Prepared by: Aditya Sunil Choksi (axc210030) **Date: July 24th 2023**

BUAN 6341 Applied Machine Learning
Assignment 2

A. Executive Summary

- **The report presents an in-depth analysis of the application of an Artificial Neural Network (ANN) to predict hourly bike rental requirements in urban cities, using a dataset that includes weather conditions and hourly bike rental data.**
- **The ANN classifier was optimized through a meticulous process of hyperparameter tuning, including parameters such as batch size, number of epochs, activation function, learning rate, momentum, drop-out rate, weight constraint, and the number of neurons in the hidden layers.**
- **The Stochastic Gradient Descent (SGD) learning algorithm was employed, along with a Sigmoid activation function and Binary Cross Entropy as the loss function, to refine the model's predictive capabilities.**
- **The report also discusses the importance of data pre-processing and exploratory data analysis, which led to key decisions such as treating the 'hour' feature as a categorical variable and excluding the 'functioning_day' feature from model training.**
- **The final model, which utilized all features, achieved the best performance, with an average CV test score of 0.86 and a test accuracy, sensitivity, and specificity of 0.92. The report concludes that a single hidden layer with 10 neurons was sufficient for optimal classification outcomes, underscoring the importance of feature selection and parameter tuning in machine learning.**
- **The insights from this analysis can inform future predictions and research in bike rental trends, particularly in the context of urban mobility and weather conditions.**

B. Introduction

In numerous urban cities, rental bikes have been implemented to improve mobility convenience. Ensuring the timely availability and accessibility of rental bikes is crucial in reducing waiting times and addressing the challenge of maintaining a stable supply. Accurately predicting the required bike count for each hour is a vital aspect of achieving this stability. The dataset comprises weather details (such as temperature, humidity, wind speed, visibility, dew point, solar radiation, snowfall, and rainfall), along with information on the number of bikes rented per hour and date.

C. ANN Classifier

Neural networks inspired by the brain's neuronal structure are called artificial neural networks. They process data individually and refine their categorization abilities by comparing their classification of a given dataset with its recognized classification. Finding the most effective parameters within numerous hyperparameters to achieve the highest metric scores is a complex task. A method known as 5-fold Grid Search Cross Validation is implemented to determine the most suitable hyperparameters to build the final artificial neural network. Here are some crucial considerations before creating the neural network:

• Every experimental model utilizes Stochastic Gradient Descent (SGD) as the learning algorithm (or backpropagation technique). SGD is an iterative method that leverages a training dataset to enhance the model.
• The output node employs a Sigmoid activation function.
• Binary Cross Entropy serves as the loss function. This function computes a score that averages the discrepancy between the actual and predicted probability distributions for predicting the first class. The goal is to minimize this score, with the optimal cross-entropy value being 0.

Following is the order of experimental models:

Model 1: Hyperparameters: Batch size and # of Epochs

Batch size and the number of epochs are two critical hyperparameters in gradient descent. Batch size influences the quantity of training examples utilized before an update is made to the model's inner parameters. On the other hand, the number of epochs dictates how many times the model will iterate over the entire training dataset.

Model 2: Activation Function of the 1st hidden layer nodes

Four specific activation functions are employed: Linear (also known as Identity), Sigmoid (sometimes referred to as Logistic), Tanh (short for Hyperbolic Tangent), and ReLU, which stands for Rectified Linear Unit.

Model 3: Learning Rate and Momentum of the 1st hidden layer nodes

The learning rate determines the extent of weight adjustment after each batch, while momentum decides the degree to which the previous weight modification affects the present weight adjustment.

Model 4: Drop-out Rate and Weight Constraint of the 1st hidden layer nodes

The dropout rate refers to the proportion of neurons randomly chosen to be disregarded during the training process. On the forward pass, their input to the activation of subsequent neurons is temporarily nullified, and any updates to their weights are omitted on the backward pass. This strategy aids in preventing excessive co-adaptation, which in neural networks, signifies a high correlation in the behavior of different hidden units, leading to overfitting. Encouraging independent feature detection by hidden units enhances computational efficiency and the model's capacity to establish a more generalized representation. The weight constraint specifies the upper limit for the norm of incoming weights. For optimal results, combining dropout with a weight constraint like the max norm constraint is often recommended.

Model 5: # of Neurons in the 1st hidden layer

Tuning the quantity of neurons in a layer is a crucial task. Typically, the number of neurons in a given layer influences the network's ability to represent data, especially at that particular point in the network structure.

