Performance in the absence of the ground truth

DIVE INTO NANNYML

SoyGema

Key Concepts of machine learning in production

- Model Decay
  - Data Drift
  - Concept Drift

Estimating performance in the absence of the ground truth

- CBEP estimator
- Data Drift Plots
- Alerts

From the tutorial to the reality

- Hacking the docs
- Unpopulated Chunks

WORK FROM HOME? → 1 YES
→ 0 NO

Binary Model

Model Decay

Model Decay

Detected a posteriori

Model Decay

Detected a posteriori

Silent

Model Decay

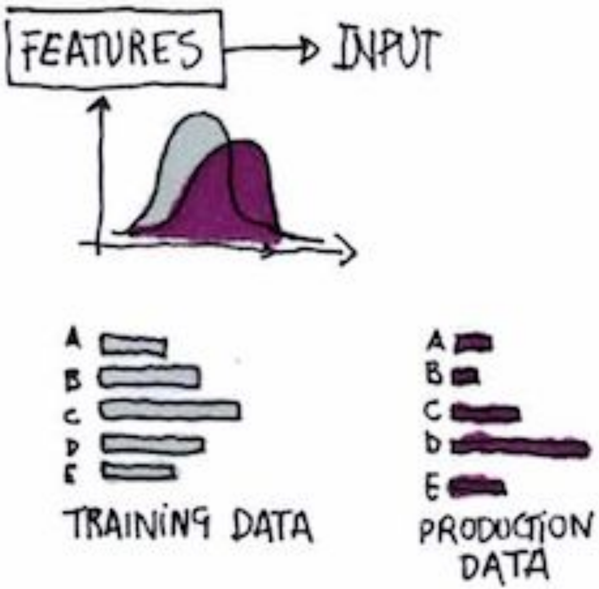
Detected a posteriori

Silent

Can take bad
decisions

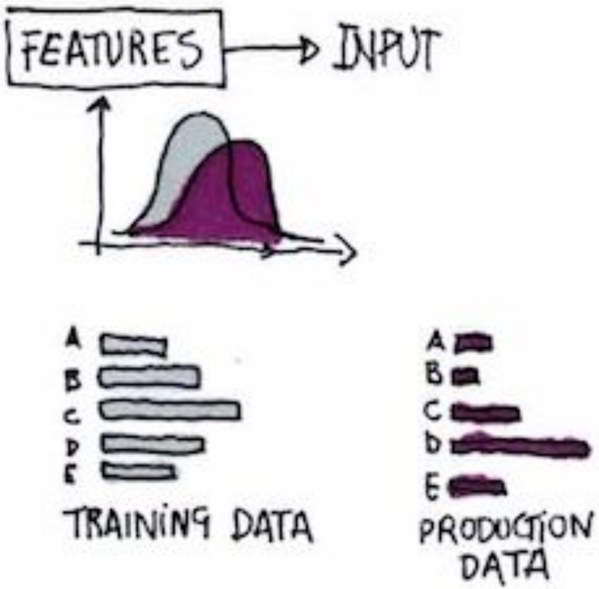
Why do Machine learning models fail ?

Data Drift



Why do Machine learning models fail ?

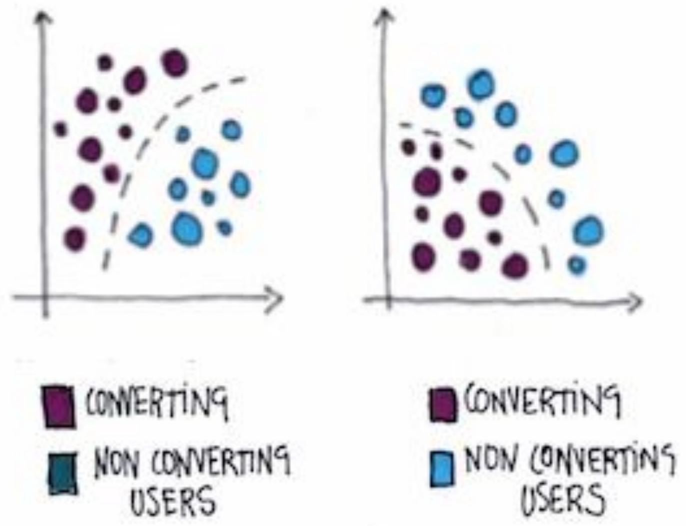
Data Drift



A/B testing

Why do Machine learning models fail ?

Concept Drift



The main difference in between data drift and concept drift is that data drift is associated with the change in the statistical properties of the inputs or independent variables associated with features, while concept drift is a change in the pattern in between the inputs and the outputs associated with the classifier or when is a major change in the statistical properties of the dependent variable

RECAP

Model Decay

Data Drift

Concept Drift

Performance in the absence of the ground truth

Generally, Model decay is normally detected a posteriori, when we do have the opportunity to compare the true label -y or actuals—with the predictions of the model -y_hat-' …So How do we measure performance estimation in the absence of the ground truth ?

