

AWS Forecast

予測精度を向上させよう！

オープントーン PARK HEE DONG

自己紹介



PARK HEE DONG

- 所属:株式会社OPENTONE
観光ビッグデータ事業部
- 好きなプログラミング言語:Python
- 趣味:旅行

観光ビッグデータ事業部では観光情報を扱うシステムを開発しています。

AWS Forecastを利用して観光施設の需要予測を行った経験があるので、その経験から得た知識を共有させていただきます。

目標

ターゲット

目標

- AWS Forecastを知る
- 関連付け予測を理解する
- 異常値の前処理を理解する

ターゲット

- AWS Forecastに興味あるエンジニアの方
- AWS Forecastを使っている方
 - 旧バージョンを使っている
 - 関連付け予測がわからない
 - 前処理をしたことがない

目次

- **AWS Forecast紹介**
 - AWS Forecastとは
 - Pythonを使った
AWS Forecast操作イメージ
 - 新バージョンAuto Predictor
- **予測精度を上げるコツ**
 - 関連付けで予測精度を向上
 - そもそも、前処理とは
 - 異常値を加工する前処理で
予測精度を向上

AWS Forecast 紹介

AWS Forecastとは

AWS ForecastはCSV形式の時系列データから需要予測を行うサービスです。

オンラインショッピングモールのAmazon.comでは、AWS Forecastと同じアルゴリズムを使って需要予測をしているそうです。

参照

https://docs.aws.amazon.com/ja_jp/forecast/latest/dg/what-is-forecast.html

Pythonを使った AWS Forecast操作 イメージ

Pythonのソースコードを見せながら、
Forecastの処理の流れを説明します。





Import your data

Datasets are required to train predictors, which are then used to generate forecasts.



Train a predictor

Train a predictor, a custom model with underlying infrastructure that Amazon Forecast trains on your datasets.



Generate forecasts

Generate forecasts using your trained predictors.



Import your data

観光施設の訪問客数データを
インポート



Train a predictor

機械学習で予測モデル作成
(特徴・パターン分析など)



Generate forecasts

予測実施

前提知識

- IAMロール:AWSリソース利用権限管理 (Forecast、S3)
- ARN:AWSリソースネーム (識別子)
- S3:AWSファイルストレージ
- Python 基本文法

boto3 (AWSライブラリ) インポート AWS Forecast用のIAMロールを準備

```
import boto3

forecast = boto3.client(service_name='forecast')

# roleはAWSのリソースの利用権限管理サービス
# s3 (AWSファイルストレージ) とforecastの権限を許可する
role_arn = "arn:aws:iam::xxxxxxx:role/SampleRole"
```

※細かい設定引数の説明は省略

AWS Forecast

データセットグループ

データセット

(空)

```
# データセットグループ作成
dataset_group_response = forecast.create_dataset_group(
    Domain="CUSTOM",
    DatasetGroupName="SampleDatasetGroupName",
    DatasetArns=[]
)
dataset_group_arn = dataset_group_response['DatasetGroupArn']

# データセット作成
response = forecast.create_dataset(
    Domain="CUSTOM",
    DatasetType="TARGET_TIME_SERIES",
    DatasetName="SampleDatasetName",
    DataFrequency="D", # 日単位
    Schema="TargetTimeSeriesDataSchema"
)
tts_dataset_arn = response['DatasetArn']

# データセットグループヘデータセットを設定
forecast.update_dataset_group(
    DatasetGroupArn=dataset_group_arn,
    DatasetArns=[tts_dataset_arn]
)
```

データセットをインポート

観光施設の客数データ

| item_id | timestamp | target_value |
|---------|------------|--------------|
| 客数 | 2022-05-01 | 100 |
| 客数 | 2022-05-02 | 81 |

AWS Forecast

データセットグループ

データセット

実績データ

実績データ



```
# データセットインポート
dataset_import_job_response = forecast.create_dataset_import_job(
    DatasetImportJobName="SampleDatasetImportJobName",
    DatasetArn=tts_data_arn,
    DataSource={
        "S3Config": {
            "Path": "s3://sample-bucket/Sample.csv", #
            "RoleArn": "arn:aws:iam::123456789012:role/SampleRole"
        }
    },
    TimestampFormat="yyyy-MM-dd"
)
dataset_import_job_arn = dataset_import_job_response['DatasetImportJobArn']
```

Predictor作成 (機械学習実施)

ForecastHorizon=10

ForecastFrequency="D"

→ 未来の10日を予測！

Train a predictor

機械学習で予測モデル作成
(特徴・パターン分析など)

