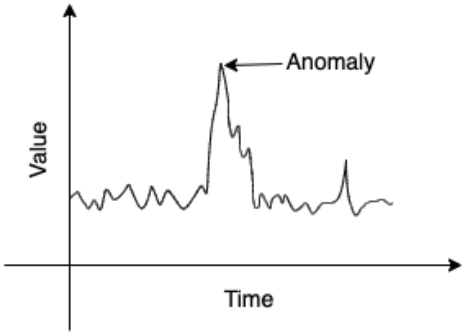
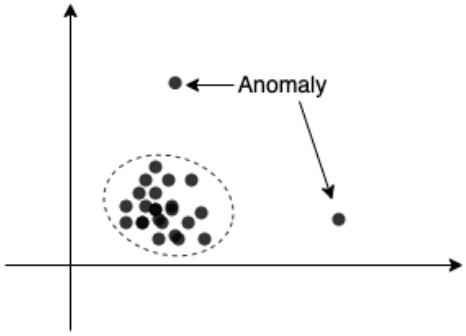
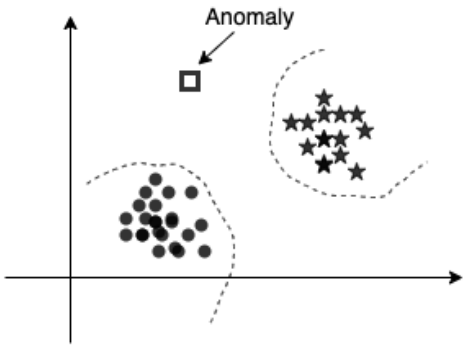
Chapter 1: Understanding Deep Learning Anomaly Detection





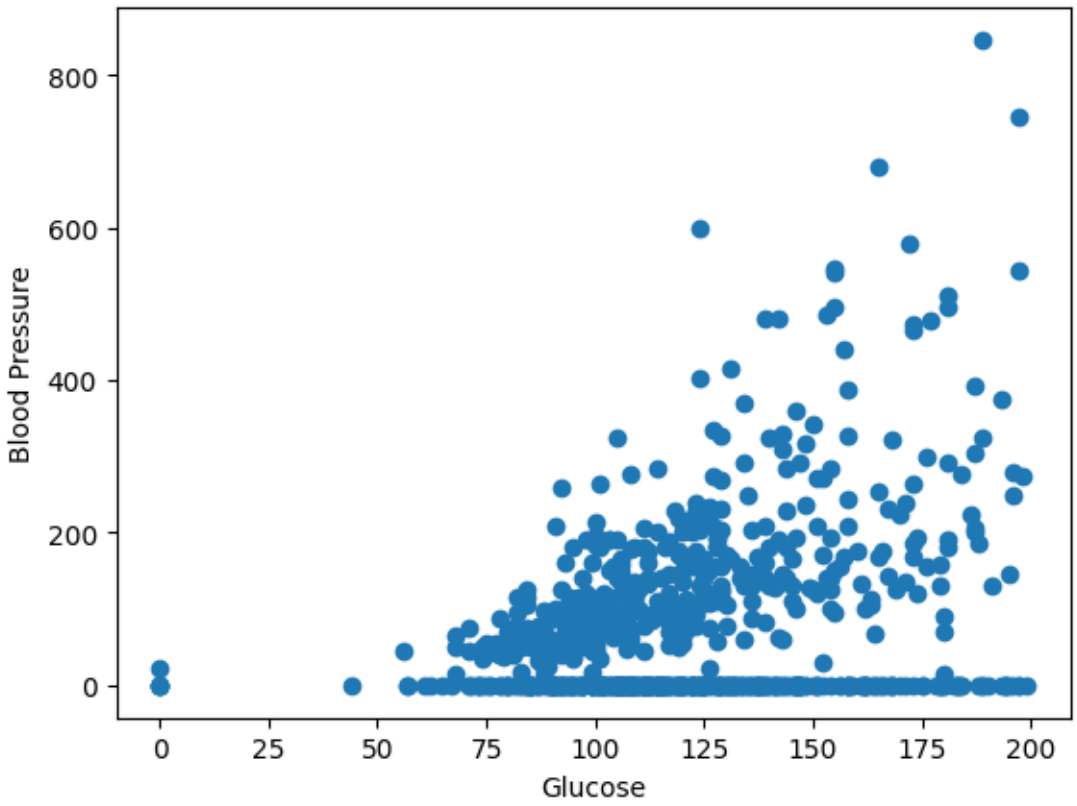
Feature 1	Feature 2	Class
115	10	1
130	15	1
300	**500**	**0**
100	12	1

Anomaly



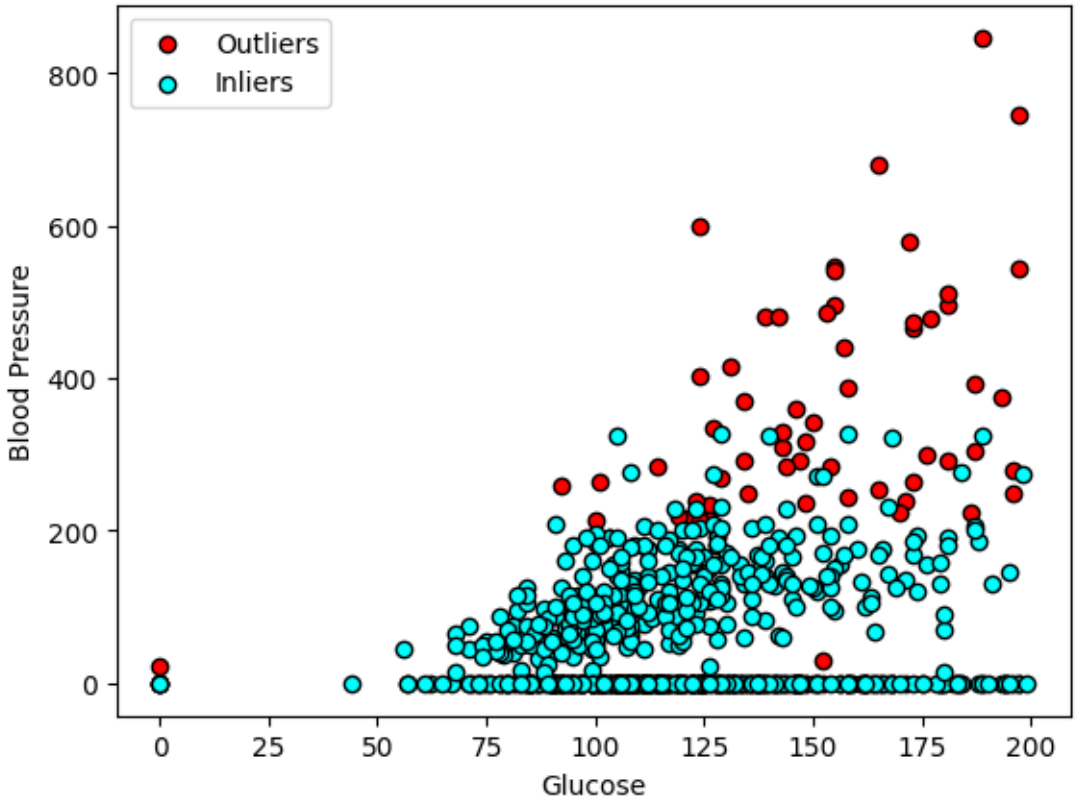
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

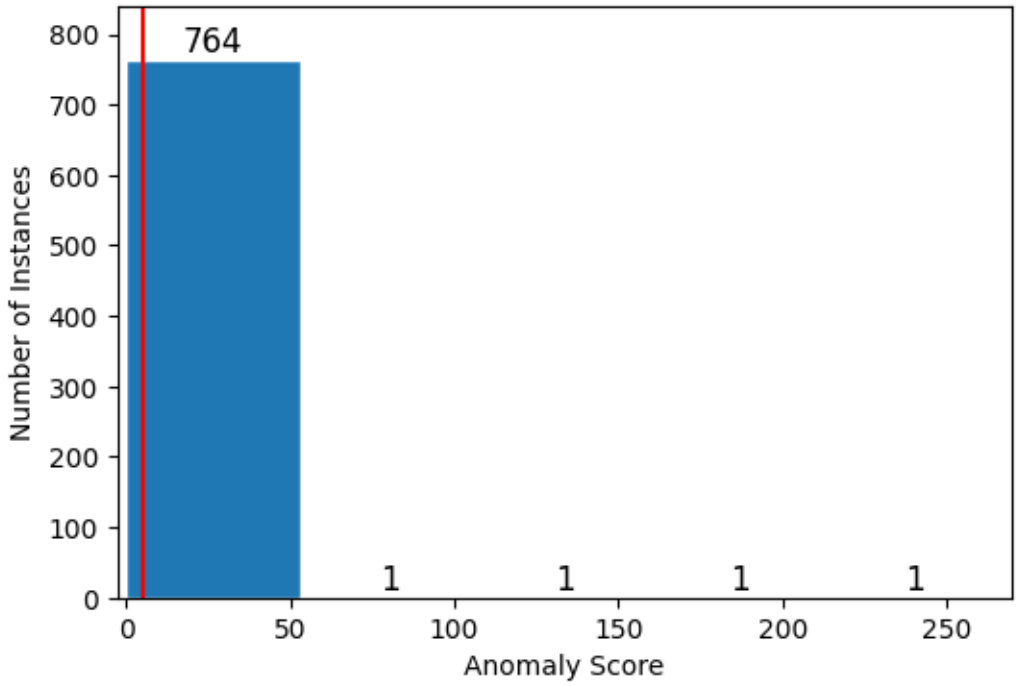
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000



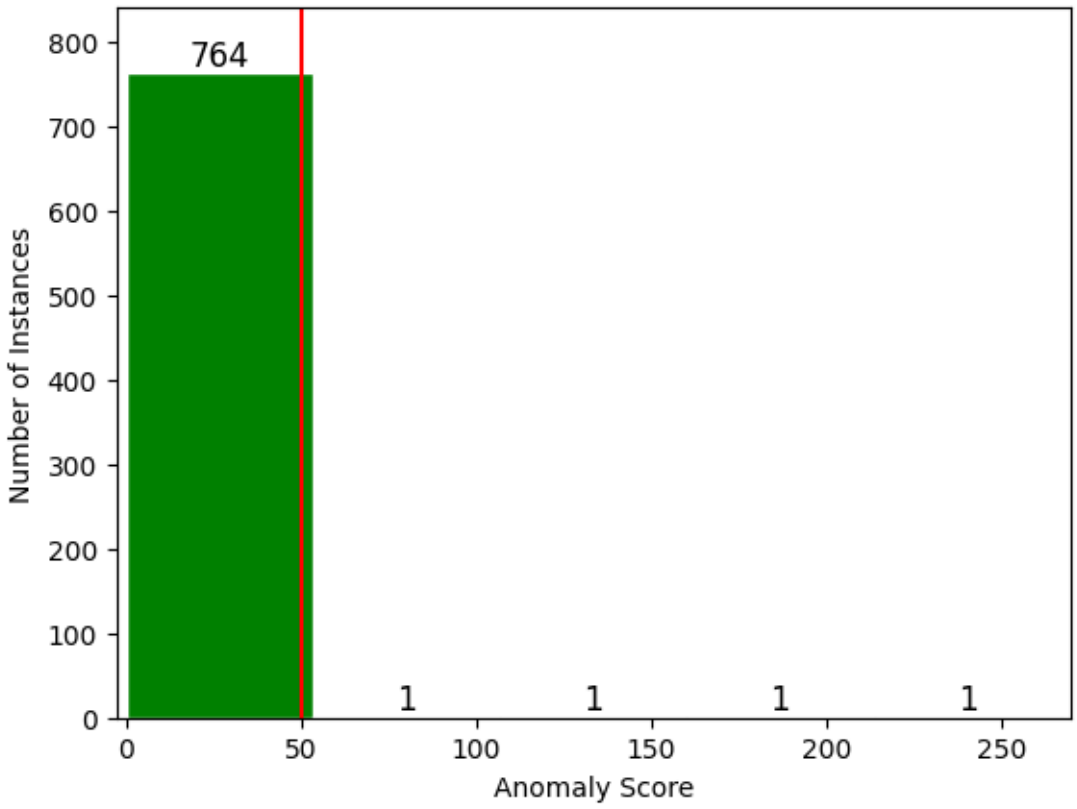

```
(array([  8,  13,  20,  31,  43,  52,  53,  54,  56,  73,  99, 111, 132,
        144, 153, 162, 182, 186, 199, 206, 220, 228, 231, 247, 248, 254,
        258, 286, 287, 296, 297, 323, 335, 359, 364, 370, 388, 392, 409,
        412, 415, 458, 485, 486, 487, 540, 545, 555, 574, 584, 606, 645,
        655, 673, 679, 695, 707, 710, 713, 715, 753]),)
```

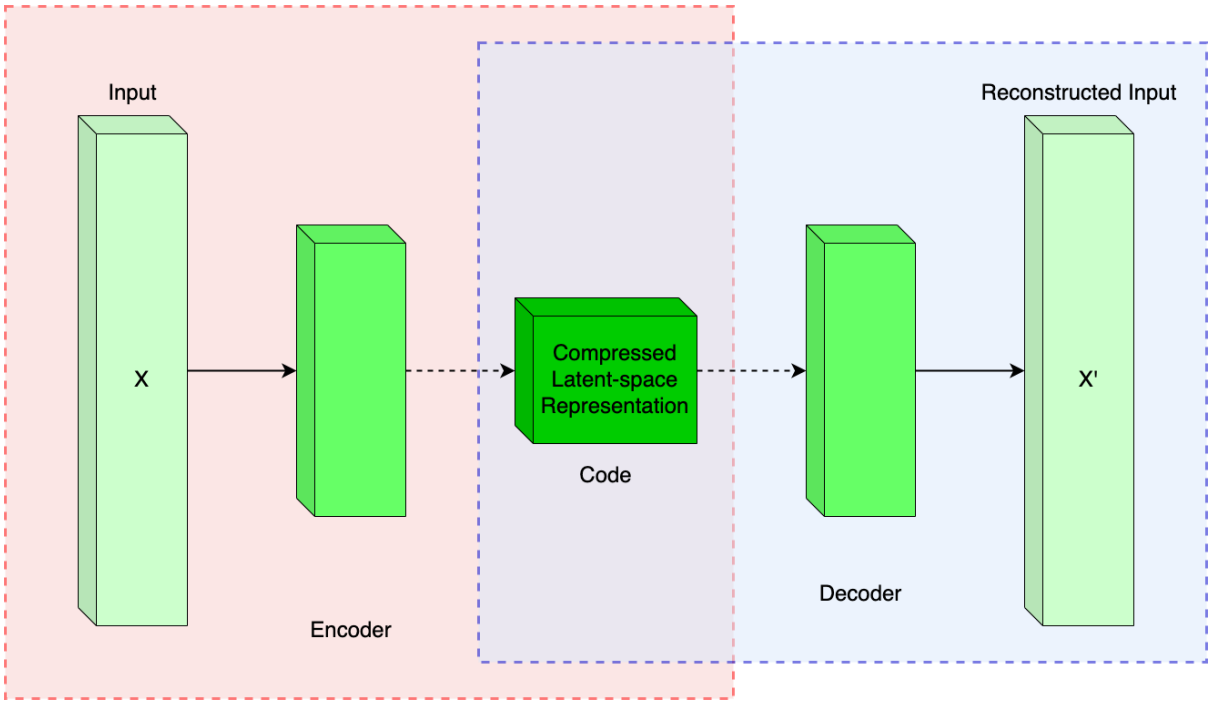
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
8	2	197	70	45	543	30.5	0.158	53	1
13	1	189	60	23	846	30.1	0.398	59	1
20	3	126	88	41	235	39.3	0.704	27	0
31	3	158	76	36	245	31.6	0.851	28	1
43	9	171	110	24	240	45.4	0.721	54	1
52	5	88	66	21	23	24.4	0.342	30	0
53	8	176	90	34	300	33.7	0.467	58	1
54	7	150	66	42	342	34.7	0.718	42	0
56	7	187	68	39	304	37.7	0.254	41	1
73	4	129	86	20	270	35.1	0.231	23	0





	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
13	1	189	60	23	846	30.1	0.398	59	1





	Time	V1	V2	V3	V4	V5	...	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	...	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	...	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	...	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	...	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	...	-0.206010	0.502292	0.219422	0.215153	69.99	0

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 30)                930

 dropout (Dropout)           (None, 30)                0

 dense_1 (Dense)             (None, 30)                930

 dropout_1 (Dropout)         (None, 30)                0

 dense_2 (Dense)             (None, 64)                1984

 dropout_2 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 30)                1950

 dropout_3 (Dropout)         (None, 30)                0

 dense_4 (Dense)             (None, 30)                930

 dropout_4 (Dropout)         (None, 30)                0

 dense_5 (Dense)             (None, 64)                1984

 dropout_5 (Dropout)         (None, 64)                0

 dense_6 (Dense)             (None, 30)                1950

=================================================================
Total params: 10,658
Trainable params: 10,658
Non-trainable params: 0
```

```
1/1 [==============================] - 0s 23ms/step
array([1])
```

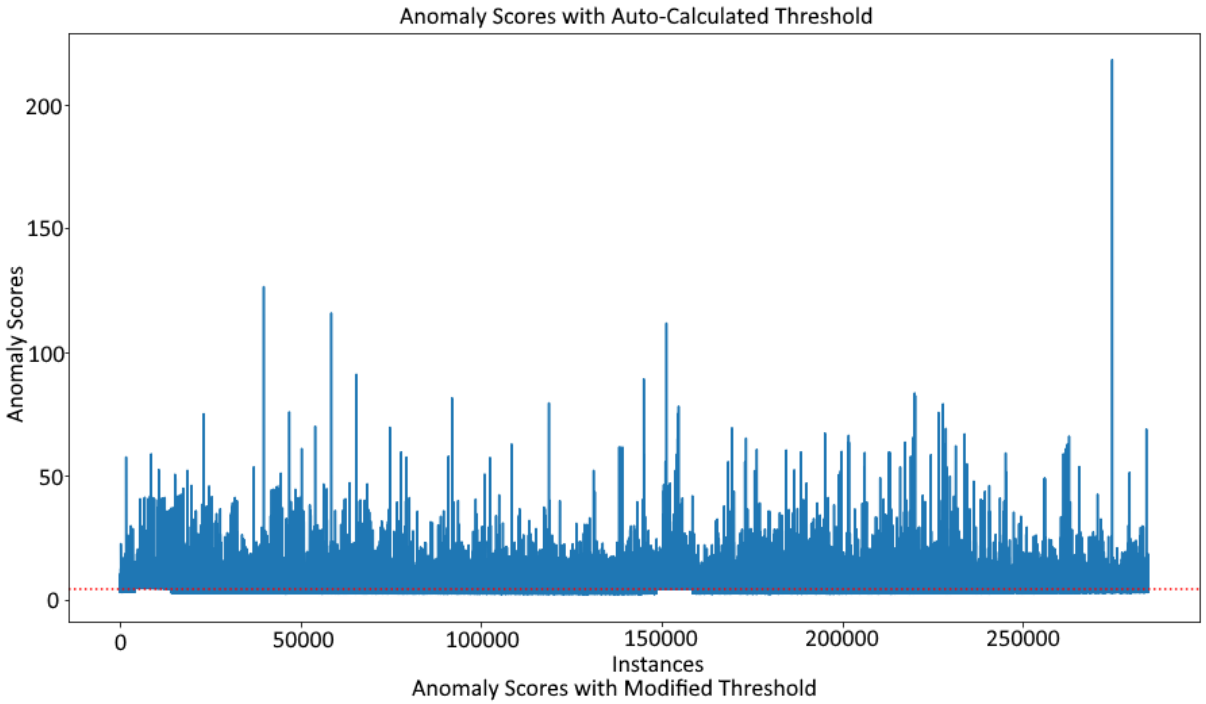
```
X.iloc[[4920]]
```

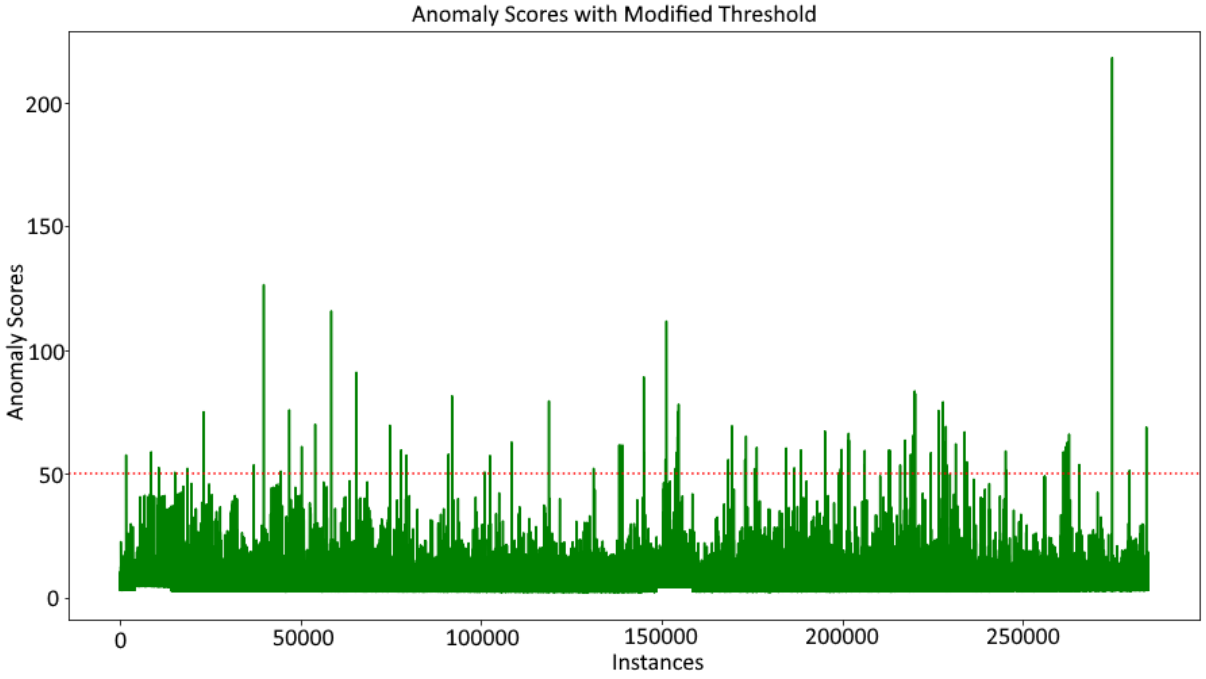
	Time	V1	V2	V3	V4	V5	...	V25	V26	V27	V28	Amount
4920	4462.0	-2.30335	1.759247	-0.359745	2.330243	-0.821628	...	-0.156114	-0.542628	0.039566	-0.153029	239.93

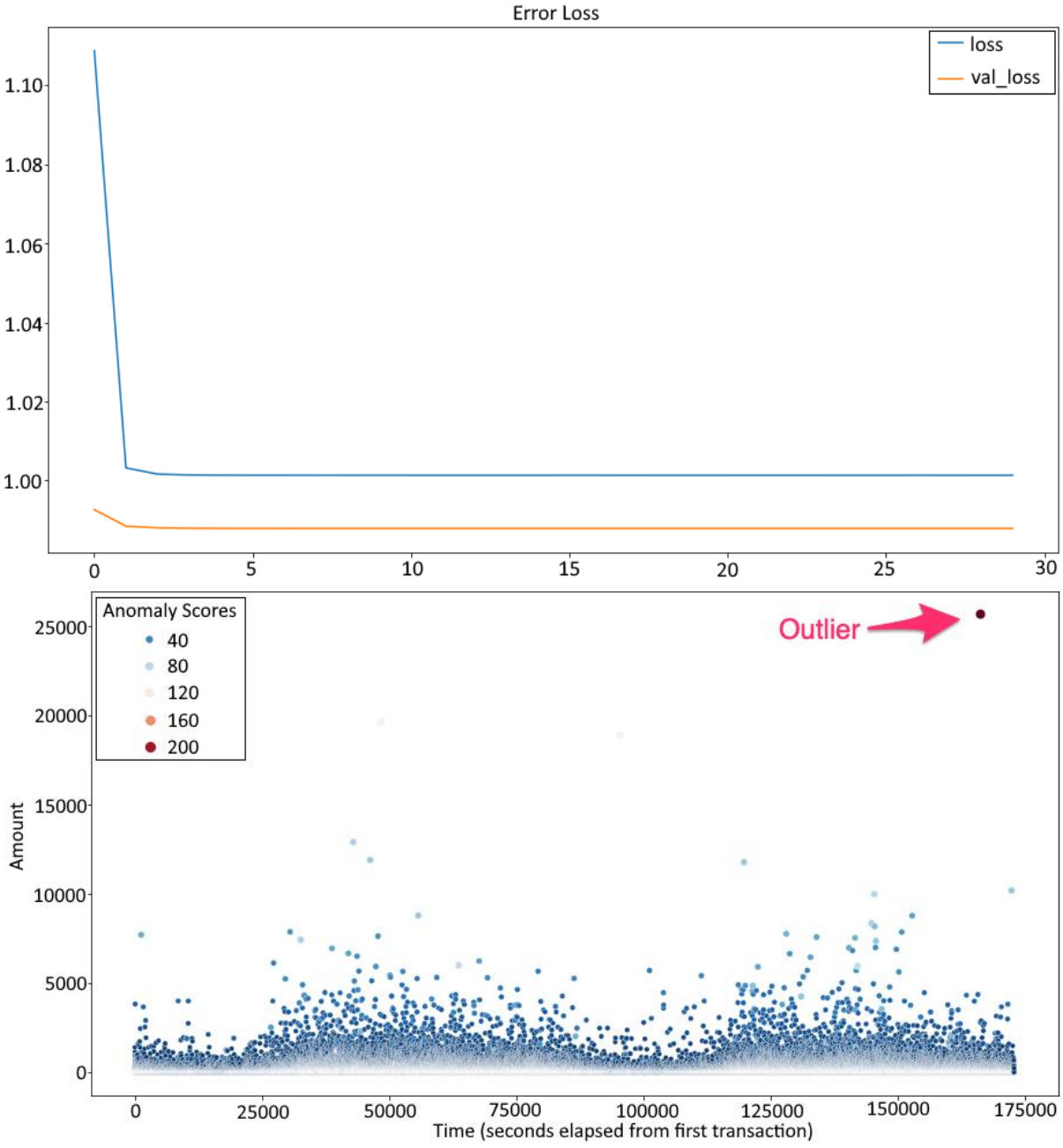
1 rows × 30 columns

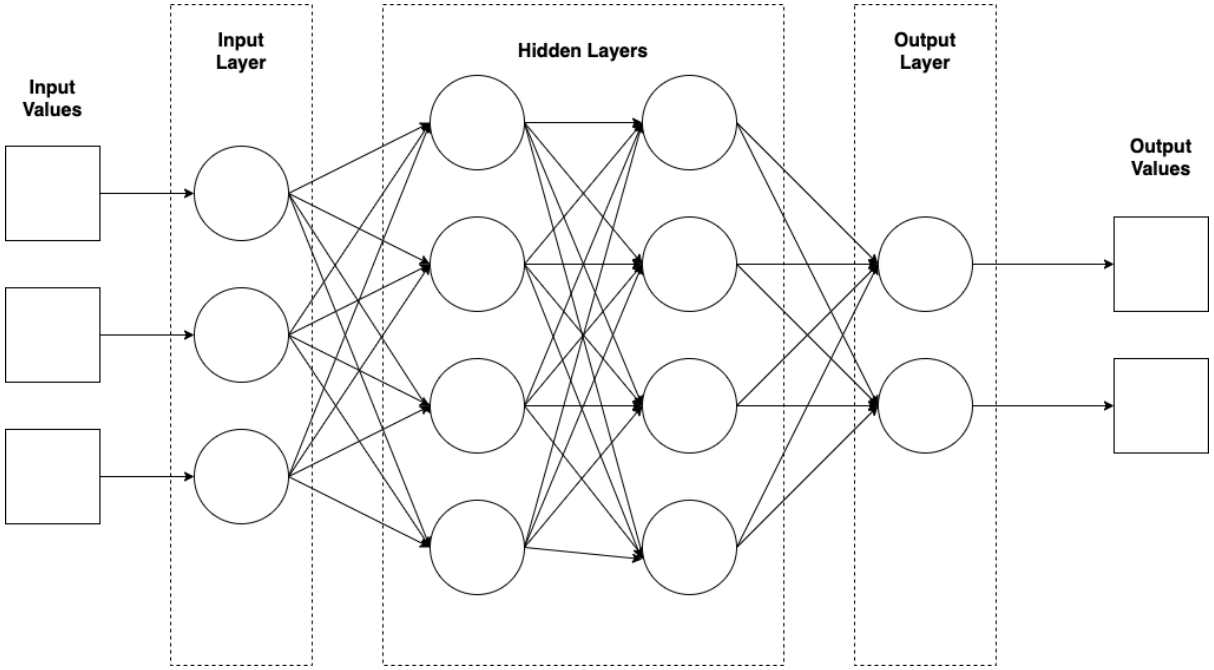
```
y.iloc[[4920]]
```

