Shahid Beheshti University
Anita Soroush
98222085

assignment 3 of Machine Learning

Service 1: Interpolation

About My Implementation:

My implementation supports both shamsi and miladi dates (as “type” in config) and the intervals can be monthly or daily (as “time” in config). The only type of interpolation that my implementation supports is linear (as “interpolation” in config).
The input dates should be in string format and year, month and day should be separated by dash (“yyyy-mm-dd”) for example: “1390-10-02” or “2020-09-01”. **there are 2 samples of input and the related output in .\samples\service1**

Input example:

```
{
 "data": {
   "time": {
     "0": "2020-01-01",
     "1": "2020-01-02",
     "2": "2020-01-04"
   },
   "vol": {
     "0": 20,
     "1": 40,
     "2": 100
   }
 },
 "config": {
   "type": "miladi",
   "time": "daily",
   "interpolation": "linear"
 }
}
```

The related output:


```
{
   "time": {
       "0": "2020-01-01",
       "1": "2020-01-02",
       "2": "2020-01-03",
       "3": "2020-01-04"
   },
   "vol": {
       "0": 20.0,
       "1": 40.0,
       "2": 70.0,
       "3": 100.0
   }
}
```

Service 2: Interpolation (input in milai and the result in shamsi)

About My Implementation:

Everything is just the same as service 1 but the only difference is that there isn't any "type" in the config part and all the input dates should be in miladi.
There is one sample of input and the related output in .\samples\service2

Input example:


```
{
 "data": {
   "time": {
     "0": "2020-01-01",
     "1": "2020-01-02",
     "2": "2020-01-04"
   },
   "vol": {
     "0": 20,
     "1": 40,
     "2": 100
   }
 },
 "config": {
   "time": "daily",
   "interpolation": "linear"
 }
}
```

The related output:


```
{
    "time": {
        "0": "1398-10-11",
        "1": "1398-10-12",
        "2": "1398-10-13",
        "3": "1398-10-14"
    },
    "vol": {
        "0": 20.0,
        "1": 40.0,
        "2": 70.0,
        "3": 100.0
    }
}
```

Service 3: Outlier detector

About My Implementation:

- The input data **is time series**:

In the data part, "time" has the index role and the input dates should be in miladi and dash separated string format ("yyyy-mm-dd") for example: "2020-09-01".
The input dataset should have at least 24 rows and in the config part "time_series" should be set to true.
outliers will be detected using 2 methods:

- STL (Seasonal-Trend decomposition using Loess)
- Isolation Forest

- The input data **is not time series**:

In the data part, "id" has the index role (integers from 1 to rows.len).
In the config part "time_series" should be set to false.

outliers will be detected using 3 following methods and at the end the overall column will be filled by voting among these 3 methods. for each row of the dataset, if 2 methods out of 3 claim that the row is outlier then the related overall column will be filled with true and false if otherwise:

- Isolation forest
- Local outlier factor
- One class SVM

There are 2 samples of input and the related output in .\samples\service3

Input example 1:


```
{
 "data": {
   "time": {
     "0": "2020-01-01",
     "1": "2020-02-01",
     "2": "2020-03-01",
     "3": "2020-04-01",
     "4": "2020-05-01",
     "5": "2020-06-01",
     "6": "2020-07-01",
     "7": "2020-08-01",
     "8": "2020-09-01",
     "9": "2020-10-01",
     "10": "2020-11-01",
     "11": "2020-12-01",
     "12": "2021-01-01",
     "13": "2021-02-01",
     "14": "2021-03-01",
     "15": "2021-04-01",
     "16": "2021-05-01",
     "17": "2021-06-01",
     "18": "2021-07-01",
     "19": "2021-08-01",
     "20": "2021-09-01",
     "21": "2021-10-01",
     "22": "2021-11-01",
     "23": "2021-12-01",
     "24": "2022-01-01",
     "25": "2022-02-01",
     "26": "2022-03-01"
   },
   "feature": {
     "0": 4,
     "1": 50,
```

```
      "2": 20,
      "3": 10,
      "4": 20,
      "5": 5,
      "6": 3,
      "7": 1,
      "8": 1,
      "9": 3,
      "10": 2,
      "11": 60,
      "12": 40,
      "13": 70,
      "14": 30,
      "15": 400,
      "16": 30,
      "17": 50,
      "18": 60,
      "19": 70,
      "20": 1,
      "21": 2,
      "22": 100,
      "23": 20,
      "24": 4,
      "25": 20,
      "26": 100
    }
  },
  "config": {
    "time_series": true
  }
}
```