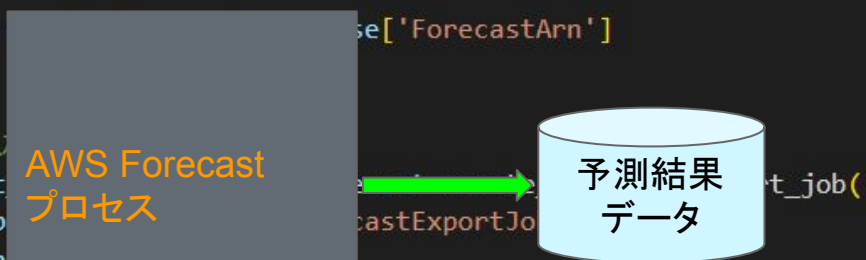
```
# Predictor作成 (機械学習)
create_predictor_response = forecast.create_auto_predictor(
    PredictorName="SamplePredictorName",
    ForecastHorizon=10,
    ForecastFrequency="D", # 日単位
    DataConfig = {
        'DatasetGroupArn': dataset_group_arn,
        'AttributeConfigs': [
            {
                'AttributeName': "target_value",
                'Transformations': {
                    "aggregation": "sum",
                    "frontfill": "none",
                    "middlefill": "zero",
                    "backfill": "zero"
                }
            }
        ],
        'AdditionalDatasets': [
            {
                'Name': 'holiday',
                'Configuration': {
                    'CountryCode': ['JP']
                }
            }
        ],
    }
)

predictor_arn = create_predictor_response['PredictorArn']
```

Forecast作成と 予測結果エクスポート

```
# Forecast作成 (予測実施)
create_forecast_response = forecast.create_forecast(
    ForecastName="SampleForecastName",
    PredictorArn=predictor_arn,
    ForecastTypes=['0.5']
)
forecast_arn = create_forecast_response['ForecastArn']

# 予測結果エクスポート
create_forecast_export_response = forecast.create_forecast_export_job(
    ForecastArn=forecast_arn,
    Destination={
        "S3Conf": {
            "Path": "s3://SampleBucket/Output",
            "RoleArn": role_arn
        }
    }
)
forecast_export_arn = create_forecast_export_response['ForecastExportJobArn']
```



The diagram illustrates the data flow from the AWS Forecast process to the prediction result data. A grey box labeled "AWS Forecast プロセス" (AWS Forecast Process) is connected by a green arrow to a light blue cylinder labeled "予測結果 データ" (Prediction Result Data). The cylinder is positioned to the right of the code block, and the arrow points from the process box to the data cylinder.

予測結果は
このような感じ

| | A | B | C |
|----|---------|----------------------|----------|
| 1 | item_id | date | p50 |
| 2 | 客数 | 2022-06-01T00:00:00Z | 181.4259 |
| 3 | 客数 | 2022-06-02T00:00:00Z | 132.0789 |
| 4 | 客数 | 2022-06-03T00:00:00Z | 92.56128 |
| 5 | 客数 | 2022-06-04T00:00:00Z | 95.16986 |
| 6 | 客数 | 2022-06-05T00:00:00Z | 86.69084 |
| 7 | 客数 | 2022-06-06T00:00:00Z | 90.992 |
| 8 | 客数 | 2022-06-07T00:00:00Z | 130.2134 |
| 9 | 客数 | 2022-06-08T00:00:00Z | 181.4482 |
| 10 | 客数 | 2022-06-09T00:00:00Z | 114.6722 |
| 11 | 客数 | 2022-06-10T00:00:00Z | 90.96508 |

AWS Forecast Handson

<https://pages.awscloud.com/rs/112-TZM-766/images/201912-AmazonForecastHandson.pdf>

boto3 Documents

<https://boto3.amazonaws.com/v1/documentation/api/latest/index.html>

実際に実装をされてみたい方は上記のリンクを参照してください。

AWS Forecast 新バージョン AutoPredictor

次は新バージョンのAutoPredictorについて紹介します。

新バージョン AutoPredictor

| | AutoPredictor |
|----------------|------------------------|
| 機械学習アルゴリズム | 自動で最適なアルゴリズムを組み合わせてくれる |
| 再トレーニング速度 | 向上 |
| 予測精度 | 向上 |
| 追加情報の影響分析 | あり |
| デフォルト(コンソール画面) | ○ |

旧バージョンでは同じ条件でも予測するたびに予測値のブレがありました。バージョンアップでブレがおさまった経験があります。お勧めは新バージョンです。

予測精度を上げるコツ

コツその1: 関連付けで予測!

コツその2: 前処理で予測!

