UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

BITI 3133 – NEURAL NETWORK

LECTURER: PROF. TS. DR. BURHANUDDIN BIN MOHD ABOOBAIDER

PROJECT: PREDICTION OF BRAKE PAD USING ANN

PREPARED BY:

NO.	NAME	MATRIX
1	AMIR FAIZ BIN MOKHTAR	B032110351
2	MUHAMMAD ADAM HAFIZI BIN HASHIM TEE	B032110306
3	NAWAL IZZAH BINTI AZHAR	B032110359
4	NUR FARIHAH BINTI NAZRI	B032110052

Table of Contents

INTRODUCTION

Predictive analytics and machine learning have revolutionized several sectors, including car preventative maintenance, in the quickly evolving technology landscape of today. This project primarily focuses on the maintenance of brake pads in huge vehicles, demonstrating how these cutting-edge technologies may improve safety, reduce downtime, and maximize operational effectiveness.

The project successfully created a predictive model that predicts the ideal time for brake pad replacements. Using the potent artificial neural network (ANN) framework TensorFlow, variables including heat, journey distance, and wear rates were thoroughly examined. This data-driven strategy provides proactive maintenance, allowing for prompt interventions and reducing the likelihood of brake failures.

Businesses that manage automobiles can greatly profit from the deployment of such a predictive maintenance approach. Better safety is of the utmost importance since timely brake pad changes significantly lower the probability of accidents brought on by braking problems. In addition, preventive maintenance reduces downtime, which boosts productivity and operational effectiveness.

A significant cost reduction opportunity is also demonstrated by this project. Businesses may steer clear of the costly repairs or replacements that may be required in the event of unanticipated brake failures by proactively taking care of maintenance needs. Optimized preventative maintenance programs also result in better resource management, a reduction in wasteful spending, and an extension of the lifespan of vehicle parts.

In summary, this project serves as an example of how predictive analytics and machine learning have had a transformational influence on the field of preventive maintenance for vehicles, notably in the maintenance of brake pads for heavy trucks. Businesses may revolutionize utilizing their maintenance procedures, improve safety, and realize large operational and financial savings by utilizing the power of data and cutting-edge technology.

