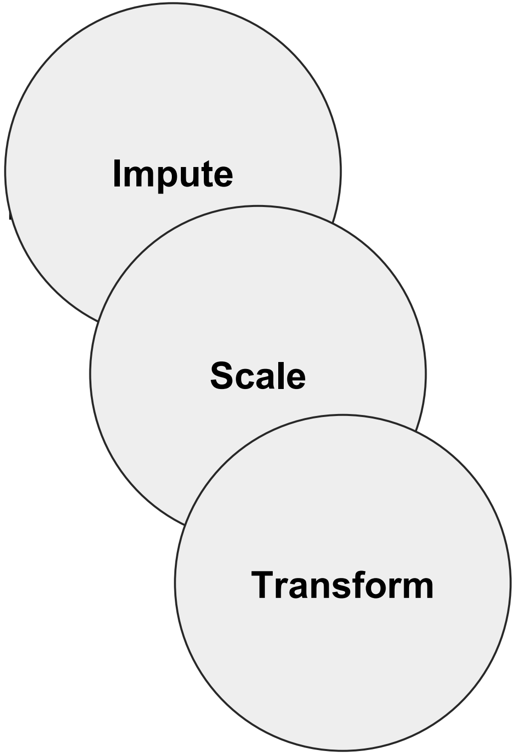
Feature Engineering - Transformation

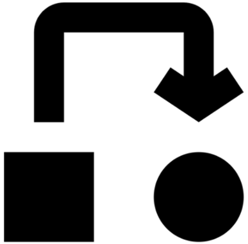


Missing feature values

Numerical features

Feature Engineering - Transformation





Missing feature values

Imputation

Numerical features

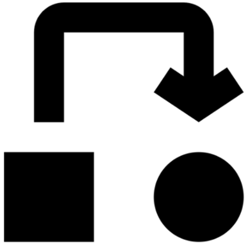
Standardization and Normalization

Non Numerical features

Non Numerical Features are text or category

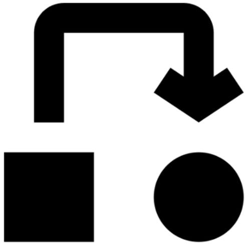
Categorical Feature can be converted into numeric features by using "one-hot encoding".
Text need to converted into vectors. more specifically "BERTvectors" or "BERT Embedding"

Feature Engineering - Transformation



Class Name	Review Text
Blouses	“I simply love it!”
Pants	“It's ok.”
Dresses	“It arrived damaged. Going to return.”

Feature Engineering - Transformation



Class Name	Review Text
Blouses	"I simply love it!"
Pants	"It's ok."
Dresses	"It arrived damaged. Going to return."

Feature Transformation

Review Text

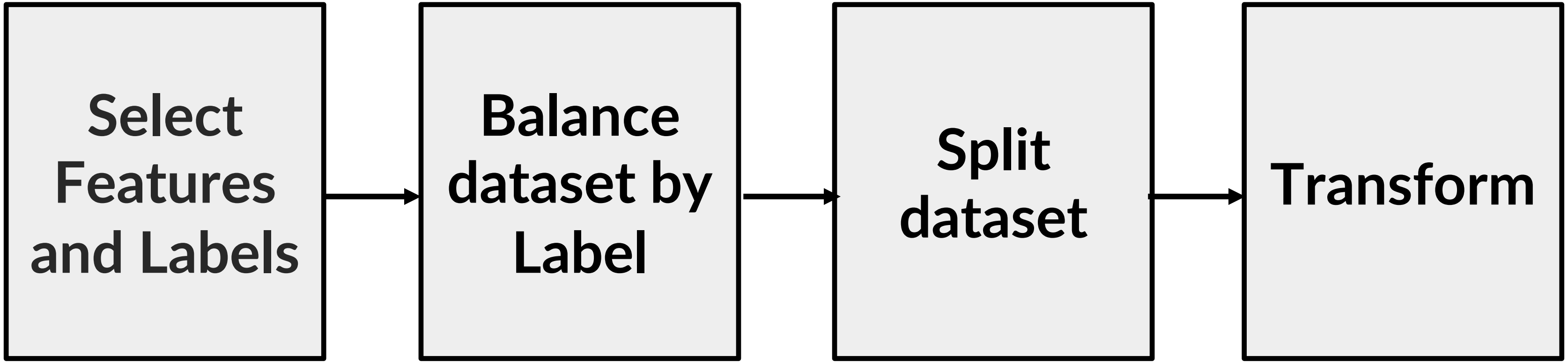
“I simply love it!”
“It's ok.”
“It arrived damaged. Going to return.”

BERT vect

101	2023	...	...
3319	1012	...	...
2003	2307	...	...