関連付けで予測精度を 上げる

AWS Forecastは、関連する時系列データセットを使用して予測を行うことが可能です。

参照

https://docs.aws.amazon.com/ja_jp/forecast/latest/dg/related-time-series-datasets.html

| 日付 | 雨の日 |
|------------|-----|
| 2022/06/01 | 1 |
| 2022/06/03 | 0 |
| 2022/06/04 | 1 |

例えば、雨の日のスキー場を想像してみましょう。

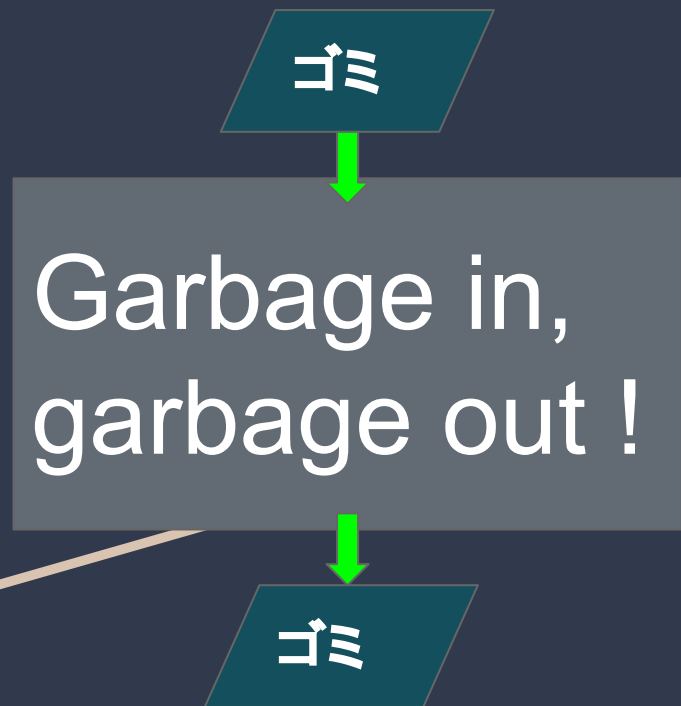
雨の日には、スキー場の客数は減るかも知れません。

天気予報を参照して、左の表のような雨の日のCSVデータを作成して、AWS Forecastに投入することで、予測精度を向上させることができます。

- ・雨が降った日 → 1
- ・雨が降らなかった日 → 0

こちらでは「雨の日」だけ設定したが、複数の項目を関連付けすることも可能です。

そもそも、 データ前処理とは？



参照:

https://en.wikipedia.org/wiki/Data_pre-processing

データ前処理は言葉の通り、何か主要な処理をする前に使おうとするデータを事前に加工する作業です。

前処理に関して

「**Garbage in, garbage out!**」という一般的な考え方があります。

無意味なデータを入力すると、無意味なデータが出力されるということです。

ですので、異常値を加工して予測精度を向上させる方法について説明します。

異常値を加工する前処理で予測精度を上げる

| 日付 | 客数 |
|------|-----|
| 6/1 | 56 |
| 6/2 | 15 |
| 6/3 | 0 |
| ... | ... |
| 6/30 | 89 |

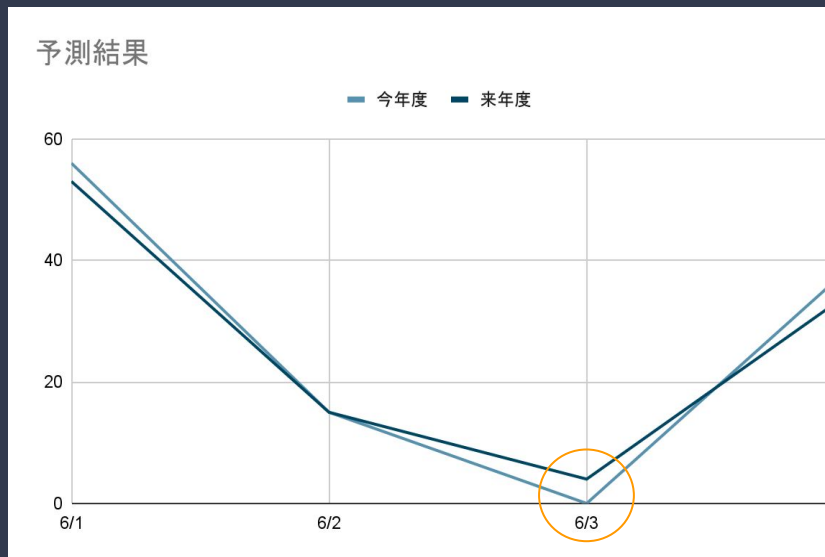
例えば左の表のような観光施設の日別の客数のCSVデータを、AWS Forecastに投入して、需要予測を行うとしましょう。

| 日付 | 客数 |
|------|-----|
| 6/1 | 56 |
| 6/2 | 15 |
| 6/3 | 0 |
| ... | ... |
| 6/30 | 89 |

ところで6月3日に大掃除をするために営業しませんでした。それで客数が0人です。

このままのデータを使用した場合、予測結果が異常値の影響を受けてしまいます。

コツその2:前処理で予測!