```
4920    1
Name: Class, dtype: int64
```

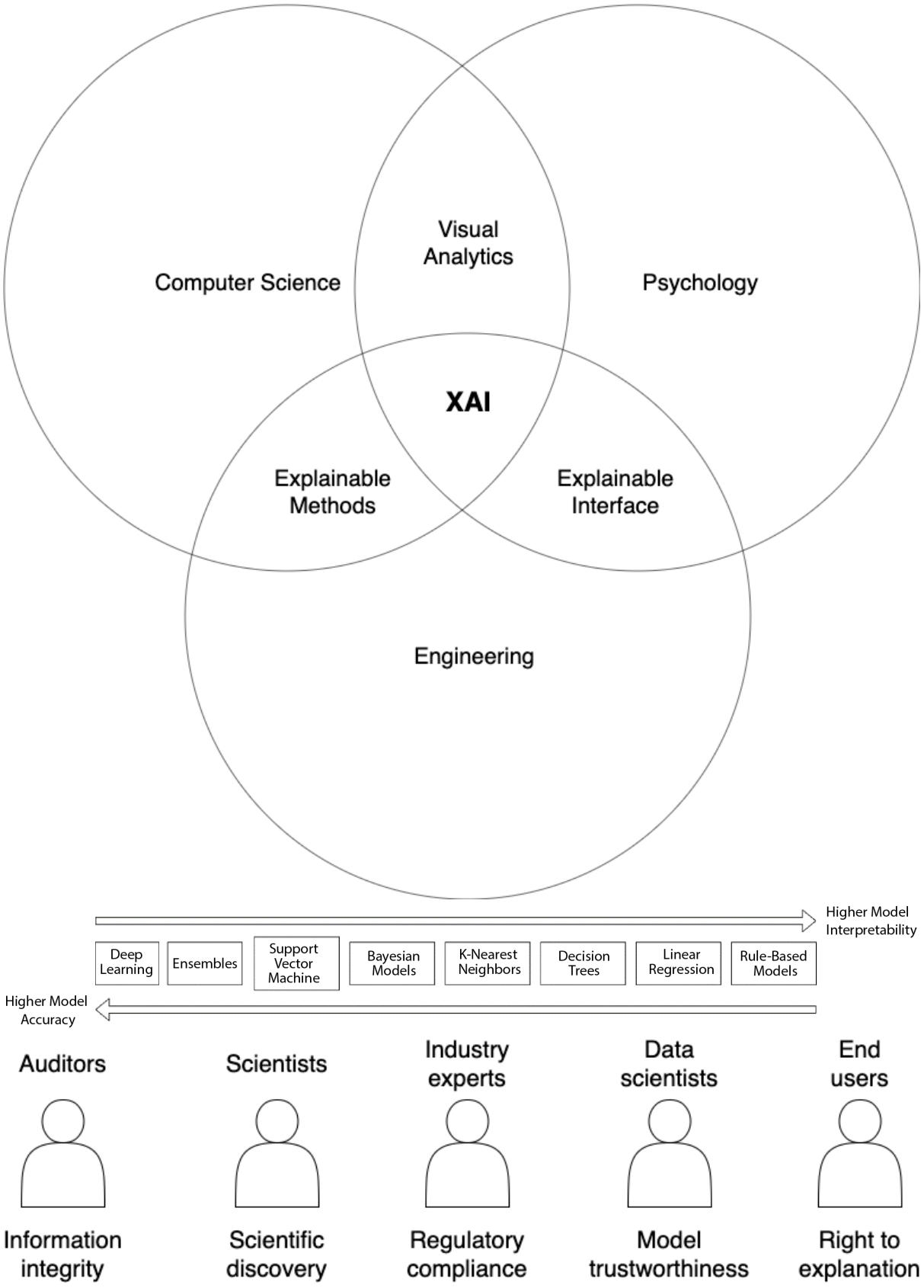

Anomaly Scores with Auto-Calculated Threshold
Anomaly Scores
Instances
0
50
100
150
200
0
50000
100000
150000
200000
250000

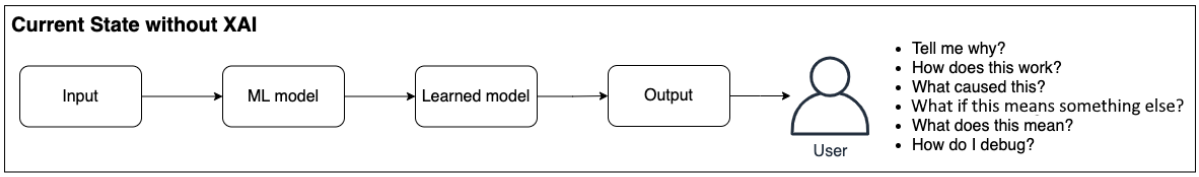

Anomaly Scores with Modified Threshold
Anomaly Scores
Instances
0
50
100
150
200
0
50000
100000
150000
200000
250000

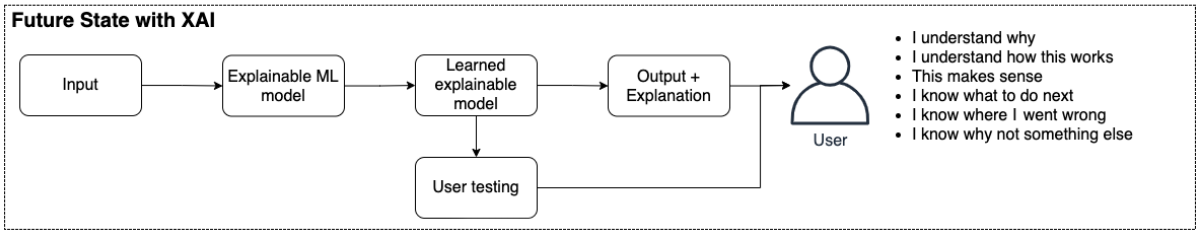

Error Loss
loss
val_loss
1.10
1.08
1.06
1.04
1.02
1.00
0
5
10
15
20
25
30
Anomaly Scores
40
80
120
160
200
Outlier
Amount
25000
20000
15000
10000
5000
0
0
25000
50000
75000
100000
125000
150000
175000
Time (seconds elapsed from first transaction)

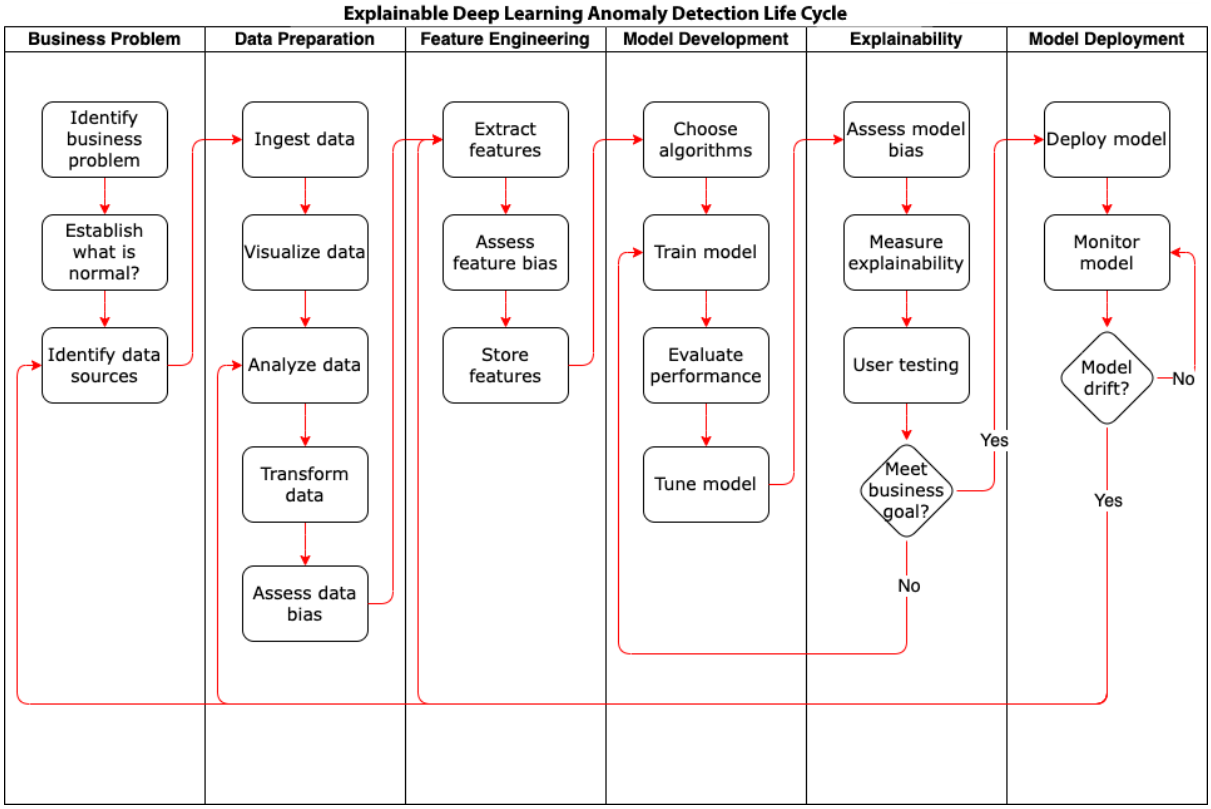

Input Values
Input Layer
Hidden Layers
Output Layer
Output Values

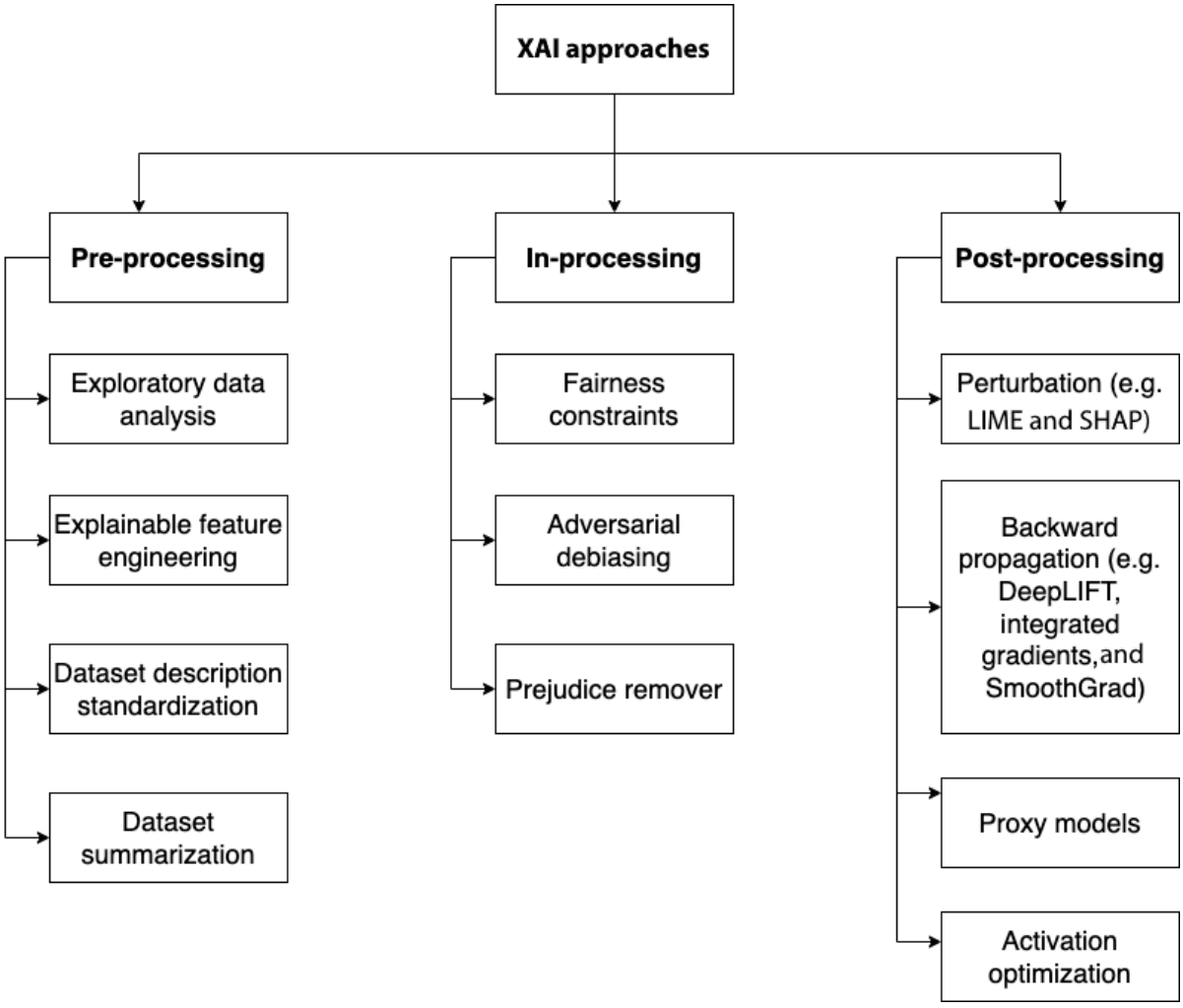
Chapter 2: Understanding Explainable AI



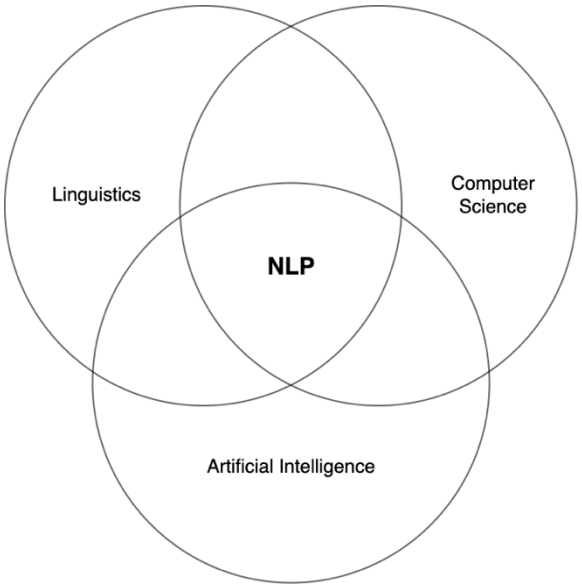

Current State without XAI
Input
ML model
Learned model
Output
User
Tell me why?
How does this work?
What caused this?
What if this means something else?
What does this mean?
How do I debug?


Future State with XAI
Input
Explainable ML model
Learned explainable model
Output + Explanation
User
User testing
I understand why
I understand how this works
This makes sense
I know what to do next
I know where I went wrong
I know why not something else


Explainable Deep Learning Anomaly Detection Life Cycle
Business Problem
Data Preparation
Feature Engineering
Model Development
Explainability
Model Deployment
Identify business problem
Establish what is normal?
Identify data sources
Ingest data
Visualize data
Analyze data
Transform data
Assess data bias
Extract features
Assess feature bias
Store features
Choose algorithms
Train model
Evaluate performance
Tune model
Assess model bias
Measure explainability
User testing
Meet business goal?
Yes
No
Deploy model
Monitor model
Model drift?
No
Yes


XAI approaches
Pre-processing
In-processing
Post-processing
Exploratory data analysis
Explainable feature engineering
Dataset description standardization
Dataset summarization
Fairness constraints
Adversarial debiasing
Prejudice remover
Perturbation (e.g. LIME and SHAP)
Backward propagation (e.g. DeepLIFT, integrated gradients,and SmoothGrad)
Proxy models
Activation optimization

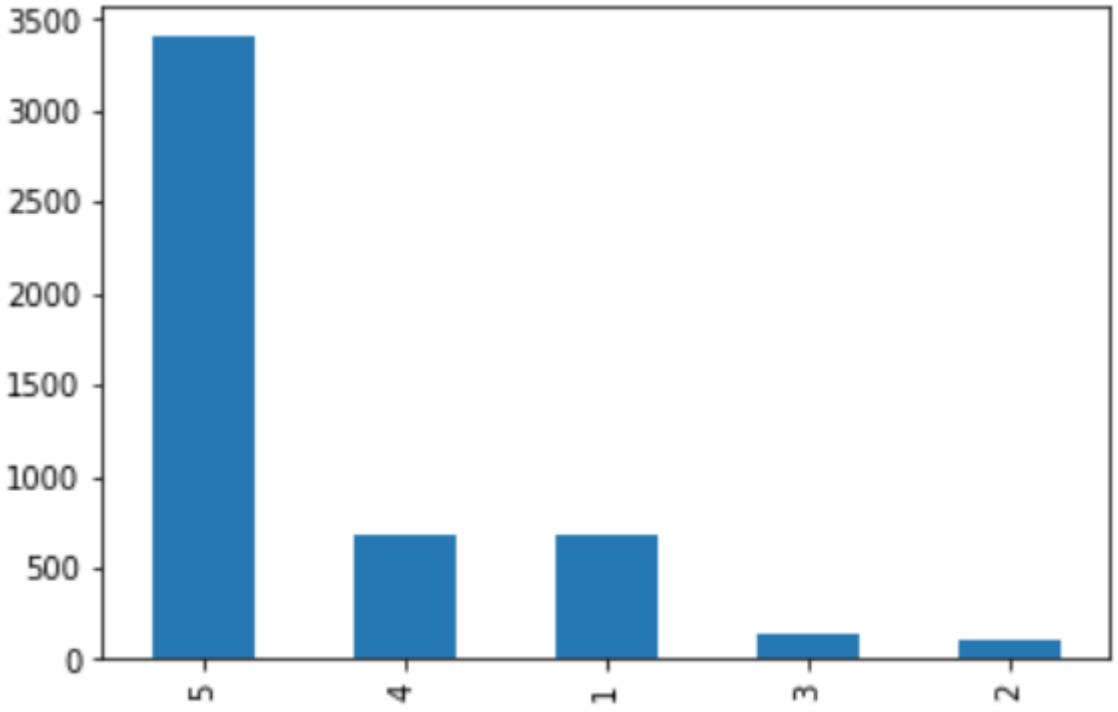
Chapter 3: Natural Language Processing Anomaly Explainability

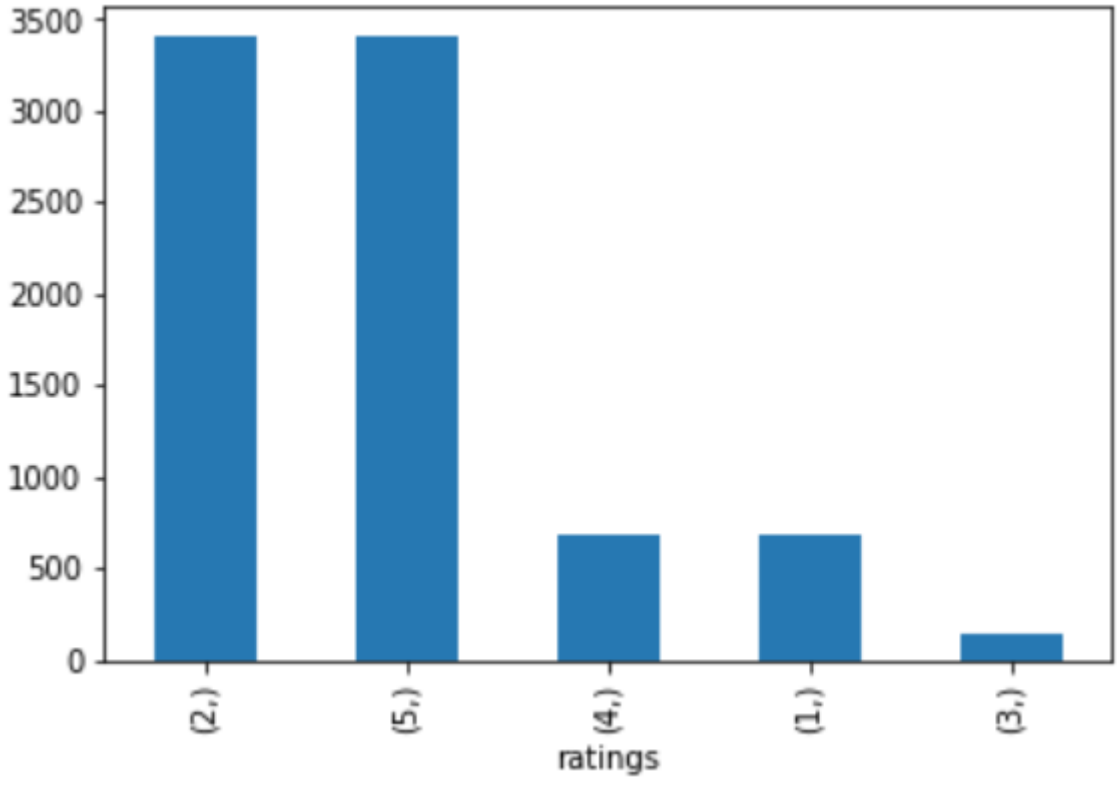


	reviews	ratings
0	I've been ordering this product monthly for over a year. These past few shipments have been off, and the packaging is different. It is obvious they are from two different manufacturers. The full bottle is the imposter and has no pink ribbon on the label, the label photo is different, and if you look at the bottle, you can see on the back that the legitimate product has a 3% Juice Content label on the back up top, and the imposter has a 1% Juice Content. There are also other differences on the label. See Photos for comparison. Two different products, and the IMPOSTER is absolutely undrinkable and tastes nothing like the original product.I contacted Amazon and they are pulling this flavor, Red Grapefruit, off the shelf while they investigate this issue. They also issued me a full credit. I hope they can get this figured out because this is my favorite flavor and I drink it everyday.	4
1	I used to drink these Sparkling Ice waters in all flavors every day. They all taste GREAT, my favorite was Black Raspberry. However, I just discovered that they contain sucralose which is just another harmful artificial sweetener. The only safe sweeteners are Stevia, Agave, and Raw Honey. I have stopped drinking/eating anything with harmful artificial sweeteners since they have been linked to many diseases, including dementia and the increase in Atzheimers disease patients.	5

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   reviews  5000 non-null   object
 1   ratings  5000 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 78.2+ KB
```

```
5    3402
4    683
1    677
3    134
2    104
array([4, 5, 1, 2, 3]) Name: ratings, dtype: int64
```





```
Train shape: (6638, 2)
Test shape: (1660, 2)
```

```
Global seed set to 123
GPU available: False, used: False
TPU available: False, using: 0 TPU cores
IPU available: False, using: 0 IPUs
HPU available: False, using: 0 HPUs

  | Name              | Type                           | Params
-------------------------------------------------------------------
0 | model             | HFAutoModelForTextPrediction   | 108 M
1 | validation_metric | Accuracy                       | 0
2 | loss_func         | CrossEntropyLoss               | 0
-------------------------------------------------------------------
108 M     Trainable params
0         Non-trainable params
108 M     Total params
435.582   Total estimated model params size (MB)
```

```
Epoch 0, global step 23: 'val_acc' reached 0.59488 (best 0.59488), saving model to '/root/Chapter3/ag_food_reviews/epoch=0-step=23.ckpt' as top 3
Epoch 0, global step 46: 'val_acc' reached 0.67620 (best 0.67620), saving model to '/root/Chapter3/ag_food_reviews/epoch=0-step=46.ckpt' as top 3
Epoch 1, global step 70: 'val_acc' reached 0.78614 (best 0.78614), saving model to '/root/Chapter3/ag_food_reviews/epoch=1-step=70.ckpt' as top 3
Epoch 1, global step 93: 'val_acc' reached 0.81627 (best 0.81627), saving model to '/root/Chapter3/ag_food_reviews/epoch=1-step=93.ckpt' as top 3
Epoch 2, global step 117: 'val_acc' reached 0.73946 (best 0.81627), saving model to '/root/Chapter3/ag_food_reviews/epoch=2-step=117.ckpt' as top 3
Epoch 2, global step 140: 'val_acc' reached 0.80873 (best 0.81627), saving model to '/root/Chapter3/ag_food_reviews/epoch=2-step=140.ckpt' as top 3
Epoch 3, global step 164: 'val_acc' was not in top 3
Epoch 3, global step 187: 'val_acc' reached 0.81024 (best 0.81627), saving model to '/root/Chapter3/ag_food_reviews/epoch=3-step=187.ckpt' as top 3
Epoch 4, global step 211: 'val_acc' was not in top 3
Epoch 4, global step 234: 'val_acc' was not in top 3
Epoch 5, global step 258: 'val_acc' was not in top 3
Epoch 5, global step 281: 'val_acc' was not in top 3
Epoch 6, global step 305: 'val_acc' was not in top 3
Epoch 6, global step 328: 'val_acc' was not in top 3
```

```
Predicting DataLoader 0: 100% 21/21 [00:02<00:00, 8.89it/s]
Predicting DataLoader 0: 100% 21/21 [00:02<00:00, 8.93it/s]
Predicting DataLoader 0: 100% 21/21 [00:02<00:00, 8.89it/s]
Predicting DataLoader 0: 100% 21/21 [00:02<00:00, 8.85it/s]
```

```
{'accuracy': 0.8012048192771084, 'f1_micro': 0.8012048192771084}
```