Model 6: # of Neurons in the 2nd hidden layer

The neurons in the second hidden layer share similar attributes with those in the first hidden layer. The sole purpose of implementing a grid search is to determine the ideal quantity of neurons for the second hidden layer.

D. Bike Sharing Dataset:

The dataset consists of 14 features and 8760 records. The dataset contains the number of bikes rented per hour and date information along with the weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall) which can be used to predict the bike count required at each hour for the stable supply of rental bikes.

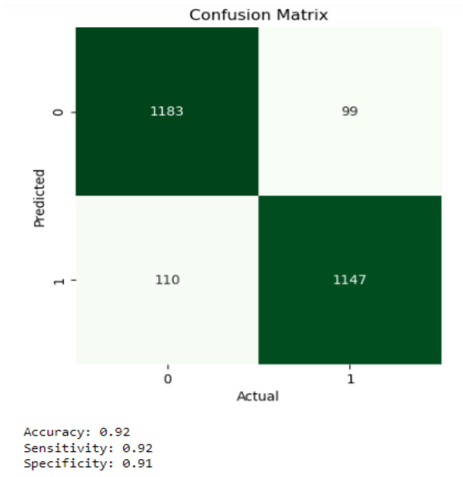
D1. EDA and Data Pre-processing:

The exploratory data analysis leads to several key findings that are incorporated into our modeling:

- The 'hour' feature is treated as a categorical variable and incorporated in our models.
- It's unnecessary to include 'functioning_day' during model training. We are not required to predict the number of bikes rented on those days.
- As the dataset lacks the day of the week as a variable, a new one is created for inclusion in machine learning models.
- The dew point temperature is closely correlated with the temperature (0.91), and integrating both variables could induce issues of multicollinearity during modeling. Furthermore, the Variance Inflation Factor (VIF) for all variables remains below 10 once the dew-point temperature is omitted. Therefore, the dew point temperature is excluded from the model development.
- The dataset is balanced.

D2. ANN Classification

Model 1: Hyperparameters: Batch size and # of Epochs



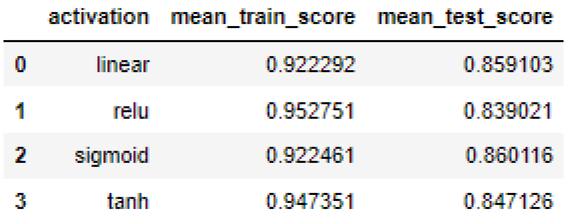
```
Accuracy: 0.92
Sensitivity: 0.92
Specificity: 0.91
```

Hyperparameter: Batch Size = [10, 20, 50] and # of Epochs = [10, 50, 100]
The network employs a sigmoid function as the activation function for the first hidden layer's 10 nodes and the output node.

Chosen parameters: Batch Size = 10 and # of Epochs = 100. The average CV accuracy is 0.86.

As a general trend, an increase in the number of epochs results in more frequent alterations to the neural network's weights, moving the model from an underfitting phase through to an optimal phase and then potentially to an overfitting phase. However, for this specific dataset, the optimal setup entails 100 epochs with a batch size of 10.

Model 2: Activation Function of the 1st hidden layer nodes



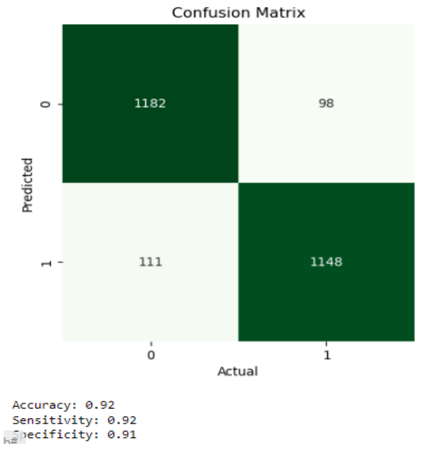
	activation	mean_train_score	mean_test_score
0	linear	0.922292	0.859103
1	relu	0.952751	0.839021
2	sigmoid	0.922461	0.860116
3	tanh	0.947351	0.847126

Hyperparameter: Activation = [Linear, Sigmoid, Tanh, ReLU]
The network has sigmoid function as the activation function for the 10 nodes in the first hidden layer and for the output node.

Chosen parameters: Activation Function = Sigmoid. The mean CV test score for Sigmoid activation function is 0.86.

The performance on test set has 0.92 as accuracy, sensitivity is 0.92, and specificity is 0.91. Sigmoid activation function in hidden layer generally leads to best classification results. The Relu activation function gives the lowest CV test score of 0.84. The average of the mean CV test scores for linear, sigmoid, relu and tanh activation functions is 0.85.