CBEP estimator

CBEP estimator

Partitions

Predicted Proba

CBEP estimator

Partitions

Predicted Proba

Reference

- Data that shows good model performance
- True label (y)
- Predicted probabilities

Analysis

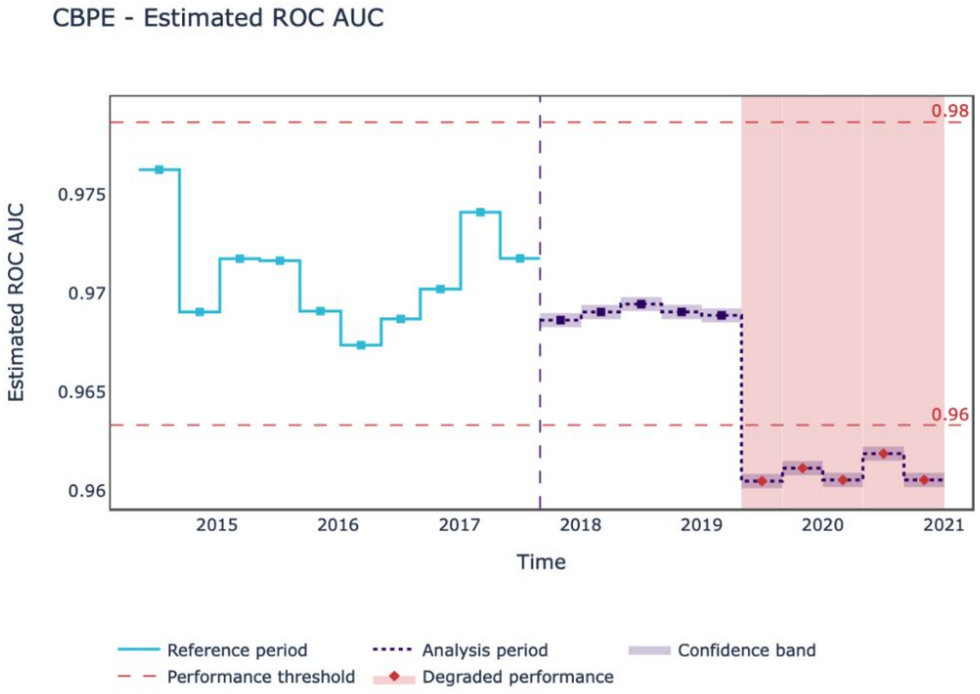
- Absence of the ground truth
- NO True label (y)
- Predicted probabilities

Metadata

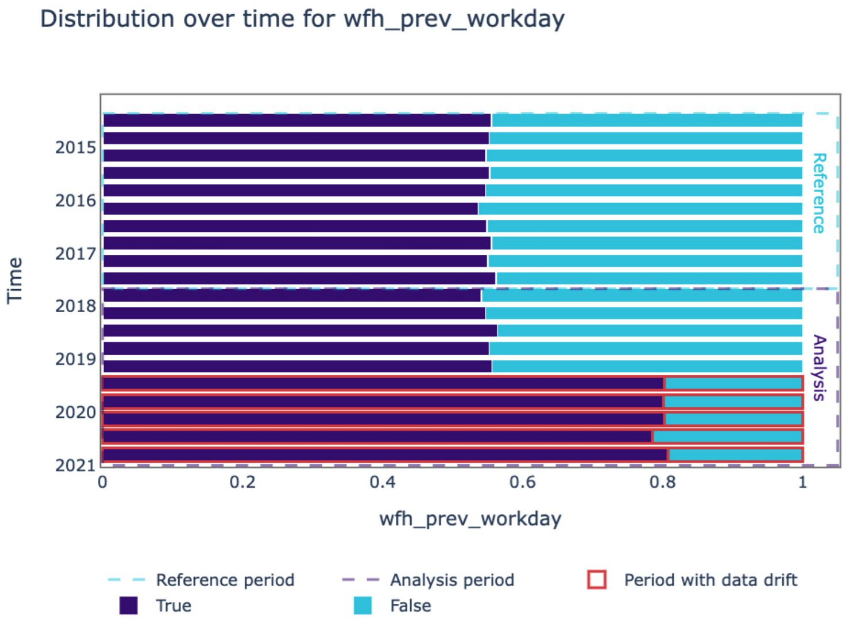
```
occurred.metadata = nml.extract_metadata(reference)
metadata.target_column_name = ‘work_home_actual’
```

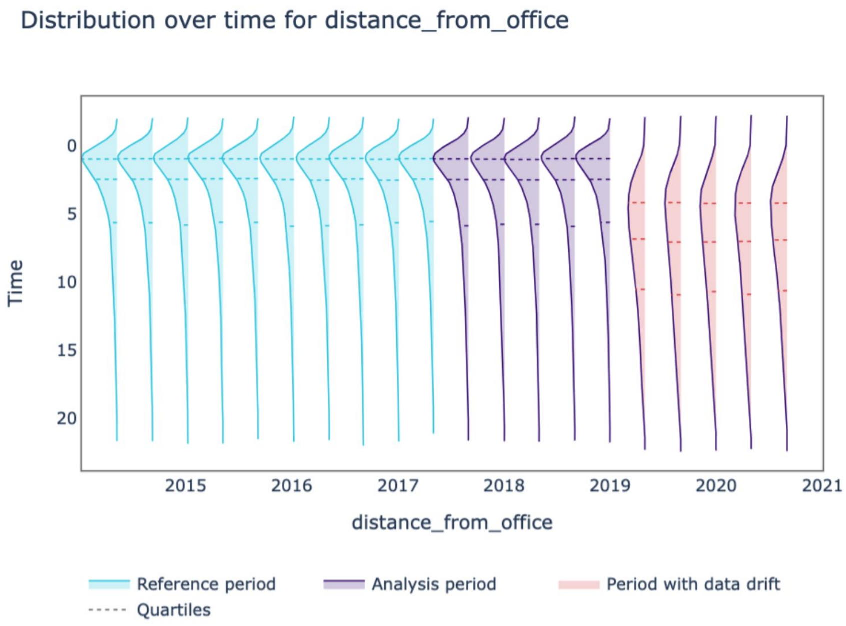
Estimator

```
cbpe = nml.CBPE(model_metadata=metadata, chunk_period="D")
cbpe.fit(reference_data=reference)
```