	reviews	ratings
4985	They taste really good and I would recommend it to anyone	4
3994	Tastes really good.	5
2820	Did not like the ginger taste.	5
2280	My favorite flavor. Like sparkling lemonade....So refreshing!	5
3149	Refreshing	5
...	...	...
620	We love this and had been paying more per bottle at the grocery. I love being able to get this for cheaper and like the combination of flavors.	5
4745	Product came pre-opened with two inches of product missing.	4
4313	good taste n price	5

809 rows × 2 columns

```
Product Review: Sparkling ICE is the best! They are tasty, calorie free, bubbly, and inexpensive. Buying on amazon is less expensive than at the grocery store, so I order these all the time.
Predicted Rating: 5
Product Review: I have been ordering this for months but this last batch was old or bad and faded I had to return
Predicted Rating: 2
```

Besides predicted classes, we can obtain predicted class-probabilities.

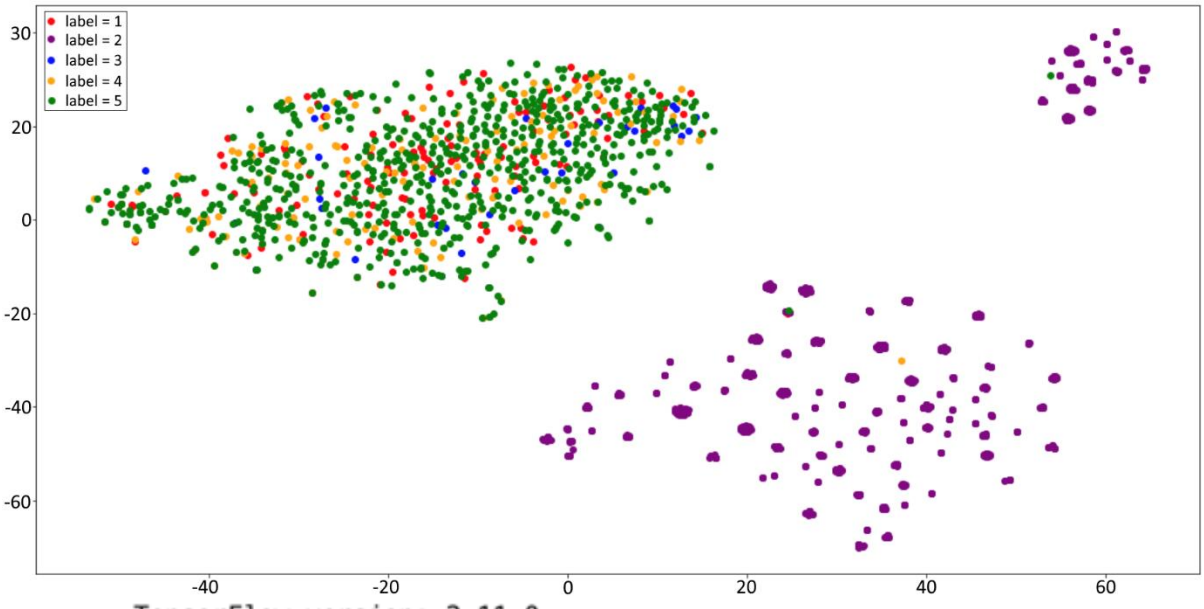
```
Product Review: Sparkling ICE is the best! They are tasty, calorie free, bubbly, and
inexpensive. Buying on amazon is less expensive than at the grocery store, so I order
these all the time.
Predicted Class-Probabilities:
1    0.158037
2    0.250578
3    0.042650
4    0.133276
5    0.415459
```

	1	2	3	4	5
0	0.158037	0.250578	0.042650	0.133276	0.415459
1	0.000047	0.999846	0.000051	0.000025	0.000030

	1	2	3	4	5
0	0.158037	0.250578	0.042650	0.133276	0.415459
1	0.000047	0.999846	0.000051	0.000025	0.000030

Predicting DataLoader 0: 100% 52/52 [00:05<00:00, 9.27it/s]

```
[[-0.6521491  -0.26700833  0.09421055 ...  0.29980472  0.44825974
   0.7461777 ]
 [ 0.00520702  0.68804175  0.02289832 ...  0.5043433   0.2496965
  -0.26859117]
 [ 0.14096348  0.6670949   0.0537451  ...  0.46771514  0.09381741
  -0.37640372]
 ...
 [ 0.2750097  -0.34764984  0.1748297  ... -0.64605737  0.5334411
   0.12582716]
 [ 0.19423327  0.51381886  0.03548399 ...  0.3546247   0.25145265
  -0.6742987 ]
 [-0.65595    -0.77710974 -0.42872536 ... -0.7155813   0.6250126
   0.1075721 ]]
```




```
TensorFlow version: 2.11.0
Python version: 3.9.10
```

```
array([4, 5, 1, 2, 3])
```

	reviews	categorical_ratings
0	I've been ordering this product monthly for over a year. These past few shipments have been off, and the packaging is different. It is obvious they are from two different manufacturers. The full bottle is the imposter and has no pink ribbon on the label, the label photo is different, and if you look at the bottle, you can see on the back that the legitimate product has a 3% Juice Content label on the back up top, and the imposter has a 1% Juice Content. There are also other differences on the label. See Photos for comparison. Two different products, and the IMPOSTER is absolutely undrinkable and tastes nothing like the original product.I contacted Amazon and they are pulling this flavor, Red Grapefruit, off the shelf while they investigate this issue. They also issued me a full credit. I hope they can get this figured out because this is my favorite flavor and I drink it everyday.	3
1	I used to drink these Sparkling Ice waters in all flavors every day. They all taste GREAT, my favorite was Black Raspberry. However, I just discovered that they contain sucralose which is just another harmful artificial sweetener. The only safe sweeteners are Stevia, Agave, and Raw Honey. I have stopped drinking/eating anything with harmful artificial sweeteners since they have been linked to many diseases, including dementia and the increase in Atzheimers disease patients.	4

	reviews	categorical_ratings
0	False	False
1	False	False
2	False	False
...	...	...
4997	False	False
4998	False	False
4999	False	False

5000 rows × 2 columns

	reviews	categorical_ratings
0	False	False
1	False	False
2	False	False
...	...	...
4997	False	False
4998	False	False
4999	False	False

5000 rows × 2 columns

```
Dataset shape: (5000, 2)
Dataset features: Index(['reviews', 'categorical_ratings'], dtype='object')
```

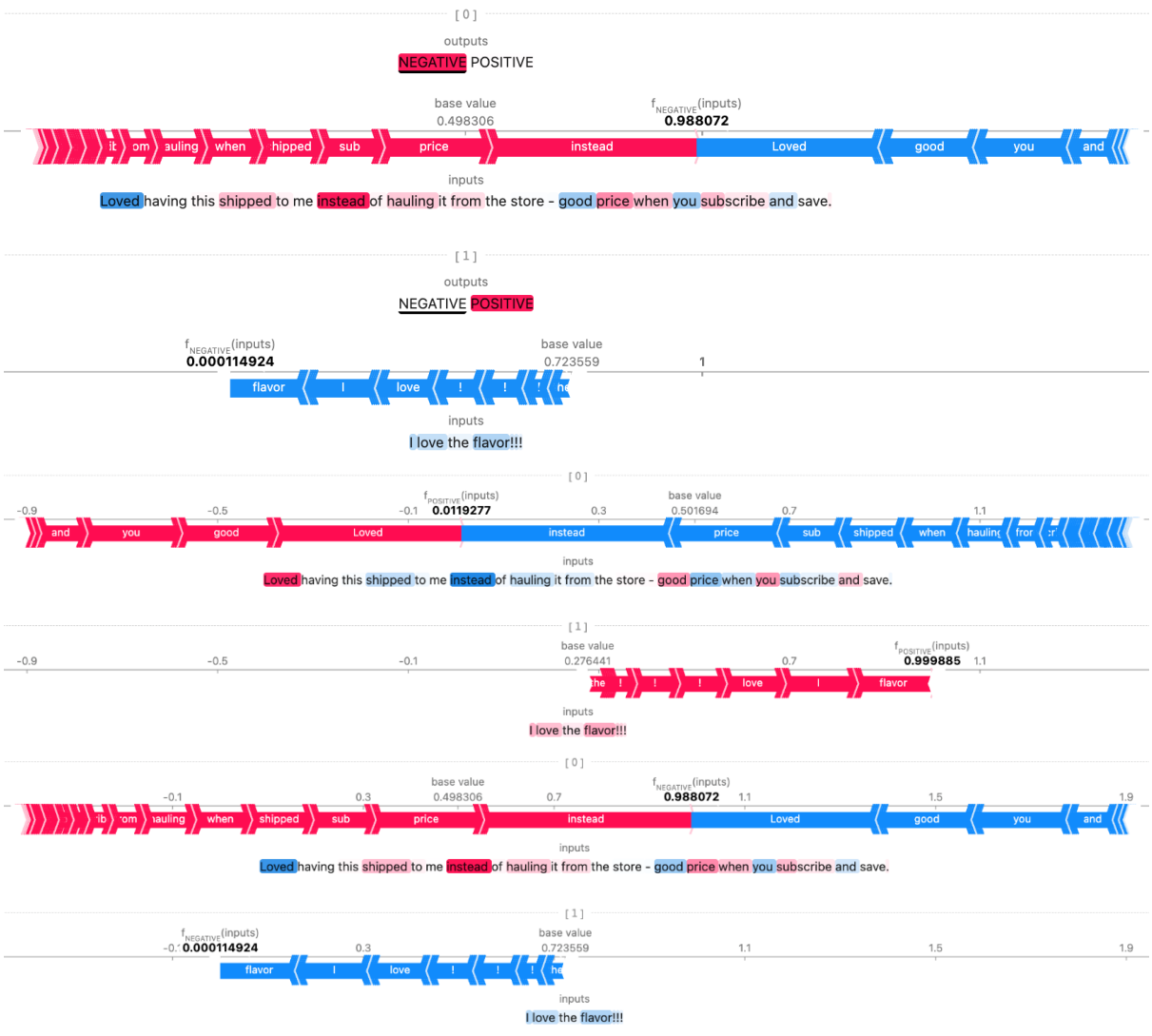
```
    Example Label: 4
    Example Text: Great but cheaper at the store
    Example Label: 4
    Example Text: Great but cheaper at the store
Cleanlab identified 1556 potential label errors.
 Indices of the top 10 most likely errors:
 [4270 4366 4443 4872 4112 4327 4804 4496 4009 3792]
```

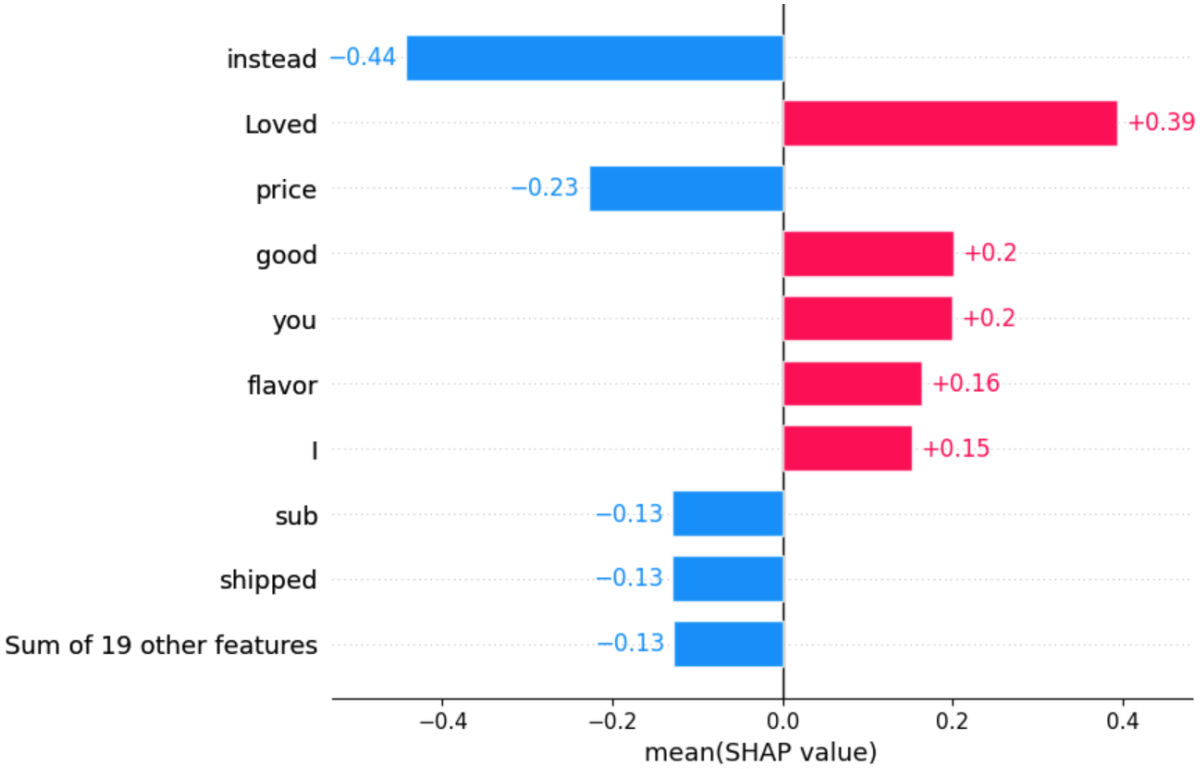
	Reviews	Labels
4270	Great price	2

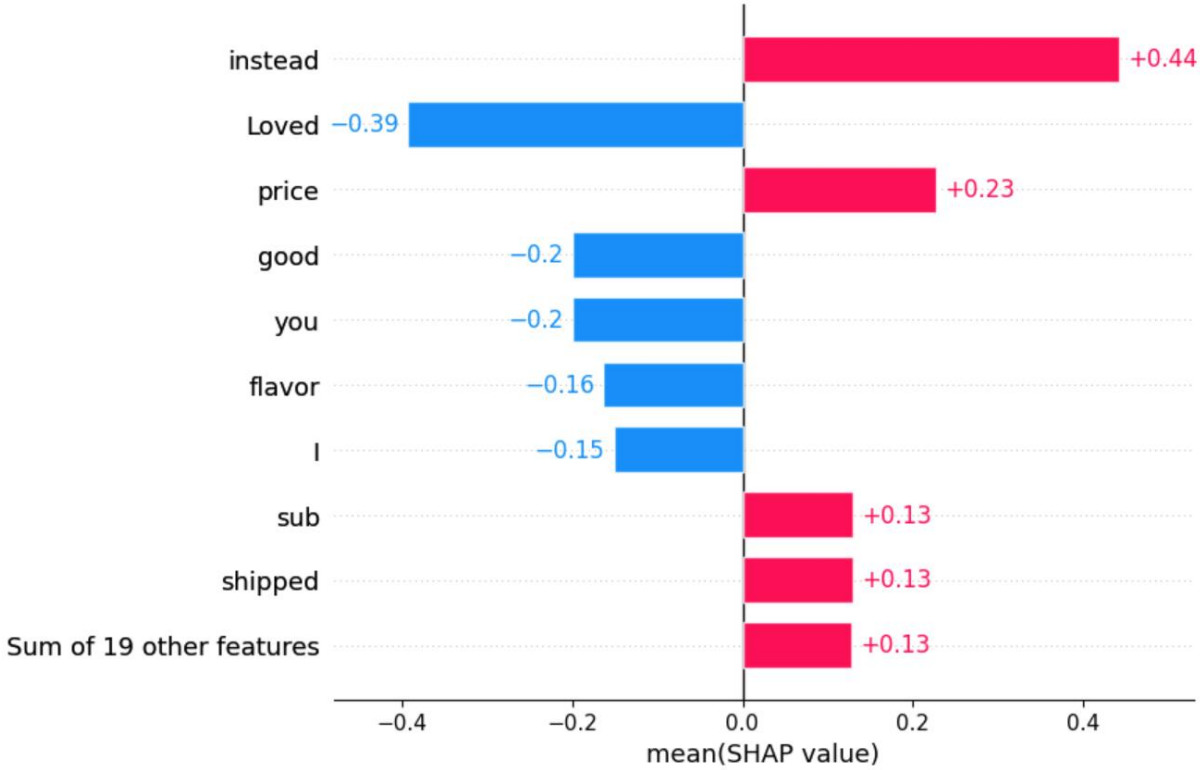
	Reviews	Labels
4443	Great alternative for soda! I love it	2

