

BUAN 6341 APPLIED MACHINE LEARNING ASSIGNMENT 3

The purpose of this assignment is to implement two learning algorithms, Artificial neural networks (ANN), K nearest neighbors, on the two datasets, "Seoul Bike-sharing Demand Data" and "IBM HR-Employee-Attrition & Performance." The tasks are to experiment and compare the different algorithms on the response of accuracy. Compare the result with the other three algorithms from the previous assignment, which are SVM, Decision tree and Boosting.

Solutions

1. Experimental results using the Seoul Bike-Sharing Demand Dataset

(Table 1. Represented Data table of the Seoul Bike-Sharing Demand Dataset)

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes

The machine learning algorithms were used to develop prediction models to predict the hourly bike renting based on the dataset (Table 1). Since the information about hourly bike renting is important for the bike-sharing company to improve its bike renting management. The "Functioning Day" column data were hourly bike renting records and were output data. The bike renting data in a specific hour has two category values: with bike rent=1 or without bike rent=0. Before applying the learning algorithms, the dataset should be processed into the desired format.

Dataset processing steps.

1. Ignored the "Date" column since the prediction model is a 24 hours model with hourly prediction and not considering a specific year and date.
2. Ignored the "Rented Bike Count" column since the "Rented Bike Count" information is positively correlated with the output, the bike renting status. The bike renting status will be "with bike renting" for count value other than zero; otherwise, the renting status will be "no bike renting." No further information is needed if considering the "Rented Bike Count" as an input attribute. Therefore, to develop a sensible hourly prediction model for predicting bike renting status, we ignore the "Rented Bike Count" column data.
3. Encoded "Functioning Day" data into two class labels, which 0 = No (295) and 1 = Yes (8465).
4. As required by the scikit package, the independent categorical variable "Seasons" was encoded into four binary columns: spring, summer, autumn, and winter. Similarly, the "Holiday" column was encoded into two binary columns.
5. Split the dataset into the training set (70% of data) and test set (30% of data). The following is the shape of split data.
[Y train: (6132,), X train:(6132, 15), Y test: (2628,), X test: (2628, 15)]
6. Normalize the dataset to diminish the scale effect of different columns data to avoid numerical difficulties and it is essential in K-NN and ANN algorithms.

Outline of the principles for applying the machine learning algorithms.

Use the Artificial neural networks (ANN), K Nearest Neighbors algorithms supported in the scikit-learn machine learning and TensorFlow packages as the learning engines.

• ***Artificial Neural Networks (ANN)***

1. Apply ANN models with different hidden layers (1 and 2) and units (4 and 6).
2. I use the rectified linear activation function for hidden layers and sigmoid activation function for the output layer in Artificial neural networks because we want to predict the binary result, and this is a prevalent combination.
3. Apply two optimizers "SGD" = Gradient descent (with momentum) method and "Adam" = stochastic gradient descent method.
4. Apply two epochs (100 and 300), the number of complete passes through the training dataset.
5. Set the ANN model predict probability > 0.5 to 1 and probability < 0.5 to 0

• ***K Nearest Neighbors***

1. Apply ANN models with different neighbors (3 and 10) and metric, Minkowski and Euclidean.
2. Apply a 10-Fold cross-validation scheme to build the models.

Summary of the machine learning experiments.

(Table 2. Summary of the accuracies for the selected machine learning algorithms)

Methods	Model No.	Hidden layers	units	optimizer	Epochs	Training Accuracy	Test Accuracy	difference
Artificial neural networks (ANN)	Model 1	2	6	Adam	100	96.90%	96.27%	0.63%
	Model 2	2	6	Adam	300	97.31%	96.19%	1.12%
	Model 3	2	6	SGD	100	97.26%	96.27%	0.99%
	Model 4	2	6	SGD	300	97.31%	96.27%	1.04%
	Model 5	2	4	Adam	100	97.00%	96.31%	0.69%
	Model 6	2	4	Adam	300	97.18%	96.19%	0.99%
	Model 7	2	4	SGD	100	97.15%	96.27%	0.88%
	Model 8	2	4	SGD	300	97.20%	96.31%	0.89%
	Model 9	1	6	Adam	100	96.87%	96.35%	0.52%
	Model 10	1	6	Adam	300	97.15%	96.16%	0.99%
	Model 11	1	6	SGD	100	97.13%	96.16%	0.97%
	Model 12	1	6	SGD	300	97.11%	96.08%	1.03%
	Model 13	1	4	Adam	100	96.79%	96.35%	0.44%
	Model 14	1	4	Adam	300	96.77%	96.27%	0.50%
	Model 15	1	4	SGD	100	96.75%	96.27%	0.48%
	Model 16	1	4	SGD	300	96.75%	96.27%	0.48%
		Methods	Model No.	neighbors	metric	Training Accuracy	Cross-validation	Test Accuracy
		K Nearest Neighbors	Model 17	3	minkowski	96.69%	96.61%	96.27%
			Model 18	10	minkowski	96.31%	97.00%	96.27%
			Model 19	3	euclidean	96.69%	96.61%	96.27%
			Model 20	10	euclidean	96.31%	97.00%	96.27%

