Child Sleep prediction:
Analysing and predicting child sleep behaviour

Members: Haider Ali, Yujie Wu

Original Dataset

```
train_events
```

✓ 0.0s

	series_id	night	event	step	timestamp
0	038441c925bb	1	onset	4992.0	2018-08-14T22:26:00-0400
1	038441c925bb	1	wakeup	10932.0	2018-08-15T06:41:00-0400
2	038441c925bb	2	onset	20244.0	2018-08-15T19:37:00-0400
3	038441c925bb	2	wakeup	27492.0	2018-08-16T05:41:00-0400
4	038441c925bb	3	onset	39996.0	2018-08-16T23:03:00-0400
...	...	...	...	...	...
14503	fe90110788d2	33	wakeup	560604.0	2017-09-06T04:07:00-0400
14504	fe90110788d2	34	onset	574620.0	2017-09-06T23:35:00-0400
14505	fe90110788d2	34	wakeup	581604.0	2017-09-07T09:17:00-0400
14506	fe90110788d2	35	onset	NaN	NaN
14507	fe90110788d2	35	wakeup	NaN	NaN

14508 rows × 5 columns

```
train_series
```

✓ 0.1s

	series_id	step	timestamp	anglez	enmo
0	038441c925bb	0	2018-08-14T15:30:00-0400	2.636700	0.0217
1	038441c925bb	1	2018-08-14T15:30:05-0400	2.636800	0.0215
2	038441c925bb	2	2018-08-14T15:30:10-0400	2.637000	0.0216
3	038441c925bb	3	2018-08-14T15:30:15-0400	2.636800	0.0213
4	038441c925bb	4	2018-08-14T15:30:20-0400	2.636800	0.0215
...	...	...	...	...	...
127946335	fe90110788d2	592375	2017-09-08T00:14:35-0400	-27.277500	0.0204
127946336	fe90110788d2	592376	2017-09-08T00:14:40-0400	-27.032499	0.0233
127946337	fe90110788d2	592377	2017-09-08T00:14:45-0400	-26.841200	0.0202
127946338	fe90110788d2	592378	2017-09-08T00:14:50-0400	-26.723900	0.0199
127946339	fe90110788d2	592379	2017-09-08T00:14:55-0400	-31.521601	0.0205

127946340 rows × 5 columns

Challenges with the dataset

1. The main challenge we faced is about the size of datasets which is too big to load in the memory and apply transformations, hence we used Polars.
2. The merging of the data from series to events to get the event for each sample takes a lot of memory. The merging takes based on each childID. However, the duration of one observation in the smaller dataset (6 hrs to 10 hrs) usually contains 4000 to 7000 observations in the larger one (5 seconds). This merge function cost the most time (4 hrs to 10 hrs) and memories.

Dataset description

Methodology 1 dataset:

1. Randomly pick 50,000,000 rows of 120,000,000 rows. The training size is 45,000,000, and the test size is 5,000,000. Due to the size of dataset, can not import all data at once.
2. Cut the data off by the children (‘series_id’), which contains 269 different children. Randomly pick 240 children and tested by other 29 children.

Methodology 2 dataset:

1. Applied averaging method on the original dataset using the window size for each child.
2. Also, assigned each child a cluster for better accuracy
3. The final dataset for training and validatioin is as below:

X_Train: (6856130, 7), Y_Train: (6856130, 1)
X_Val: (1714033, 7), Y_Val: (1714033, 1)

Training for Methodology 1

KNN:

The first choice of me, because of the continuity and strongly correlation of the dataset. The best hyperparameter is n_neighbors=3, weights='uniform'

	precision	recall	f1-score	support
0	0.98	0.96	0.97	2886911
1	0.94	0.97	0.96	2113089
accuracy			0.96	5000000
macro avg	0.96	0.96	0.96	5000000
weighted avg	0.96	0.96	0.96	5000000

Training on Methodology 1

XGBoost:

An efficient algorithm. n_estimator = [120, 100, 75, 50], max_depth = [10, 7, 5], learning_rate = [0.1, 0.05, 0.01]

After finetuning, when the n_estimator = 75, max_depth = 10, learning_rate = 0.1, the accuracy of the accuracy is the best.

Fit

	precision	recall	f1-score	support
0	0.92	0.92	0.92	2886911
1	0.89	0.89	0.89	2113089
accuracy			0.91	5000000
macro avg	0.91	0.90	0.91	5000000
weighted avg	0.91	0.91	0.91	5000000

Partial Fit

	precision	recall	f1-score	support
1	0.60	0.15	0.24	3238488
2	0.76	0.96	0.85	8919972
accuracy			0.75	12158460
macro avg	0.68	0.56	0.54	12158460
weighted avg	0.72	0.75	0.69	12158460

Methodology 2

- Window Function: Creates a 'window' feature based on non-null events, assigning unique identifiers to each window, facilitating time-series analysis.
- Inactive Periods Removal: Identifies and eliminates inactive periods by computing differences in 'anglez' and filtering based on conditions, resulting in a DataFrame without inactive periods.
- Clustering: Applies KMeans clustering to 'anglez' and 'enmo' columns after scaling, providing labels indicating the cluster for each data point.
- Rolling Statistics: Computes 1-minute rolling std for 'anglez' and 'enmo,' and 2-minute rolling mean, filling missing values for robust time-series data.
- Scaling: Standardizes input data using StandardScaler, ensuring consistent scales for effective analysis and modeling.