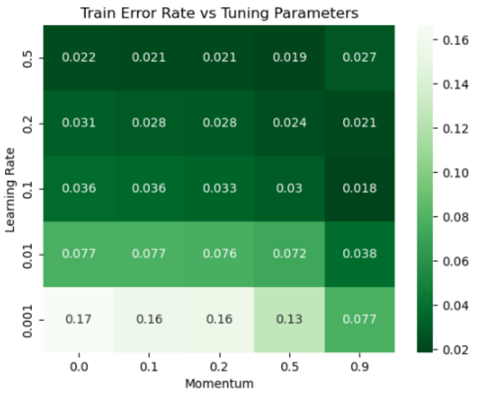
The test set performance is comparable to Model 1.

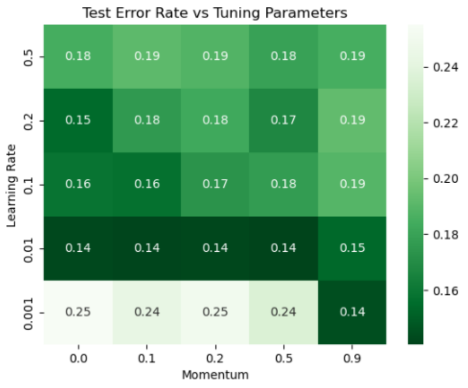


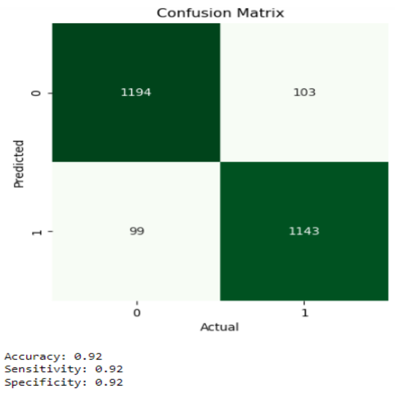
```
Accuracy: 0.92
Sensitivity: 0.92
Specificity: 0.91
```

Model 3: Learning Rate and Momentum of the 1st hidden layer nodes

Hyperparameter: Learning Rate = [0.001, 0.01, 0.1, 0.2, 0.5] and Momentum = [0.0, 0.1, 0.2, 0.5, 0.9]







```
Accuracy: 0.92
Sensitivity: 0.92
Specificity: 0.92
```

In the network, the Sigmoid function is employed as the activation function for the first hidden layer's 10 nodes as well as the output node.

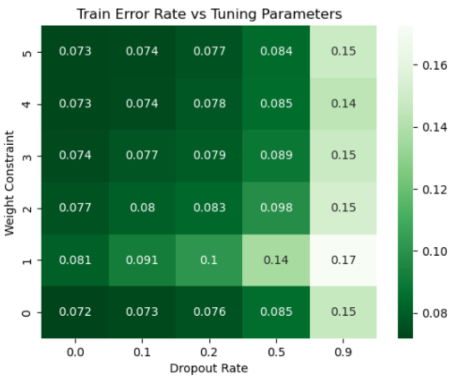
Chosen parameters: Learning Rate = 0.01 and Momentum = 0.5. When the Learning Rate is 0.01 and the Momentum is 0.5 the average CV test score is 0.86.

The performance on test set has 0.92 as accuracy, sensitivity is 0.92, and specificity is 0.92. Learning rate of 0.01 and momentum of 0.5 leads to best classification results. Learning rate 0.001 and momentum 0.0 gives the lowest CV test score of 0.745. The average of the mean CV test scores is 0.82. We also notice that there is a substantial change in the average CV test score with the changes in Learning Rates and Momentums.

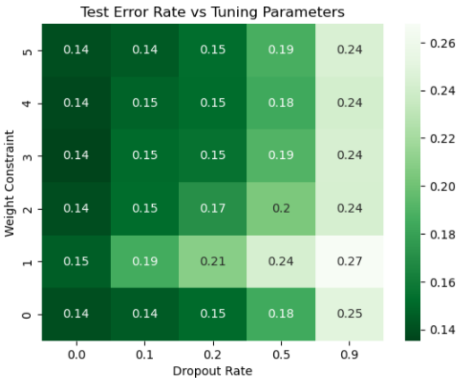
The test set performance is comparable to Model 2.