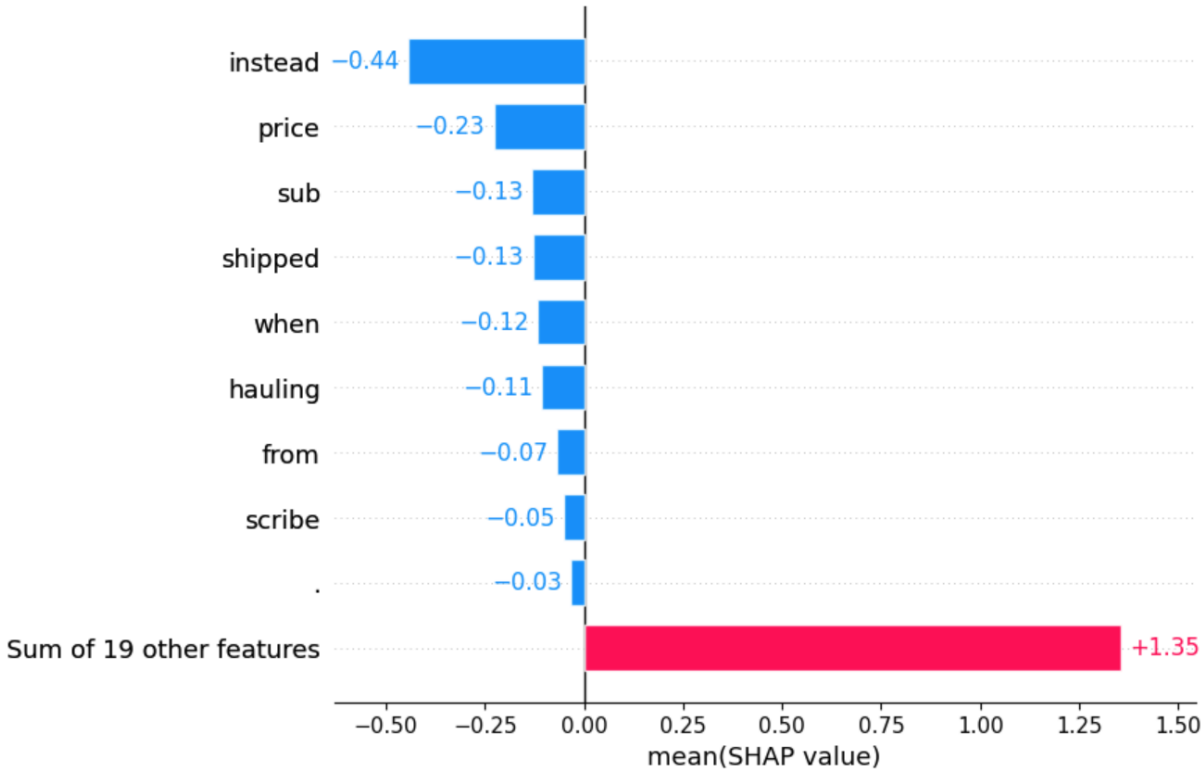
```
raw_train_texts shape: (4000,)
train_labels shape: (4000,)
raw_test_texts shape: (1000,)
test_labels shape: (1000,)
```

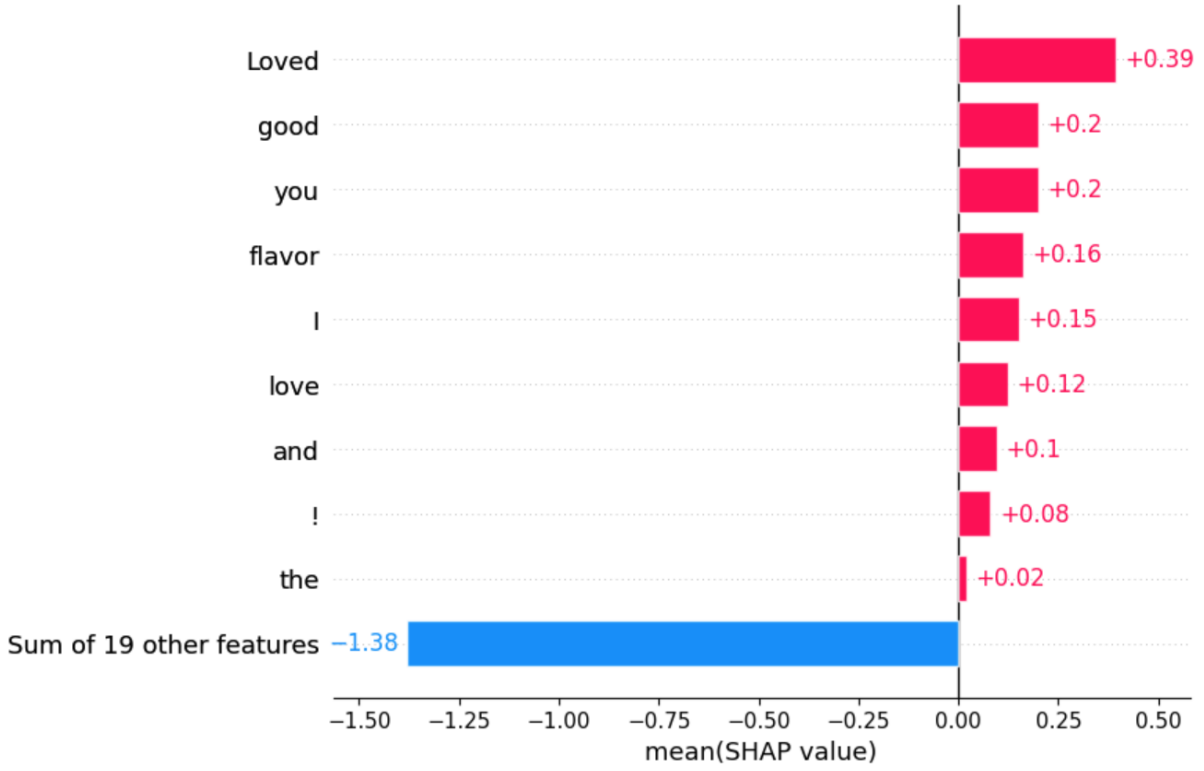
```
Epoch 1/10
125/125 [==============================] - 1s 3ms/step - loss: 1.3862 - categorical_accuracy: 0.0075
Epoch 2/10
125/125 [==============================] - 0s 3ms/step - loss: 1.0653 - categorical_accuracy: 0.0000e+00
Epoch 3/10
125/125 [==============================] - 0s 3ms/step - loss: 1.0118 - categorical_accuracy: 0.0000e+00
Epoch 4/10
125/125 [==============================] - 0s 3ms/step - loss: 0.9991 - categorical_accuracy: 0.0000e+00
Epoch 5/10
125/125 [==============================] - 0s 4ms/step - loss: 0.9967 - categorical_accuracy: 0.0000e+00
Epoch 6/10
125/125 [==============================] - 0s 4ms/step - loss: 0.9994 - categorical_accuracy: 0.0000e+00
Epoch 7/10
125/125 [==============================] - 0s 4ms/step - loss: 0.9957 - categorical_accuracy: 0.0000e+00
Epoch 8/10
125/125 [==============================] - 0s 3ms/step - loss: 0.9910 - categorical_accuracy: 0.0000e+00
Epoch 9/10
125/125 [==============================] - 0s 4ms/step - loss: 0.9958 - categorical_accuracy: 0.0000e+00
Epoch 10/10
125/125 [==============================] - 0s 4ms/step - loss: 0.9940 - categorical_accuracy: 0.0000e+00
32/32 [==============================] - 0s 1ms/step
Test accuracy of original neural net: 0.698
Epoch 10/10
87/87 [==============================] - 0s 4ms/step - loss: 0.5372 - categorical_accuracy: 0.0043
32/32 [==============================] - 0s 1ms/step
Test accuracy of Cleanlab's neural net: 0.696
```




instead
Loved
price
good
you
flavor
I
sub
shipped
Sum of 19 other features
−0.44
+0.39
−0.23
+0.2
+0.2
+0.16
+0.15
−0.13
−0.13
−0.13
−0.4
−0.2
0.0
0.2
0.4
mean(SHAP value)


instead
Loved
price
good
you
flavor
I
sub
shipped
Sum of 19 other features
+0.44
−0.39
+0.23
−0.2
−0.2
−0.16
−0.15
+0.13
+0.13
+0.13
−0.4
−0.2
0.0
0.2
0.4
mean(SHAP value)


instead −0.44
price −0.23
sub −0.13
shipped −0.13
when −0.12
hauling −0.11
from −0.07
scribe −0.05
. −0.03
Sum of 19 other features +1.35
−0.50 −0.25 0.00 0.25 0.50 0.75 1.00 1.25 1.50
mean(SHAP value)

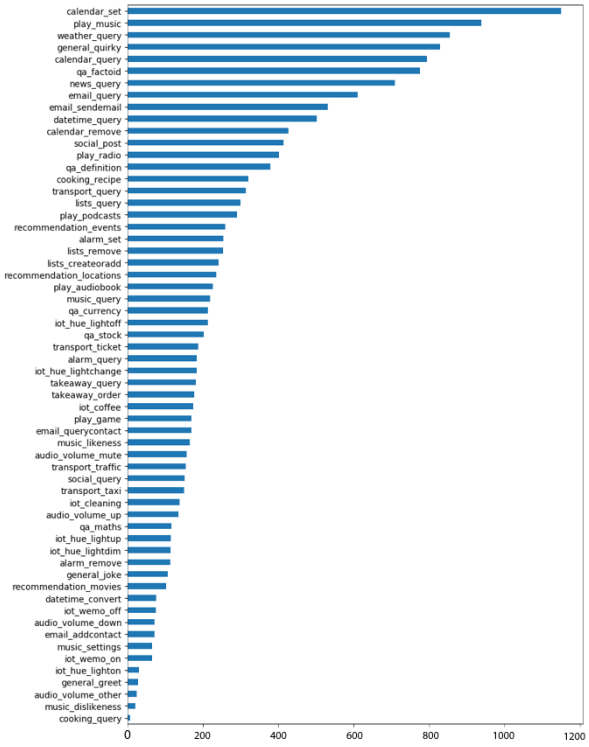

Loved +0.39
good +0.2
you +0.2
flavor +0.16
I +0.15
love +0.12
and +0.1
! +0.08
the +0.02
Sum of 19 other features −1.38
−1.50 −1.25 −1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50
mean(SHAP value)

```
[{'id': '0',
  'locale': 'en-US',
  'partition': 'test',
  'scenario': 'alarm',
  'intent': 'alarm_set',
  'utt': 'wake me up at five am this week',
  'annot_utt': 'wake me up at [time : five am] [date : this week]',
  'worker_id': '1'},
 {'id': '1',
  'locale': 'en-US',
  'partition': 'train',
  'scenario': 'alarm',
  'intent': 'alarm_set',
  'utt': 'wake me up at nine am on friday',
  'annot_utt': 'wake me up at [time : nine am] on [date : friday]',
  'worker_id': '1'},
 {'id': '2',
  'locale': 'en-US',
  'partition': 'train',
  'scenario': 'alarm',
  'intent': 'alarm_set',
  'utt': 'set an alarm for two hours from now',
  'annot_utt': 'set an alarm for [time : two hours from now]',
  'worker_id': '1'}]
```

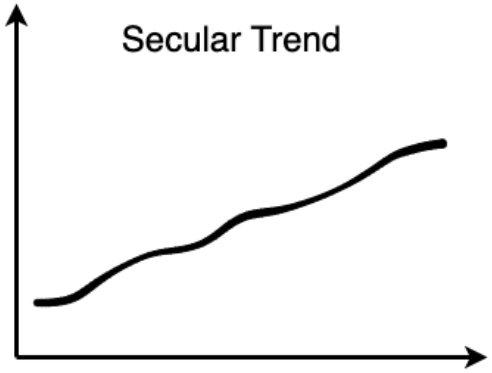
	id	locale	partition	scenario	intent	utt	annot_utt	worker_id
0	0	en-US	test	alarm	alarm_set	wake me up at five am this week	wake me up at [time : five am] [date : this week]	1
1	1	en-US	train	alarm	alarm_set	wake me up at nine am on friday	wake me up at [time : nine am] on [date : friday]	1
2	2	en-US	train	alarm	alarm_set	set an alarm for two hours from now	set an alarm for [time : two hours from now]	1
3	3	en-US	test	audio	audio_volume_mute	quiet	quiet	1
4	4	en-US	train	audio	audio_volume_mute	olly quiet	olly quiet	1

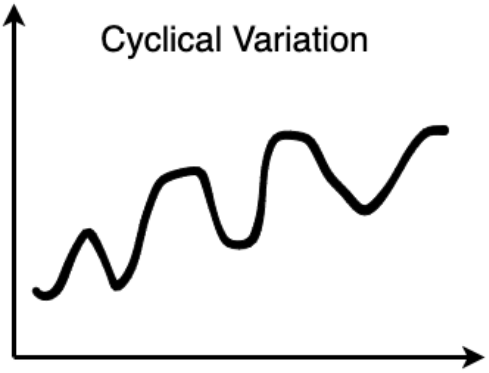
	id	locale	partition	scenario	intent	utt	annot_utt	worker_id
count	16521	16521	16521	16521	16521	16521	16521	16521
unique	16521	1	3	18	60	16432	16434	691
top	0	en-US	train	calendar	calendar_set	do i have any new email	do i have any new email	0
freq	1	16521	11514	2370	1150	3	3	228

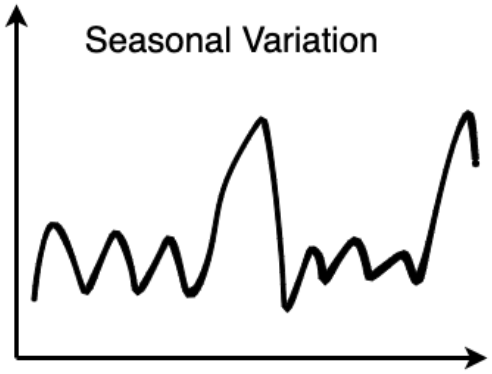
```
array(['alarm_set', 'audio_volume_mute', 'iot_hue_lightchange',
       'iot_hue_lightoff', 'iot_hue_lighton', 'iot_hue_lightdim',
       'iot_cleaning', 'calendar_query', 'play_music', 'general_quirky',
       'general_greet', 'datetime_query', 'datetime_convert',
       'takeaway_query', 'alarm_remove', 'alarm_query', 'news_query',
       'music_likeness', 'music_query', 'iot_hue_lightup',
       'takeaway_order', 'weather_query', 'music_settings',
       'audio_volume_down', 'general_joke', 'music_dislikeness',
       'audio_volume_other', 'iot_coffee', 'audio_volume_up',
       'iot_wemo_on', 'iot_wemo_off', 'qa_stock', 'play_radio',
       'social_post', 'recommendation_locations', 'cooking_recipe',
       'qa_factoid', 'recommendation_events', 'calendar_set',
       'play_audiobook', 'play_podcasts', 'social_query',
       'transport_query', 'email_sendemail', 'transport_ticket',
       'recommendation_movies', 'lists_query', 'play_game', 'email_query',
       'transport_traffic', 'cooking_query', 'qa_definition',
       'calendar_remove', 'lists_remove', 'email_querycontact',
       'lists_createoradd', 'email_addcontact', 'transport_taxi',
       'qa_maths', 'qa_currency'], dtype=object)
```

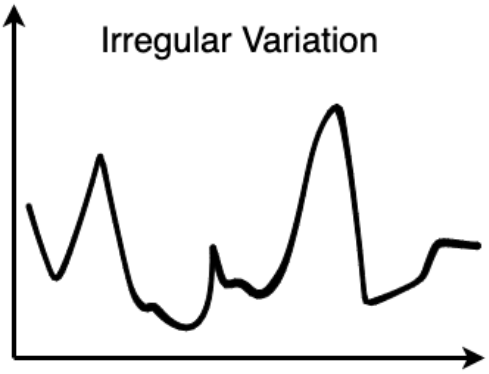


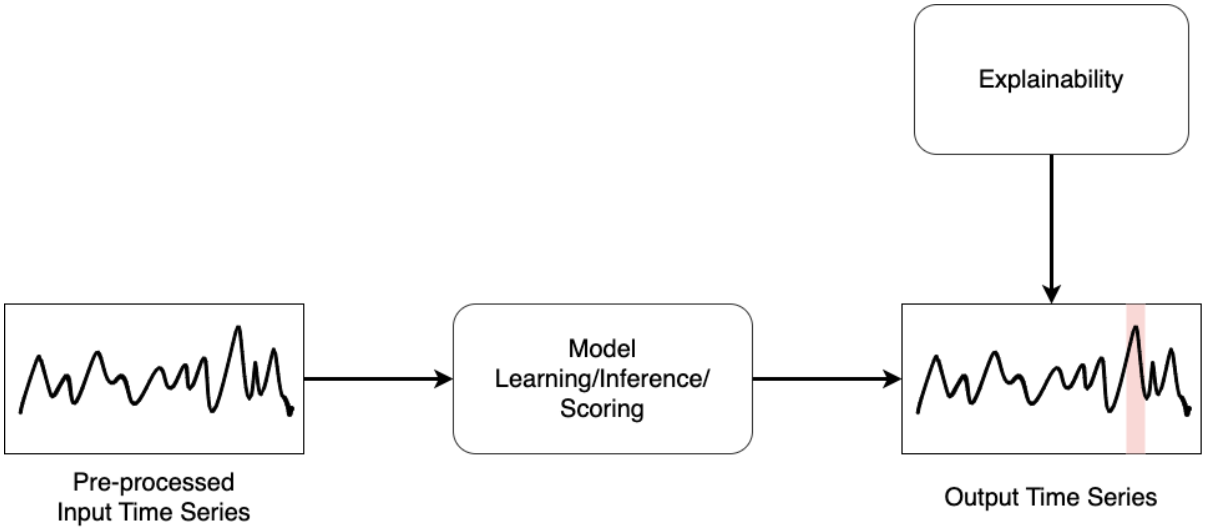
Chapter 4: Time Series Anomaly Explainability









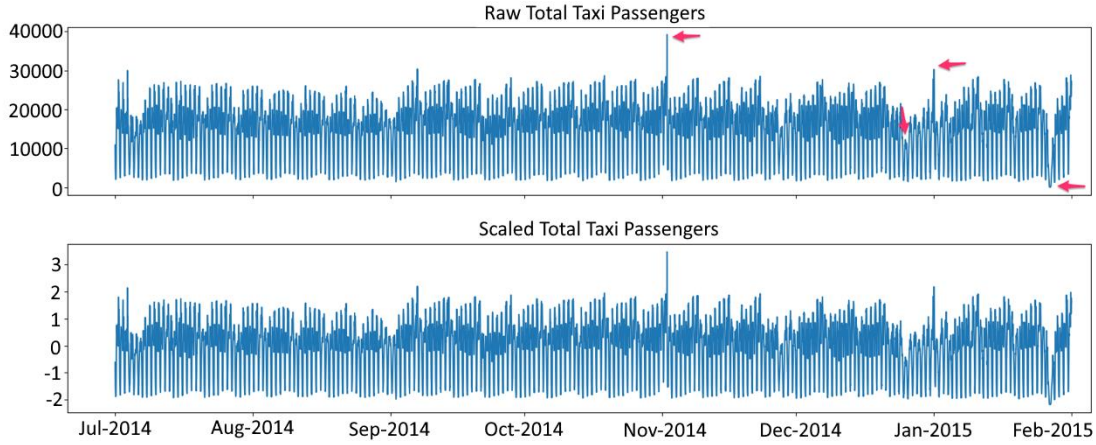


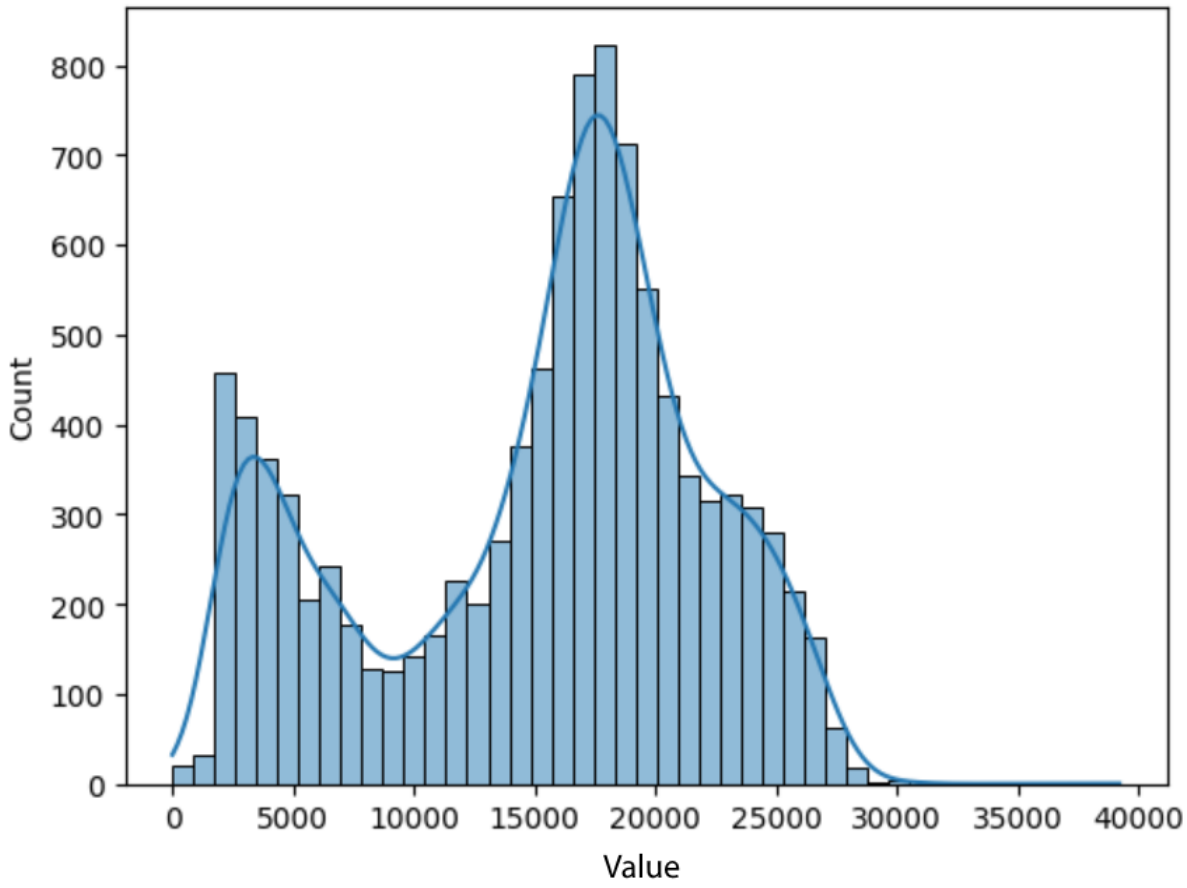

```
TensorFlow version: 2.11.0
Python version: 3.9.10
```

	date	value
0	2014-07-01 00:00:00	10844
1	2014-07-01 00:30:00	8127
2	2014-07-01 01:00:00	6210
3	2014-07-01 01:30:00	4656
4	2014-07-01 02:00:00	3820

	value
count	10320.000000
mean	15137.569380
std	6939.495808
min	8.000000
25%	10262.000000
50%	16778.000000
75%	19838.750000
max	39197.000000

	date	value	scaled_value
0	2014-07-01 00:00:00	10844	-0.618745
1	2014-07-01 00:30:00	8127	-1.010291
2	2014-07-01 01:00:00	6210	-1.286549
3	2014-07-01 01:30:00	4656	-1.510496
4	2014-07-01 02:00:00	3820	-1.630971