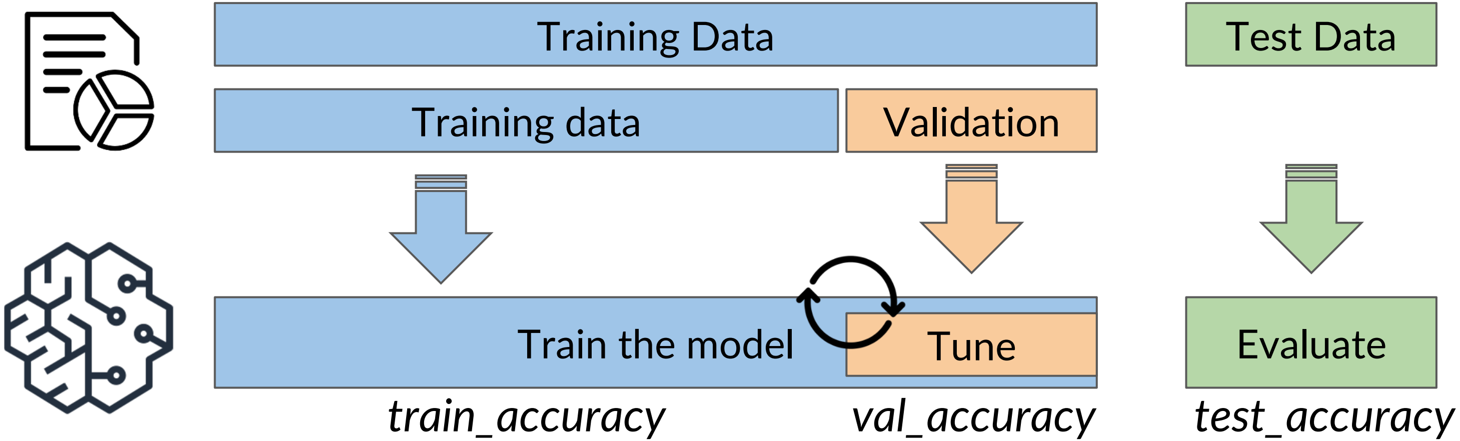
Feature Engineering Pipeline

Feature engineering pipeline

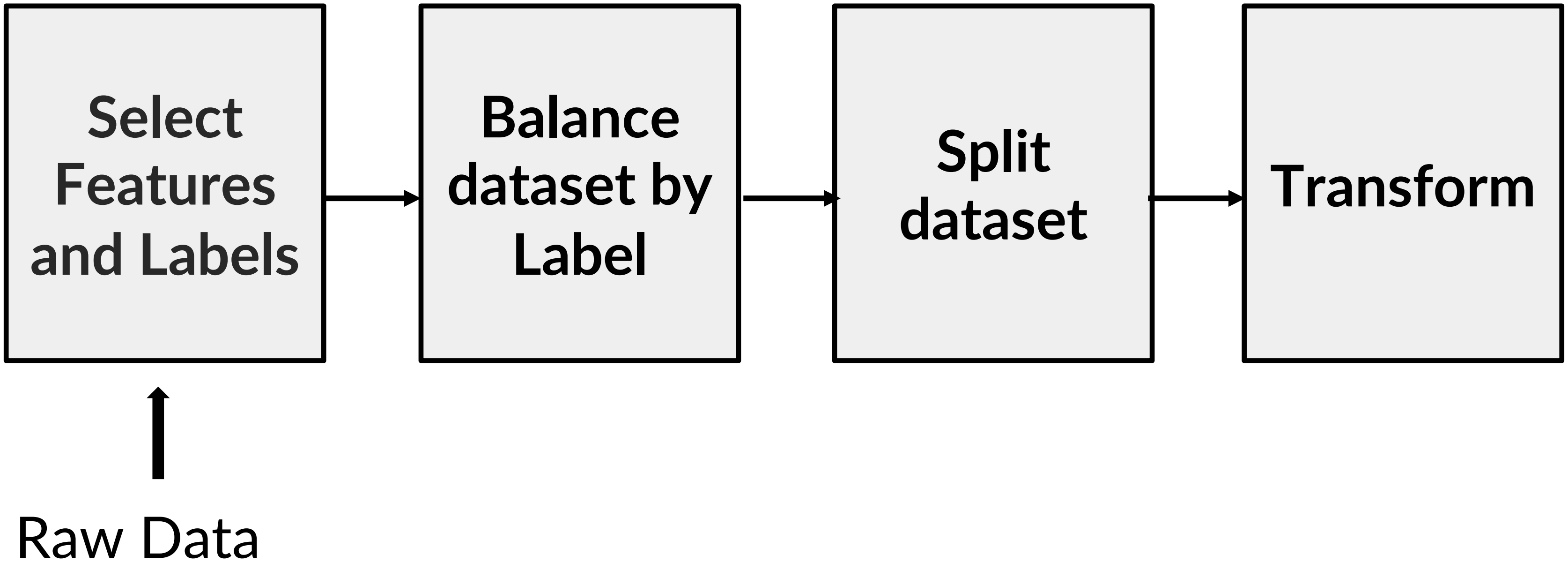


Split Dataset

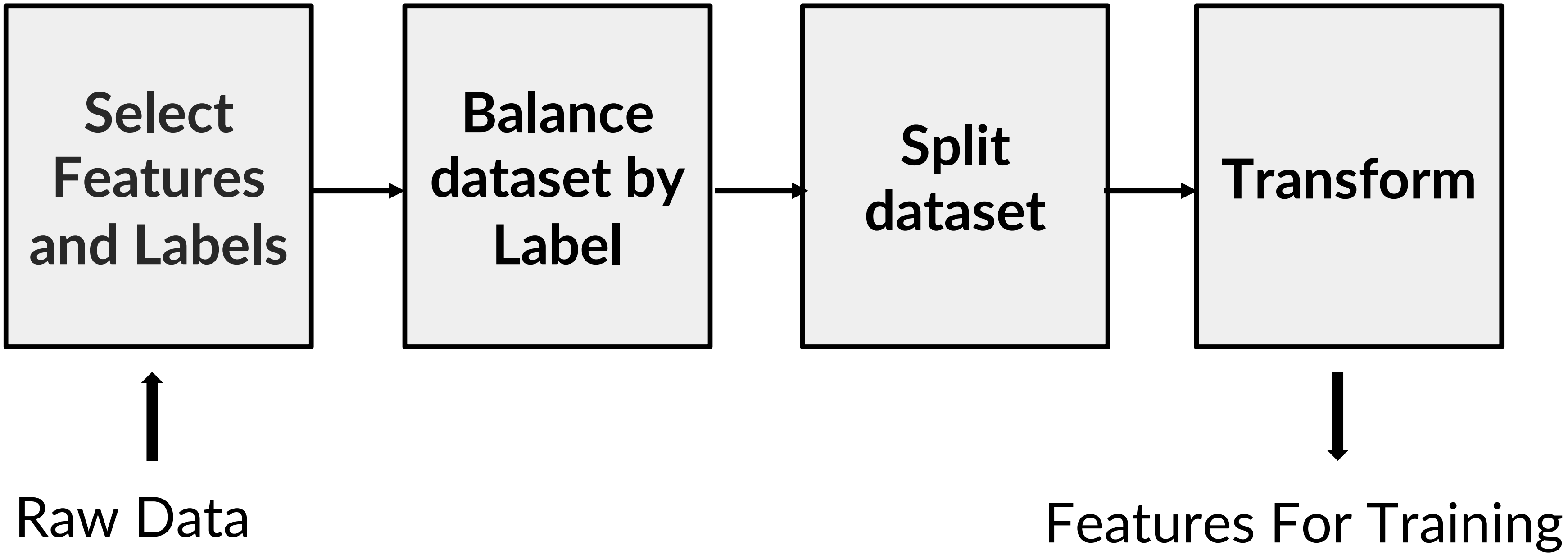
- Training, validation and test data



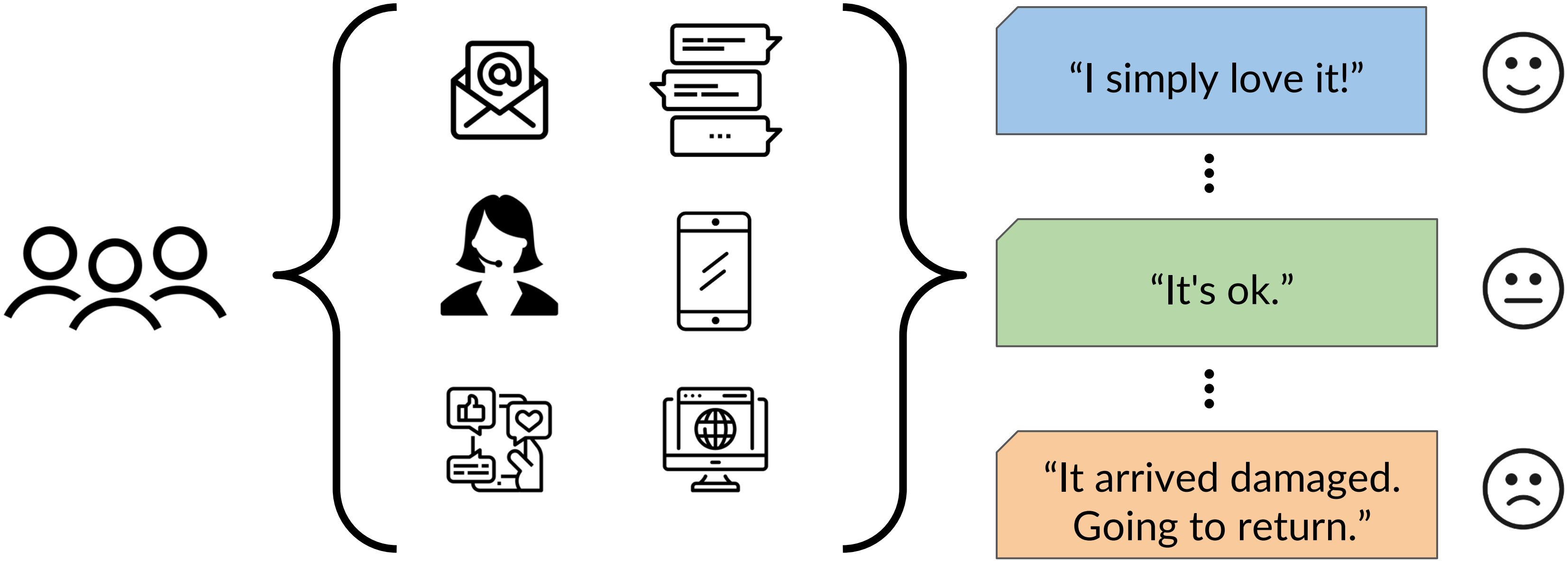
Feature Engineering Pipeline



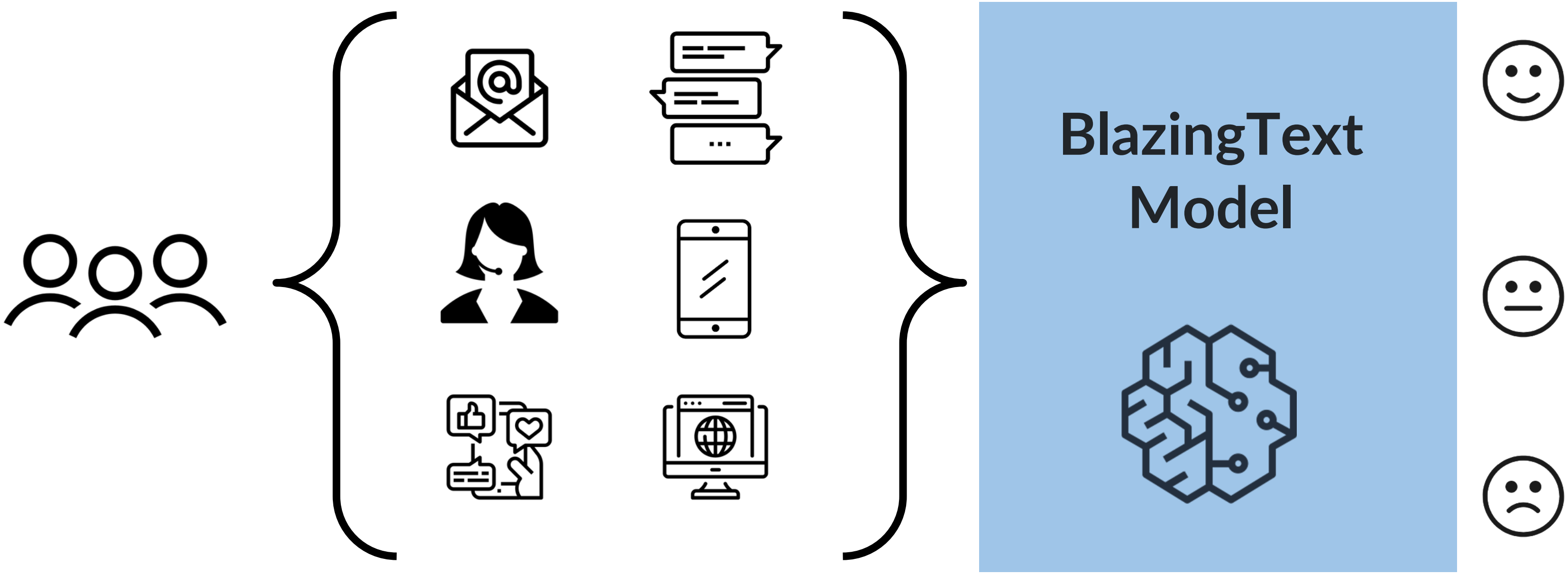
