Data Science with Spark

X – ECE – Telecom ParisTech – ENSAE – Paris SUD

ING5

Data Science

2017/2018

Machine Learning with Spark

Salim NAHLE

Organization:

- You can work on any Spark environment
- A **PDF report** is expected. It shall contain the code, explanations and necessary screenshots. Alternatively, you can use your notebook to generate an **HTML report**.
- Please work in **pairs**! Each group (composed of 2 persons at most) shall submit one report. Do not forget to indicate your names in the report.
- The report shall be uploaded on the campus page before **Monday 18/12/2017 midnight.**
- Late reports will be penalized (3 points/day)

Abstract:

- The objective of this mini-project is to use the different Spark machine learning libraries to build predictive models.
- Two open data sets are provided. In both cases, the correct answers are given. Supervised learning algorithms are thus used.
- In the first data set, the output is continuous, you shall build several regression models, tune them and compare them
- In the second, the output is discrete. You shall build several classifiers, tune and compare them

I. Part I: Combined Cycle Power Plant Data Set

1. Understanding the dataset

The first step for building a machine learning application is understanding the data and its application domain.

For this first data set, we want to predict the power output of a gas-fired power generation plant based on the readings from various sensors.

Peaking power plants are plants that supply power only occasionally. i.e. when there is high demand (peak demand) for electricity. Hence, the generated electricity is much more expensive than base load power plants. So it is important to understand and predict the power output to manage the plant connection to the power grid.

More information in Peaking power plants: https://en.wikipedia.org/wiki/Peaking_power_plant

a) Data Set Variables

Features consist of hourly average ambient variables

- Temperature (T) in the range 1.81°C and 37.11°C,
- Ambient Pressure (AP) in the range 992.89-1033.30 millibar (mbar),
- Relative Humidity (RH) in the range 25.56% to 100.16%
- Exhaust Vacuum (V) in the range 25.36-81.56 cm Hg
- Net hourly electrical energy output (PE) 420.26-495.76 MW The averages are taken from various sensors located around the plant that record the ambient variables every second. The variables are given without normalization.

This is a supervised machine learning problem since we have a labelled data set. The data set is taken from UCI Machine Learning Repository Direct link: https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant

For learning purposes, only a subset of this data set is used in this project. Moreover, some values are intentionally removed. You need hence to deal with missing data problem.

b) References

- Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
- Pınar Tüfekci, Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods, International Journal of Electrical Power & Energy Systems, Volume 60, September 2014, Pages 126-140, ISSN 0142-0615

- Heysem Kaya, Pınar Tüfekci , Sadık Fikret Gürgen: Local and Global Learning Methods for Predicting Power of a Combined Gas & Steam Turbine, Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE 2012, pp. 13-18 (Mar. 2012, Dubai)

2. Consulting project

You have been contacted by power plant operators to help them build a predictive model to predict the output power (PE). They provided you data with the features described above:

- AT = Atmospheric Temperature in C
- V = Exhaust Vaccum Speed
- AP = Atmospheric Pressure
- RH = Relative Humidity
- PE = Power Output

9568 data points are given. A lot of values, however, are missing. The missing values include features but also the labels (PE).

As any real world project, there are no 100% correct answers, you have to try your best to build the best model!

a) Part 1: Data preparation

The first step hence consists in understanding the data, finding relationships between different features, getting some insights from them, treating missing data. **Your objective when handling missing data shall be to maintain and use as much useful data as you can.**

- Please include in your report, the steps and the necessary arguments and explanations of decisions you made in this preliminary data preparation stage.
- Join also a screen shot of the clean data you used in your pipeline as well as the original row data:
  Hint use [rowData].describe().show() and [cleanData].describe().show()

b) Part 2: Build you linear regression model

You shall train and evaluate several regression models Linear Regression, Ridge (L2) and Lasso (L1). To find the best model you shall also tune your model by varying the hyper parameters you judge the most relevant

- Again, you shall justify all the choices

Suggested steps:

- Use the method **randomSplit** of the class DataFrame (pyspark.sql.dataframe) to split your data as follows:
  - 30% for the test set
  - 70% for the training set

- Start by fitting a LinearRegression model and evaluate it.
- Then tune over the regularization parameter and possibly other parameters. Use RMSE (Root-Mean-Square Error Metric) as evaluation metric to evaluate the predictions on the test data set.
    - Hint: in this part you have to use a CrossValidator
    - Set its estimator (your linear regression model)
    - Set its evaluator (a RegressionEvaluator instance with RMSE as metric)
    - And finally set its estimator parameter maps (shall be an instance of ParamGridBuilder where you set the grid for the regularization parameter)

Required libraries:

- pyspark.ml.feature.VectorAssembler
- pyspark.ml.Pipeline
- pyspark.ml.regression.LinearRegression
- pyspark.ml.tuning.ParamGridBuilder
- pyspark.ml.tuning.CrossValidator
- pyspark.ml.evaluation.Evaluator

c) Decision Tree and Random Forest

It is now time to try other learning algorithms to find if it is possible to obtain a better model. Train:

- A Random Forest Regressor
- A Decision Tree Regressor

For each learning algorithm, you shall tune for at least one parameter using CrossValidator and ParamGridBuilder

Required libraries:

- pyspark.ml.regression.DecisionTreeRegressor
- pyspark.ml.regression.RandomForestRegressor

d) Conclusions

Compare and comment the obtained results (you may use a comparison table).

