Data Processing ¶

Created By: Aldo Iturrios and Alyssa Wisk

In this notebook, we'll be reading in all txt files, and stacking each file into one single (rec
dataframe).

In [1]:
```
import numpy as np
import pandas as pd
import os
from sklearn.linear_model import LinearRegression
```

Below, we'll be stacking all txt files into one single dataframe. There are two other import
happening in the for loop below:

- If a patient has more than 1 measurement recorded for a single measure in a one ha
interval, we take the average of all those measurements in that half hour interval anc
average. Ex. If in hour 1:30, patient had blood pressure taken 3 times, we take the a
those 3 measurements, and that is the number that ends up in the data frame for tha
- We will also be taking a linear regression for each measure (across all time) for each
Coefficent associated with this measure will serve as a summary of the trend for tha
over time.

In [2]:

```
# Set folder where all data is stored
folder = "data/x_all/"

# Construct empty dataframe to store all data
train_data = pd.DataFrame()

for i, patient in enumerate(os.listdir(folder)):

    # Foundation
    patient_id = patient[:-4]
    file = os.path.join(folder, patient)
    df = pd.read_csv(file, delimiter = ",", dtype={'Time': str

    # Regression for each variable
    linear_regressor = LinearRegression()
    df_reg = df.copy().dropna()
    df_reg['Time2'] = df_reg['Time'].apply(lambda x: int(x[0:2

    vars = df_reg.Variable.unique()
    dict_reg = {}

    for v in vars:
        data = df_reg[df_reg['Variable'] == v]
        try:
            reg = linear_regressor.fit(data['Time2'].to_numpy(
            dict_reg[v] = reg.coef_
        except:
            pass
            dict_reg[v] = 0



    # Form for final data
    df1 = df.melt(id_vars=['Variable', 'Time'], value_vars=['V
    df1['Feature'] = df1['Time'] + "_" + df1['Variable']
    df2 = df1.groupby('Feature')['value'].mean().to_frame().T
    df2 = df2.set_index(pd.Series([patient_id]))

    patient_data = pd.concat([df2, pd.DataFrame(dict_reg).set_

    if i == 0:
        train_data = patient_data
    else:
        train_data = train_data.append(patient_data, sort=Fals
```

In [6]:
```
train_data.head(5)
```

Out[6]:

	00:00_AdmissionType	00:00_Age	00:00_Gender	00:00_RecordID	00:30_GCS	00:30_
3644	3.0	42.0	0.0	3644.0	15.67	8
5235	3.0	59.0	1.0	5235.0	NaN	10
1053	1.0	59.0	1.0	1053.0	NaN	
8711	3.0	57.0	1.0	8711.0	NaN	
7422	3.0	58.0	1.0	7422.0	6.11	6

5 rows × 3525 columns

Further adjustments

- Remove Time stamp from 4 initial variables
- Name index into "id" (to match train_outcome.csv file)
- Order columns

In [4]:
```
train_data_dc = train_data.copy()
train_data_dc = train_data_dc.drop(columns=["AdmissionType", "
train_data_dc = train_data_dc.sort_values(by = ["00:00_RecordI
train_data_dc = train_data_dc.rename(columns={"index": "id",
                                "00:00_AdmissionType": "Admissic
                                "00:00_Age": "Age",
                                "00:00_Gender": "Gender",
                                "00:00_RecordID": "RecordID"})
```

In [5]:
```
train_data_dc = train_data_dc.reindex(sorted(train_data_dc.col
cols_at_beg = ["id", "AdmissionType", "Age", "Gender", "Record
train_data_dc = train_data_dc[[c for c in cols_at_beg if c in
                               + [c for c in train_data_dc if c
train_data_dc.sort_values(by = ["RecordID"])
train_data_dc.head(5)
```

Out[5]:

	id	AdmissionType	Age	Gender	RecordID	00:00_ALP	00:00_ALT	00:00_AST	00:00_
0	1	4.0	64.0	1.0	1.0	NaN	NaN	NaN	
1	2	2.0	76.0	1.0	2.0	NaN	NaN	NaN	
2	3	4.0	65.0	0.0	3.0	NaN	NaN	NaN	
3	4	4.0	44.0	0.0	4.0	NaN	NaN	NaN	
4	5	3.0	48.0	1.0	5.0	NaN	NaN	NaN	

5 rows × 3522 columns

Save Data Frame

In [7]:
```
1 train_data_dc.to_csv("data/patient_data/patient_dataframe.csv"
```

In [ ]:
```
1
```

In [ ]:
```
1
```