Feature Engineering Pipeline



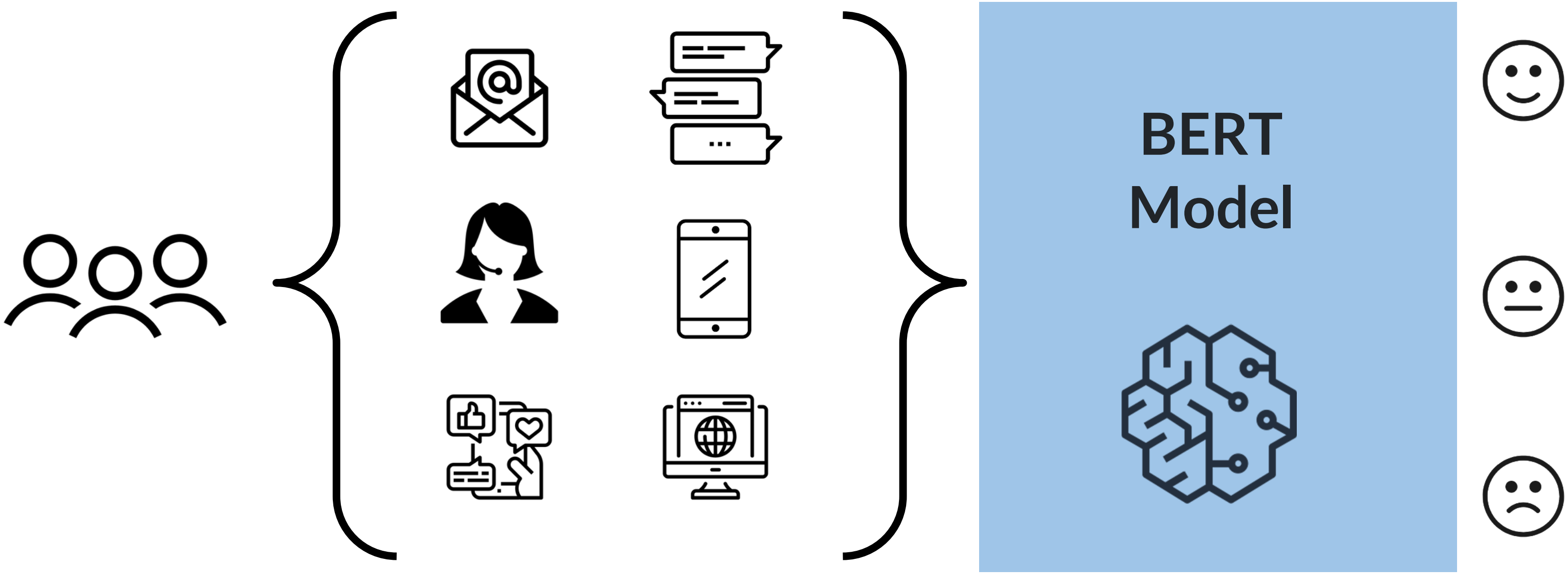
Multi-class Classification for Sentiment Analysis of Product Reviews



Multi-class Classification for Sentiment Analysis of Product Reviews



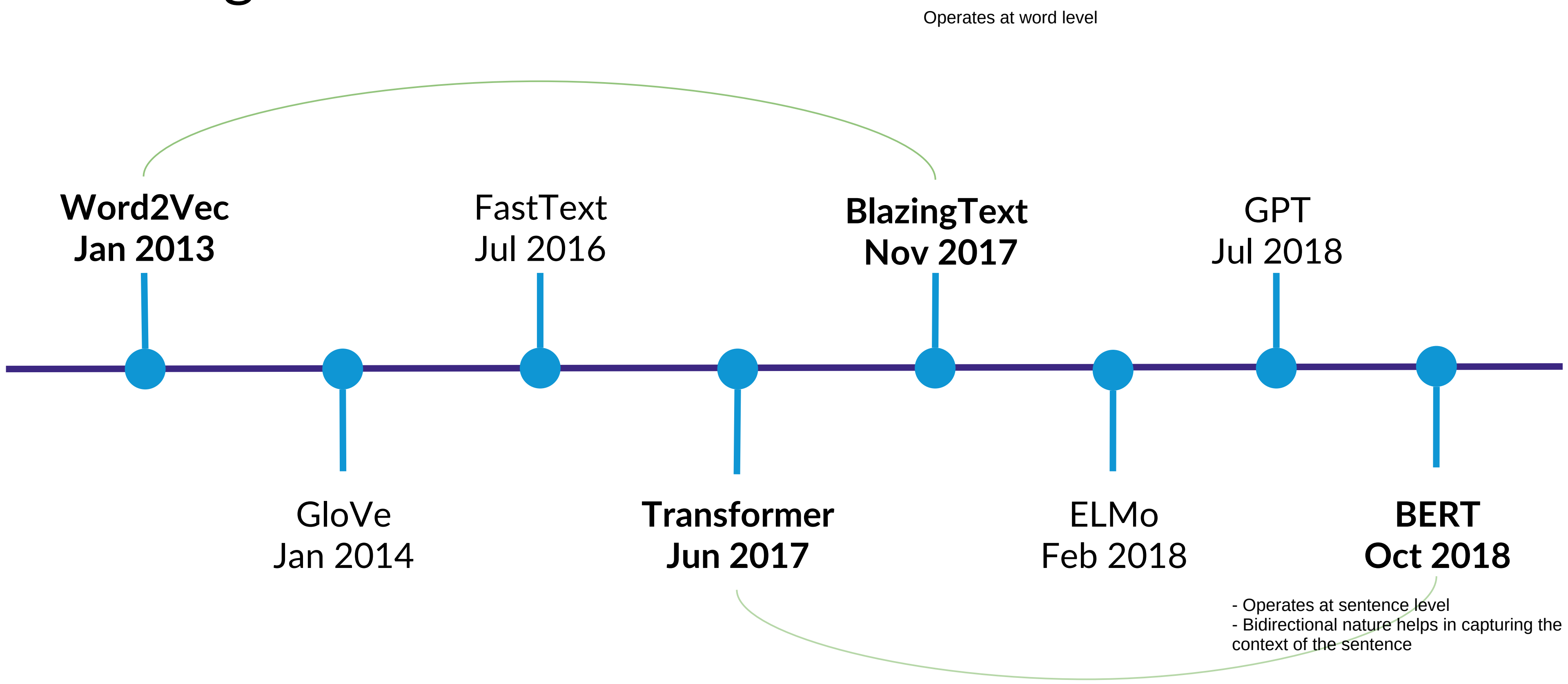
Multi-class Classification for Sentiment Analysis of Product Reviews