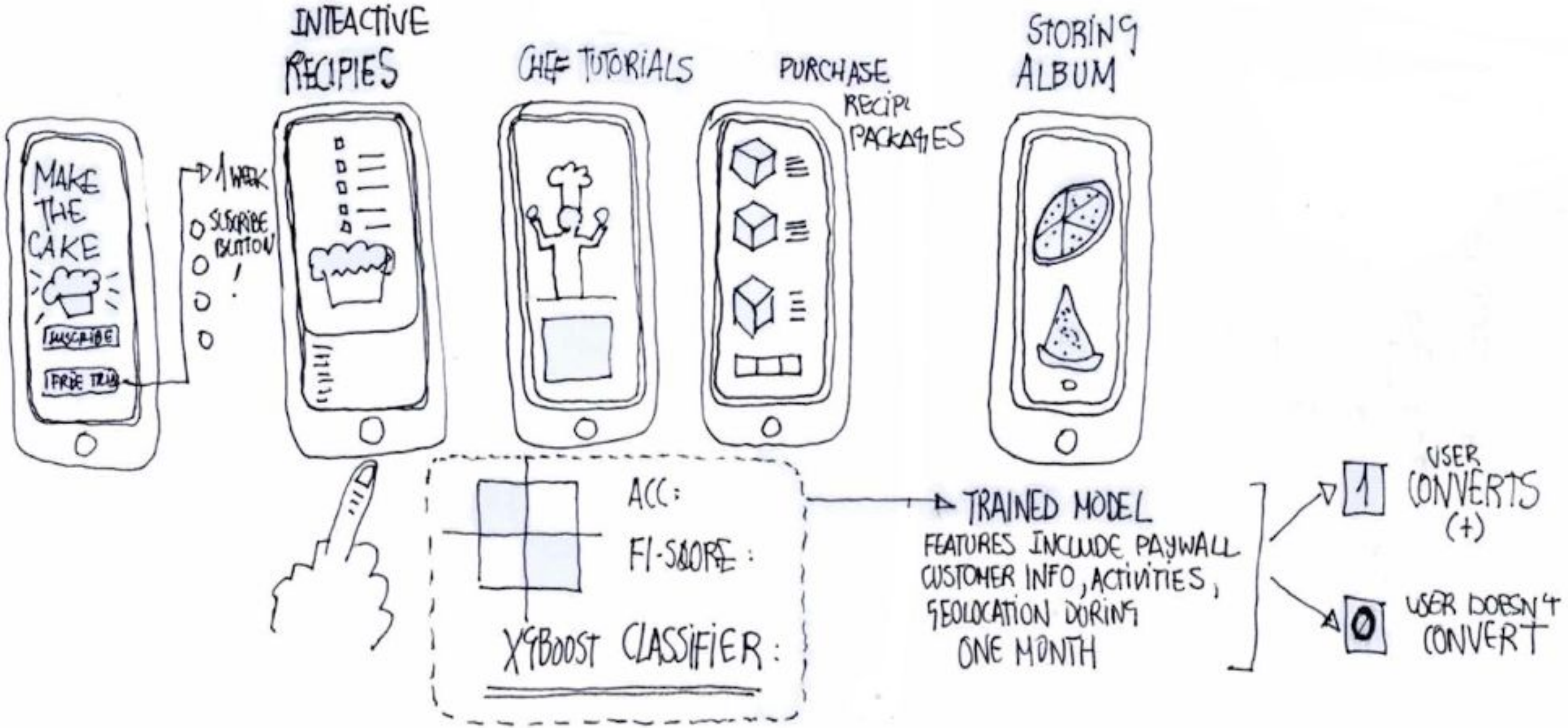


Data Drift Plots






IT ESTIMATES THE PERFORMANCE WITHOUT THE REAL LABEL
IT WILL WORK WITH MY DATA FOR MY BUSINESS CASE, RIGHT?
RIGHT?