ANN Models

- Compared to 100-time epochs, 300-times epochs have lower bias and higher variance. In other words, 100-time have high test accuracy and 300-times have high training accuracy, at the same hidden layers, units, and optimizer.
- Four unites ANN models have higher test accuracy than 6 unites models, and one hidden layer models also have higher test accuracy than two layers models. Thus, assuming that more neural might create the better fit model on the training set but increase the test set variance.
- In this dataset, changing hidden layers, units, optimizers, and epochs do little impact on the accuracy score. All the models are appropriate to predict the result.
- Comparing all Models, Model 9 has the best result. By cutting back the hidden layers to reduce the model complexity and built more perceptrons to achieve more complex decision boundaries, we have a more balanced model.

K Nearest Neighbors

- According to the result, metrics do not impact the model, and the number of neighbors decided the training accuracy and the average training accuracy.
- K-NN models' test accuracies are the same, but we can check the cross-validation accuracy to choose the better models.
- Model 17 and Model 19 are the best models; in this case, higher training accuracy and lower cross-validation accuracy.

Model comparison

- All the Models are proper estimated models that have more than 96% accuracy.
- The classification results had a little different by checking the confusion matrix, but the accuracy is similar. *The following show confusion matrix results.*

M1	M2	M3	M4
[[6 91] [6 2525]] Accuracy: 96.31 %	[[12 85] [13 2518]] Accuracy: 96.27 %	[[12 85] [14 2517]] Accuracy: 96.23 %	[[13 84] [16 2515]] Accuracy: 96.19 %
M5	M6	M7	M8
[[3 94] [5 2526]] Accuracy: 96.23 %	[[8 89] [16 2515]] Accuracy: 96.00 %	[[7 90] [10 2521]] Accuracy: 96.19 %	[[5 92] [10 2521]] Accuracy: 96.12 %

2. Experimental results using the IBM HR Analytics Employee Attrition & Performance

(Table 3. Represented Data table of the IBM Dataset)

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel	JobRole	JobSatisfaction	MaritalStatus	Mo
0	41	Yes	Travel_Rarely	1102	Sales		1	2	Life Sciences	1	1	2	Female	94	3	2	Sales Executive	4	Single
1	49	No	Travel_Frequently	279	Research & Development		8	1	Life Sciences	1	2	3	Male	61	2	2	Research Scientist	2	Married
2	37	Yes	Travel_Rarely	1373	Research & Development		2	2	Other	1	4	4	Male	92	2	1	Laboratory Technician	3	Single

Dataset link: <https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset>

The machine learning algorithms were used to develop prediction models to predict employee performance based on the dataset (Table 3). Since the information about employee performance is essential for all organizations, it helps to improve their human resource management. The "PerformanceRating" column data were number rating records and were output data. The Performance Rating in a specific number has two category values: with lower rating=3 or higher rating=4. Let's explore the reasons for job performance, and the result might help in my future career performance. Before applying the learning algorithms, the dataset should be processed into the desired format.

Dataset processing steps.

1. Ignored the "PercentSalaryHike" column because it will direct to classify "Performance Rating" and will lose the meaning of exercise
2. Encoded " PerformanceRating " data into two class labels, which 0 = 3 (1224) and 1 = 4 (226).
3. As required by the scikit package, the independent categorical variable "Attrition", "BusinessTravel", "Department", "EducationField", "Gender", "JobRole", "MaritalStatus" and "OverTime"...etc, are encoded into appropriate columns.
4. Split the dataset into the training set (70% of data) and test set (30% of data). The following is the shape of split data.
[Y train: (1029,), X train:(1029, 55), Y test: (441,), X test: (441, 55)]
5. Normalize the dataset to diminish the scale effect of different columns data to avoid numerical difficulties, and it is essential in K-NN and ANN algorithms.