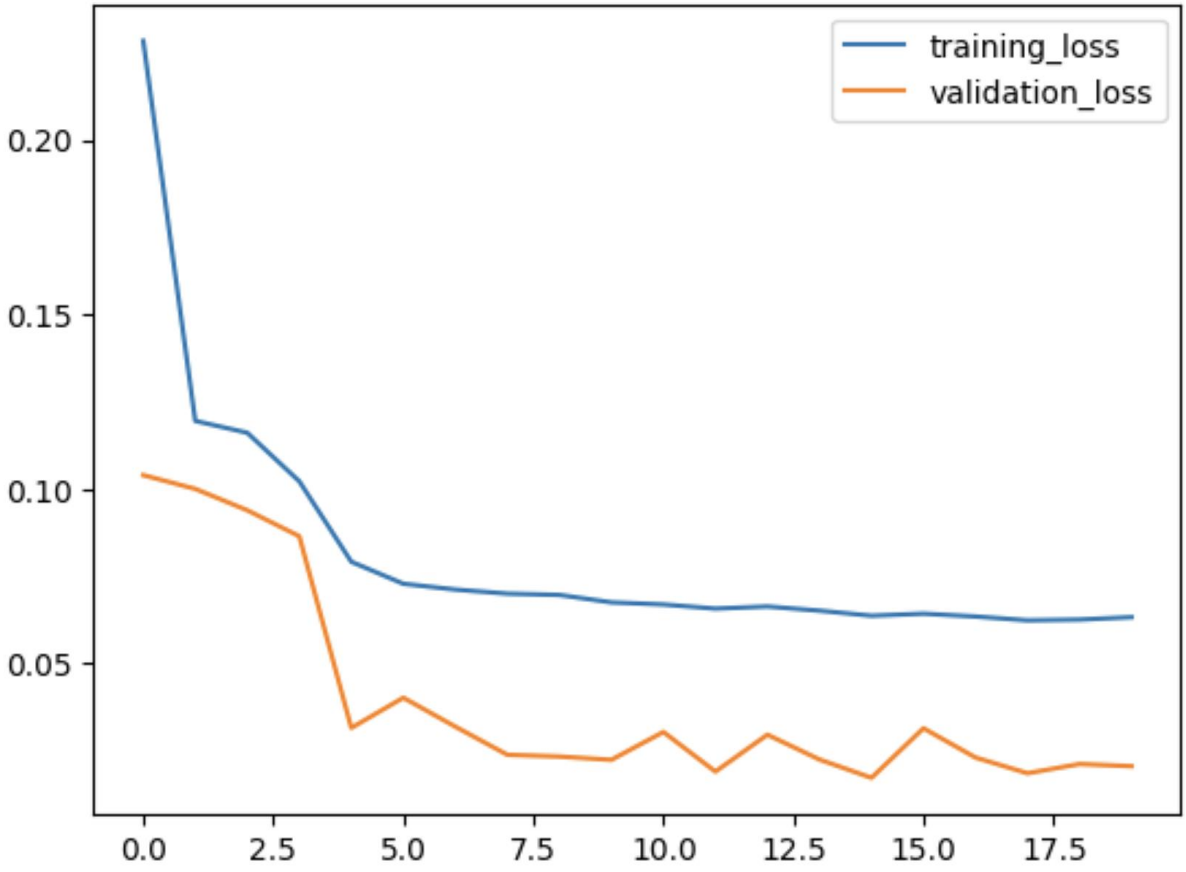




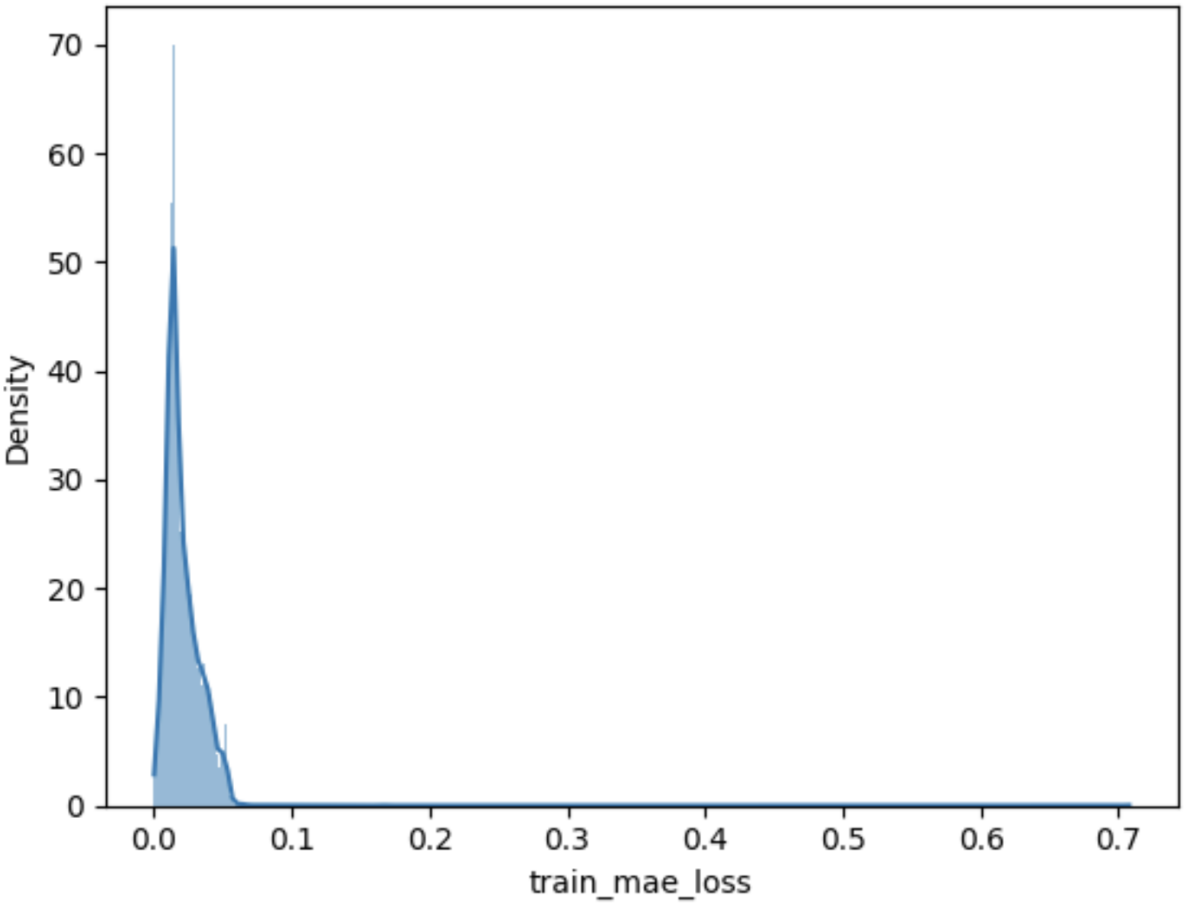

```
Train shape: (8256, 3)
Test shape: (2064, 3)
```

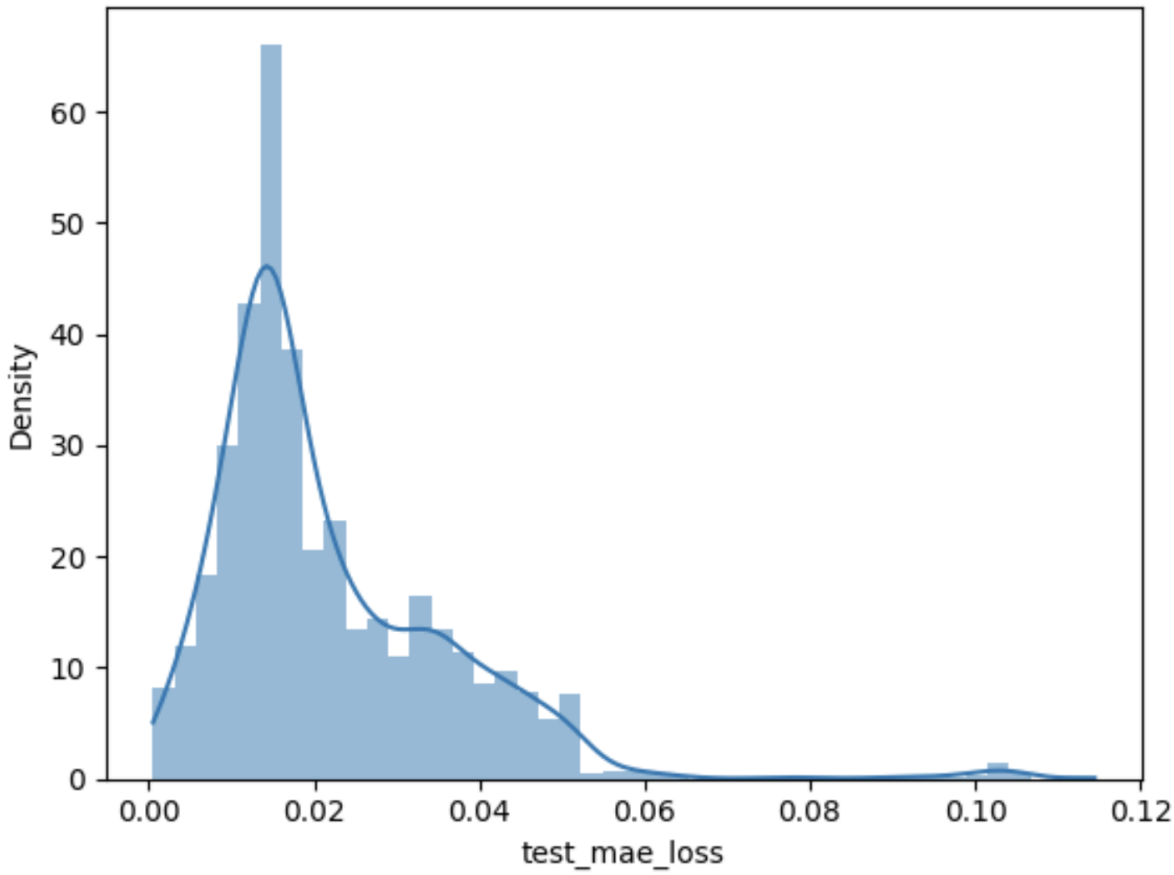
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 encoder_lstm (LSTM)         (None, 64)                16896

 encoder_dropout (Dropout)   (None, 64)                0

 decoder_repeater (RepeatVec  (None, 2, 64)            0
 tor)

 decoder_lstm (LSTM)         (None, 2, 64)             33024

 decoder_dropout (Dropout)   (None, 2, 64)             0

 time_distributed (TimeDistr  (None, 2, 1)             65
 ibuted)

=================================================================
Total params: 49,985
Trainable params: 49,985
Non-trainable params: 0
_________________________________________________________________
```

```
CPU times: user 3 µs, sys: 0 ns, total: 3 µs
Wall time: 8.34 µs
Epoch 1/20
233/233 [==============================] - 10s 8ms/step - loss: 0.2356 - val_loss: 0.1076
Epoch 2/20
233/233 [==============================] - 1s 5ms/step - loss: 0.1199 - val_loss: 0.1023
Epoch 3/20
233/233 [==============================] - 1s 5ms/step - loss: 0.1162 - val_loss: 0.0996
Epoch 4/20
233/233 [==============================] - 1s 5ms/step - loss: 0.1090 - val_loss: 0.0793
Epoch 5/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0876 - val_loss: 0.0458
Epoch 6/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0713 - val_loss: 0.0387
Epoch 7/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0693 - val_loss: 0.0182
Epoch 8/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0663 - val_loss: 0.0265
Epoch 9/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0658 - val_loss: 0.0193
Epoch 10/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0648 - val_loss: 0.0219
Epoch 11/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0635 - val_loss: 0.0166
Epoch 12/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0647 - val_loss: 0.0185
Epoch 13/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0623 - val_loss: 0.0189
Epoch 14/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0633 - val_loss: 0.0177
Epoch 15/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0614 - val_loss: 0.0226
Epoch 16/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0626 - val_loss: 0.0257
Epoch 17/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0614 - val_loss: 0.0294
Epoch 18/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0622 - val_loss: 0.0156
Epoch 19/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0611 - val_loss: 0.0256
Epoch 20/20
233/233 [==============================] - 1s 5ms/step - loss: 0.0606 - val_loss: 0.0218
```



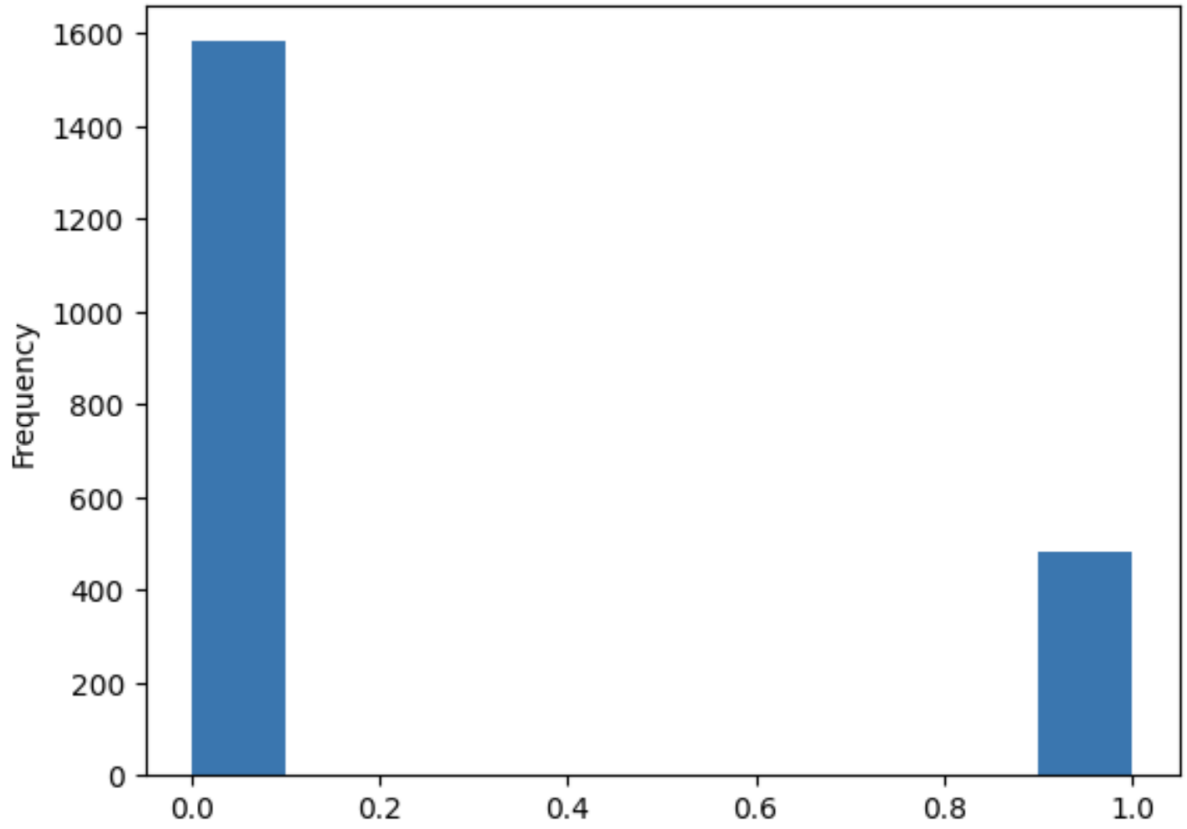

```
Reconstruction error threshold: 0.0335676140354171116
```

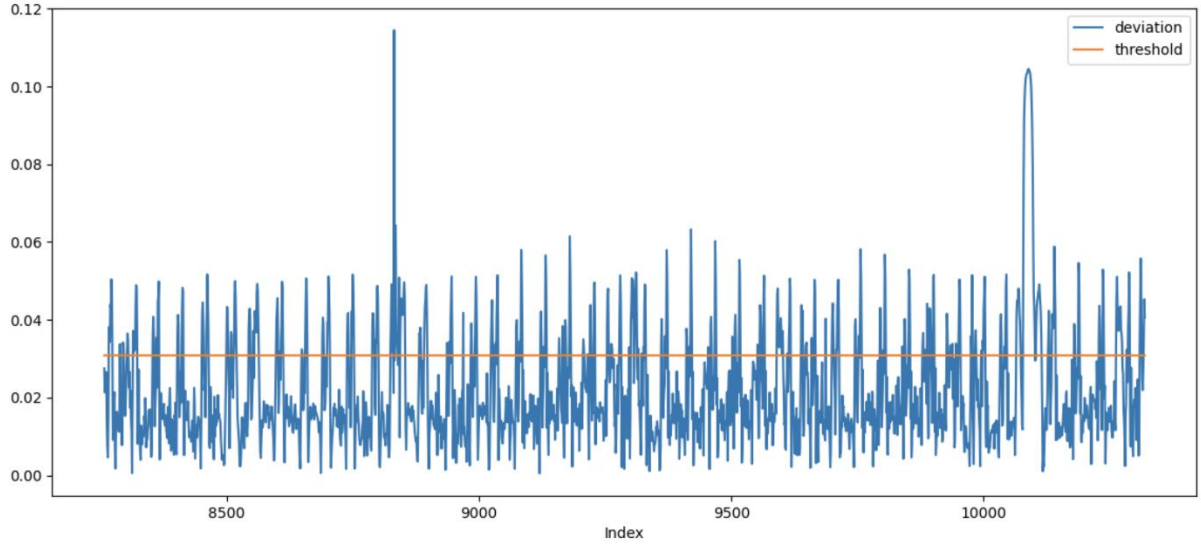

Density
train_mae_loss
0
10
20
30
40
50
60
70
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

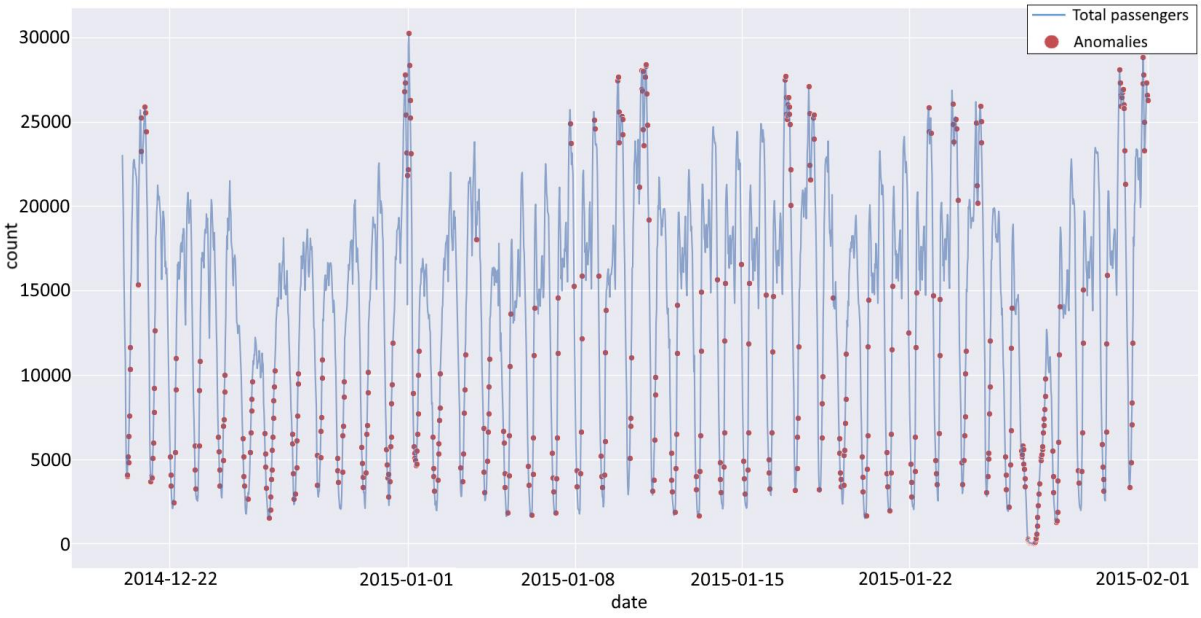

Density
test_mae_loss
0
10
20
30
40
50
60
0.00
0.02
0.04
0.06
0.08
0.10
0.12

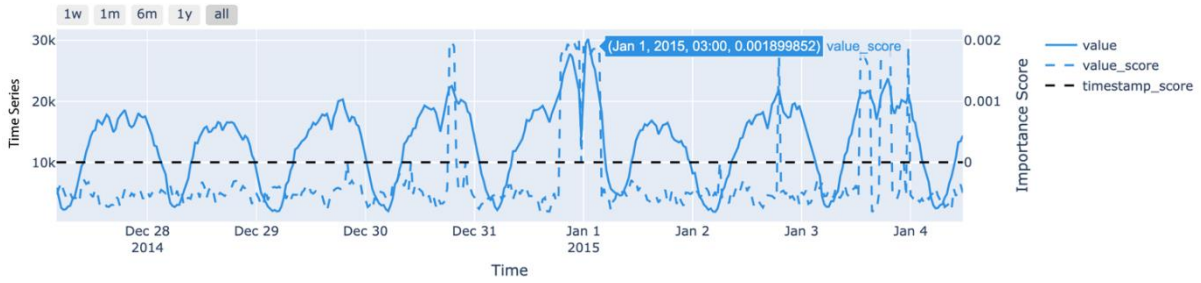
```
Total normal: 1639
Total anomalies: 423
```

	value	scaled_value	date	deviation	threshold	anomaly
8258	22993	1.132044	2014-12-20 01:00:00	0.034212	0.033568	1
8294	25685	1.519987	2014-12-20 19:00:00	0.037085	0.033568	1
8295	25252	1.457587	2014-12-20 19:30:00	0.044607	0.033568	1
8296	23238	1.167350	2014-12-20 20:00:00	0.039972	0.033568	1
8301	24614	1.365645	2014-12-20 22:30:00	0.034693	0.033568	1

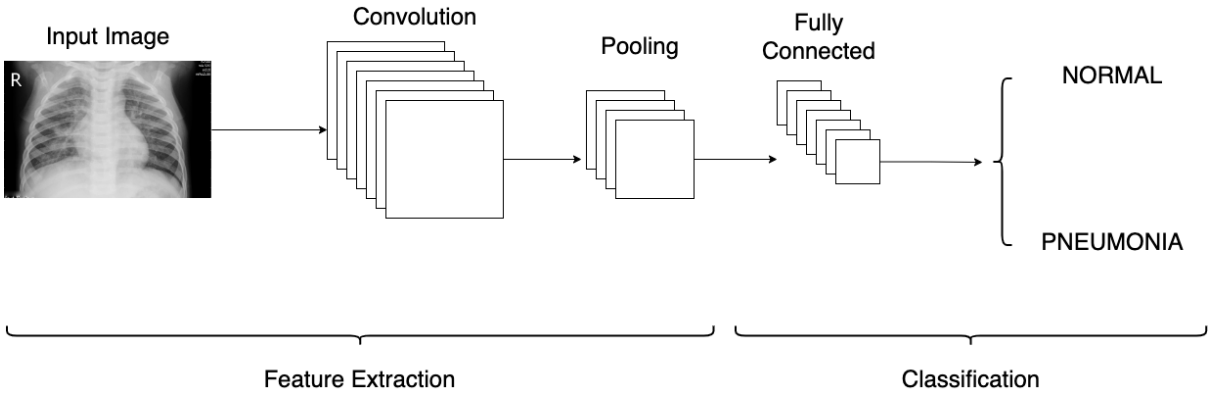



deviation
threshold
0.12
0.10
0.08
0.06
0.04
0.02
0.00
8500
9000
9500
10000
Index


Total passengers
Anomalies
count
30000
25000
20000
15000
10000
5000
0
2014-12-22
2015-01-01
2015-01-08
2015-01-15
2015-01-22
2015-02-01
date

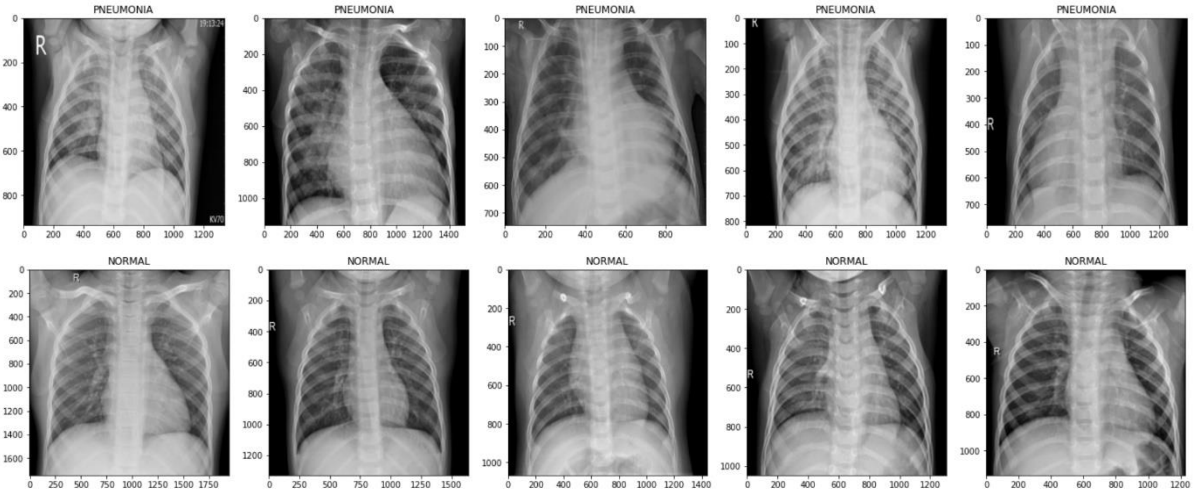

1w
1m
6m
1y
all
(Jan 1, 2015, 03:00, 0.001899852) value_score
Time Series
30k
20k
10k
0.002
0.001
0
Importance Score
value
value_score
timestamp_score
Dec 28
2014
Dec 29
Dec 30
Dec 31
Jan 1
2015
Jan 2
Jan 3
Jan 4
Time

Chapter 5: Computer Vision Anomaly Explainability