Methodology 2

DecisionTree Classifier

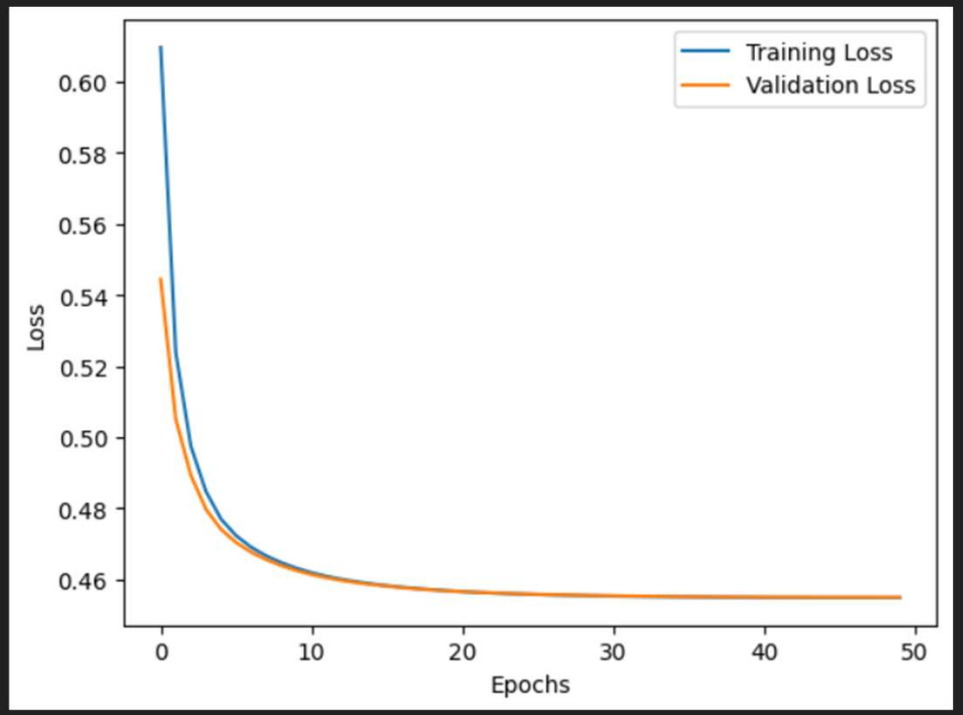
Parameters:criterion= Gini
max_depth=None,
min_samples_split=2,
min_samples_leaf=1

```
DecisionTreeClassifier: report
              precision    recall  f1-score   support

         0.0       0.89      0.88      0.88    960222
         1.0       0.85      0.86      0.85    753811

    accuracy                           0.87   1714033
   macro avg       0.87      0.87      0.87   1714033
weighted avg       0.87      0.87      0.87   1714033

DecisionTreeClassifier: 0.8530411801882274
CPU times: user 5min 9s, sys: 27.2 ms, total: 5min 9s
Wall time: 5min 9s
```

Decision Tree had the fastest convergence with a better accuracy.

Neural Network

1. The Neural Network.has 2 layers for faster training.
2. The model converges in 12 epochs and best F1 score is 85%.



2 Layers NN Classification Report:

	precision	recall	f1-score	support
Class 0	0.86	0.89	0.87	960222
Class 1	0.85	0.82	0.83	753811
accuracy			0.86	1714033
macro avg	0.86	0.85	0.85	1714033
weighted avg	0.86	0.86	0.86	1714033

Weighted F1 Score: 0.8571

Ensemble classifier for Methodology 2

Stacked classifiers:

- AdaBoost: 10 estimators
- GradientBoost: 30 estimators, min_samples_leaf=10
- Gaussian Naive Bias
- Logistic Regression
- Final Estimators: Decision Tree Classifier: Max_depth = 5

```
StackingClassifier: report
              precision    recall  f1-score   support

         0.0       0.86      0.90      0.88    960222
         1.0       0.86      0.81      0.84    753811

    accuracy                           0.86   1714033
   macro avg       0.86      0.85      0.86   1714033
weighted avg       0.86      0.86      0.86   1714033

StackingClassifier: 0.8361878718915041
CPU times: user 2h 18min 21s, sys: 43.2 s, total: 2h 19min 4s
Wall time: 2h 17min 40s
```

Model comparison

Method	Model Name	Validation F1 Score
Methodology 1	KNN	96
	XGBOOST	91
Methodology 2	Decision Tree	85.3
	Stacked: AdaBoost, GradientBoost, Logistic, GaussianNB.	83.6
	NN	85.71

Conclusion

- As the dataset size is big it takes a lot of time to train most of the models.
- The best model is KNN, just like what we predicted
- Partial fitting will influence the result, and may cause low metric. However, sometimes we have to use it cause the size of data
- Further there is a scoper for more data engineering steps, such as merging, averaging and KPI generation for improving the results.
- Also, larger models can be applied such as GRU to understand time sequence.