Outline of the principles for applying the machine learning algorithms.

Use the Artificial neural networks (ANN), K Nearest Neighbors algorithms supported in the scikit-learn machine learning, and TensorFlow packages as the learning engines.

- ***Artificial Neural Networks (ANN)***

1. Apply ANN models with different hidden layers (1 and 2) and units (4 and 6).
2. I use the rectified linear activation function for hidden layers and sigmoid activation function for the output layer in Artificial neural networks because we want to predict the binary result, and this is a prevalent combination.
3. Apply two optimizers "SGD" = Gradient descent (with momentum) method and "Adam" = stochastic gradient descent method.
4. Apply two epochs (100 and 300), the number of complete passes through the training dataset.
5. Set the ANN model predict probability > 0.5 to 1 and probability < 0.5 to 0

- ***K Nearest Neighbors***

1. Apply ANN models with different neighbors (3 and 10) and metric, Minkowski, and Euclidean.
2. Apply a 10-Fold cross-validation scheme to build the models.

Summary of the machine learning experiments.

(Table 4. Summary of the accuracies for the selected machine learning algorithms)

Methods	Model No.	Hidden layers	units	optimizer	Epochs	Training Accuracy	Test Accuracy	difference
Artificial neural networks (ANN)	Model 1	2	6	Adam	100	88.14%	82.77%	5.37%
	Model 2	2	6	Adam	300	94.66%	74.38%	20.28%
	Model 3	2	6	SGD	100	95.04%	74.38%	20.66%
	Model 4	2	6	SGD	300	96.02%	75.51%	20.51%
	Model 5	2	4	Adam	100	84.94%	84.13%	0.81%
	Model 6	2	4	Adam	300	85.53%	81.18%	4.35%
	Model 7	2	4	SGD	100	85.62%	81.18%	4.44%
	Model 8	2	4	SGD	300	86.59%	80.27%	6.32%
	Model 9	1	6	Adam	100	85.52%	83.45%	2.07%
	Model 10	1	6	Adam	300	90.57%	80.50%	10.07%
	Model 11	1	6	SGD	100	90.96%	79.37%	11.59%
	Model 12	1	6	SGD	300	92.52%	76.19%	16.33%
	Model 13	1	4	Adam	100	84.94%	84.58%	0.36%
	Model 14	1	4	Adam	300	89.02%	78.91%	10.11%
	Model 15	1	4	SGD	100	89.60%	78.46%	11.14%
	Model 16	1	4	SGD	300	89.80%	78.68%	11.12%
		Methods	Model No.	neighbors	metric	Training Accuracy	Cross-validation	Test Accuracy
		K Nearest Neighbors	Model 17	3	minkowski	80.05%	80.08%, sd=1.56%	78.68%
			Model 18	10	minkowski	84.58%	84.45%, sd=0.4%	78.68%
			Model 19	3	euclidean	80.05%	80.08%, sd=1.56%	78.68%
			Model 20	10	euclidean	84.58%	84.45%, sd=0.4%	78.68%

ANN Models

- Models with one hidden layer and four units have the best test accuracy and reasonable training accuracy.
- Compared to 100-time epochs, 300-times epochs have lower bias and higher variance. In other words, 100-time have high test accuracy, and 300-times have high training accuracy, at the same hidden layers, units, and optimizer.
- At the same optimizer, units, and optimizer, increasing hidden layers will increase the training accuracy but lower the test accuracy
- When the same optimizer and epochs, increasing unites will increase the training accuracy but lower the test accuracy at two hidden layers but not in 1 hidden layers models.
- Changing numbers of layers, unite and epoch, and optimizer function will impact the test accuracy, and we need to try different combinations to find the best model.

K Nearest Neighbors

- According to the results, the number of neighbors has the most considerable impact on accuracy.
- The dataset distribution might cause the method of metric to have no impact on the prediction models.
- Increase the number of neighbors will help to improve the training accuracy.

