Big Data Management Assignment 1

Jiashuo Li jial@itu.sk

10

Overview

In this report I divided into three parts. In the first part I will give you a bird peek of my predict model, while the second part would be some topics I'd like to share and explain, and in the last part I will explain the important questions that are asked in the assignment file.

System Modules

My system mainly consists of three parts: preprocessing, pipeline-training, and future prediction.

First of all, I retrieve the most recent data (90days) from the database. And the instantaneous power consumption is stitched together with the corresponding wind data to form the initial data set.

Afterwards, in the pipeline, I split the dataset into 5 parts using time series segmentation, train and validate them in a constant forward chronological stepwise manner, and repeat the above steps using different models.

Finally, in the prediction part, I tested the different models individually on their predictions of the wind power function for the next 7 days of weather forecast information, calculating the error from the real statics. The test results prove that my models are able to predict the power relatively accurately.

Key Points

1. **Reads the latest data from the InuxDB:**
   I use SQL language to search and load the data from 90 days before to now.
2. **Prepares the data for model training:** ?
   I choose to drop the records that has none values and concat the two table into one single dataframe.
3. **column transformer**: I use min-max Normalization to Normalize the column which represent the wind speed and wind direction, fin order to transform features to be on a similar scale.
4. **Trains a (few) regression model(s) of your choice:**
   I choose several models and train them for predicting the most recent weeks power, and below is the error results.

```
y_pred_linear
Mean Square Error: 16.76
Root Mean Square Error: 4.09
y_pred_SVR
Mean Square Error: 12.33
Root Mean Square Error: 3.51
y_pred_tree
Mean Square Error: 9.4
Root Mean Square Error: 3.07
y_pred_random_forest
Mean Square Error: 8.7
Root Mean Square Error: 2.95
y_pred_KNN
Mean Square Error: 12.44
Root Mean Square Error: 3.53
y_pred_NN
Mean Square Error: 13.98
Root Mean Square Error: 3.74
```

This needs interpretation. What does RMSE = 2.95 mean in this context? Is it high/low compared to the energy production?

5. **Saves the best performing model to disk**
   I use time split to split the data in a “Walk-Forward” validation, and compare the predict scores using 10 days ago trained models with the most recent one, which I find the most recent on has a better performance. But this is not always true because if the newly coming data is full of noise and has very bad quality, it could affect the model that lead to a bad trained one.
6. **Compares the newly trained model with the currently saved model, and picks the best performing model**
7. **Saves the best performing model to disk** ← not clear if you did this
8. **Uses the current best model to forecast future (wind) power production**

Critical Questions

1. **Pipeline details:** My pipeline consists of two parts: the data transformation of the columns and the declaration of the model. I use the pipeline .fit for model training, pipeline.predict for prediction. ✓
2. **The input data format:** The inputs to the model are dataframes after concated.
3. **How to align the data:** I took the energy generation data at three hour intervals by downsampling and combined it with the wind data, I think it's more accurate to downsample than upsample because we already have a relatively rich amount of data.
4. **Type of model and hyperparameter:** I choose linear regression, SVM, decision tree, random forest, KNN, and also neural networks to train and predict, and use grid search to tune the parameters.
5. **Compare the newly trained model with stored version:** I use time split to split the data in a “Walk-Forward” validation, and compare the predict scores using 10 days

ago trained models with the most recent one, which I find the most recent on has a better performance. But this is not always true because if the newly coming data is full of noise and has very bad quality, it could affect the model that lead to a bad trained one.

the data is not high-dim?

6. **How could the system be improved:** I think the pipeline could also be added some more features and functions, for example use PCA to reduce the data dimensions.
7. **Wind direction:** By comparing a model trained with wind direction with a model trained without wind direction, the former can show better results and I therefore consider wind direction to be a useful feature. In addition to this I have converted the wind direction from a degree to a two-dimensional vector, as I believe that the onehot coding method introduces too much redundant information.
8. **90 days' interva problem:** In fact I have found experimentally that using a longer time interval, for example 180 days, does not perform any better in predicting wind power than using 90 days, and I think there are two reasons to explain this: First, with longer time period we have more outliers, which could affect the model precision; Second, the relationship between wind and power consumption may vary continuously from season to season, so it is not reasonable to use spring data, which is out of date and useless, to train and predict power generation in autumn.
9. **Tradeoff between time period length and test result:** Longer periods of time result in more data, but they also lead to more outliers and the impact of greater historical information that is already out of date. According to the analyse before, increasing the amount of data does not necessarily increase prediction accuracy.

You need to explain more clearly how you transformed the data, when you split into train and test, how you used the pipelines.
The report should be self-explanatory. For the next ones please include some plots, some important parts of the code etc.