We first connect to our PSQL, take the data we want in our DataFrame.
We pick gender, age, max and min temperature, total snow and rain mm, and city as our features. Because gender and age will affect individual body fitness, temperature and mm also has a great influence on the body, city decide the medical level. I think it is reasonable.

handling missing values:

Then we check how many NaN we have and decide a way to fill them. We use X.isnull().sum() and find we have 2692 cases with NULL value.

We also want to know how many cases we have, using len(X) and check out we have 107349 cases in our DataFrame, it means we can just drop the null cases since they are in a small group. Another reason is the null value is max and min temperature, if we fill them by mean, or media, we think it affect our model. So we use X.dropna() to drop all the null cases.

data summarization using histograms:

To have a "feel" of the data, we generate histograms for all the selected features.

handling categorical attributes:

Then we use X.get_dummies function to do the one-hot recording for the categorical data, here we recording the age group, gender and city.

Normalization of numeric attributes:

I used X.apply(lambda x:(x-x.min(axis=0))/(x.max(axis=0)-x.min(axis=0))) for the normalization.
So whole the features will in the same range 0 to 1.

feature selection:

We take out the label fatal as y. And the remind of the data frame as X.

undersampling of the majority class(es):

Using train_test_split to divide the data into 4 part.

We check the y_train and y_test

```
Training set Counter({0: 82527, 1: 1196})
Test set Counter({0: 20643, 1: 288})
```

We find they are high unbalance.
Using Nearniss() function to fit X_train and y_train

We get a new y_train_ns

```
0    1196
1    1196
Name: fatal, dtype: int64
```

By under-sampling, that is how we preprocessed the data