INTEACTIVE RECIPIES
CHEF TUTORIALS
PURCHASE RECIPE PACKAGES
STORING ALBUM
MAKE THE CAKE
SUBSCRIBE
FREE TRIAL
1 WEEK
SUBSCRIBE BUTTON !
ACC:
F1-SCORE:
XGBOOST CLASSIFIER:
TRAINED MODEL
FEATURES INCLUDE PAYWALL CUSTOMER INFO, ACTIVITIES, GEOLOCATION DURING ONE MONTH
1
USER CONVERTS (+)
0
USER DOESN'T CONVERT

user_id	days_since_trial_start	n_sessions	os_name	country	locale	device_type	source	level	...	time_chef	time_album
23326879	5	1	1	16	3	2	3	0	...	0.000000	0.0
23251961	2	1	1	36	3	2	3	1	...	1.000000	0.0
23412815	6	3	1	43	4	2	3	1	...	0.142857	0.0
23464377	5	1	1	39	17	2	3	1	...	0.055556	0.0
23482370	1	1	0	43	3	63	3	1	...	1.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...
23534685	1	2	1	36	3	2	0	2	...	0.000000	0.0
23416575	6	3	1	43	3	2	3	1	...	0.250000	0.0
23549766	0	1	1	16	3	2	3	1	...	1.000000	0.0
23538438	1	2	1	16	3	2	3	1	...	0.000000	0.0
23525662	0	1	1	16	3	2	3	2	...	0.250000	0.0

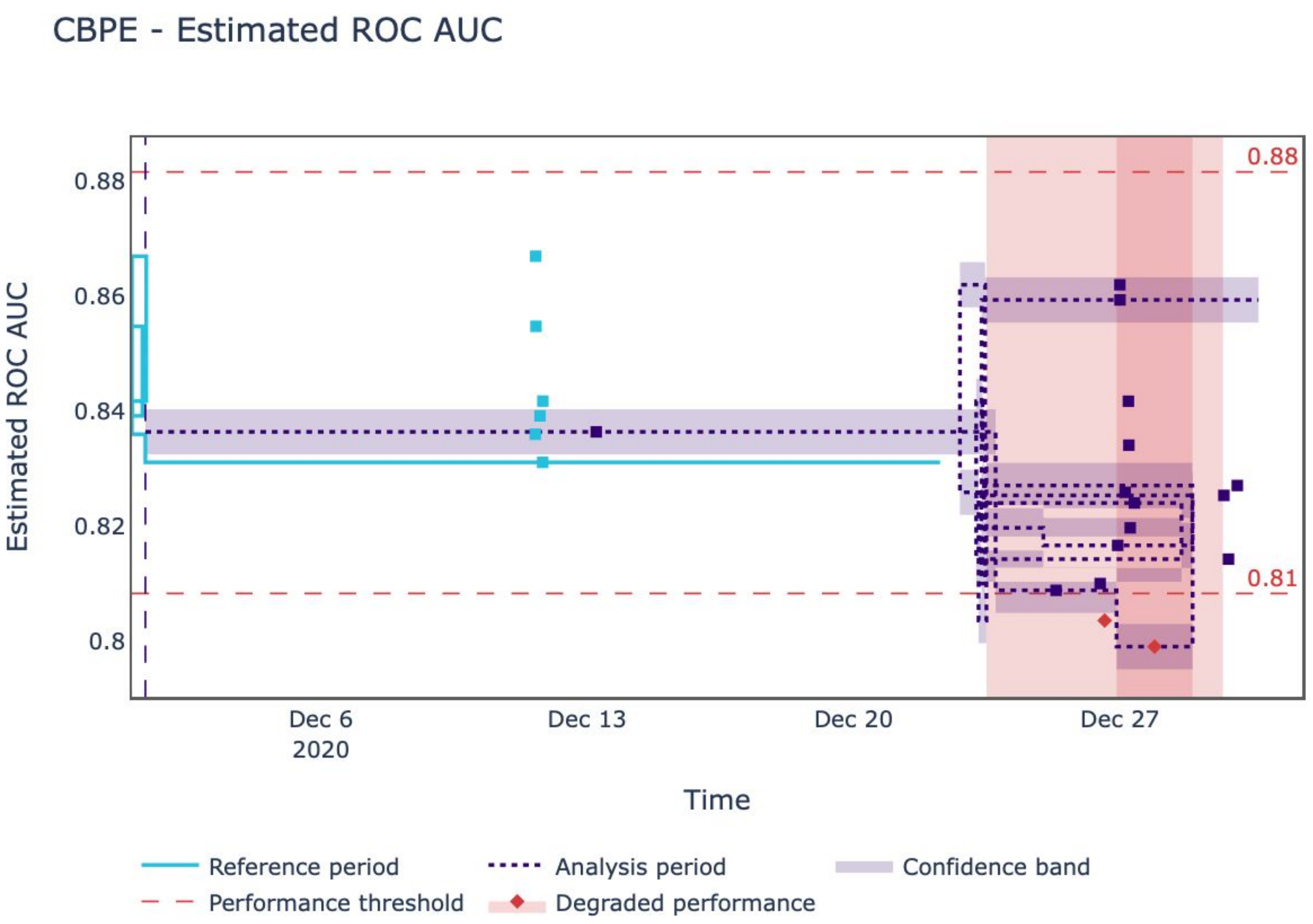
Metadata

```
metadata = nml.extract_metadata(reference)
metadata.target_column_name = 'customer'
```