```
TensorFlow version: 2.11.0
Python version: 3.9.10
```

```
train_path: images/chest_xray/train/
test_path: images/chest_xray/test/
val_path: images/chest_xray/val/
```

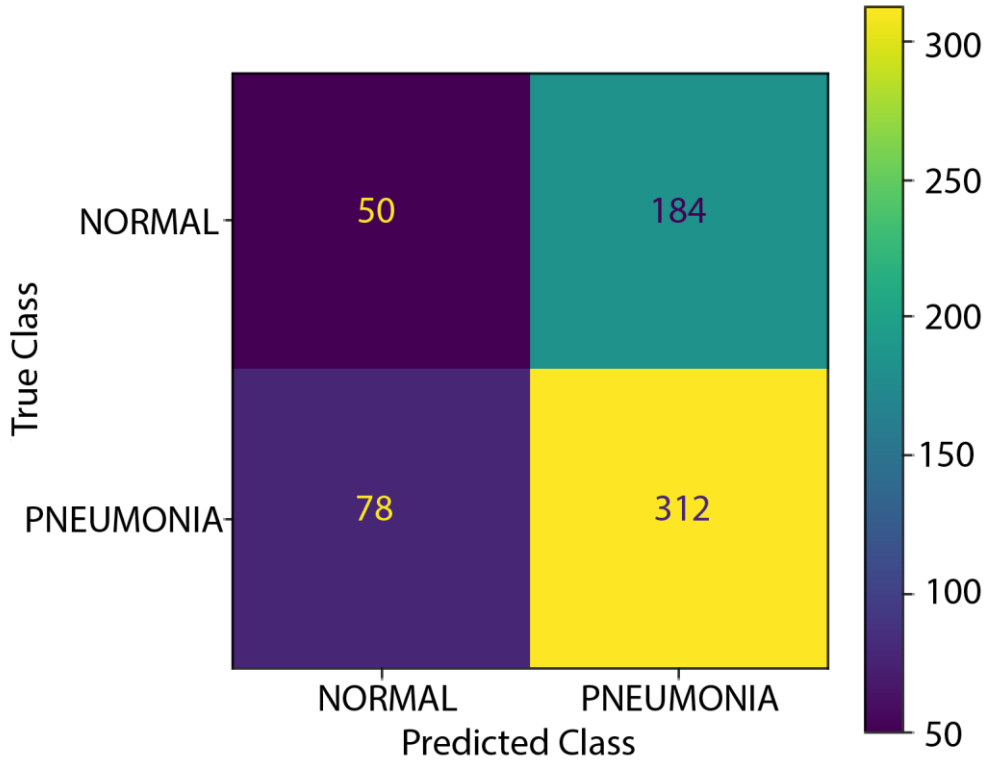


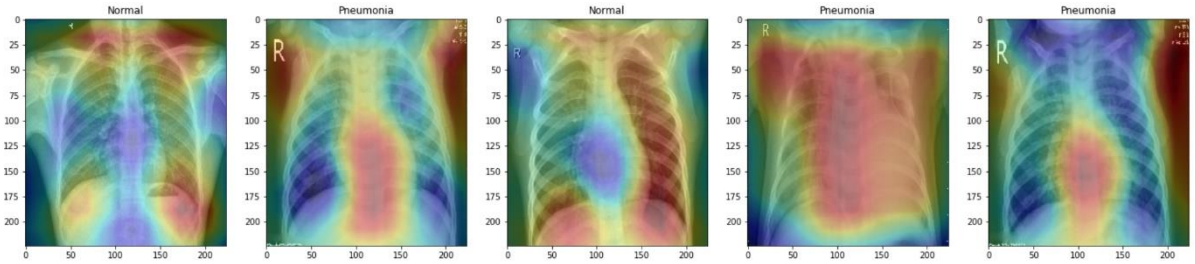
```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 global_average_pooling2d (G  (None, 512)              0
 lobalAveragePooling2D)

 dense (Dense)               (None, 2)                 1026

=================================================================
Total params: 14,715,714
Trainable params: 12,980,226
Non-trainable params: 1,735,488
_________________________________________________________________
```

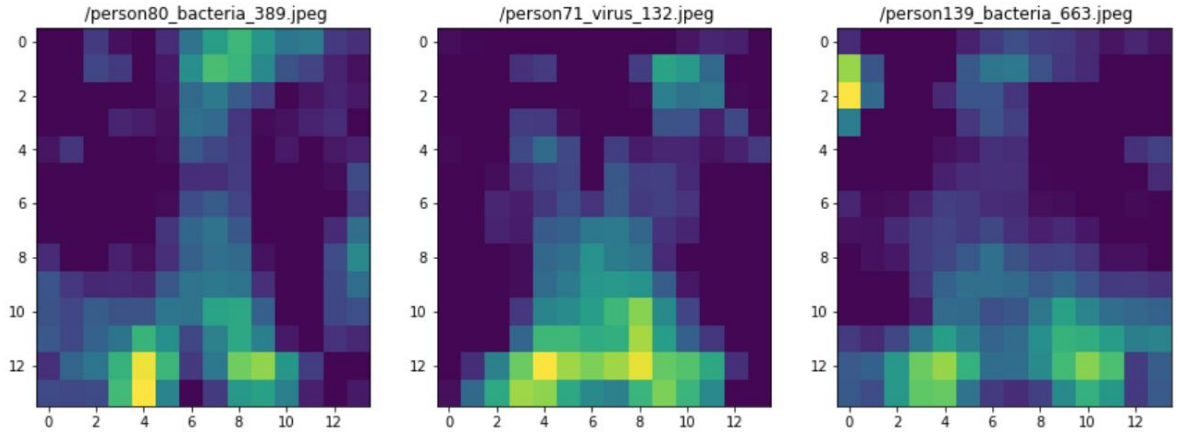
```
Epoch 1/8
20/20 [==============================] - 244s 12s/step - loss: 0.5509 - accuracy: 0.7429
Epoch 2/8
20/20 [==============================] - 243s 12s/step - loss: 0.5010 - accuracy: 0.7429
Epoch 3/8
20/20 [==============================] - 242s 12s/step - loss: 0.4415 - accuracy: 0.7500
Epoch 4/8
20/20 [==============================] - 244s 12s/step - loss: 0.3763 - accuracy: 0.8202
Epoch 5/8
20/20 [==============================] - 243s 12s/step - loss: 0.3097 - accuracy: 0.8921
Epoch 6/8
20/20 [==============================] - 244s 12s/step - loss: 0.2551 - accuracy: 0.9112
Epoch 7/8
20/20 [==============================] - 243s 12s/step - loss: 0.2152 - accuracy: 0.9220
Epoch 8/8
20/20 [==============================] - 244s 12s/step - loss: 0.1901 - accuracy: 0.9293
```

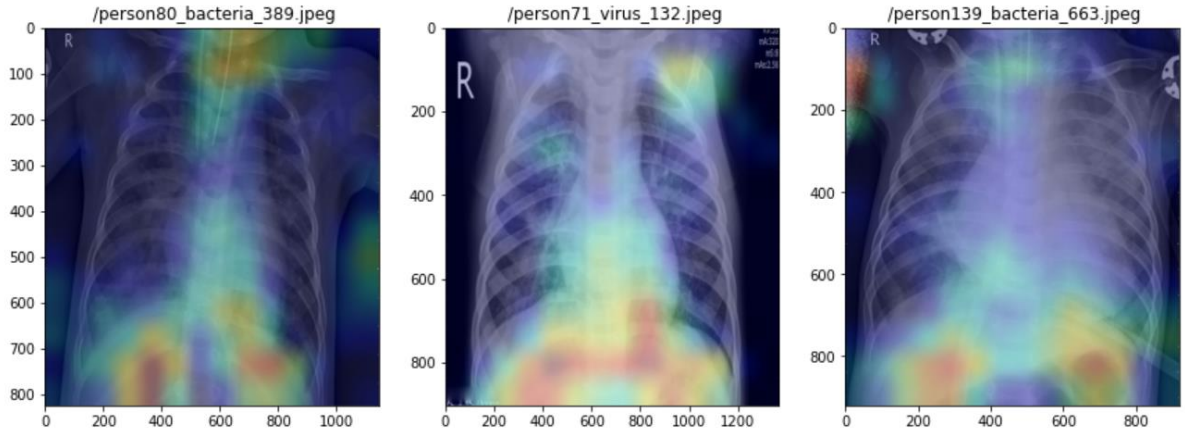
```
3/3 [==============================] - 16s 4s/step
```

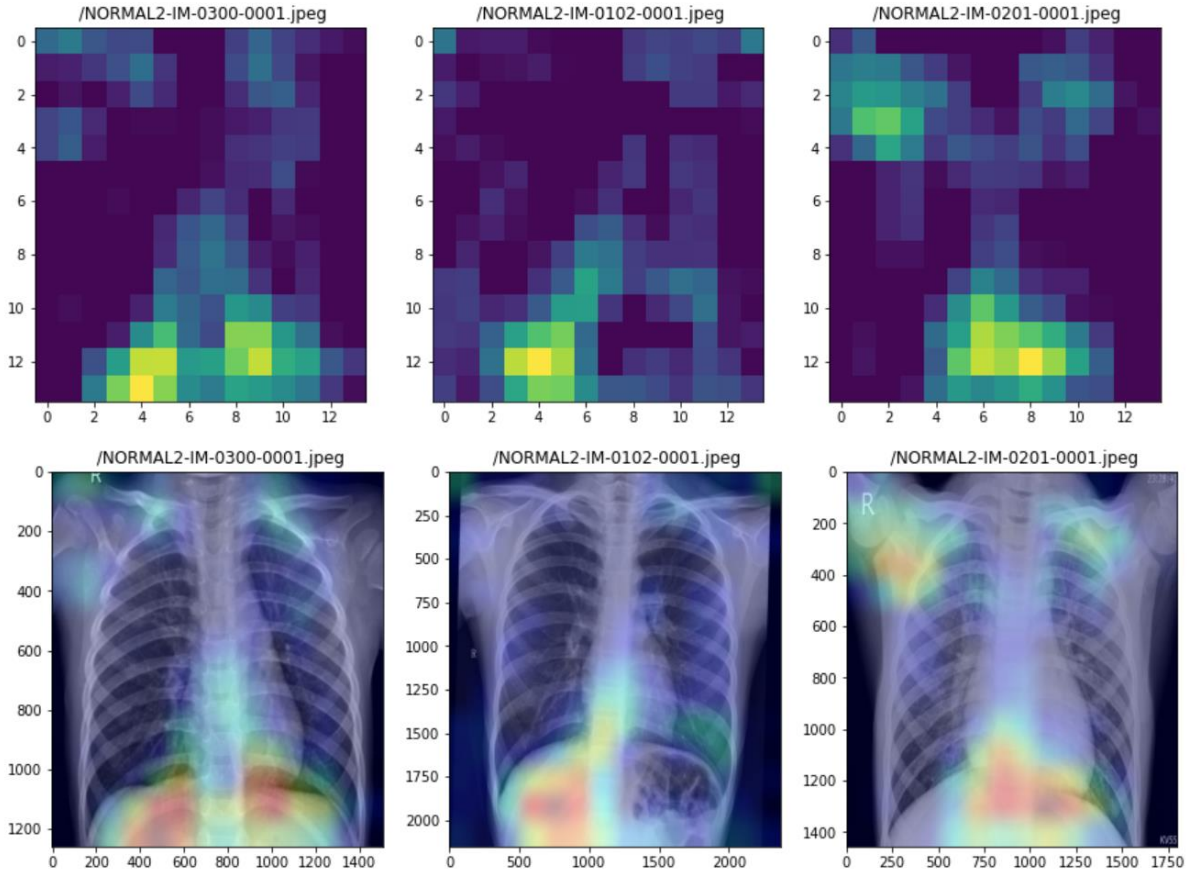
	precision	recall	f1-score	support
0	0.93	0.51	0.66	234
1	0.77	0.98	0.86	390
accuracy			0.80	624
macro avg	0.85	0.74	0.76	624
weighted avg	0.83	0.80	0.78	624




Normal
Pneumonia
Normal
Pneumonia
Pneumonia


/person80_bacteria_389.jpeg
/person71_virus_132.jpeg
/person139_bacteria_663.jpeg


/person80_bacteria_389.jpeg
/person71_virus_132.jpeg
/person139_bacteria_663.jpeg


/NORMAL2-IM-0300-0001.jpeg
/NORMAL2-IM-0102-0001.jpeg
/NORMAL2-IM-0201-0001.jpeg
/NORMAL2-IM-0300-0001.jpeg
/NORMAL2-IM-0102-0001.jpeg
/NORMAL2-IM-0201-0001.jpeg

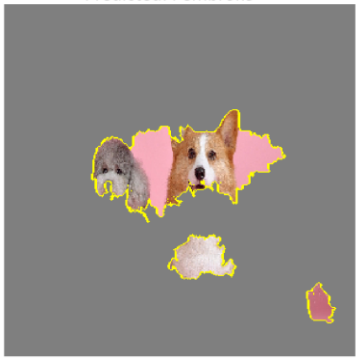
Chapter 6: Differentiating Intrinsic versus Post Hoc Explainability

```
TensorFlow version: 2.11.0
Python version: 3.8.2
```

Original Image

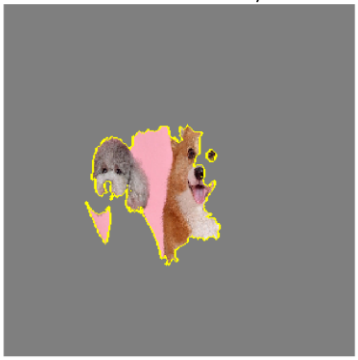
```
Top 5 classes:
('n02113023', 'Pembroke', 0.28453475)
('n02342885', 'hamster', 0.09641102)
('n04399382', 'teddy', 0.020572951)
('n02113186', 'Cardigan', 0.011049673)
('n02094258', 'Norwich_terrier', 0.010510642)
```

"Predicted: Pembroke"



"Predicted: Hamster"

"Predicted: Teddy"

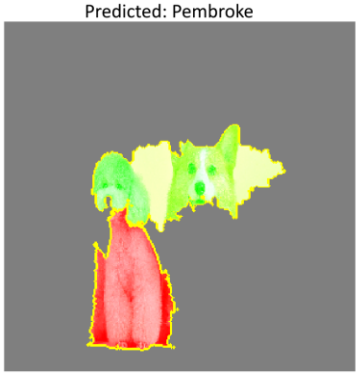


Predicted: Pembroke

Predicted: Hamster

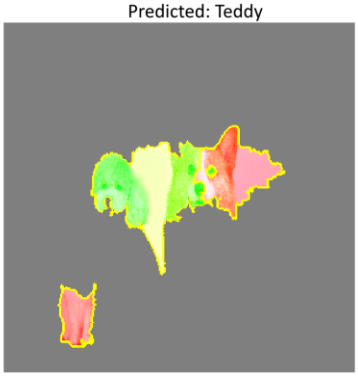
Predicted: Teddy

Predicted: Pembroke



Predicted: Hamster

Predicted: Teddy

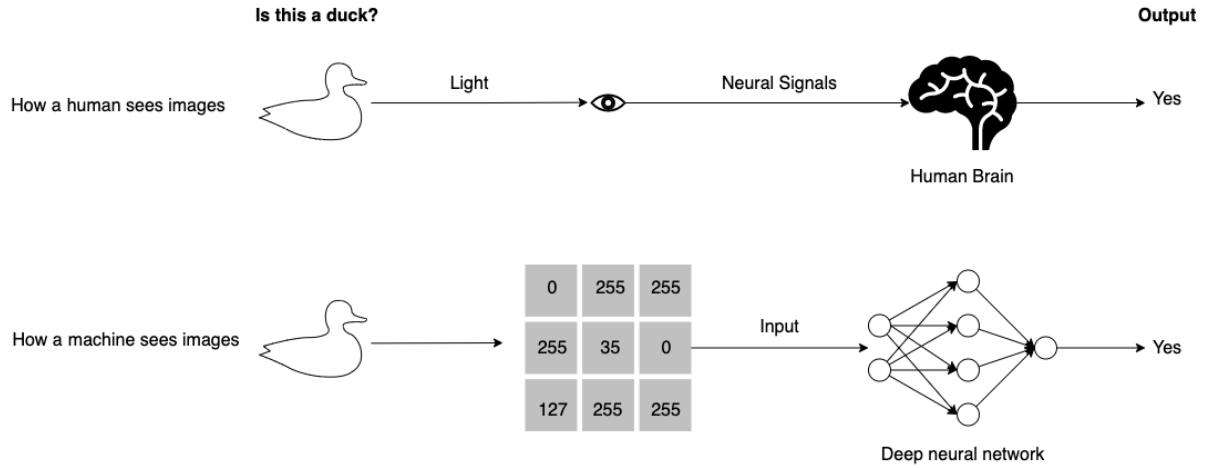


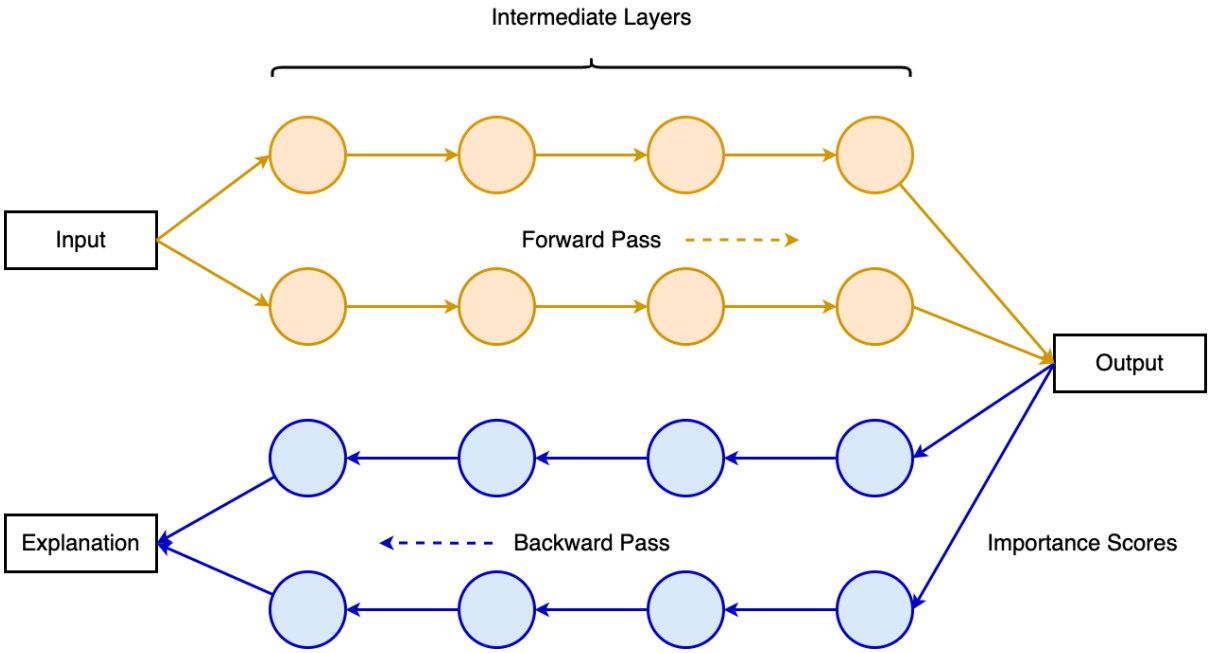
Predicted: Pembroke

Predicted: Hamster

Predicted: Teddy

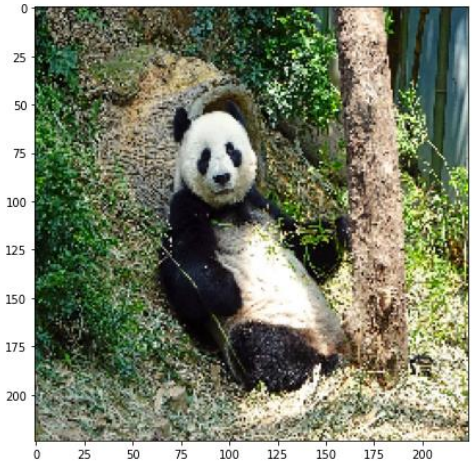
Chapter 7: Backpropagation versus Perturbation Explainability

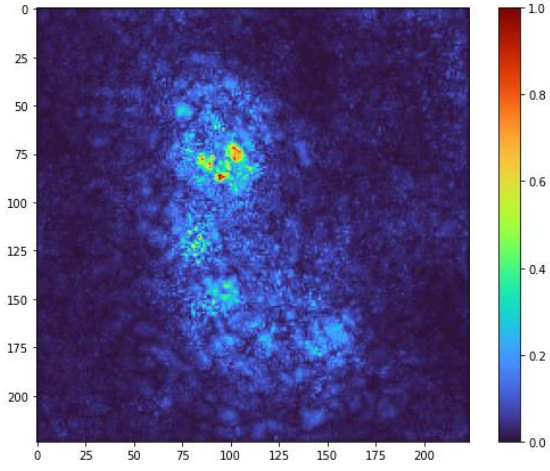




```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 input_1 (InputLayer)         [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0

 flatten (Flatten)            (None, 25088)             0

 fc1 (Dense)                  (None, 4096)              102764544

 fc2 (Dense)                  (None, 4096)              16781312

 predictions (Dense)          (None, 1000)              4097000

=================================================================
Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0
_________________________________________________________________
```

```
1/1 [==============================] - 1s 540ms/step
Top 5 classes:
[[('n02510455', 'giant_panda', 0.9559301),
  ('n02509815', 'lesser_panda', 0.039848775),
  ('n02483362', 'gibbon', 0.0019217625),
  ('n02443114', 'polecat', 0.000856469),
  ('n02488702', 'colobus', 0.0003992283)]]
```





	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...

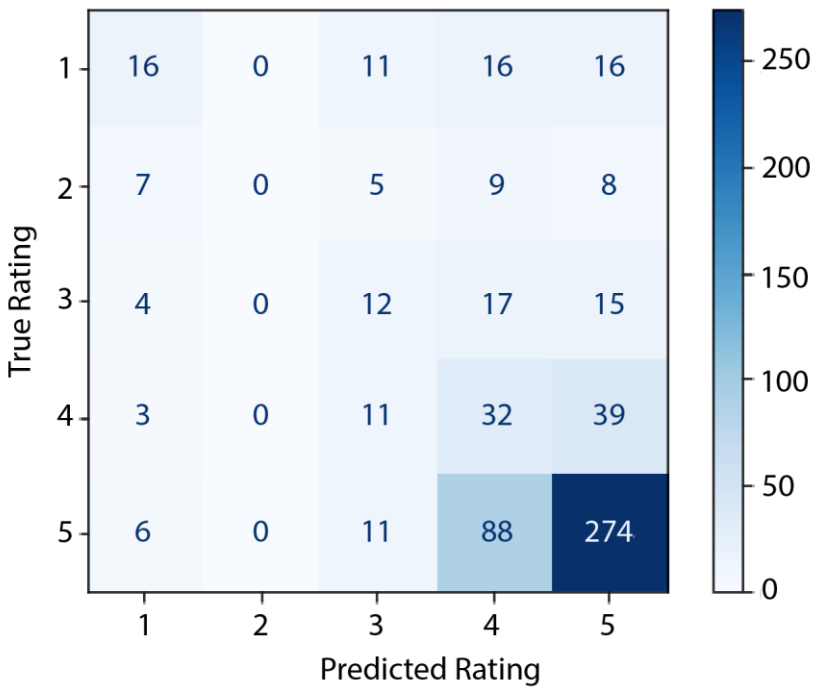
	Text	Score
0	I have bought several of the Vitality canned d...	5
1	Product arrived labeled as Jumbo Salted Peanut...	1
2	This is a confection that has been around a fe...	4
3	If you are looking for the secret ingredient i...	2
4	Great taffy at a great price. There was a wid...	5

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   Text    568454 non-null  object
 1   Score   568454 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 8.7+ MB
```

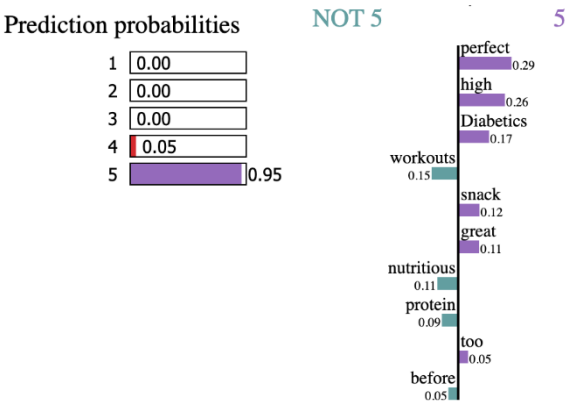
Test Accuracy: 55.67%

Classification Report:

	precision	recall	f1-score	support
1	0.44	0.27	0.34	59
2	0.00	0.00	0.00	29
3	0.24	0.25	0.24	48
4	0.20	0.38	0.26	85
5	0.78	0.72	0.75	379
accuracy			0.56	600
macro avg	0.33	0.32	0.32	600
weighted avg	0.58	0.56	0.56	600




```
Sample Text:
This nutritious bar is great for before or after workouts.  Low in sugar, high in protein makes this a perfect snack for Diabetics too.

1/1 [==============================] - 0s 30ms/step
Probability: [[0.    0.    0.    0.002 0.998]]
True class: 5
```



Text with highlighted words

This nutritious bar is great for before or after workouts. Low in sugar, high in protein makes this a perfect snack for Diabetics too.