Model comparison

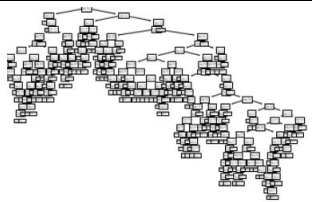
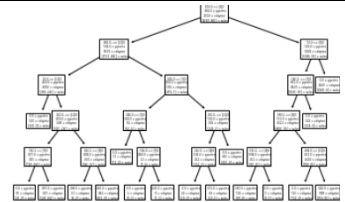
- According to the experiment result, the ANN and K-NN algorithms might not suit the IBM dataset because it created the low test, training, and validation accuracy.
- ANN creates a better model than K-NN on the IBM dataset.

3. Conclusion of ANN and K-NN

- The ANN algorithm takes more time to calculate than the KNN algorithm.
- Not every ANN combination is a good prediction model. They are finding the best combination after multiple implementations can get better results than the KNN algorithm.
- Both ANN and KNN algorithms are suitable for the Seoul model, but neither is ideal for the IBM model.

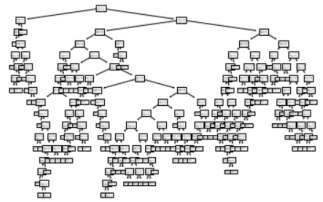
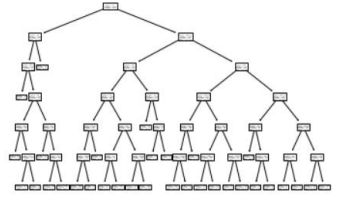
4. Comparison of Five algorithms

(Table 5. Summary of the SBD Dataset for the selected machine learning algorithms)

Model No.	Methods	Model Training Accuracy	10-Fold Cross-Validation Accuracy	Model Testing Accuracy
Model 1	SVM with RBF	96.77 %	96.77 %, SD = 0.06 %	96.31 %
Model 2	SVM with Sigmoid	96.38 %	95.89%, SD = 0.48 %	95.74 %
Model 3	SVM with Polynomial	96.98 %	96.85%, SD = 0.19 %	96.19 %
Model 4	Decision Trees with best Tree	100 %	95.84%, SD = 0.65 %	95.93 %
Model 5	Decision Trees with 6 Depths	96.93 %	96.67%, SD = 0.24 %	96.23 %
Model 6	AdaBoost	96.74 %	96.54%, SD = 0.19 %	96.35 %
Decision Trees		Model 4		Model 5
Plot				

(Table 6. Summary of the IBM Dataset for the selected machine learning algorithms)

Model No.	Methods	Model Training Accuracy	10-Fold Cross-Validation Accuracy	Model Testing Accuracy
Model 1	SVM with RBF	84.55%	84.55%, SD = 0.25%	84.81%
Model 2	SVM with Sigmoid	83.67%	84.65%, SD = 0.71 %	84.58%

Model 3	SVM with Polynomial	88.92%	84.35%, SD = 0.19 %	84.13%
Model 4	Decision Trees with “best” tree	100.0%	72.89%, SD = 2.97 %	73.24%
Model 5	Decision Trees with 6 Depths	87.46%	80.28%, SD = 2.10 %	79.14%
Model 6	AdaBoost	84.35%	82.80%, SD = 1.22%	82.31%
Decision Trees		Model 4		Model 5
Plot				

- You can use these five algorithms to build a predictive model for SBDD, and the accuracy of testing and training is outstanding.
- The models of five algorithms to predict for IBM have a low accuracy of testing and training. I assume some reasons might help to build or influenced the better-predicted model. First, the IBM dataset is not a good dataset; it has a mess of data. Second, we need to use more data to train the data because our training accuracy is high, but test accuracy is low, so increasing the training data can help to decrease the variance.
- The running time has a difference between the five algorithms, in my experiment, ANN > Decision Tree > SVM = Boosting > KNN.
- The best algorithm model for the "Seoul Bike-Sharing Demand Dataset" is the ANN algorithm with one hidden layer, six units, and adam optimizer. It has the best test accuracy (96.35%) and training accuracy (96.87%).
- The best algorithm model for "IBM HR Analytics Employee Attrition & Performance" is the SVM algorithm with RBF kernel. It has the best test accuracy (84.81%) and adequate training accuracy (84.55%).