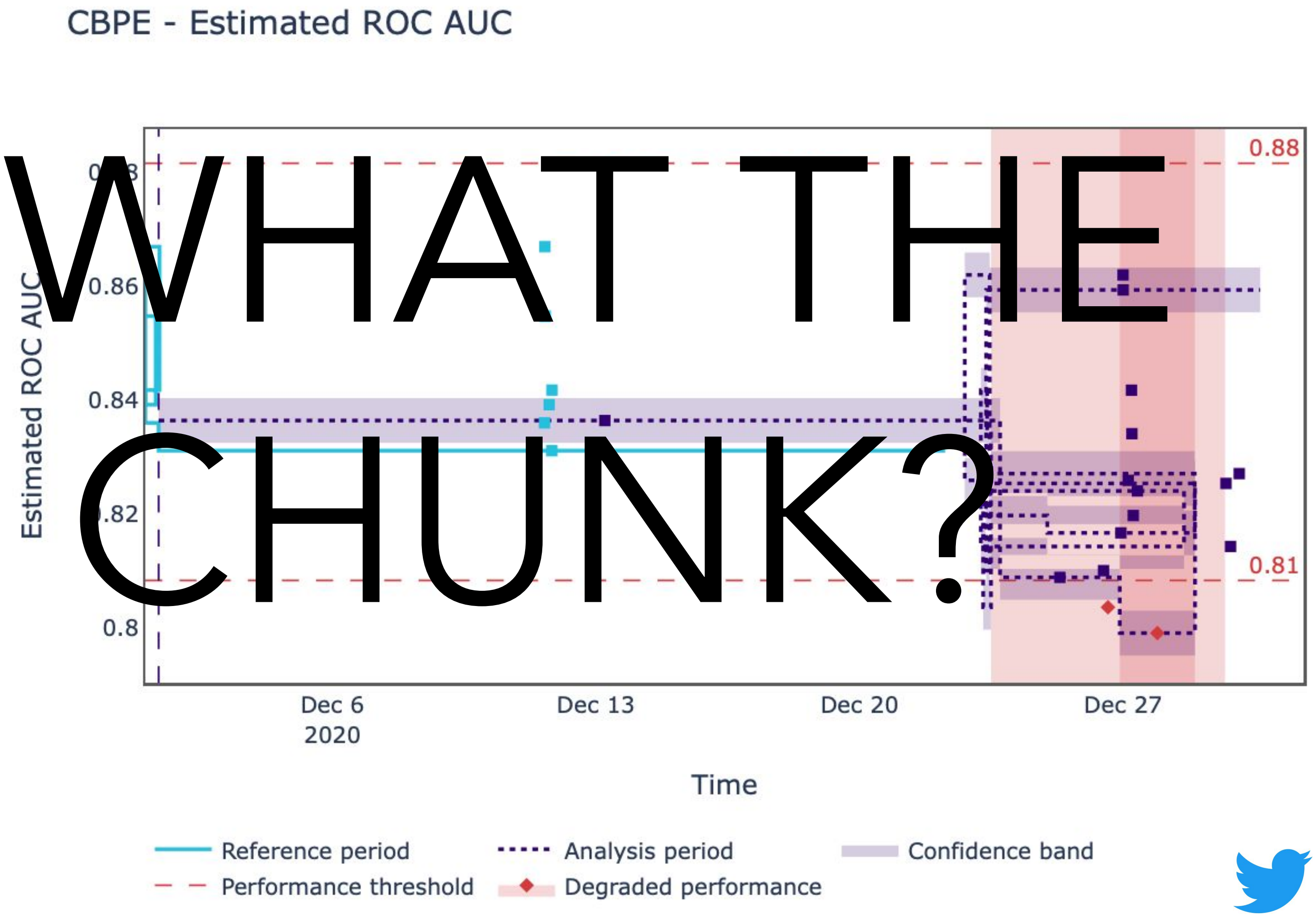
```
Model problem                        binary_classification

Identifier column                    uid
Timestamp column                     date
Partition column                     partition
Prediction column                    y_pred
Predicted probability column         y_pred_proba
Target column                        customer
```

Estimator

```
cbpe = nml.CBPE(model_metadata=metadata)
cbpe.fit(reference_data=reference)
est_perf = cbpe.estimate(pd.concat([reference, analysis], ignore_index=True))
est_perf.data.head()
fig = est_perf.plot(kind='performance')
fig.show()
```