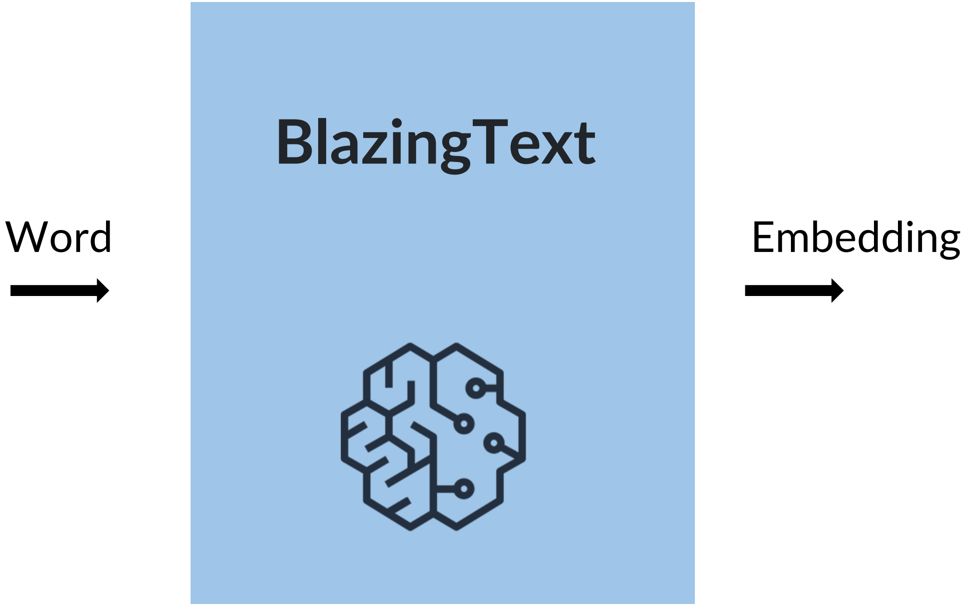
BERT

Bidirectional Encoder
Representations
from Transformers

BlazingText vs BERT



BlazingText - Word Level Embeddings



BlazingText - Word Level Embeddings

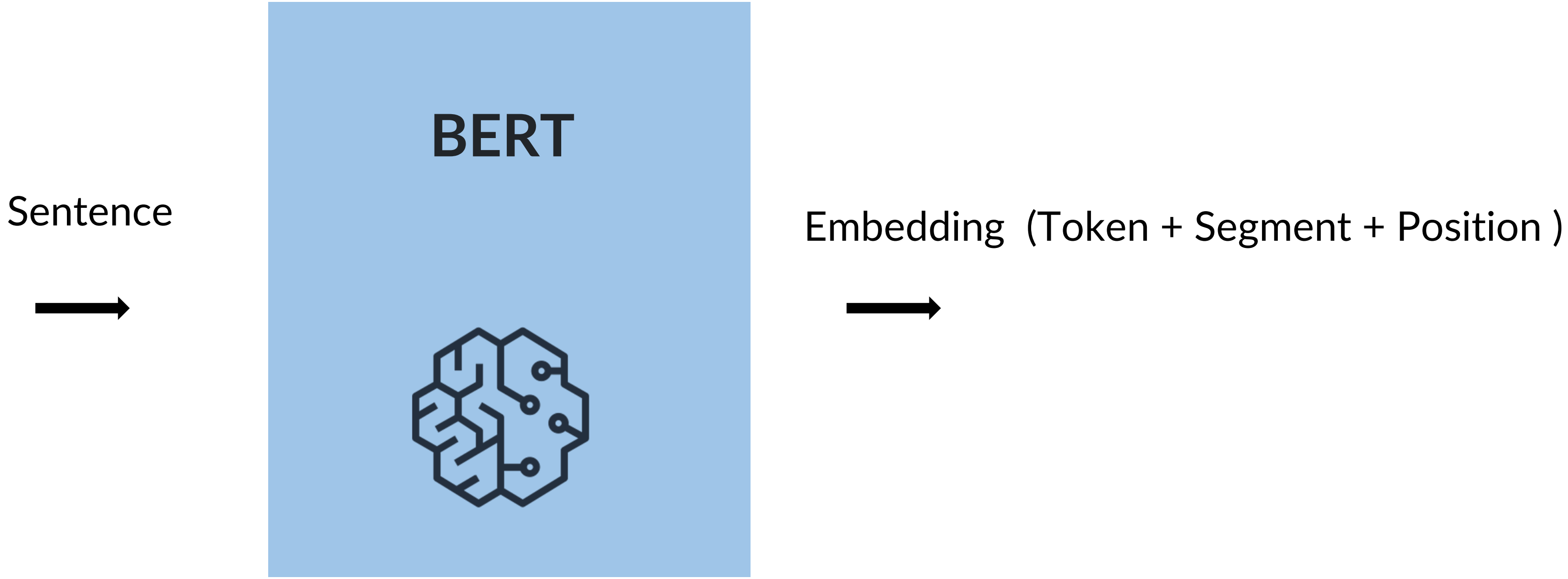
Word → *dress*

BlazingText

Embedding →

.04 .01 .15 -.00 .08 - 0.09 - 0.03

BERT - Contextual Embeddings



BERT - Contextual Embeddings

Sentence → *I love the dress*

BERT

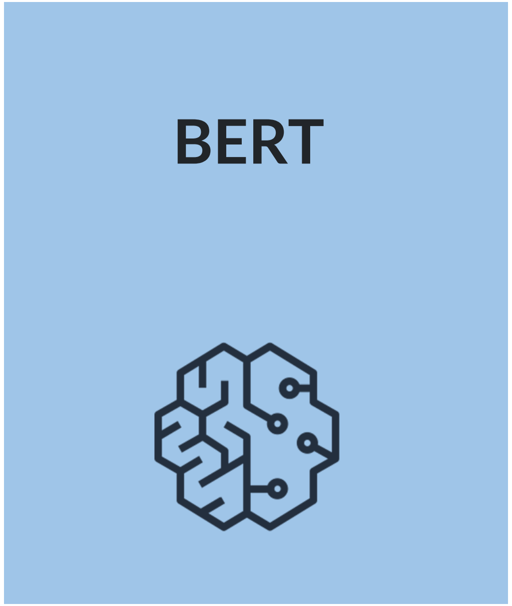
Embedding (Token + Segment + Position) → *Fixed length embedding*

BERT - Contextual Embeddings

Sentence

I love the dress

I love the dress, but not the price



Embedding (Token + Segment + Position)

Fixed length embedding

BERT Embeddings

Input for BERT Model	(1, 4, 768)	**Element wise sum of position, segment and token embedding**
POSITION EMBEDDING Input ID	0 | 1 | 2 | 3 (1, 4, 768)	**Index position in input sequence**
SEGMENT EMBEDDING Segment ID	0 | 0 | 0 | 0 (1, 4, 768)	**0 = Sentence 1** **1 = Sentence 2**
TOKEN EMBEDDING Input ID	101 | 2293 | 2023 | 4377 (1, 4, 768)	**Lookup the 768 dimension vector dimension**
Word Piece Tokenization	[CLS], Love, this, dress	**1 input sequence (consisting of 4 tokens)**
Raw Input **sequence**	Love this dress	**1 input**

BERT Embeddings

Raw Input **sequence**	Love this dress	**1 input**

BERT Embeddings

CLS -> indicates Classification problem
SEP -> token that separates the individual sentences

Word Piece Tokenization (segment words into sub-words with the dimension of 768) + CLS | [CLS], Love, this, dress | **1 input sequence (consisting of 4 tokens)**