Model 4: Drop-out Rate and Weight Constraint of the 1st hidden layer nodes

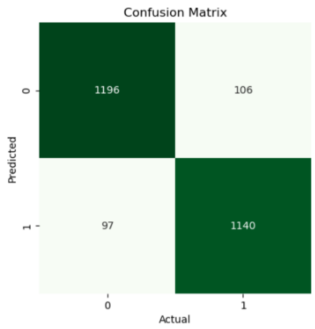
Hyperparameter: Drop-out Rate = [0.001, 0.01, 0.1, 0.2, 0.5] and Max Norm Weight = [0.0, 0.1, 0.2, 0.5, 0.9]



Train Error Rate vs Tuning Parameters



Test Error Rate vs Tuning Parameters



Confusion Matrix

```
Accuracy: 0.92
Sensitivity: 0.91
Specificity: 0.92
```

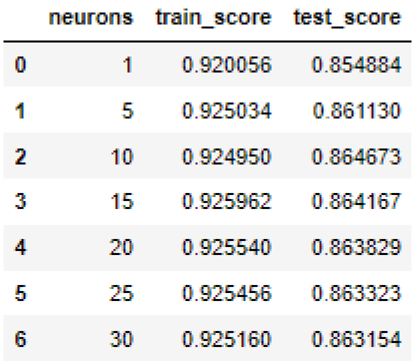
The network employs a Sigmoid function for activation for the initial hidden layer's 10 nodes and also for the output node. The specified learning rate is 0.01, while the momentum is set at 0.5.

Chosen parameters: Drop-out Rate = 0.0 and Max Norm Weight = 3. The average CV test score is 0.86.

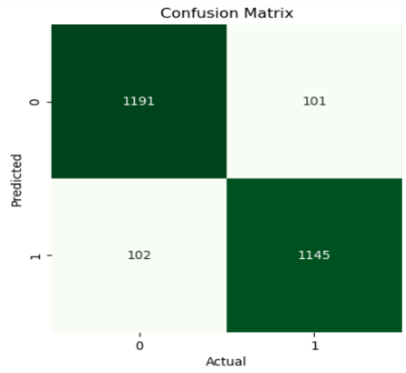
The performance on test set has 0.92 as accuracy, sensitivity is 0.91, and specificity is 0.92. Drop-out rate of 0.0 and Weight Constraint of 3 leads to best classification results. Drop out rate 0.9 and Weight Constraint 1 gives the lowest CV test score of 0.732. The average of the mean CV test scores is 0.8187.

Model 5: # of Neurons in the 1st hidden layer

Hyperparameter: # of neurons = [1, 5, 10, 20, 25, 30]



	neurons	train_score	test_score
0	1	0.920056	0.854884
1	5	0.925034	0.861130
2	10	0.924950	0.864673
3	15	0.925962	0.864167
4	20	0.925540	0.863829
5	25	0.925456	0.863323
6	30	0.925160	0.863154



Confusion Matrix

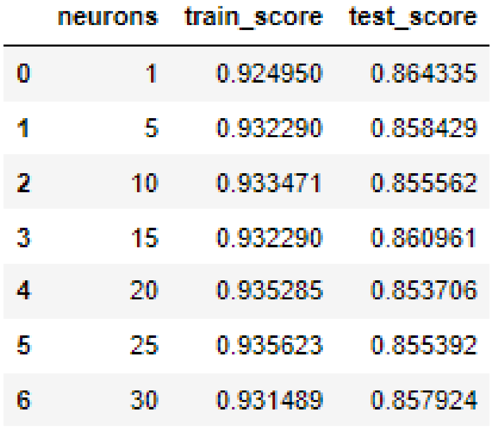
```
Accuracy: 0.92
Sensitivity: 0.92
Specificity: 0.92
```

The performance on test set has 0.92 as accuracy, sensitivity is 0.92, and specificity is 0.92. When the number of neurons is 10 leads to best classification results. When the number of neurons is 1 gives the lowest CV test score of 0.732. The average of the mean CV test scores is 0.862. We notice that there is not a major effect of the number of neurons on the mean CV test score.

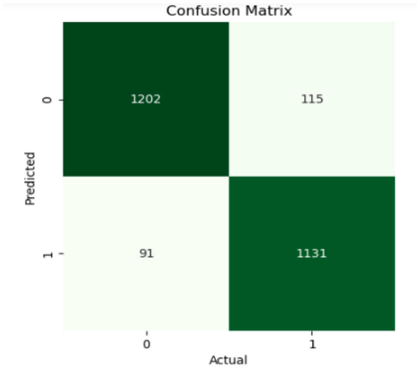
The performance on test set has 0.92 as accuracy, sensitivity is 0.92, and specificity is 0.92. When the number of neurons is 10 leads to best classification results. When the number of neurons is 1 gives the lowest CV test score of 0.732. The average of the mean CV test scores is 0.862. We notice that there is not a major effect of the number of neurons on the mean CV test score.

