IBM Watson Machine Learning Platform

Ziyuan Shen

November 12, 2018

1 Implementing Watson ML Project: Heartrate Failure

To get familiar with how IBM Watson ML platform works and what it can offer specifically, I plan to repeat some ML related projects. As I have mentioned weeks ago, there is an open code pattern named "Create and deploy a scoring model to predict heartrate failure"[1]. In this project, an already well coded Node.js application, which implements a heartrate failure predictive model, is called by IBM cloud for making online predictions. A tool chain is provided by the author so that the local application somehow has a link to IBM cloud. Next steps will be easy. Create a Watson studio and add machine learning service. Then create a notebook to store python codes for loading data, creating online machine learning model, etc. Some prediction outputs can be give (Figure 1), as well as accuracy and error (Figure 2).

AVGHEARTBEATSPERMIN	PALPITATIONSPERDAY	CHOLESTEROL	BMI	HEARTFAILURE	AGE	SEX	FAMILYHISTORY	SMOKERLAST5YRS	EXERCISEMINPERWEEK	label	SEX_IX	FAMILYHISTORY_IX	SMOKERLAST5YRS_IX	features
93	22	163	25	N	49	F	N	N	110	0.0	1.0	0.0	0.0	[93.0,22.0,163.0,...
108	22	181	24	N	32	F	N	N	192	0.0	1.0	0.0	0.0	[108.0,22.0,181.0...
86	0	239	20	N	60	F	N	N	121	0.0	1.0	0.0	0.0	[86.0,0.0,239.0,2...
80	36	164	31	Y	45	F	Y	N	141	1.0	1.0	1.0	0.0	[80.0,36.0,164.0,...
66	36	185	23	N	39	F	N	N	63	0.0	1.0	0.0	0.0	[66.0,36.0,185.0,...
125	27	201	31	N	47	M	N	N	13	0.0	0.0	0.0	0.0	[125.0,27.0,201.0...
83	27	169	20	N	71	F	Y	N	124	0.0	1.0	1.0	0.0	[83.0,27.0,169.0,...
107	31	199	32	N	55	F	N	N	22	0.0	1.0	0.0	0.0	[107.0,31.0,199.0...
92	28	174	22	N	44	F	N	N	107	0.0	1.0	0.0	0.0	[92.0,28.0,174.0,...
84	12	206	25	N	50	M	N	N	199	0.0	0.0	0.0	0.0	[84.0,12.0,206.0,...
60	1	194	28	N	71	M	N	N	27	0.0	0.0	0.0	0.0	[60.0,1.0,194.0,2...
134	7	228	34	Y	63	F	Y	N	92	1.0	1.0	1.0	0.0	[134.0,7.0,228.0,...
103	0	237	24	N	64	F	Y	N	34	0.0	1.0	1.0	0.0	[103.0,0.0,237.0,...
101	39	157	20	N	49	M	N	N	33	0.0	0.0	0.0	0.0	[101.0,39.0,157.0...
92	2	169	26	N	36	M	N	N	217	0.0	0.0	0.0	0.0	[92.0,2.0,169.0,2...
80	27	234	27	N	50	M	N	N	28	0.0	0.0	0.0	0.0	[80.0,27.0,234.0,...
82	14	155	30	N	70	F	N	N	207	0.0	1.0	0.0	0.0	[82.0,14.0,155.0,...
63	9	204	26	N	42	M	N	N	88	0.0	0.0	0.0	0.0	[63.0,9.0,204.0,2...
83	12	209	29	N	38	M	Y	N	220	0.0	0.0	1.0	0.0	[83.0,12.0,209.0,...
80	37	157	20	N	48	M	N	N	54	0.0	0.0	0.0	0.0	[80.0,37.0,157.0,...

Figure 1: Heartrate failure prediction.

```
Accuracy = 0.871012
Test Error = 0.128988
```

Figure 2: Test accuracy.

Unfortunately, the code of getting the API token fails and I cannot figure out why. So the application cannot be accessed through API. It is not a big deal, though. We can save the API realization for the latter stage of BioCyBig. As I have got some knowledge of how to create a ML project on IBM cloud, I can dig more into projects about realtimeness for serving the BioCyBig goal.