Raw Input | Love this dress | **1 input sequence**

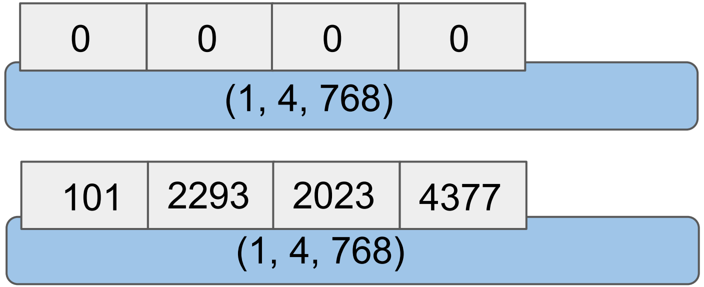
BERT Embeddings

TOKEN EMBEDDING Input ID	101 | 2293 | 2023 | 4377 (1, 4, 768)	**Lookup the 768 dimension vector dimension**
Word Piece Tokenization	[CLS], Love, this, dress	**1 input sequence (consisting of 4 tokens)**
Raw Input **sequence**	Love this dress	**1 input**

BERT Embeddings

SEGMENT EMBEDDING
Segment ID

TOKEN EMBEDDING
Input ID



0 = Sentence 1
1 = Sentence 2

Lookup the 768 dimension vector dimension

Word Piece Tokenization	[CLS], Love, this, dress	**1 input sequence (consisting of 4 tokens)**
Raw Input **sequence**	Love this dress	**1 input**

BERT Embeddings

POSITION EMBEDDING
Input ID

0	1	2	3

(1, 4, 768)

Index position in input sequence

SEGMENT EMBEDDING
Segment ID

0	0	0	0

(1, 4, 768)

0 = Sentence 1
1 = Sentence 2

TOKEN EMBEDDING
Input ID

101	2293	2023	4377

(1, 4, 768)

Lookup the 768 dimension vector dimension

Word Piece Tokenization — [CLS], Love, this, dress — **1 input sequence (consisting of 4 tokens)**

Raw Input **sequence** — Love this dress — **1 input**

BERT Embeddings

1-> 1 input sequence
4-> 4 tokens in the sequence
768 -> 768 dimension vector

Input for BERT Model

(1, 4, 768)

Element wise sum of position, segment and token embedding

POSITION EMBEDDING
Input ID

0	1	2	3

(1, 4, 768)

Index position in input sequence

SEGMENT EMBEDDING
Segment ID

0	0	0	0

(1, 4, 768)

0 = Sentence 1
1 = Sentence 2

TOKEN EMBEDDING
Input ID

101	2293	2023	4377

(1, 4, 768)

Lookup the 768 dimension vector dimension

Word Piece Tokenization

[CLS], Love, this, dress

1 input sequence (consisting of 4 tokens)

Raw Input **sequence**

Love this dress

1 input

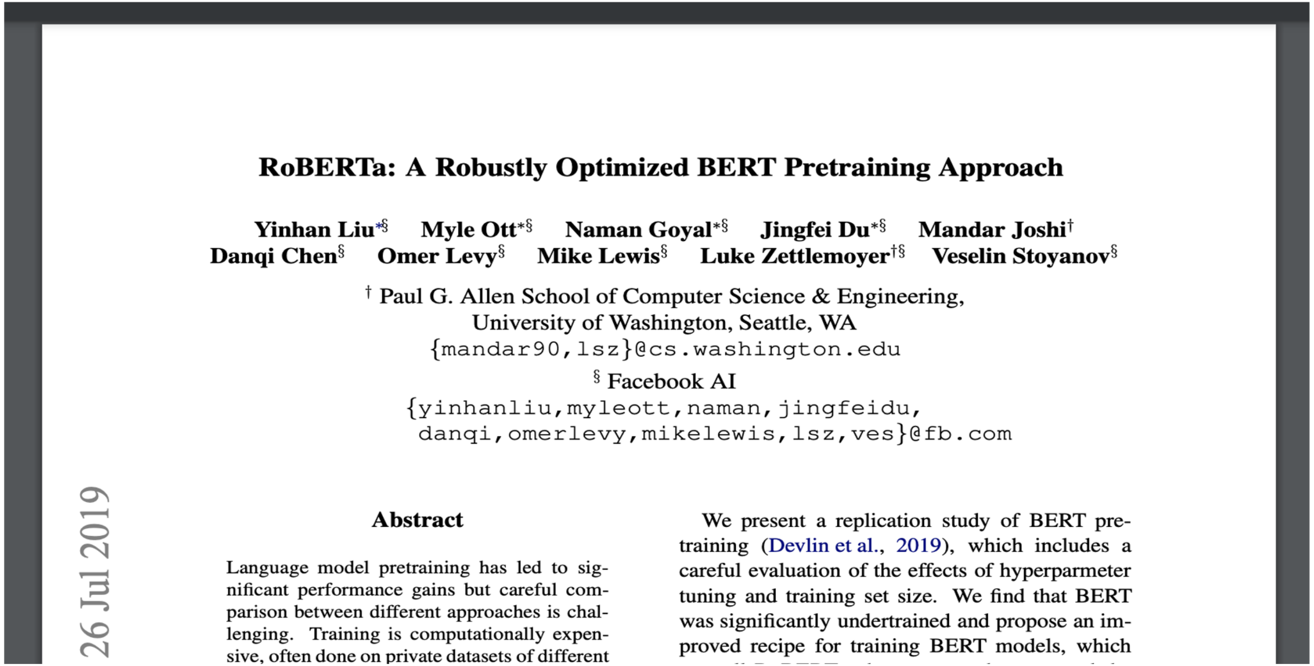
Feature Engineering

At scale with Amazon SageMaker Processing Jobs

RoBERTa model

RoBERTa is built on top of BERT model, but it modifies a few hyper parameters and the way the model is trained.

It also uses a lot more training data than the original BERT model.



RoBERTa: A Robustly Optimized BERT Pretraining Approach

Yinhan Liu*§ Myle Ott*§ Naman Goyal*§ Jingfei Du*§ Mandar Joshi†
Danqi Chen§ Omer Levy§ Mike Lewis§ Luke Zettlemoyer†§ Veselin Stoyanov§

† Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA
{mandar90,lsz}@cs.washington.edu

§ Facebook AI
{yinhanliu,myleott,naman,jingfeidu,
danqi,omerlevy,mikelewis,lsz,ves}@fb.com

26 Jul 2019

Abstract

Language model pretraining has led to significant performance gains but careful comparison between different approaches is challenging. Training is computationally expensive, often done on private datasets of different

We present a replication study of BERT pretraining (Devlin et al., 2019), which includes a careful evaluation of the effects of hyperparmeter tuning and training set size. We find that BERT was significantly undertrained and propose an improved recipe for training BERT models, which

BERT Embeddings with RoBERTa

```
from transformers import RobertaTokenizer

PRE_TRAINED_MODEL_NAME = 'roberta-base'


tokenizer =
RobertaTokenizer.from_pretrained(PRE_TRAINED_MODEL_NAME)
```

Import the Tokenizer class

Create the tokenizer to use based on pre trained model

BERT Embeddings with scikit-learn

encode_plus method

```
def convert_to_bert_input_ids(...):
    encode_plus = tokenizer.encode_plus(
        review,
        add_special_tokens=True,
        max_length=128,
        return_token_type_ids=False,
        padding='max_length',
        return_attention_mask=True,
        return_tensors='pt',
        truncation=True

    return encode_plus['input_ids'].flatten().tolist()
```

Review to be encoded

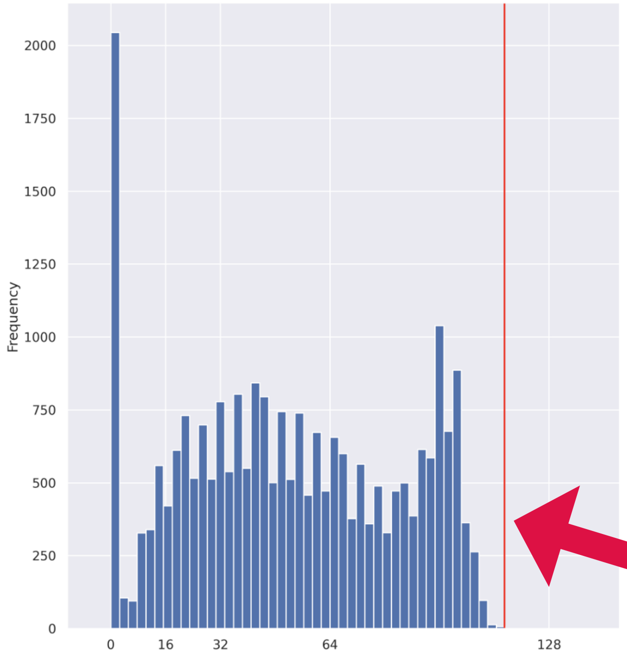
Max sequence length

Defines the max length sequence.