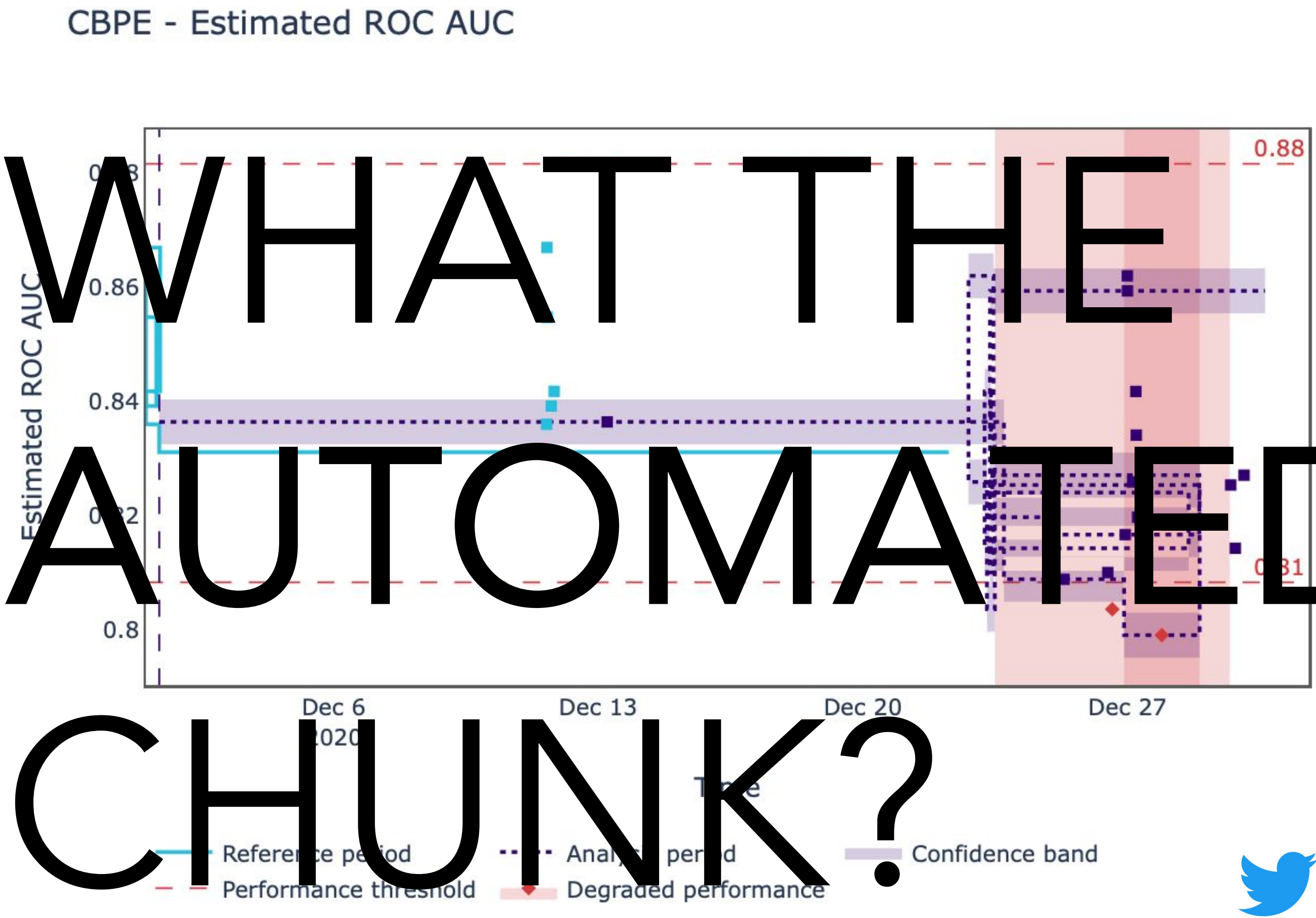

CBPE - Estimated ROC AUC
Estimated ROC AUC
0.88
0.86
0.84
0.82
0.8
0.88
0.81
Dec 6
2020
Dec 13
Dec 20
Dec 27
Time
Reference period
Analysis period
Confidence band
Performance threshold
Degraded performance

WHAT THE CHUNK?



SoyGema

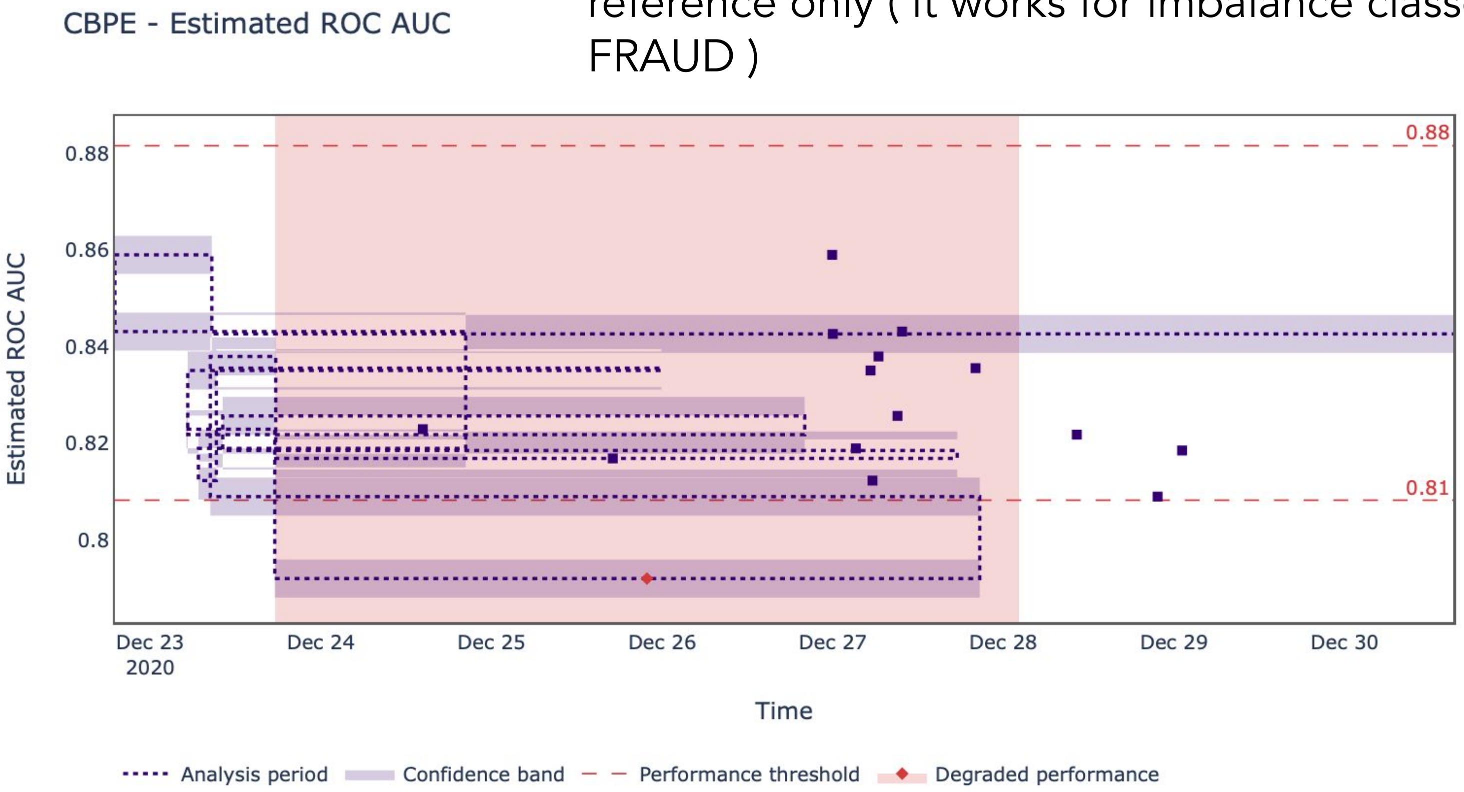
WHAT THE AUTOMATED CHUNK?