2 Machine Learning in Watson Studio

It is good that IBM cloud ML is compatible with python, scikit-learn etc., because python is easy to implement due to many existing resources. The project above is still complex. Very simple ML models can be created directly in Watson Studio. There are several ways of building models in Watson Studio ML: model builder, notebook and flow editor. I tried the first two.

2.1 Building a Model Using Model Builder

WSML (Watson Studio Machine Learning) has the mostly widely used ML estimators (algorithms) built inside the platform so that they can be utilized directly. By uploading a data file (with feature columns and label columns), we can create and train a ML model either automatically (choose binary classification, or multiclass classification or regression) or manually (specify an estimator). After the model is trained, performance will be evaluated and accuracy will be given. After deploying the model, we can also type in the feature data to get prediction results (Figure 3).

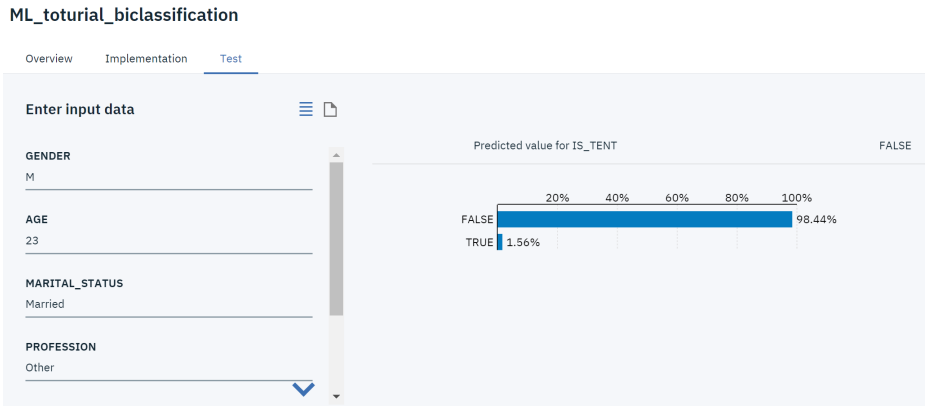

Figure 3: Predict whether a customer will buy a tent.(Tutorial[2])

2.2 Building a Model Using a Notebook

As shown above, using a model builder is rather easy and straightforward. However, the estimators that can be chosen from are limited. Using a notebook for creating a model means to demonstrate and deploy a model all by coding (eg. Python). By running the sample notebook (Figure 4), I created the digits prediction model [3].

Figure 4: Add notebooks for creating a model.

In summary, the disadvantage of using a notebook is that complete coding is required for specifying an algorithm, deploying the model, evaluating the model etc. The advantage is that the implemented algorithm can be much more flexible.

3 Testing My Own Dataset

As I have acquired the basic skills of creating a model in WSML, I can test my own dataset independent of the tutorials. One important thing is missed by me about the case study I tried to implement weeks ago [4]. The algorithm the author proposed is clustering, which is unsupervised learning. Thus the algorithm is not evaluated by a value between [0,1] (accuracy). By far, I haven't seen implementation of unsupervised learning in the WSML documentation I read. So I tried an even simpler supervised learning case.

Wisconsin Diagnostic Breast Cancer (WDBC) [5] is a dataset widely used by ML researchers [6, 7]. The dataset includes 10 feature columns (tumor features) and one label column (class: whether breast cancer or not) Figure 5. The entire dataset includes 699 patients. After deleting rows with invalid data, 683 rows are left.

	id	thickness	size	shape	adhesion	single	nuclei	chromatin	nucleoli	mitosis	class
0	1000025	5	1	1	1	2	1.0	3	1	1	0
1	1002945	5	4	4	5	7	10.0	3	2	1	0
2	1015425	3	1	1	1	2	2.0	3	1	1	0
3	1016277	6	8	8	1	3	4.0	3	7	1	0
4	1017023	4	1	1	3	2	1.0	3	1	1	0

Figure 5: Wisconsin Diagnostic Breast Cancer (WDBC).

Through scikit-learn, a LogisticRegression function can be applied to train a model (Figure 6). The accuracy is 0.92.