Special tokens

Add special tokens or not

BERT hyper-parameter: **max_seq_length**



mean	52.51
std	31.38
min	1.00
10%	10.00
20%	22.00
30%	32.00
40%	41.00
50%	51.00
60%	61.00
70%	73.00
80%	88.00
90%	97.00
100%	**115.00**

BERT Embeddings with scikit-learn

```
def convert_to_bert_input_ids(...):
    encode_plus = tokenizer.encode_plus(
        review,
        add_special_tokens=True,
        max_length=128,
        return_token_type_ids=False,
        padding='max_length',
        return_attention_mask=True,
        return_tensors='pt',
        truncation=True

    return encode_plus['input_ids'].flatten().tolist()
```

Review to be encoded

Max sequence length

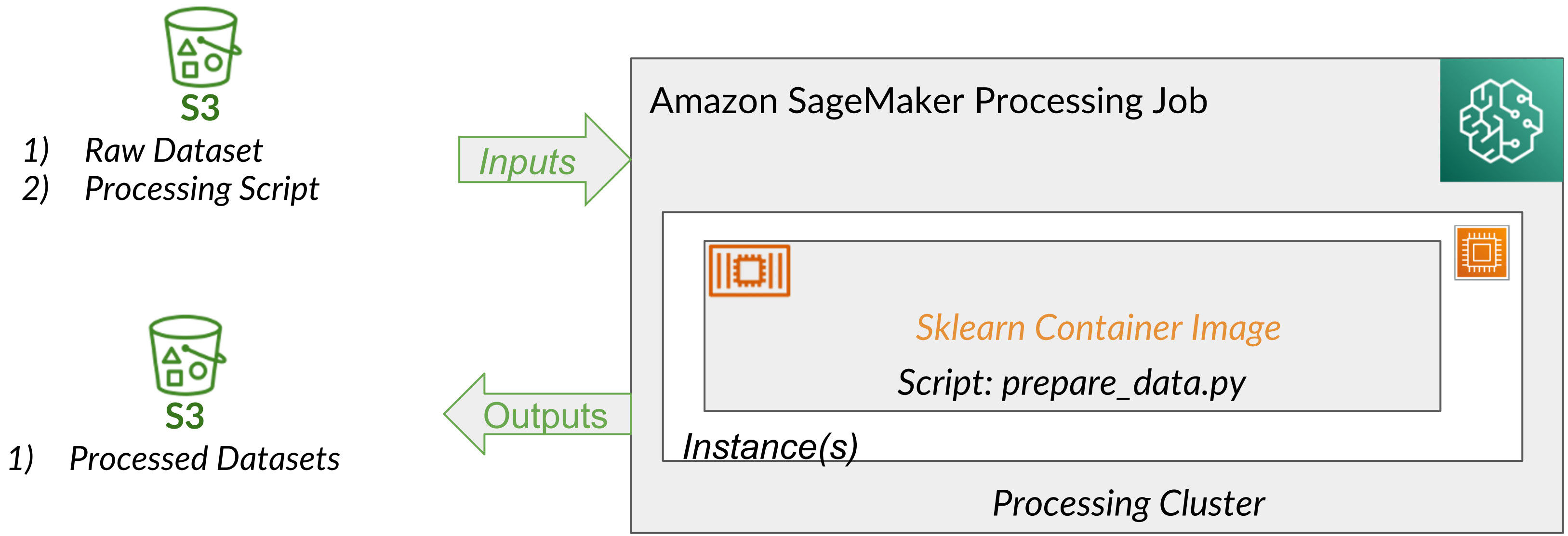
Special tokens

Amazon SageMaker Processing

Allows us to perform feature engineering at scale

Allows us to perform preprocessing, post-processing and Model evaluation at scale by using a distributed cluster.

Execute preprocessing, post processing, model evaluation



You can define how many nodes and the types of nodes that you want to include in a cluster.

Amazon SageMaker Processing with scikit-learn

```
from sagemaker.sklearn.processing import SKLearnProcessor
from sagemaker.processing import ProcessingInput, ProcessingOutput


processor = SKLearnProcessor(
        framework_version='<SCIKIT_LEARN_VERSION>',
        role=role,
        instance_type='ml.c5.4xlarge',
        instance_count=2)

processor.run(<parameters>)
```

Setup processing cluster

Run the processing job

Amazon SageMaker Processing with scikit-learn

```
...
code='preprocess-scikit-text-to-bert.py',

inputs=[
    ProcessingInput(
        input_name='raw-input-data',
        source=raw_input_data_s3_uri,
        ...)
],
```

Scikit-learn script to execute

Input data to transform

Amazon SageMaker Processing with scikit-learn

```
...
outputs=[
    ProcessingOutput(
        output_name='bert-train',
        s3_upload_mode='EndOfJob',
        source='/opt/ml/processing/output/bert/train'),
    ...,
],
```

Output from the processing job

Amazon SageMaker Processing with scikit-learn

Sentiment	Review
1	*this is a great item!*
-1	*not a good product.*
0	*dress is ok*
-1	*do not use! awful. blah*

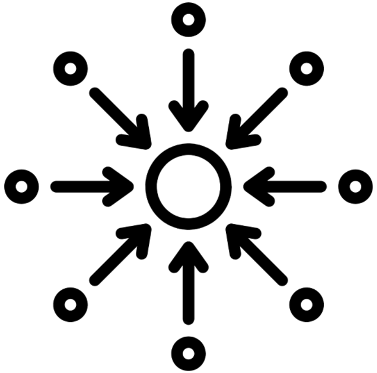
SageMaker Processing

label_id	input_ids		
1	*101*	*2023*	...
-1	*3319*	*1012*	...
0	*2003*	*2307*	...
-1	*102*	*3212*	...

Feature Store

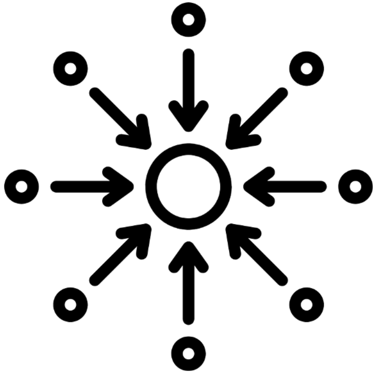
Store the results of Feature engineering efforts and reuse those results, so you don't have to run the feature engineering pipeline again and again.

Feature Store

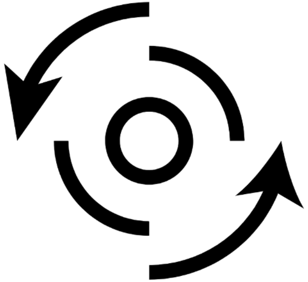


Centralized

Feature Store



Centralized



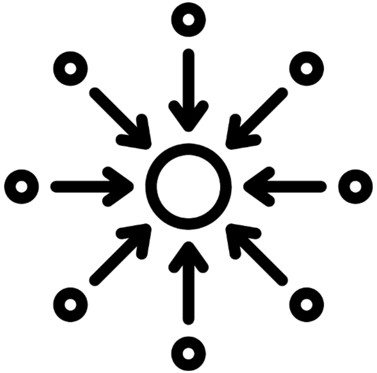
Reusable

Feature Store

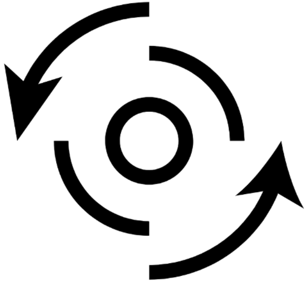
Multiple teams can contribute their features to this centralized repository

Reuse of engineered features, not just across multiple phases of a single machine learning project, but across multiple learning projects.