予測結果が出ました。

大掃除をした6/3の予測値が低くなっています。

このような場合、どんな対応ができるかを考えてみましょう。

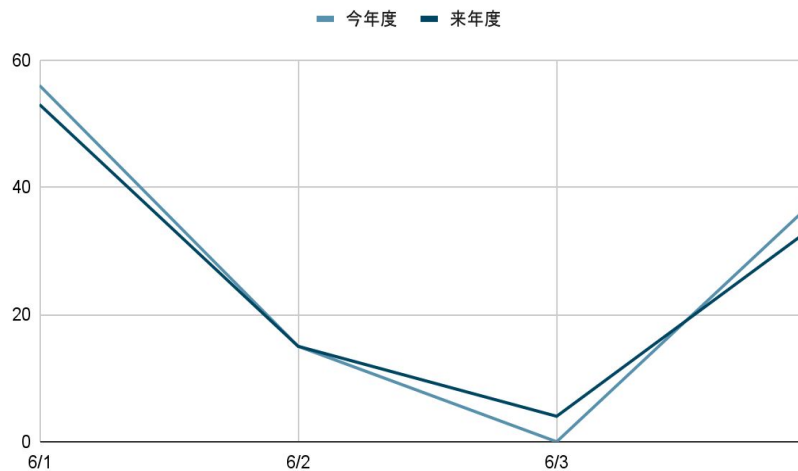
| 日付 | 実績値 | 加工値 |
|------|-----|-----|
| 6/1 | 56 | 56 |
| 6/2 | 15 | 15 |
| 6/3 | 0 | 24 |
| ... | ... | ... |
| 6/30 | 89 | 89 |

大掃除をしなかったら、どのぐらいのお客さまが来ただろうか、推算した**加工値**を作ります。推算方法として、例えば単純に去年のその日の値を使うこともできます。

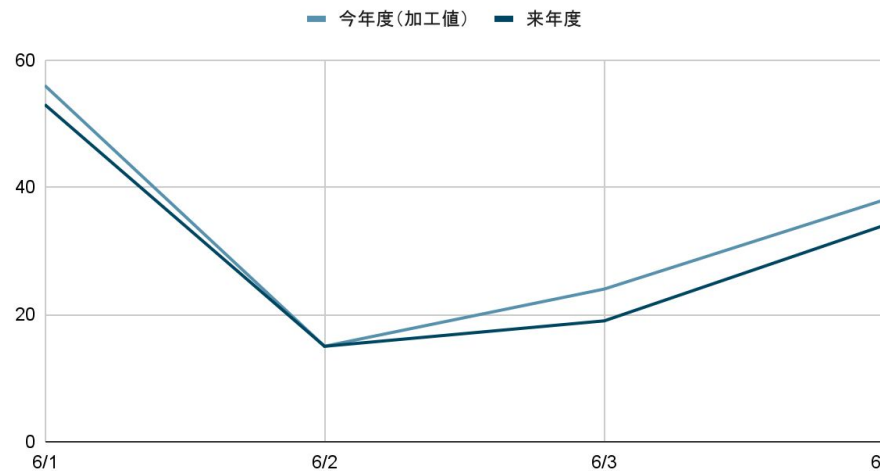
それでは、加工値を使って予測を行います。

コツその2: 前処理で予測!

生データを使った予測結果



加工データを使った予測結果



加工値を使ったら別の形のグラフが出ました。

左が生のデータを入れたとき、右が加工したデータを入れたときのグラフです。

ご覧のように、予測結果が変わるので、異常値の加工を検討する必要があると考えます。

おわりに

あくまでも予測は予測にすぎないので、関連付け予測や前処理をしたときにも、むしろ予測精度が下がる場合があるかも知れません。

オープトーンではAWS Forecastを利用した需要予測システムの開発で数回の実験と試行錯誤を通して予測精度をある程度上げることができました。

自分もいまだAWS Forecastを100%までは使いこなしていないと思います。もし、もっといい要領が見つかったら、共有してみるのはいかがでしょうか。

ご静聴ありがとうございました。

質疑応答