Project 2

MPCS 51042 - Python Programming - Fall 2021

Deadline: December 7th at 5:00 pm CDT

In this assignment, you will be working with data in NumPy. You will perform some simple modeling and analysis, display your results using Matplotlib, and discuss your findings in a Jupyter Notebook. This will give you experience with NumPy and more practice using classes and functions to support code reuse.

You need to turn in the following files for this assignment:

- `time_series.py`: A Python module with class and function defintions for Part 1.
- `time_series.ipynb`: A Jupyter Notebook that runs your Part 1 code, displays results, and where you discuss your findings.
- `prediction_data.py`: A Python module with class and function definitions for Part 2.
- `prediction_data.ipynb`: A Jupyter Notebook that runs your Part 2 code, displays results, and where you discuss your findings.

Additional requirements are:

- You may not use Pandas or Seaborn for this assignment.
- Your Python modules, `time_series.py` and `prediction_data.py`, must be documented with NumPy, Google, or Sphinx style docstrings. They must also contain type annotations.
- Your Jupyter Notebooks should be well documented, but they do not need to have specific docstring styles or contain type annotations.

Data sets

In this assignment, you will work with two different data sets.

- In Part 1, you will work with time series data. Time series data is data that is tracked over time. With this kind of data, you can use history to predict the future.

  One example of time series data is the "Libraries - WiFi Usage" data set, which tracked the number of WiFi sessions per month at Chicago Public Library locations from 2011 to 2014. You can find this data set on Canvas.
- In Part 2, you will work with prediction data. Prediction data is tabular data (e.g., looks like a table), where each row is a *sample* and each column is a *feature*. Features are used to predict a *target* column. For example, in a study to predict California house pricing, each sample unit is a house, and each house is represented by features such as house age, number of bedrooms, etc. You can predict a house's price (target) based on the house age or number of bedrooms (features). You can find this data set on Canvas.

You are welcome to pick out different data sets that are of interest to you. The City of Chicago maintains many data sets on its Chicago Data Portal website: https://data.cityofchicago.org/. Kaggle, a machine learning and data science community, also hosts data sets, too: https://www.kaggle.com/datasets. You are not restricted to these two websites. Your prediction data set should contain at least three features and a target. Both data sets should contain at least 20 samples (rows).

1 Part 1: Time Series Data

In this part, you will implement a `TimeSeries` class that encapsulates the file parsing, computation, and plotting described below. When you are implementing your class, think carefully about its design. What should be stored as an attribute of the class? What should be implemented as a method?

1.1 Plot the time series data

Load your data into a NumPy array and plot it over time. Your plot should be clearly labeled.

1.2 Compute the moving average

Constructing the moving average is a common type of smoothing that is often applied to time series data. Say your time series data set has N data points:

$$y_0, y_1, y_2, \ldots, y_{N-1}.$$

We will refer to the whole series as y. The moving average of each time step n of order m is given by:

$$\bar{y}_n = \frac{1}{m} \sum_{i=-k}^{k} y_{n+j}$$

where $m = 2k + 1$. Compute the moving average of order m of your time series data set. To achieve this, you will construct another NumPy array with moving average values:

$$\bar{y}_k, \bar{y}_{k+1}, \ldots, \bar{y}_{N-1-k}.$$

We will refer to the whole moving average series as $\bar{y}$. Plot it with the original data. Be careful to line up the time steps correctly!

1.3 Fit the moving average to a linear model

In this task, you will use linear regression to fit the moving average to a linear model. This model can be used to predict the value of a data point y_n beyond the time steps included in your data set (i.e., in the future).

1.3.1 Model 1

Perform linear regression to compute β in order to attain the linear model:

$$\bar{y} = \beta n$$

where $\bar{y}$ is the moving average time series and n is a time step. You can use NumPy's `linalg.lstsq` function to perform the linear regression.

1.3.2 Model 2

Notice that in your first model, the y-intercept of the linear model is fixed at zero. This is not necessarily realistic for your data set. Use linear regression and find β_0 and β_1 such that:

$$\bar{y} = \beta_0 + \beta_1 n.$$

Again, you can use NumPy's `linalg.lstsq` function to perform the linear regression.

1.4 Determine which model is better

In order to judge the effectiveness of a model, you can compare the actual data points (y_n) to what is predicted by your linear model (we'll call those $\hat{y}_n$).

To find $\hat{y}_n$, you will need to apply the values of β to the time steps n. For example, to compute the $\hat{y}_n$ values for Model 1, you will need to find:

$$\hat{y}_n = \beta n$$

for all $n \in \{0, \dots, N-1\}$.

To compute the $\hat{y}_n$ values for Model 2, you will need to find:

$$\hat{y}_n = \beta_0 + \beta_1 n$$

also for all n.

One common way to evaluate a model is to compute the sum of squared residuals:

$$R = \sum_{n=0}^{N-1} (y_n - \hat{y}_n)^2$$

The better model (Model 1 or Model 2) is the one that minimizes this value.

1.5 Execute each task in `time_series.ipynb`

- Start by describing your data set. What data does it contain? What is the source of the data? What makes this data interesting to you?
- Next, execute each task outlined above and describe the output. For example, does the plot of your time series data look as you expected? Does it follow a general upwards or downwards trend? Do you see any other patterns in the data? Which model performed better?
- Compute the moving average with $m = 7$ and plot it with the original data. How does the moving average change when you modify m?
- Use your linear models to predict the data point y_N. Do you think these are reasonable predictions? What if you used your model to forecast even further into the future, e.g., y_{N+10}? Note that we will not be grading you on whether a linear model correctly models your data set. What would make the model better?

2 Part 2: Prediction Data

In this part, you will implement a `PredictionData` class that encapsulates the file parsing, computation, and plotting described below.

2.1 Plot the prediction data

Load your data into a NumPy array and create a scatter plot that shows the relationship between one feature of your data set and the target. Your plot should be clearly labeled.

2.2 Standardize the data

When working with data, it is common to *standardize* the data so that it has a mean of zero and a standard deviation of one. Standardize each feature of your prediction data set, so that each feature individually has a mean of zero and a standard deviation of one. Standardize the target, too.

2.3 Find the best model

Like you did in Part 1, you will use a linear model to model the relationship between a feature and the target. Say your data set has N samples (rows), x is one feature (column), and y is the target. Use linear regression to find β_0 and β_1 in the following:

$$y = \beta_0 + \beta_1 x.$$

Again, you will use the sum of squared residuals to determine which feature out of all the possible features in your data set best models the target:

$$R = \sum_{n=0}^{N-1} (y_n - \hat{y}_n)^2$$

The best model is the model that minimizes R. Like before, you will need to compute a model's predictions $\hat{y}_n$ before computing its R value. You can compute the $\hat{y}_n$ values for a model by finding:

$$\hat{y}_n = \beta_0 + \beta_1 x_n$$

for all $n \in \{0, \ldots, N-1\}$.

2.4 Execute each task in `prediction_data.ipynb`

- Again start by describing your data set. What are the features of your data set? What is the target? Why do you think these features would be good at predicting the target? What is the source of the data? What makes this data interesting to you?
- Next, execute each task outlined above and describe the output. For example, create scatter plots with different features of your data set. Do your scatter plots show a correlation between the features and the target? Does one feature show a higher correlation than another? Does the "best" model make sense intuitively?

Optional Extra Credit

For 10 extra credit points (out of 200 points possible), you can pick your own data set and answer an additional question about it (beyond what is asked of you above). The question should be of interest to you and should require some computation to answer. You will present your data set and extra question in class during Week 9. You do not have to have implemented your solution yet, you just need to describe the data set and extra question.