Any team member can come in and search for the features they want, and use the search results in their own machine learning projects.



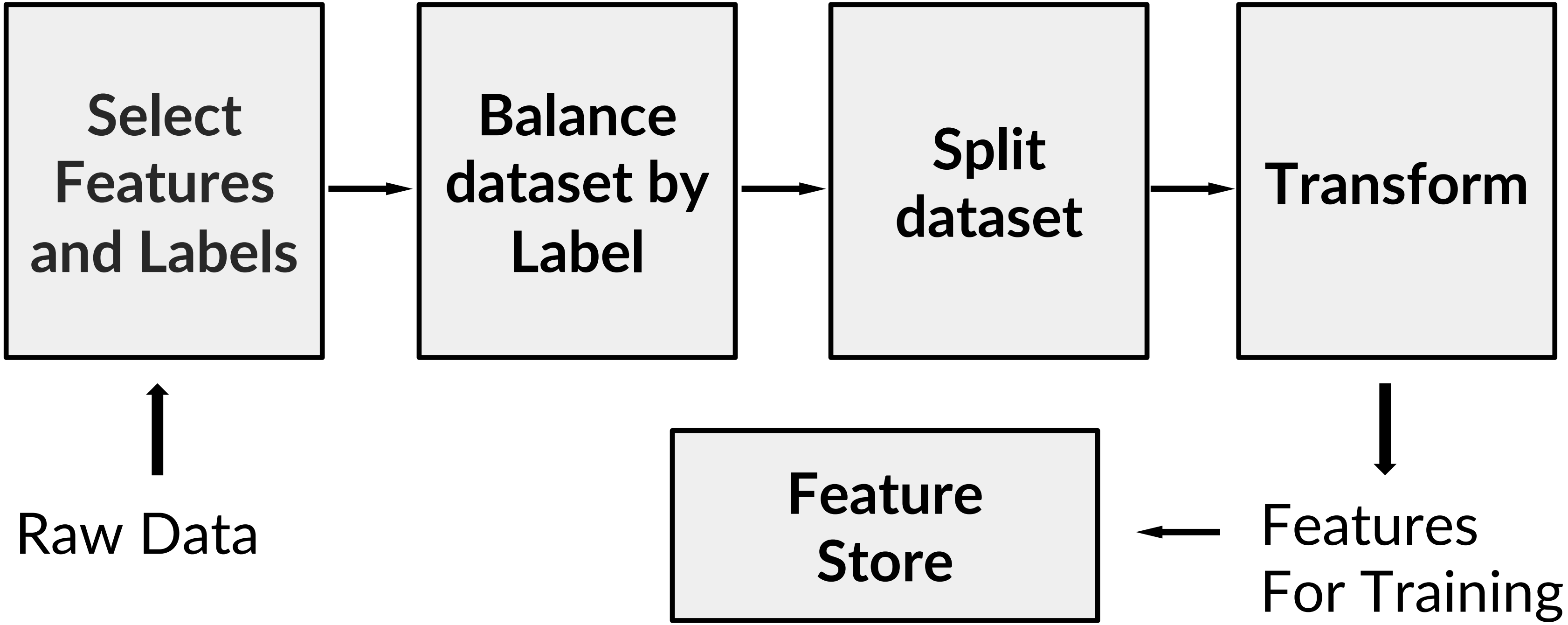
Centralized



Reusable

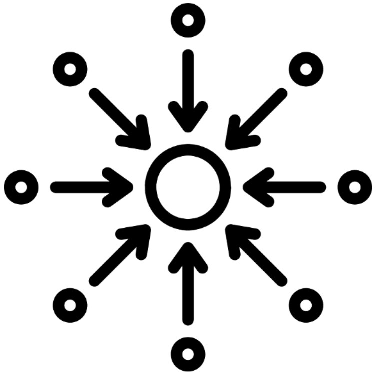
Discoverable

Feature Engineering Pipeline Extended



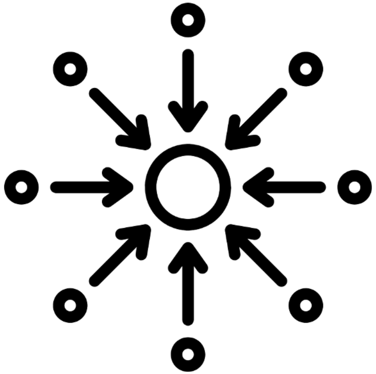
Amazon SageMaker Feature Store

Amazon SageMaker Feature Store

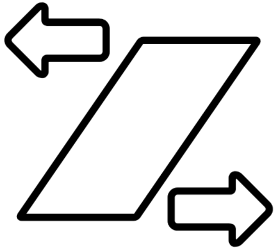


Store and Serve Features

Amazon SageMaker Feature Store

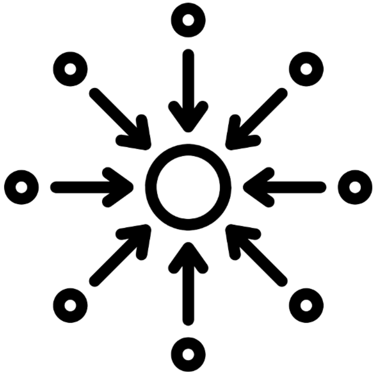


Store and Serve Features

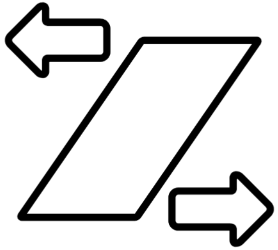


Reduce skew

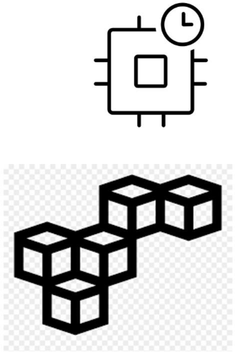
Amazon SageMaker Feature Store



Store and Serve Features



Reduce skew



Real time & Batch

Amazon SageMaker Feature Store - Create

```
from sagemaker.feature_store.feature_group import FeatureGroup

reviews_feature_group_name = "reviews_distilbert_max_seq_length_128"

reviews_feature_group = FeatureGroup(
     name=...,
     feature_definitions=...,
     sagemaker_session=sagemaker_session)

reviews_feature_group.create(
    s3_uri="s3://{}/{}".format(bucket, prefix),
    record_identifier_name=record_identifier_feature_name,
    event_time_feature_name=event_time_feature_name,
    role_arn=role)
```

Name

Create

Feature Group is a construct that allows you to group multiple features together and treat them as a set.

First, you define a feature group.Name, definitions, and sagemaker session
definition -> name and type of the features.

Create method expects an s3 location where the feature group, along with the individual features will be saved.

Amazon SageMaker Feature Store - Ingest

```
reviews_feature_group.ingest(
    data_frame=df_records,
    max_workers=3,
    wait=True)
```

Ingest

Ingest API is used to ingest feature into the feature group in a multi-threaded fashion