The related output:


```
{
   "time": {
       "0": "2020-01-01",
       "1": "2020-02-01",
       "2": "2020-03-01",
       "3": "2020-04-01",
       "4": "2020-05-01",
       "5": "2020-06-01",
       "6": "2020-07-01",
       "7": "2020-08-01",
       "8": "2020-09-01",
       "9": "2020-10-01",
       "10": "2020-11-01",
       "11": "2020-12-01",
       "12": "2021-01-01",
```

```
        "13": "2021-02-01",
        "14": "2021-03-01",
        "15": "2021-04-01",
        "16": "2021-05-01",
        "17": "2021-06-01",
        "18": "2021-07-01",
        "19": "2021-08-01",
        "20": "2021-09-01",
        "21": "2021-10-01",
        "22": "2021-11-01",
        "23": "2021-12-01",
        "24": "2022-01-01",
        "25": "2022-02-01",
        "26": "2022-03-01"
    },
    "STL": {
        "0": false,
        "1": false,
        "2": false,
        "3": false,
        "4": false,
        "5": false,
        "6": false,
        "7": false,
        "8": true,
        "9": false,
        "10": false,
        "11": false,
        "12": false,
        "13": false,
        "14": false,
        "15": false,
        "16": false,
        "17": false,
        "18": false,
        "19": true,
        "20": true,
        "21": false,
        "22": false,
        "23": false,
        "24": false,
        "25": false,
        "26": false
    },
    "IsolationForest": {
        "0": false,
        "1": false,
        "2": false,
```

```
        "3": false,
        "4": false,
        "5": false,
        "6": false,
        "7": false,
        "8": false,
        "9": false,
        "10": false,
        "11": false,
        "12": false,
        "13": false,
        "14": false,
        "15": true,
        "16": false,
        "17": false,
        "18": false,
        "19": false,
        "20": false,
        "21": false,
        "22": false,
        "23": false,
        "24": false,
        "25": false,
        "26": false
    }
}
```


Input example 2:


```
{
 "data": {
   "id": {
     "0": 1,
     "1": 2,
     "2": 3,
     "3": 4,
     "4": 5,
     "5": 6
   },
   "feature": {
     "0": 100,
     "1": 1,
     "2": 35,
     "3": 67,
     "4": 89,
     "5": 90
   }
 },
 "config": {
   "time_series": false
 }
}
```

The related output:


```
{
    "id": {
        "0": 1,
        "1": 2,
        "2": 3,
        "3": 4,
        "4": 5,
        "5": 6
    },
    "IsolationForest": {
        "0": false,
        "1": true,
        "2": false,
        "3": false,
        "4": false,
        "5": false
    },
    "LocalOutlierFactor": {
        "0": false,
        "1": false,
        "2": false,
        "3": false,
        "4": false,
        "5": false
    },
    "OneClassSVM": {
        "0": false,
        "1": true,
        "2": false,
        "3": true,
        "4": false,
        "5": false
    },
    "OverallResult": {
        "0": false,
        "1": true,
        "2": false,
        "3": false,
        "4": false,
        "5": false
    }
}
```

Service 4: Imbalanced data handling

About My Implementation:

The data part has 3 columns which are "id" (integers from 1 to rows.len), "feature1" (integer) and "class" (1 or 0)
The config part has only 1 parameter which is "method" and can be filled with "under_sampling", "over_sampling" or "SMOTE".
There are 3 samples of input and the related output in .\samples\service4