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=0)
```

```
# train logistic regression model
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(x_train,y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

```
y_pred_class=logreg.predict(x_test) # make predictions based on x_test and store it to y_pred_class
```

```
# Now let us see how our model is performing.
# We will start with accuracy
from sklearn import metrics
print (metrics.accuracy_score(y_test, y_pred_class))
```

```
0.9298245614035088
```

```
logreg.score(x_test,y_test)
```

```
0.9298245614035088
```

Figure 6: Predict breast cancer using machine learning in Python.

By using the model builder in WSML, a Logistic Classification model can be built using this dataset (Figure 7). Since the Python code is also done, I can try building a notebook later.

4 Relativity to BioCyBig

By far, my idea of relating implementation of these small cases to BioCyBig is that, with a realtime system, whenever a new data type is detected, the could platform can retrain the model using different selections of data types, evaluate the new models and give conclusion whether the new data type is useful or maybe some old data type can be eliminated. This is just like feature selection in ML so the idea is of little novelty actually. However, advantage of BioCyBig lies in realtimeness. This can be further implemented by continuous learning in IBM Watson ML platform, which has the function of retraining model on new data (require a charged service).

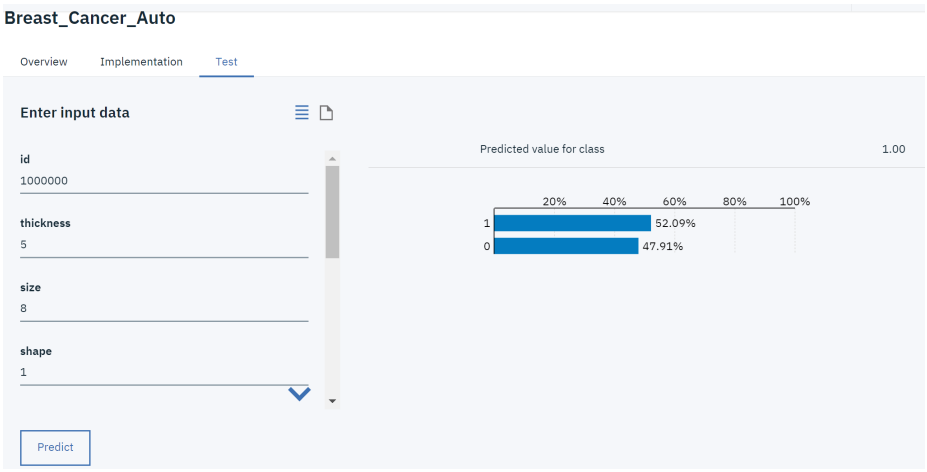

Figure 7: Building a breast cancer model in WSML.

References

[1] J. McCoy and D. Carew, "Create and deploy a scoring model to predict heartrate failure," https://developer.ibm.com/patterns/create-and-deploy-a-scoring-model-to-predict-heartrate-failure/, 2018, [Online; last updated October 11, 2018; accessed Oct 19, 2018].

[2] "Model builder tutorial: Build a binary classifier model automatically," https://dataplatform.cloud.ibm.com/docs/content/analyze-data/ml-model-builder-tutorial-01-binary-classification-auto.html, 2018, [Online; last updated June 27, 2018; accessed Nov 7, 2018].

[3] "Use scikit-learn to predict hand-written digits," https://dataplatform.cloud.ibm.com/exchange/public/entry/view/168e65a9e8d2e6174a4e2e2765aa4df1, 2018, [Online; last updated May 28, 2018; accessed Nov 8, 2018].

[4] Y. Guo, J. Zheng, X. Shang, and Z. Li, "A similarity regression fusion model for integrating multi-omics data to identify cancer subtypes," *Genes*, vol. 9, no. 7, p. 314, 2018.

[5] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[6] A. F. M. Agarap, "On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset," in *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing*. ACM, 2018, pp. 5–9.

[7] K. Liu, G. Kang, N. Zhang, and B. Hou, "Breast cancer classification based on fully-connected layer first convolutional neural networks," *IEEE Access*, vol. 6, pp. 23 722–23 732, 2018.