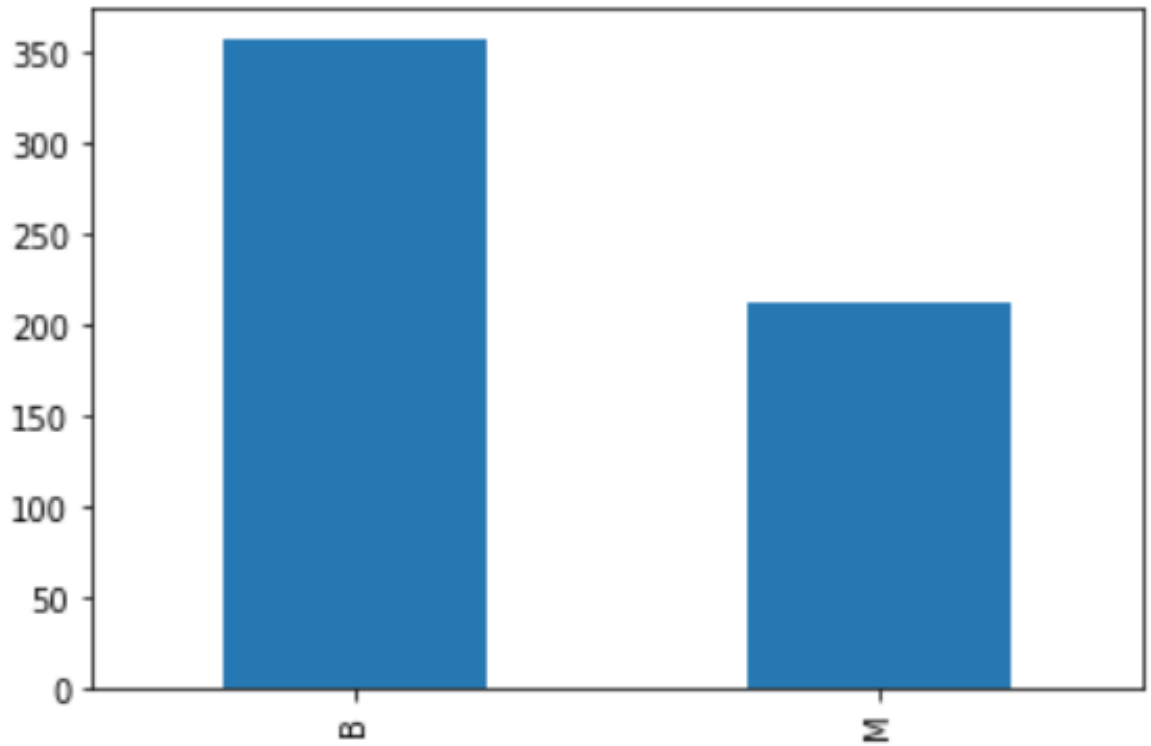
```
Modified Sample Text:
This nutritious bar is   for before or after workouts.  Low in sugar,   in protein   this a   snack for Diabetics too.

1/1 [==============================] - 0s 37ms/step
Probability: [[0.071 0.05  0.164 0.22  0.495]]
```

Chapter 8: Model-Agnostic versus Model-Specific Explainability

```
TensorFlow version: 2.11.0
Python version: 3.9.10
AutoGluon version: 0.6.2
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430

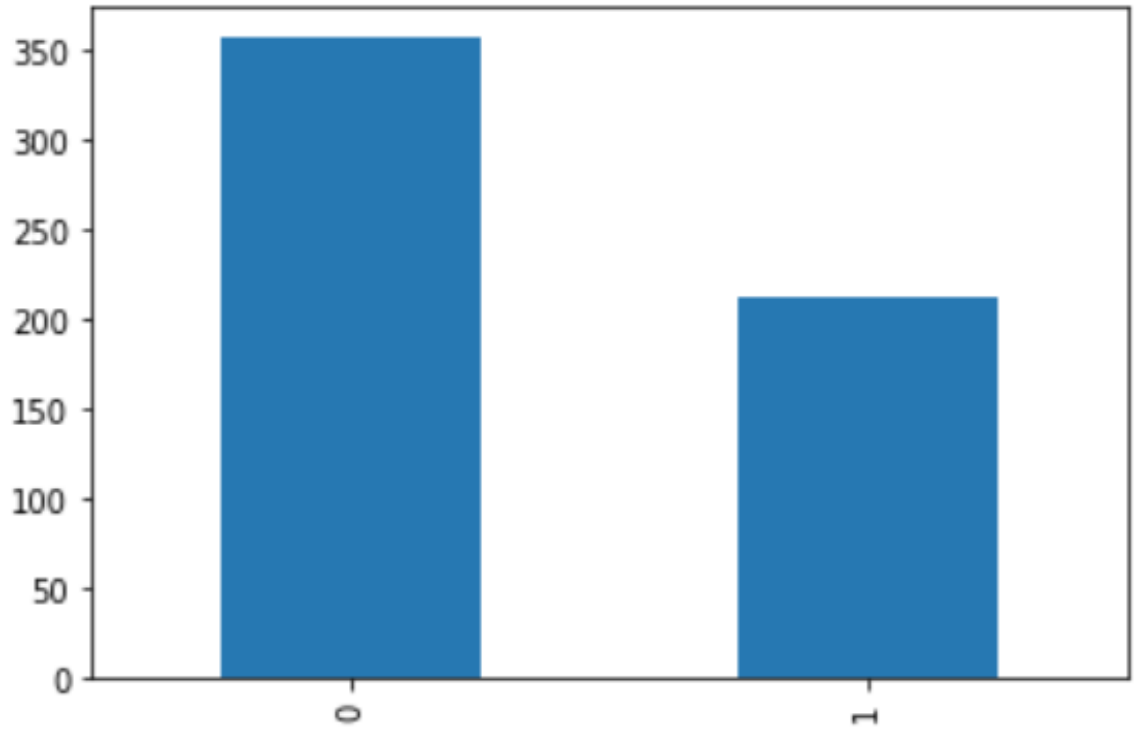
5 rows × 32 columns



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	fractal_dimension_worst	diagnosis
184.6	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890	M
158.8	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902	M
152.5	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758	M

```
Index(['id', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
       'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'diagnosis'],
      dtype='object')
```




```
Train data: 455
Test data: 114
```

```
Class variable summary:
 count    455.000000
mean       0.371429
std        0.483719
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max        1.000000
Name: diagnosis, dtype: float64
```