Input example:


```
{
 "data": {
   "id": {
    "0": 1,
     "1": 2,
     "2": 3,
     "3": 4,
     "4": 5,
     "5": 6,
     "6": 7,
     "7": 8,
     "8": 9,
     "9": 10,
     "10": 11,
     "11": 12,
     "12": 13,
     "13": 14,
     "14": 15,
     "15": 16,
     "16": 17,
     "17": 18,
     "18": 19,
     "19": 20,
     "20": 21,
     "21": 22,
     "22": 23,
     "23": 24,
     "24": 25,
     "25": 26,
     "26": 27
   },
```

```
"feature1": {
  "0": 20,
  "1": 10,
  "2": 50,
  "3": 3,
  "4": 200,
  "5": 300,
  "6": 300,
  "7": 2,
  "8": 78,
  "9": 389,
  "10": 456,
  "11": 356,
  "12": 456,
  "13": 89,
  "14": 23,
  "15": 2,
  "16": 34,
  "17": 45,
  "18": 67,
  "19": 45,
  "20": 567,
  "21": 33,
  "22": 333,
  "23": 345,
  "24": 34,
  "25": 45,
  "26": 56
},
"class": {
  "0": 0,
  "1": 1,
  "2": 0,
  "3": 1,
  "4": 1,
  "5": 1,
  "6": 0,
  "7": 0,
  "8": 0,
  "9": 0,
  "10": 1,
  "11": 1,
  "12": 1,
  "13": 1,
  "14": 1,
  "15": 1,
  "16": 0,
  "17": 0,
```

```
      "18": 1,
      "19": 1,
      "20": 0,
      "21": 1,
      "22": 0,
      "23": 1,
      "24": 0,
      "25": 1,
      "26": 1
    }
  },
  "config": {
    "method": "over_sampling"
  }
}
```


The related output:


```
{
   "id": {
       "0": 1,
       "1": 2,
       "2": 3,
       "3": 4,
       "4": 5,
       "5": 6,
       "6": 7,
       "7": 8,
       "8": 9,
       "9": 10,
       "10": 11,
       "11": 12,
       "12": 13,
       "13": 14,
       "14": 15,
       "15": 16,
       "16": 17,
       "17": 18,
       "18": 19,
       "19": 20,
       "20": 21,
       "21": 22,
       "22": 23,
       "23": 24,
       "24": 25,
       "25": 26,
       "26": 27,
       "27": 28,
```

```
    "28": 29,
    "29": 30,
    "30": 31,
    "31": 32
},
"feature1": {
    "0": 20,
    "1": 10,
    "2": 50,
    "3": 3,
    "4": 200,
    "5": 300,
    "6": 300,
    "7": 2,
    "8": 78,
    "9": 389,
    "10": 456,
    "11": 356,
    "12": 456,
    "13": 89,
    "14": 23,
    "15": 2,
    "16": 34,
    "17": 45,
    "18": 67,
    "19": 45,
    "20": 567,
    "21": 33,
    "22": 333,
    "23": 345,
    "24": 34,
    "25": 45,
    "26": 56,
    "27": 2,
    "28": 78,
    "29": 50,
    "30": 20,
    "31": 333
},
"class": {
    "0": 0,
    "1": 1,
    "2": 0,
    "3": 1,
    "4": 1,
    "5": 1,
    "6": 0,
    "7": 0,
```

```
        "8": 0,
        "9": 0,
        "10": 1,
        "11": 1,
        "12": 1,
        "13": 1,
        "14": 1,
        "15": 1,
        "16": 0,
        "17": 0,
        "18": 1,
        "19": 1,
        "20": 0,
        "21": 1,
        "22": 0,
        "23": 1,
        "24": 0,
        "25": 1,
        "26": 1,
        "27": 0,
        "28": 0,
        "29": 0,
        "30": 0,
        "31": 0
    }
}
```