SoyGema

```
cbpe = nml.CBPE(model_metadata=metadata)
cbpe.fit(reference_data=reference)
est_perf = cbpe.estimate(pd.concat([analysis], ignore_index=True))
est_perf.data.head()
fig = est_perf.plot(kind='performance')
fig.show()
```

The estimation sometimes is performed on reference only (it works for imbalance classes : Ej : FRAUD)



```
est_perf = cbpe.estimate(pd.concat([reference, analysis], ignore_index=True))
```

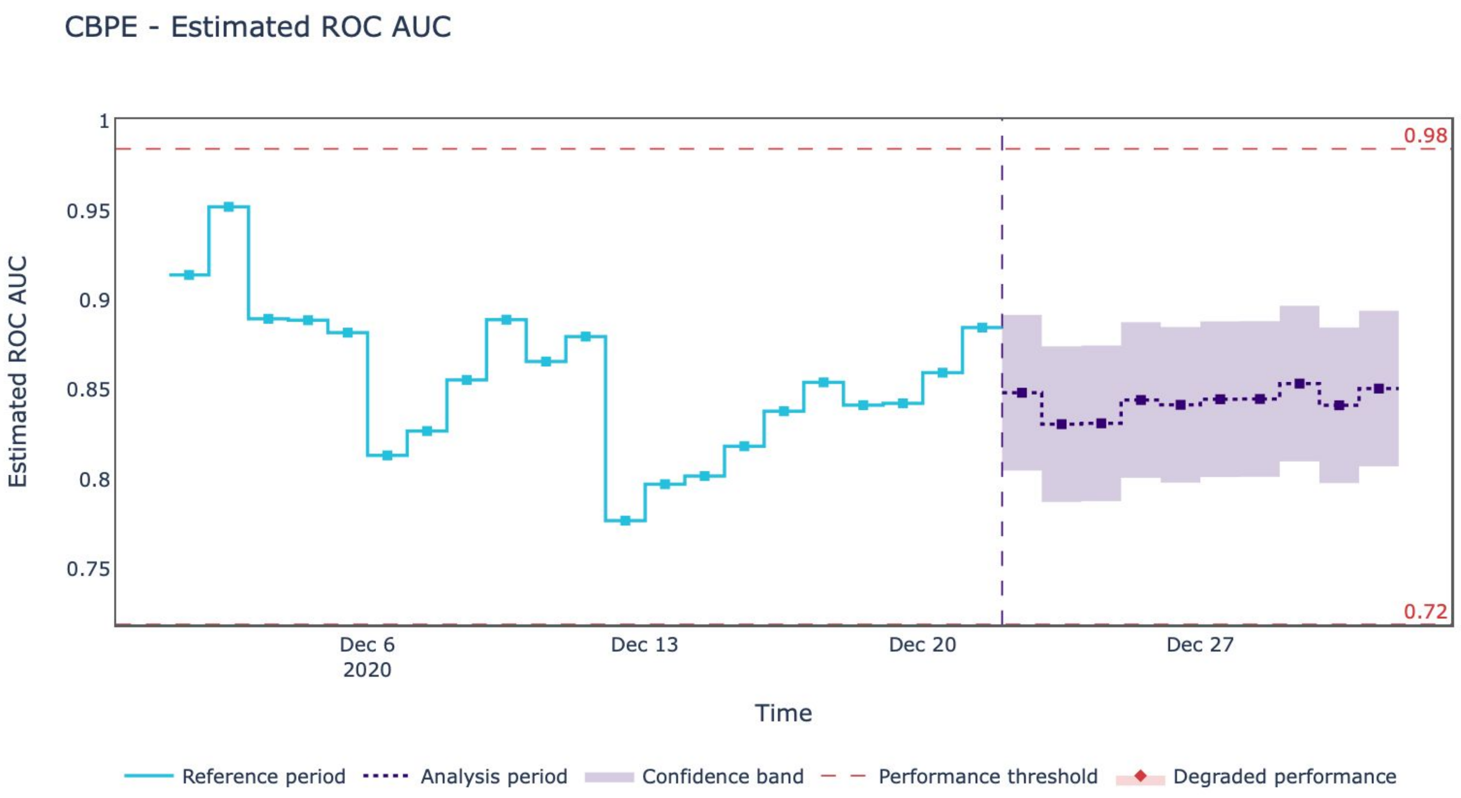
Confidence based performance estimation is ALWAYS done in ANALYSIS PARTITION , we concatenate them for visual completeness .