II. Part 2: SMS Spam Collection Data Set

1. Understanding the data set

a) Direct Link

Direct link on UCI's machine learning repository:

https://archive.ics.uci.edu/ml/machine-learning-databases/00228/

b) Data Set Information :

This corpus has been collected from free or free for research sources at the Internet:

- A collection of 425 SMS spam messages was manually extracted from the Grumbletext Web site. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the very spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully scanning hundreds of web pages.
- A subset of 3,375 SMS randomly chosen ham messages of the NUS SMS Corpus (NSC), which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. The messages largely originate from Singaporeans and mostly from students attending the University. These messages were collected from volunteers who were made aware that their contributions were going to be made publicly available.
- A list of 450 SMS ham messages collected from Caroline Tag's PhD Thesis
- Finally, we have in corporated the SMS Spam Corpus v.0.1 Big. It has 1,002 SMS ham messages and 322 spam messages and it is public. This corpus has been used in the following academic researches:

[1] GÃ³mez Hidalgo, J.M., Cajigas Bringas, G., Puertas Sanz, E., Carrero GarcÃa, F. Content Based SMS Spam Filtering. Proceedings of the 2006 ACM Symposium on Document Engineering (ACM DOCENG'06), Amsterdam, The Netherlands, 10-13, 2006.

[2] Cormack, G. V., GÃ³mez Hidalgo, J. M., and Puertas SÃ¡nz, E. Feature engineering for mobile (SMS) spam filtering. Proceedings of the 30th Annual international ACM Conference on Research and Development in information Retrieval (ACM SIGIR'07), New York, NY, 871-872, 2007.

[3] Cormack, G. V., GÃ³mez Hidalgo, J. M., and Puertas SÃ¡nz, E. Spam filtering for short messages. Proceedings of the 16th ACM Conference on Information and Knowledge Management (ACM CIKM'07). Lisbon, Portugal, 313-320, 2007.

c) Attribute Information:

The collection is composed by just one text file, where each line has the correct class followed by the raw message. We offer some examples bellow:

ham What you doing?how are you?

ham Ok lar... Joking wif u oni...

ham dun say so early hor... U c already then say...

ham MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*

spam FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop

spam Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital of Australia? Text MQUIZ to 82277. B

spam URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call 0871-872-9758 BOX95QU

Note: the messages are not chronologically sorted.

d) Relevant Papers:

We offer a comprehensive study of this corpus in the following paper. This work presents a number of statistics, studies and baseline results for several machine learning methods.

Almeida, T.A., GÃ³mez Hidalgo, J.M., Yamakami, A. Contributions to the Study of SMS Spam Filtering: New Collection and Results. Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11), Mountain View, CA, USA, 2011.

e) Citation

http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

2. Consulting Project

You have been contacted to build a predictive model to classify an incoming sms as Spam or Safe. You are provided the dataset described above. For each observation in the dataset, the row sms as well as the correct class is given.

a) The first step is to change the text features into numeric using the suitable classes (StringIndexer, Tokenizer, StopWordsRemover, CountVectorizer, IDF, VectorAssembler).

b) Then You shall train 4 classifiers and compare them. These are:
   1. LogisticRegression,
   2. DecisionTreeClassifier
   3. RandomForestClassifier
   4. NaiveBayes

c) For one of these classifiers, you shall tune at least on important hyper parameter using ParamGridBuilder and CrossValidator

d) Conclusions: Compare and comment the obtained results (you may use a comparison table).

Required libraries:

- pyspark.ml.feature.StringIndexer
- pyspark.ml.feature.Tokenizer
- pyspark.ml.feature.StopWordsRemover
- pyspark.ml.feature.CountVectorizer
- pyspark.ml.feature.IDF
- pyspark.ml.feature.VectorAssembler
- pyspark.ml.Pipeline
- pyspark.ml.classification.LogisticRegression
- pyspark.ml.classification.RandomForestClassifier
- pyspark.ml.classification.DecisionTreeClassifier
- pyspark.ml.classification.NaiveBayes
- pyspark.ml.tuning.ParamGridBuilder
- pyspark.ml.tuning.CrossValidator
- pyspark.ml.evaluation.MulticlassClassificationEvaluator