Amazon SageMaker Feature Store - Retrieve

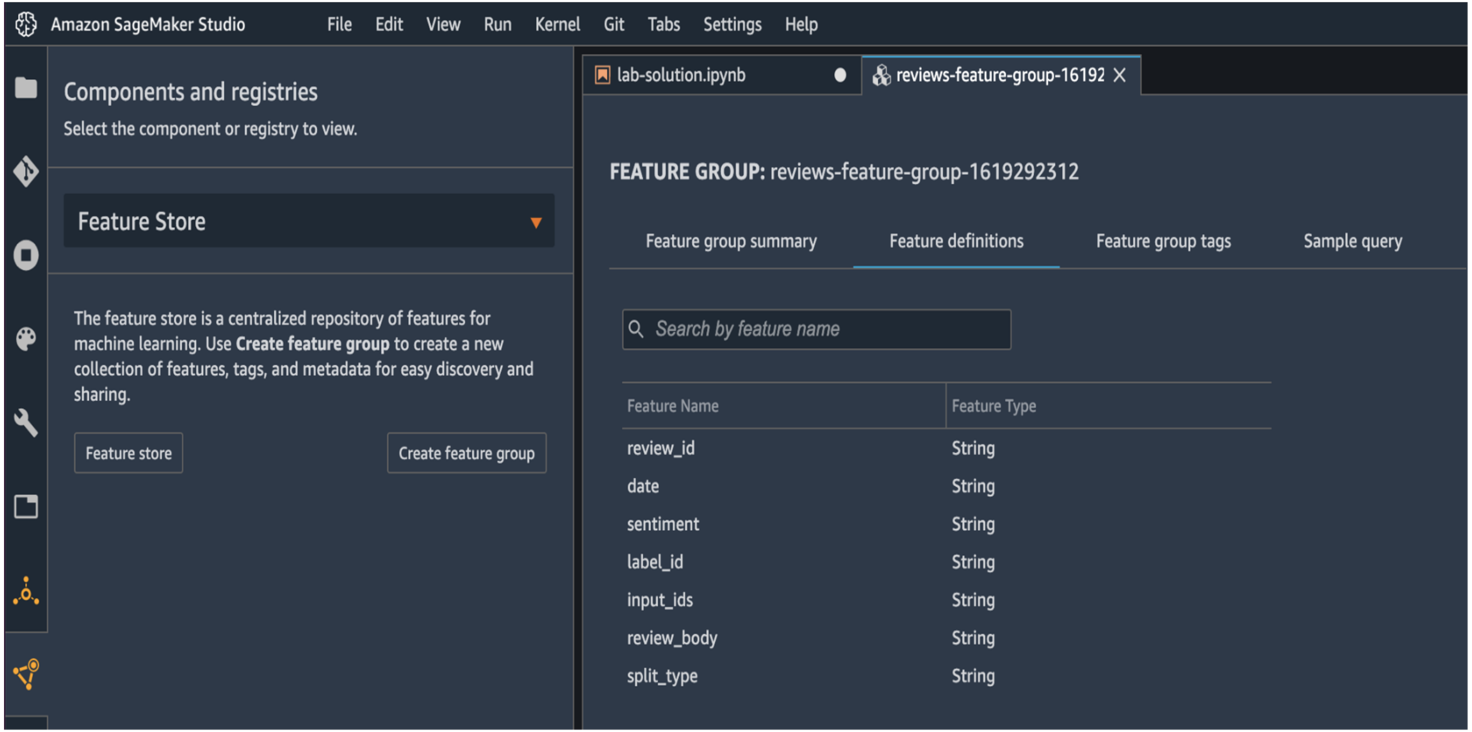
```
reviews_feature_store_query =
    reviews_feature_group.athena_query()

reviews_feature_store_table =
    reviews_feature_store_query.table_name

query_string = 'SELECT review_body, input_ids, input_mask, segment_ids,
label_id FROM "{}" LIMIT 5'.format(reviews_feature_store_table)

reviews_feature_store_query.run(
    query_string=..., ...)
```

Query S3

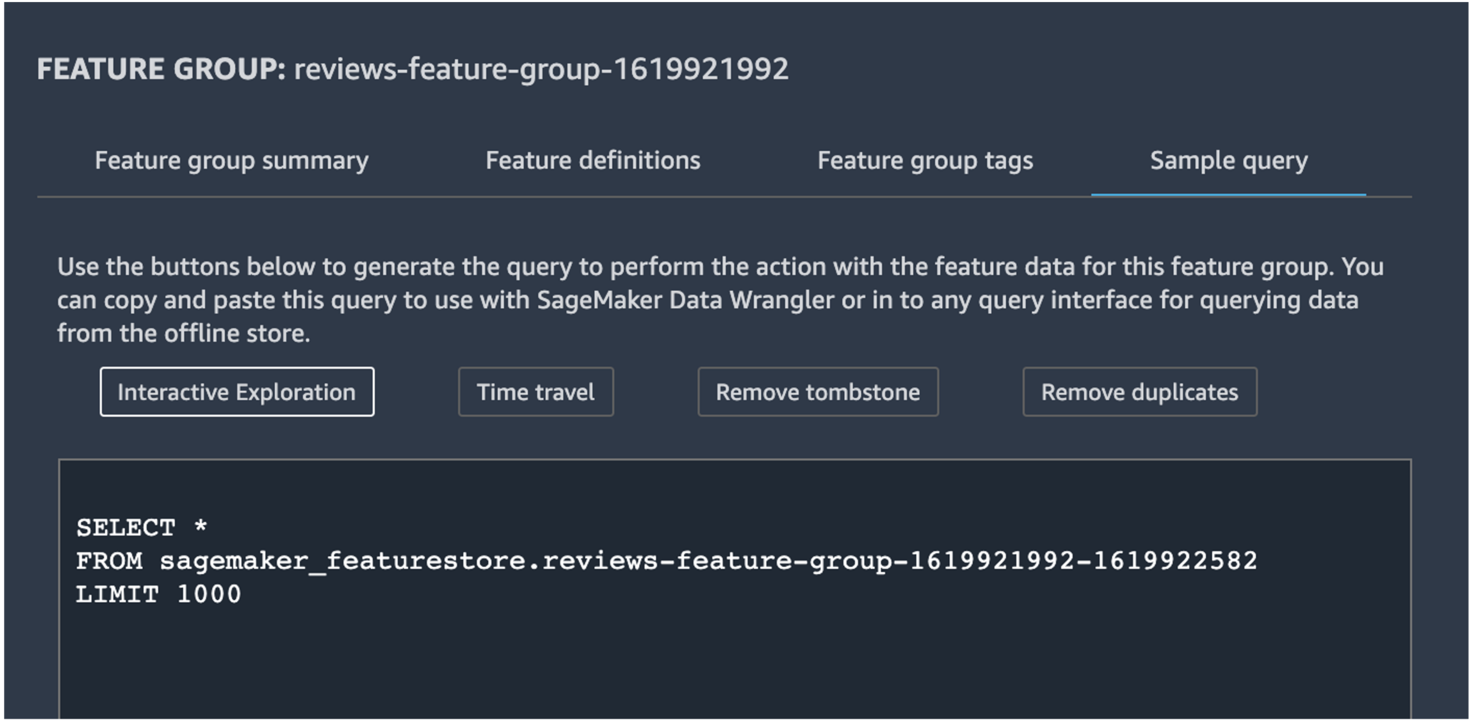
Query string

Execute the query

Amazon SageMaker Feature Store In SageMaker Studio



Amazon SageMaker Feature Store In SageMaker Studio



Amazon SageMaker Feature Store In SageMaker Studio

	date	review_id	sentiment	label_id	input_ids	review_body
0	2021-04-29T18:34:07Z	14136	1	2	[0, 713, 16, 10, 182, 22, 4903, 3760, 254, 22, 2125, 4, 939, 657, 24, 328, 939, 2813, 6215, 74, ...	This is a very "retailer " piece. i love it! i wish retailer would bring back more pieces like t...
1	2021-04-29T18:34:07Z	4026	0	1	[0, 100, 1432, 5, 6173, 8, 2162, 10, 2514, 1836, 11, 5, 2440, 33953, 4, 939, 524, 2333, 10, 1836...	I followed the reviews and bought a larger size in the blue stripe. i am usually a size 8 but or...
2	2021-04-29T18:34:07Z	7522	-1	0	[0, 713, 8443, 16, 98, 11962, 8, 10698, 1969, 8, 939, 657, 5, 32847, 4, 959, 1437, 24, 24232, 90...	This jacket is so cute and fits perfect and i love the motif. however it deposited black linty ...
3	2021-04-29T18:34:07Z	7618	-1	0	[0, 133, 19111, 738, 8, 1421, 738, 32, 2198, 430, 4, 24, 18, 101, 45, 190, 5, 276, 3588, 4, 5, 5...	The catalog shot and model shot are completely different. it's like not even the same dress. the...
4	2021-04-29T18:34:07Z	11942	1	2	[0, 100, 269, 101, 5, 356, 9, 42, 8443, 1437, 53, 939, 206, 24, 1237, 650, 8, 939, 531, 671, 5, ...	I really like the look of this jacket but i think it runs small and i must return the one i rec...