ML on FHIR

Release 0.0.1

Jul 26, 2023

Development

CHAPTER 1

Installation

TODO

CHAPTER 2

Getting Started

2.1 Connecting to a FHIR Server

To connect to a FHIR server, create a `FHIRClient` object and provide its `BaseURL`:

```
from fhir_client import FHIRClient
client = FHIRClient(service_base_url='https://r3.smarthealthit.org')
```

The server's compatibility statement is queried to determine whether the connection was sucessful established.

2.2 Get an Overview of your Data

Before querying patients that belong to a specific cohort, we can get an overview of available **procedures** and via:

```
import pandas as pd
procedures = client.get_all_procedures()
pd.DataFrame([prod.code['coding'][0] for prod in procedures]).drop_duplicates().sort_
↪values(by=['display']).head()
```

This might take a while but will give you an overview of available procedures. E.g.

ID	code	display	system
893	183450002	Admission to burn unit	http://snomed.info/sct
83	305428000	Admission to orthopedic department	http://snomed.info/sct
13687	35637008	Alcohol rehabilitation	http://snomed.info/sct

Similarily, we can receive a list of available **conditions** via:

```
conditions = client.get_all_conditions()
pd.DataFrame([cond.code['coding'][0] for cond in conditions]).drop_duplicates(subset=[
↪'display']).sort_values(by='display', ascending=True).head()
```

ID	code	display	system
488	30473006	Abdominal pain	http://snomed.info/sct
140	102594003	Abnormal ECG	http://snomed.info/sct
6801	26079004	Abnormal involuntary movement	http://snomed.info/sct

2.3 Query Patients

With a list of available conditions we can query patients for which a certain condition was diagnosed. To do so we can either use the code of a coding nomenclature (e.g. *SNOMED*) or its readable name:

```
patients_by_condition_text = client.get_patients_by_condition_text("Abdominal pain")
patients_by_procedure_code = client.get_patients_by_procedure_code("http://snomed.
↪info/sct","73761001")
```

2.4 Machine Learning

TODO

2.4.1 Server Interoperability

ObservationProcessors

To use patient features in a machine learning task, we will extract them from the FHIR Observation Resource. Depending on how your server is set up, the way we can extract desired observation values might differ. In the following example we will use LOINC codes to extract the latest **BMI measurement** of a patient.

```
1  from ml_on_fhir.preprocessing import AbstractObservationProcessor, get_coding_
   ↪condition
2
3  class ObservationLatestBmiProcessor(AbstractObservationProcessor):
4          """
5          Class to transform the FHIR observation resource with loinc code 39156-5 (BMI)
6          to be usable as patient feature.
7          """
8          def __init__(self):
9                  super().__init__('bmiLatest')
10
11         def transform(self, X, **transform_params):
12                 conditions = get_coding_condition([{'system': 'http://loinc.org',
13                                                     'code': '39156-5'}])
14                 bmis = list(filter(conditions, X))
15                 bmis = sorted(bmis, reverse=True)
16                 if len(bmis) >= 1:
17                         return self.patient_attribute_name, float(bmis[0].
   ↪valueQuantity['value'])
18                 else:
19                         return self.patient_attribute_name, 0.0
```

In line 9, we define the name of the feature, so we can use it as if it was a patient attribute. In line 12, we define the conditions that an observation needs to fulfill to be considered. An observation has to fulfill **all** conditions (e.g. coding system has to be `http://loinc.org` **and** its code must be `39156-5`). In line 13, we apply the conditions on all observations of a given patient (`X`). Now, we reversely sort the `Observation` objects that are left. Observation objects are always sorted by FHIR's `effectiveDateTime` attribute. Finally, we use the FHIR quantity datatype to return the patient's latest BMI measurement.

We can now use the new `bmiLatest` feature in a `MLOnFHIRClassifier` after registering in with the `FHIRClient`.

```python
from ml_on_fhir.fhir_client import FHIRClient

client = FHIRClient(service_base_url='https://r3.smarthealthit.org')
client.preprocessor.register_preprocessor(ObservationLatestBmiProcessor)
ml_fhir = MLOnFHIRClassifier(Patient, feature_attrs=['bmiLatest'],
                             label_attrs=['gender'], preprocessor=client.preprocessor)
```

Note that some patient attributes like `gender` are already provided through a similar mechanism with `PatientProcessors`. Read more about them here.

PatientProcessors