```
Presets specified: ['best_quality']
Stack configuration (auto_stack=True): num_stack_levels=0, num_bag_folds=5, num_bag_sets=20
Beginning AutoGluon training ... Time limit = 120s
AutoGluon will save models to "ag_breast_cancer/"
AutoGluon Version:  0.6.2
Python Version:     3.9.10
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP Wed Oct 26 20:36:53 UTC 2022
Train Data Rows:    455
Train Data Columns: 31
Label Column: diagnosis
Preprocessing data ...
AutoGluon infers your prediction problem is: 'binary' (because only two unique label-values observed).
        2 unique label values:  [0, 1]
```

	model	score_test	score_val	pred_time_test	pred_time_val	fit_time	pred_time_test_marginal	pred_time_val_marginal	fit_time_marginal	stack_level	can_infer	fit_order
0	LightGBMLarge_BAG_L1	0.973684	0.962637	0.035319	0.009481	5.607196	0.035319	0.009481	5.607196	1	True	13
1	ExtraTreesGini_BAG_L1	0.973684	0.969231	0.080204	0.115706	0.529064	0.080204	0.115706	0.529064	1	True	8
2	ExtraTreesEntr_BAG_L1	0.973684	0.973626	0.080616	0.129573	0.552604	0.080616	0.129573	0.552604	1	True	9
3	NeuralNetTorch_BAG_L1	0.973684	0.980220	0.126236	0.159303	7.019799	0.126236	0.159303	7.019799	1	True	12
4	CatBoost_BAG_L1	0.964912	0.967033	0.065612	0.007055	10.603860	0.065612	0.007055	10.603860	1	True	7
5	RandomForestEntr_BAG_L1	0.964912	0.964835	0.073206	0.113540	0.588865	0.073206	0.113540	0.588865	1	True	6
6	RandomForestGini_BAG_L1	0.964912	0.967033	0.074253	0.114979	0.545684	0.074253	0.114979	0.545684	1	True	5
7	NeuralNetFastAI_BAG_L1	0.964912	0.986813	0.230629	0.180764	6.781015	0.230629	0.180764	6.781015	1	True	10
8	WeightedEnsemble_L2	0.964912	0.986813	0.234094	0.182042	7.373578	0.003464	0.001278	0.592562	2	True	14
9	LightGBMXT_BAG_L1	0.964912	0.982418	0.390843	0.009857	4.205060	0.390843	0.009857	4.205060	1	True	3
10	LightGBM_BAG_L1	0.956140	0.969231	0.025051	0.009724	4.470727	0.025051	0.009724	4.470727	1	True	4
11	XGBoost_BAG_L1	0.956140	0.971429	0.055231	0.020815	3.006829	0.055231	0.020815	3.006829	1	True	11
12	KNeighborsDist_BAG_L1	0.789474	0.806593	0.028558	0.020687	0.009137	0.028558	0.020687	0.009137	1	True	2
13	KNeighborsUnif_BAG_L1	0.754386	0.771429	0.023936	0.011761	0.007596	0.023936	0.011761	0.007596	1	True	1

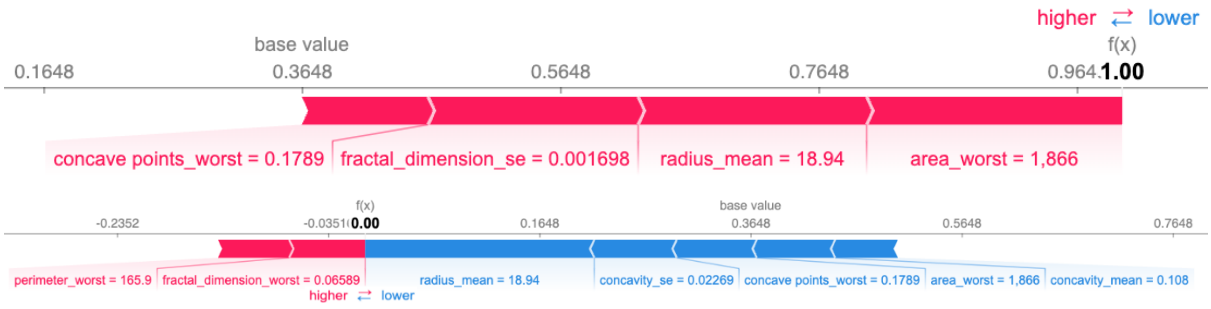
```
Evaluation: accuracy on test data: 0.9649122807017544
Evaluations on test data:
{
    "accuracy": 0.9649122807017544,
    "balanced_accuracy": 0.9626596790042581,
    "mcc": 0.9253193580085162,
    "f1": 0.9534883720930233,
    "precision": 0.9534883720930233,
    "recall": 0.9534883720930233
}
Predictions:
 204    0
70     1
131    1
431    0
540    0
      ..
486    0
75     1
249    0
238    0
265    1
Name: diagnosis, Length: 114, dtype: int64
```

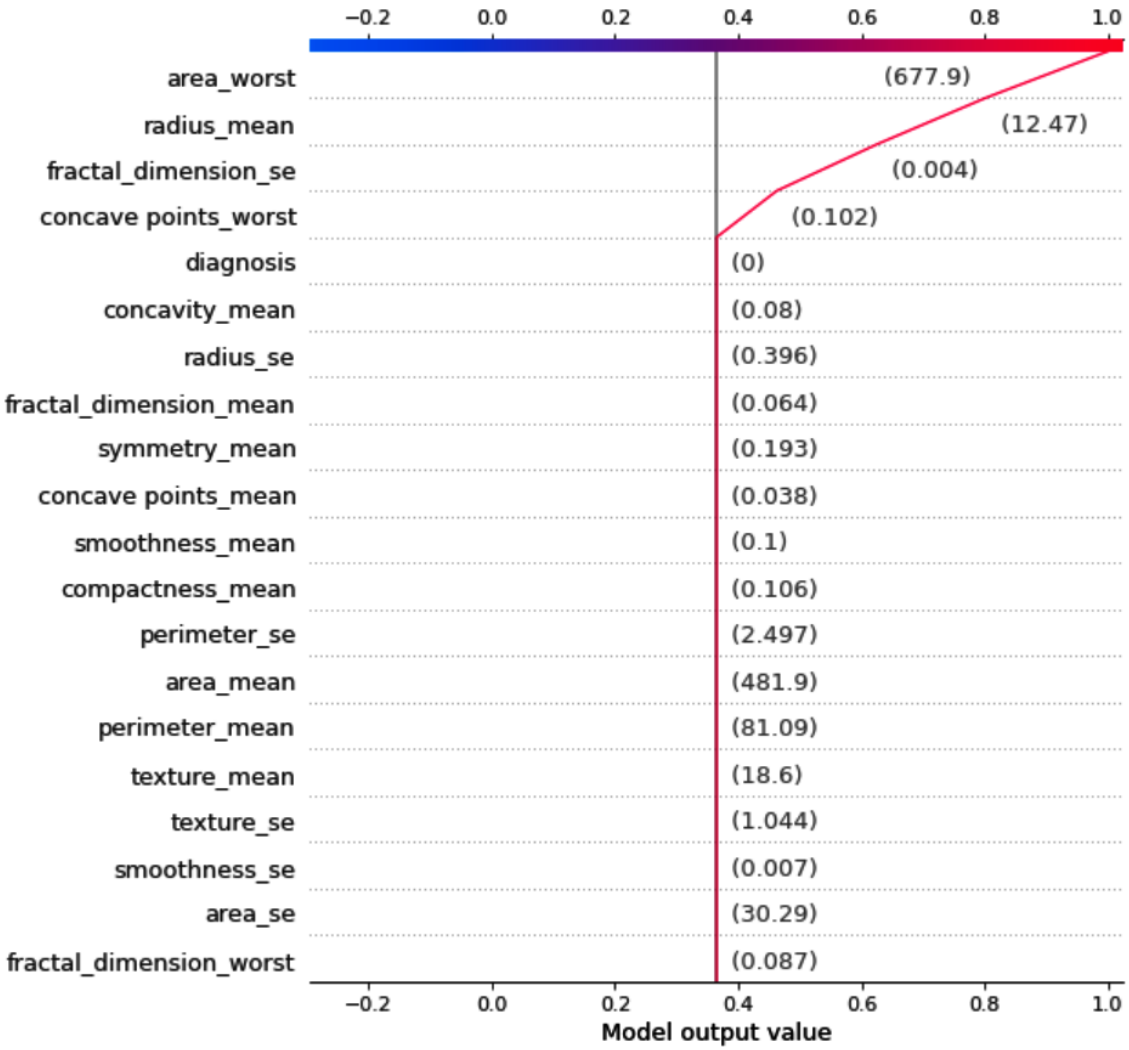
	0	1
204	0.950747	0.049253
70	0.001357	0.998643
131	0.017722	0.982278
431	0.985273	0.014727
540	0.991829	0.008171

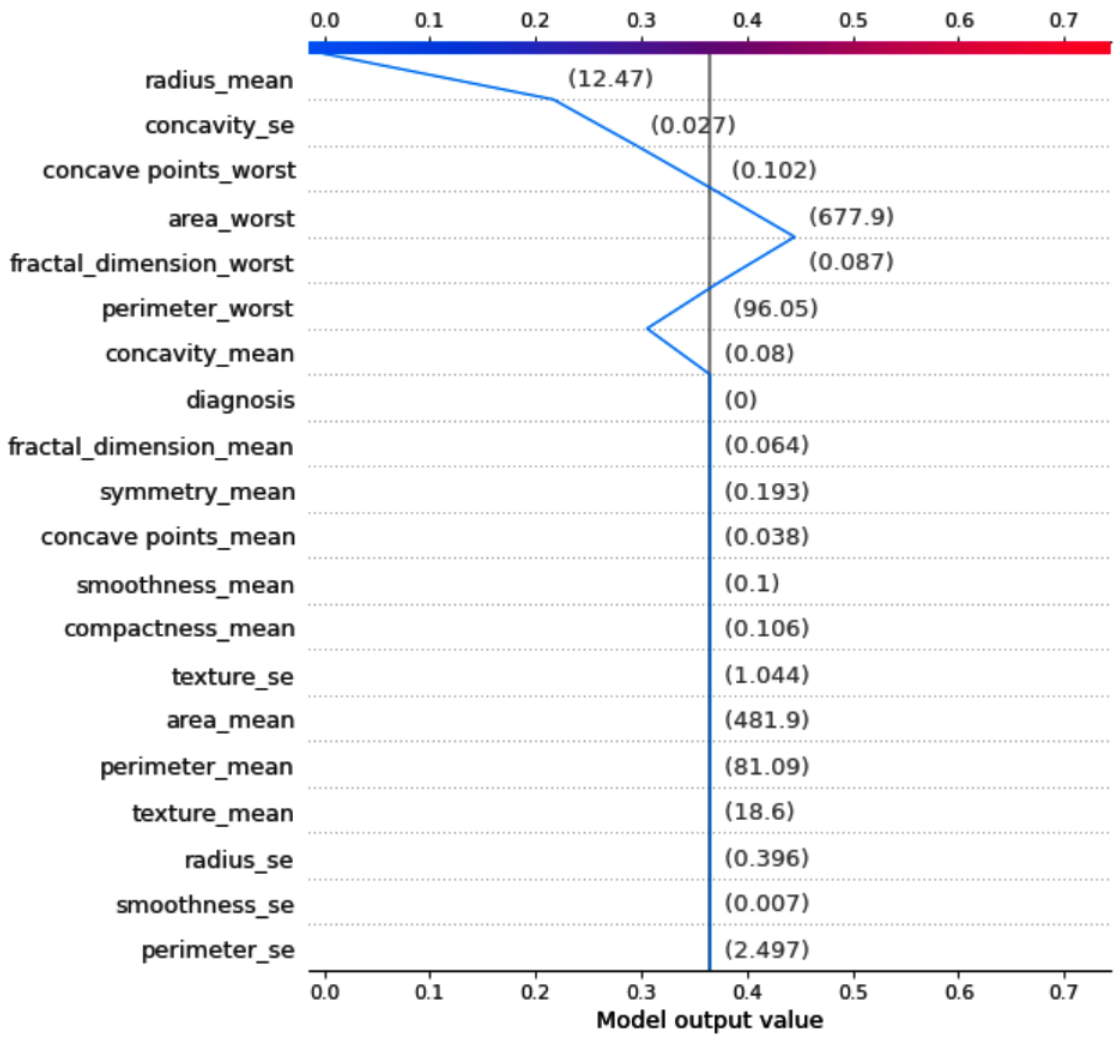
```
*** Summary of fit() ***
Estimated performance of each model:
                     model  score_val  pred_time_val   fit_time  pred_time_val_marginal  fit_time_marginal  stack_level  can_infer  fit_order
0    NeuralNetFastAI_BAG_L1   0.986813       0.151035   6.893745                0.151035           6.893745            1       True         10
1       WeightedEnsemble_L2   0.986813       0.152269   7.472078                0.001234           0.578334            2       True         14
2         LightGBMXT_BAG_L1   0.982418       0.010299   4.088349                0.010299           4.088349            1       True          3
3     NeuralNetTorch_BAG_L1   0.980220       0.126647   7.163876                0.126647           7.163876            1       True         12
4     ExtraTreesEntr_BAG_L1   0.973626       0.122370   0.581059                0.122370           0.581059            1       True          9
5           XGBoost_BAG_L1   0.971429       0.019473   3.023371                0.019473           3.023371            1       True         11
6         LightGBM_BAG_L1   0.969231       0.009121   4.655586                0.009121           4.655586            1       True          4
7     ExtraTreesGini_BAG_L1   0.969231       0.113346   0.517663                0.113346           0.517663            1       True          8
8          CatBoost_BAG_L1   0.967033       0.007781  10.785936                0.007781          10.785936            1       True          7
9   RandomForestGini_BAG_L1   0.967033       0.112606   0.536492                0.112606           0.536492            1       True          5
10  RandomForestEntr_BAG_L1   0.964835       0.112577   0.573340                0.112577           0.573340            1       True          6
11    LightGBMLarge_BAG_L1   0.962637       0.009531   5.738750                0.009531           5.738750            1       True         13
12    KNeighborsDist_BAG_L1   0.806593       0.016240   0.005436                0.016240           0.005436            1       True          2
13    KNeighborsUnif_BAG_L1   0.771429       0.018518   0.006104                0.018518           0.006104            1       True          1
Number of models trained: 14
Types of models trained:
{'StackerEnsembleModel_RF', 'StackerEnsembleModel_LGB', 'StackerEnsembleModel_XT', 'StackerEnsembleModel_TabularNeuralNetTorch', 'StackerEnsembleModel_CatBoost',
'StackerEnsembleModel_XGBoost', 'WeightedEnsembleModel', 'StackerEnsembleModel_NNFastAiTabular', 'StackerEnsembleModel_KNN'}
Bagging used: True  (with 5 folds)
Multi-layer stack-ensembling used: False
Feature Metadata (Processed):
(raw dtype, special dtypes):
('float', []) : 30 | ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', ...]
('int', [])   :  1 | ['id']
Plot summary of models saved to file: ag_breast_cancer/SummaryOfModels.html
*** End of fit() summary ***
```

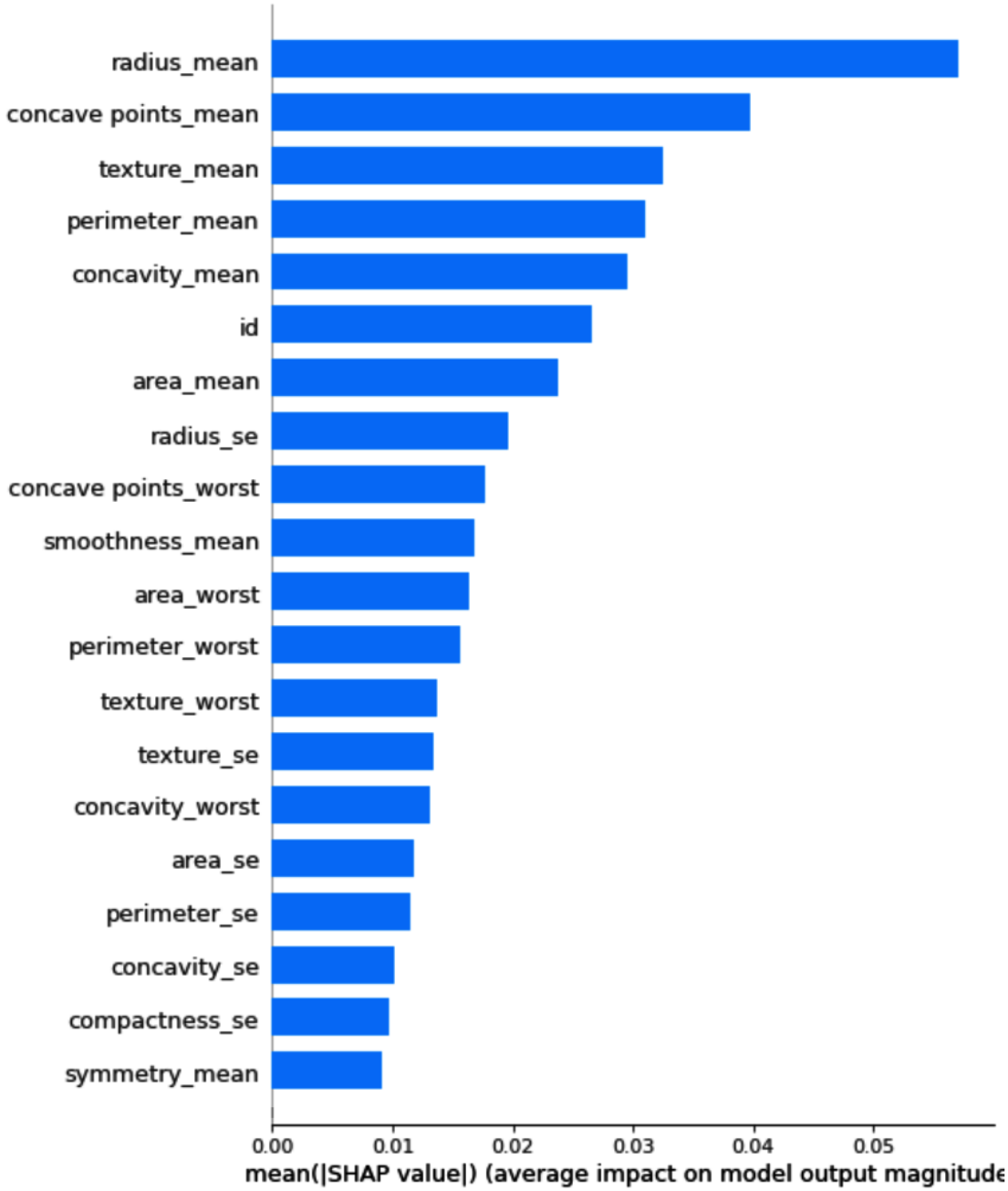
```
AutoGluon infers problem type is:  binary
AutoGluon identified the following types of features:
('float', []) : 30 | ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', ...]
('int', [])   :  1 | ['id']
```

SHAP values length: 114

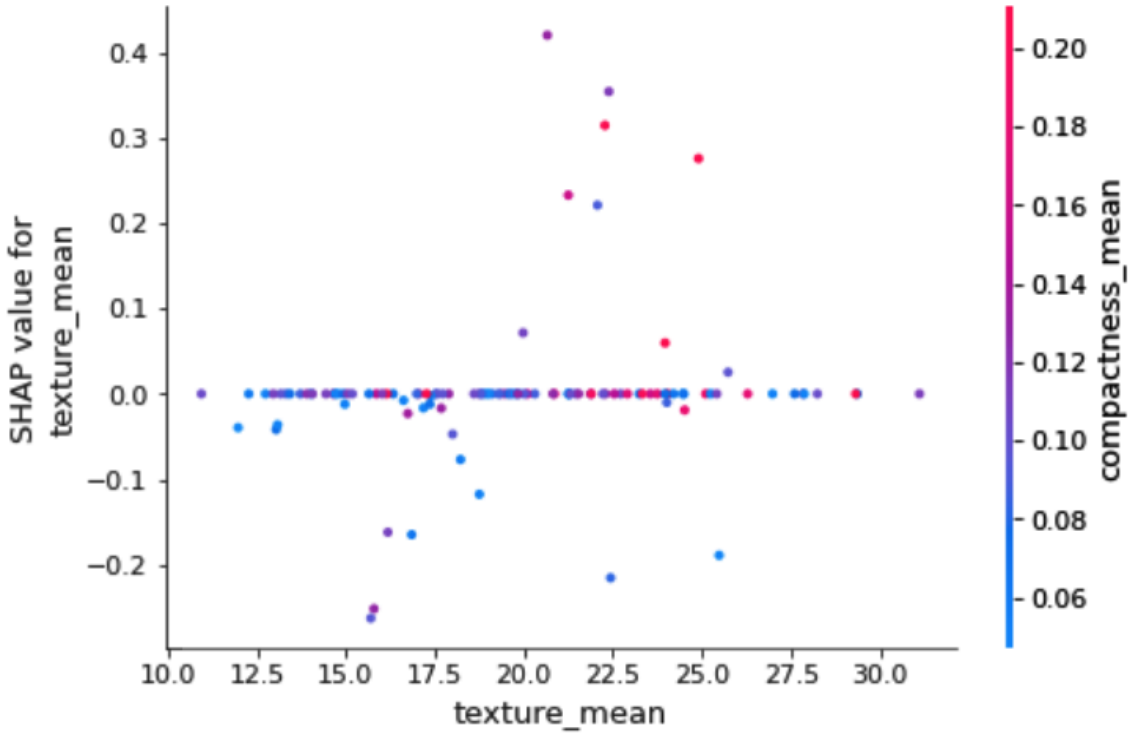


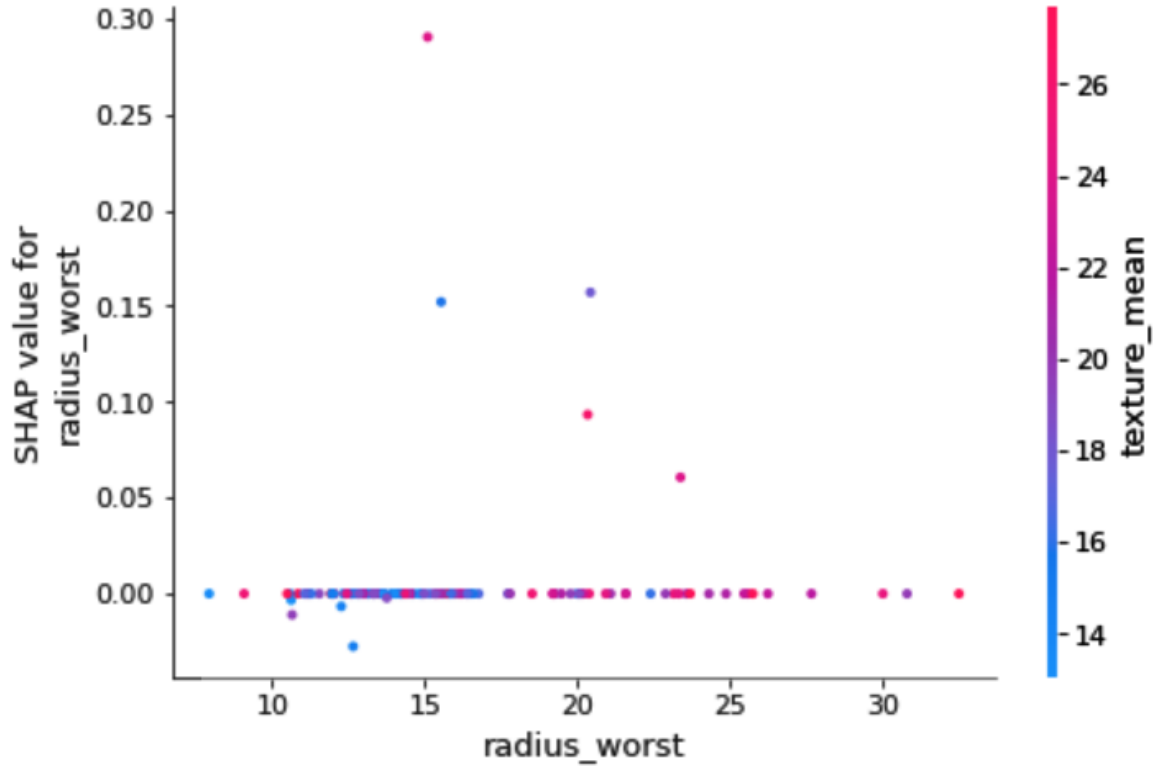

−0.2
0.0
0.2
0.4
0.6
0.8
1.0
area_worst
(677.9)
radius_mean
(12.47)
fractal_dimension_se
(0.004)
concave points_worst
(0.102)
diagnosis
(0)
concavity_mean
(0.08)
radius_se
(0.396)
fractal_dimension_mean
(0.064)
symmetry_mean
(0.193)
concave points_mean
(0.038)
smoothness_mean
(0.1)
compactness_mean
(0.106)
perimeter_se
(2.497)
area_mean
(481.9)
perimeter_mean
(81.09)
texture_mean
(18.6)
texture_se
(1.044)
smoothness_se
(0.007)
area_se
(30.29)
fractal_dimension_worst
(0.087)
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Model output value


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
radius_mean
concavity_se
concave points_worst
area_worst
fractal_dimension_worst
perimeter_worst
concavity_mean
diagnosis
fractal_dimension_mean
symmetry_mean
concave points_mean
smoothness_mean
compactness_mean
texture_se
area_mean
perimeter_mean
texture_mean
radius_se
smoothness_se
perimeter_se
(12.47)
(0.027)
(0.102)
(677.9)
(0.087)
(96.05)
(0.08)
(0)
(0.064)
(0.193)
(0.038)
(0.1)
(0.106)
(1.044)
(481.9)
(81.09)
(18.6)
(0.396)
(0.007)
(2.497)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Model output value


radius_mean
concave points_mean
texture_mean
perimeter_mean
concavity_mean
id
area_mean
radius_se
concave points_worst
smoothness_mean
area_worst
perimeter_worst
texture_worst
texture_se
concavity_worst
area_se
perimeter_se
concavity_se
compactness_se
symmetry_mean
0.00
0.01
0.02
0.03
0.04
0.05
mean(|SHAP value|) (average impact on model output magnitude


radius_mean
concave points_mean
texture_mean
perimeter_mean
concavity_mean
id
area_mean
radius_se
concave points_worst
smoothness_mean
area_worst
perimeter_worst
texture_worst
texture_se
concavity_worst
area_se
perimeter_se
concavity_se
compactness_se
symmetry_mean
−0.2
0.0
0.2
0.4
0.6
SHAP value (impact on model output)
High
Low
Feature value



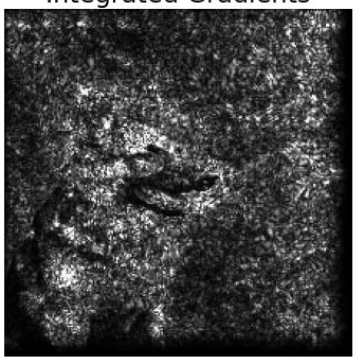


TensorFlow version: 2.11.0
Python version: 3.9.10

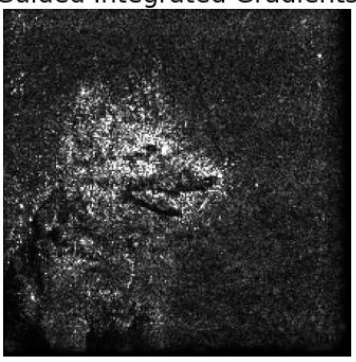
Prediction: German Shepherd
Class: 236

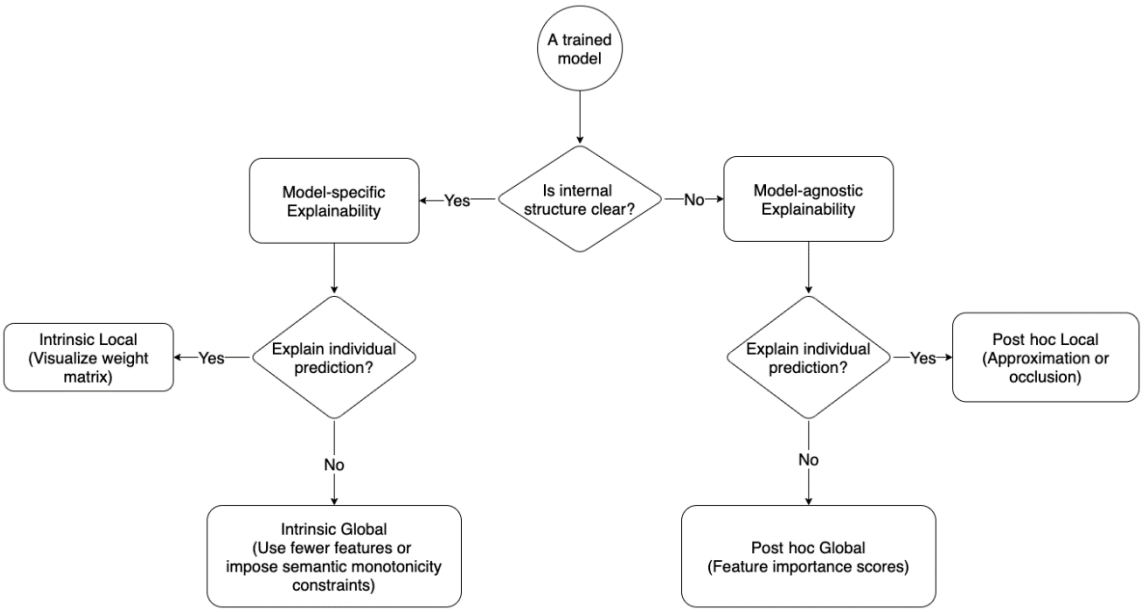
Original Image

Integrated Gradients



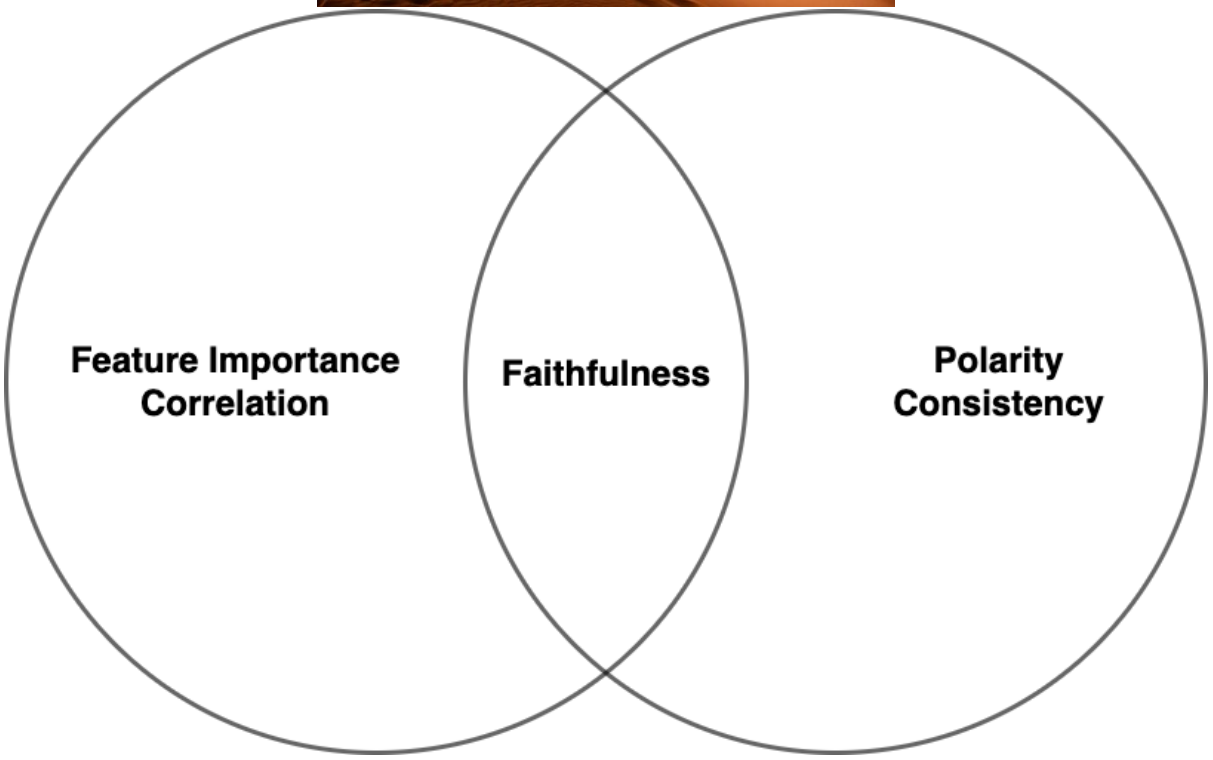
Guided Integrated Gradients

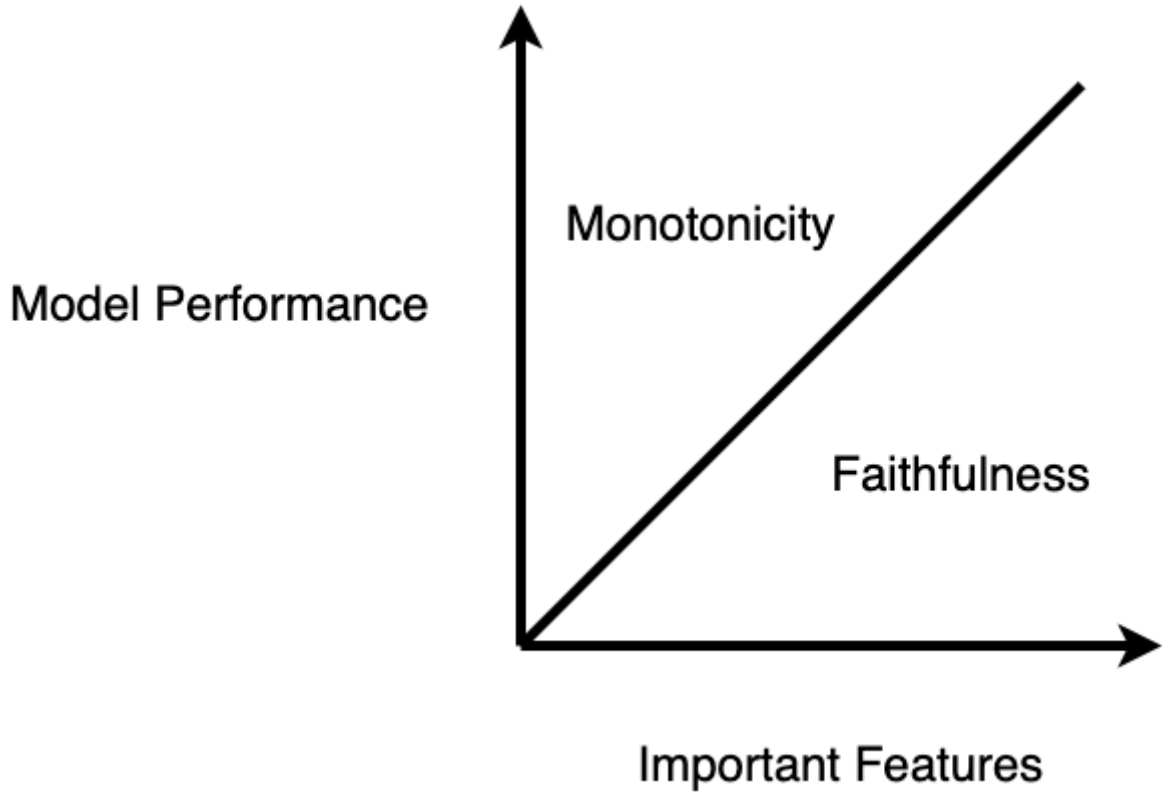


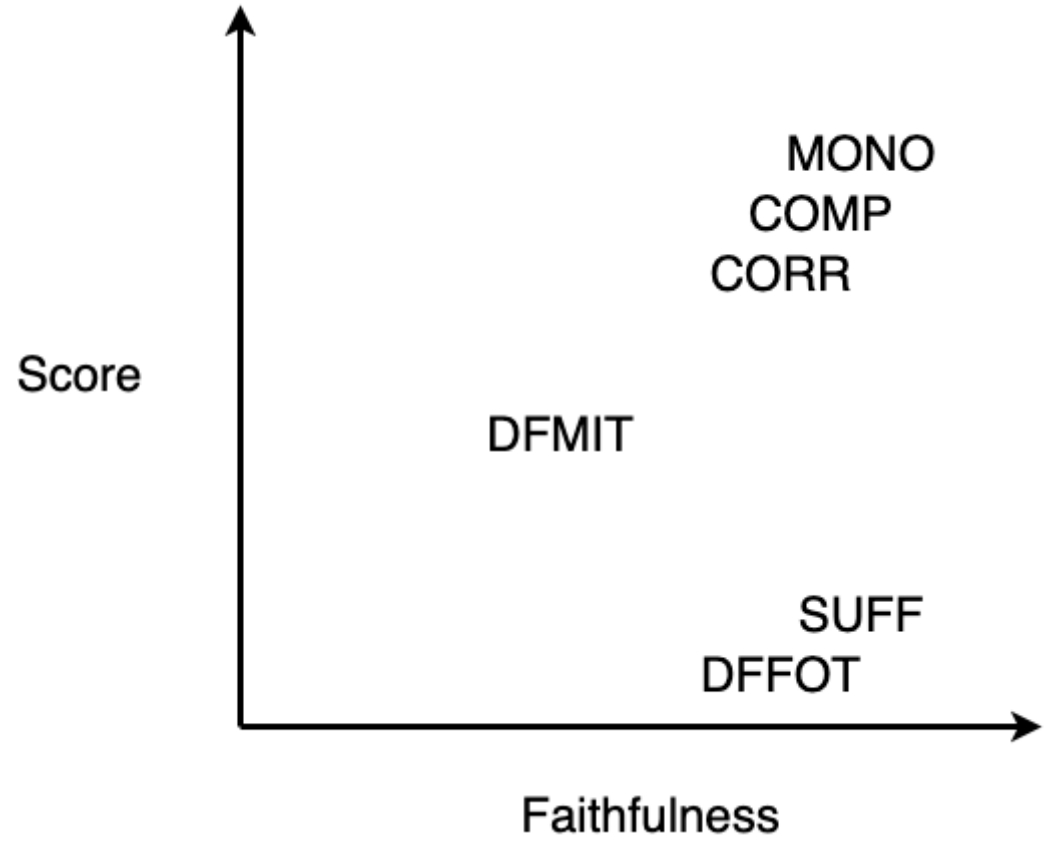

A trained model
Is internal structure clear?
Yes
No
Model-specific Explainability
Model-agnostic Explainability
Explain individual prediction?
Explain individual prediction?
Yes
Yes
No
No
Intrinsic Local (Visualize weight matrix)
Post hoc Local (Approximation or occlusion)
Intrinsic Global (Use fewer features or impose semantic monotonicity constraints)
Post hoc Global (Feature importance scores)

Chapter 9: Explainability Evaluation Schemes

User	XAI Goals	Information	Benchmark
Who is the target audience?	What to explain?	How to explain?	How to evaluate explanations?




Monotonicity
Model Performance
Faithfulness
Important Features


MONO
COMP
CORR
Score
DFMIT
SUFF
DFFOT
Faithfulness