Model 6: # of Neurons in the 2nd hidden layer

Hyperparameter: # of neurons = [1, 5, 10, 20, 25, 30]



	neurons	train_score	test_score
0	1	0.924950	0.864335
1	5	0.932290	0.858429
2	10	0.933471	0.855562
3	15	0.932290	0.860961
4	20	0.935285	0.853706
5	25	0.935623	0.855392
6	30	0.931489	0.857924



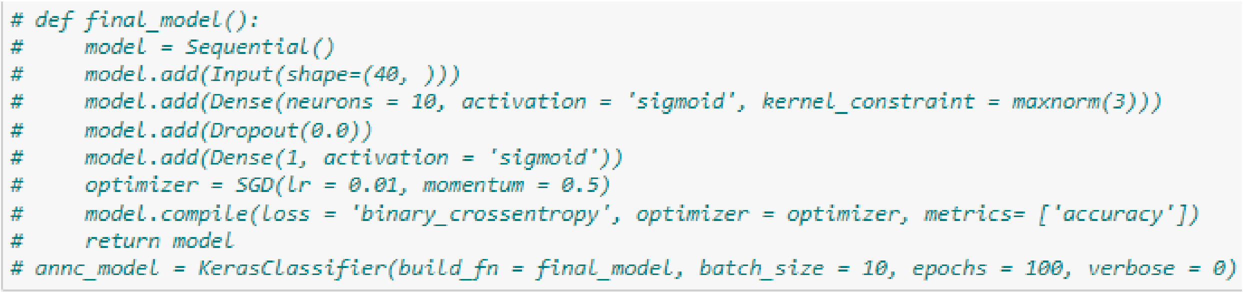
```
Accuracy: 0.92
Sensitivity: 0.91
Specificity: 0.93
```

The network utilizes the Sigmoid function as the activation strategy in the initial hidden layer, which comprises 10 neurons, and also for the output node. The parameters for learning rate and momentum are set at 0.01 and 0.5, respectively. In terms of architecture, the network features a drop-out rate of 0% and a weight restriction with a maximum norm of 3.

Chosen parameters: # of Neurons = 1. The average CV test score is 0.86.

The performance on test set has 0.92 as accuracy, sensitivity is 0.91, and specificity is 0.93. When the number of neurons is 1 leads to best classification results. When the number of neurons is 20 gives the lowest CV test score of 0.8537. The average of the mean CV test scores is 0.858. Upon evaluating the cross-validation accuracy of Model 5 and Model 6, it's evident that introducing an additional hidden layer has not significantly boosted performance. Thus, a single hidden layer encompassing 10 neurons suffices to yield optimal classification outcomes.

Final Model



```
# def final_model():
#     model = Sequential()
#     model.add(Input(shape=(40, )))
#     model.add(Dense(neurons = 10, activation = 'sigmoid', kernel_constraint = maxnorm(3)))
#     model.add(Dropout(0.0))
#     model.add(Dense(1, activation = 'sigmoid'))
#     optimizer = SGD(lr = 0.01, momentum = 0.5)
#     model.compile(loss = 'binary_crossentropy', optimizer = optimizer, metrics= ['accuracy'])
#     return model
# annc_model = KerasClassifier(build_fn = final_model, batch_size = 10, epochs = 100, verbose = 0)
```

D3. Final Model Scores

Model	Hyperparameters	5-fold CV score	Test Accuracy	Test Sensitivity	Test Specificity
ANN	batch_size=10 , epochs=100 Hidden Layer 1: neurons=10, activation=sigmoid, dropout=0.0 , weight_constraint=maxnorm(3), Output Layer: neurons=1, activation=sigmoid SGD Optimizer: learn_rate=0.01, momentum=0.5	0.86	0.92	0.92	0.92

E. Conclusion

The final model achieved an average CV test score of 0.86, with a test accuracy, sensitivity, and specificity of 0.92. The model used a batch size of 10, 100 epochs, 10 neurons in the first hidden layer with a Sigmoid activation function, a drop-out rate of 0.0, a weight constraint of maxnorm(3), and an SGD optimizer with a learning rate of 0.01 and momentum of 0.5. The report concludes that introducing an additional hidden layer did not significantly boost performance, and a single hidden layer with 10 neurons was sufficient to yield optimal classification outcomes.