Time-based chuncks

```
cbpe = nml.CBPE(model_metadata=metadata, chunk_period="D")
cbpe.fit(reference_data=reference)
est_perf = cbpe.estimate(pd.concat([reference, analysis], ignore_index=True))
fig = est_perf.plot(kind='performance')
fig.show()
```

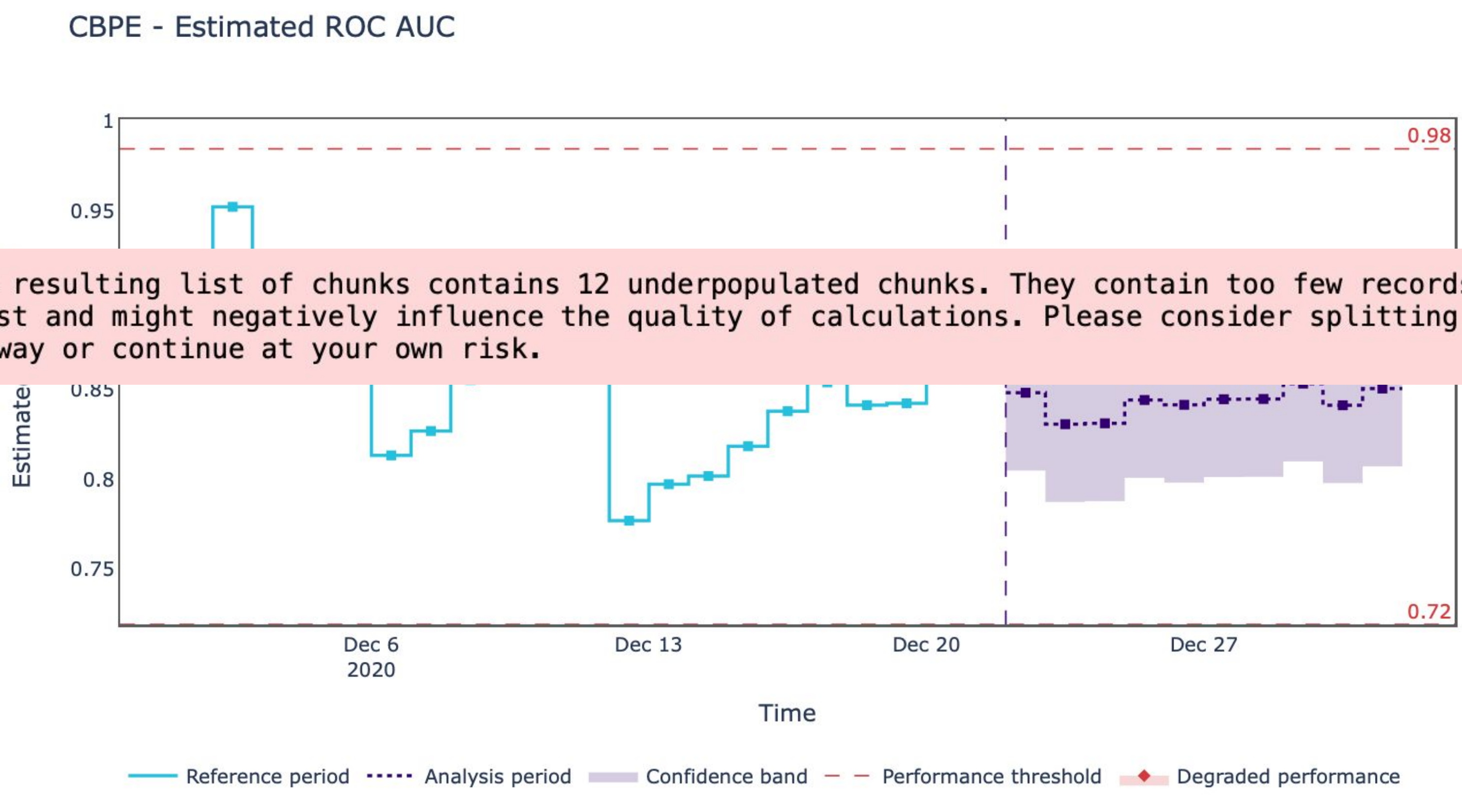
SoyGema

Time-based chuncks



SoyGema

Time-based chuncks




```
The resulting list of chunks contains 12 underpopulated chunks. They contain too few records to be statistically r
obust and might negatively influence the quality of calculations. Please consider splitting your data in a differe
nt way or continue at your own risk.
```

SoyGema

Size based chunks based on
Time-based intervals

Size-based Knowledge Time-based

SoyGema

CONCLUSIONS

Model Decay

CBPE Estimator

Chunks

https://github.com/NannyML/nannyml/

A STAR !

SoyGema