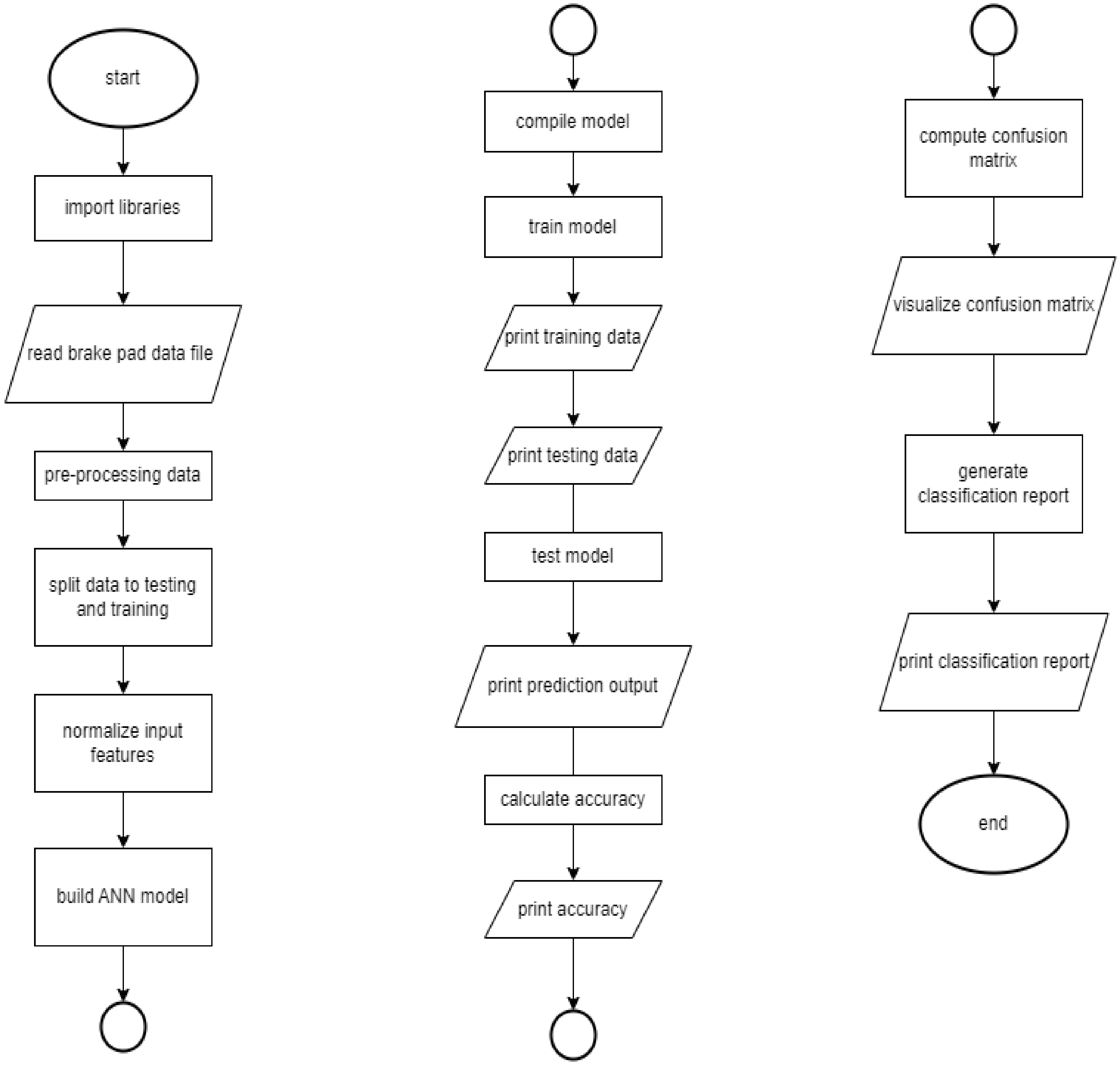
DETAIL DESCRIPTIONS OF SAMPLE DATA

The dataset used in this project is from brake_Modifieds.csv which consists of data on brake pad condition from various backgrounds which is total distance until it worn (km), brake pad heat, worn (0: not worn, 1: worn), the wear rate (z) and brake status (pr). Based on the backgrounds, this dataset has 338 samples with 5 columns (worn, km, heat, z, pr). The data is continuous for the km, heat, z, and pr while nominal data is for worn status. Based on the data source page, this dataset will be useful in creating a model that could estimate the condition of the brake pad of the vehicle based on the background given. However, the main source of this dataset only has minimal data, so our team decided to do some dummy data for the original dataset. This dummy data will help in improving the accuracy of our system.

The original dataset can be downloaded from the link:

https://github.com/werowe/mist_preventive_maintenance_ml/blob/master/brakedata.csv

FLOW CHART



start
import libraries
read brake pad data file
pre-processing data
split data to testing and training
normalize input features
build ANN model
compile model
train model
print training data
print testing data
test model
print prediction output
calculate accuracy
print accuracy
compute confusion matrix
visualize confusion matrix
generate classification report
print classification report
end

LEARNING PROCESS

The model used to predict brake pad condition is the Feedforward Neural Network model. This model was trained by using backpropagation which is implicitly used in the codes. This model contemplates all features in the dataset which are worn, km, heat, z, and pr.

Before starting the process to train the model, we need to import the necessary libraries such as pandas to manipulate data, "numpy" for numerical computations, and TensorFlow to build and train the neural network. Next, we must load the dataset which is brake_Modifieds.csv and store the dataset in the pandas data frame. Then, we need to ensure that the data is ready to train by doing the pre-processing data. In this case, we need to remove the commas for data in the 'km' column and convert them to float data type.

After doing the data pre-processing, we split the data into training and testing data. For this project, we have approximately 80% data for training and 20% for testing. Then before we train the model, we need to normalize the feature to ensure that the input features have zero mean and unit variance thus this will help in preventing any single feature from dominating the learning process.

Next, we build the neural network model. This model consists of two dense layers which are dense layer 1 and output layer. We obtained dense layer 1 from the "model.add(Dense(16, activation='relu', input_shape=(X_train.shape[1],)))" statement. This dense layer has 16 units of neurons which means the model will learn 16 different patterns of the data. As every neural network must have an activation function, thus this layer used rectified linear unit (ReLu) to introduces non-linearity and helps the model to recognize complex relationships in the data.

In order to complete the neural network architecture, this model has one output layer which has been added using the "model.add(Dense(1, activation = 'sigmoid'))" statement. This output layer has 1 unit neuron to represent the binary classification output either worn (1) or not worn (0). For the activation function, this layer used sigmoid function to compress the output between 1 and 0.

This model was compiled using optimizer and loss function. The optimizer used is Adaptive Moment Estimation (Adam) Optimization Technique to update the model's weight during the training process while the loss function used during compilation is "binary_crossentropy" which commonly being used to solve binary classification problems. Lastly, the performance of the training model was evaluated using accuracy.

This model is trained by using backpropagation algorithm used to compute the weights and biases of neural networks which will allow the optimization during the training. For the code, we can see that the model trained using "model.fit(X_train, y_train, epochs=50, batch_size=32, verbose=0)" statement which determined that the backpropagation has been handled by the 'fit' function under the TensorFlow framework. This function used computations to update the model's weight and biases on the calculated gradients. The running times for the training model is 50 epochs which means that the model will go through all the dataset 50 times and the batch size for the model is 32 which indicates that the model's weight is updated after 32 samples being processed.

After training the model, the model was tested in the testing data. The threshold for the testing data has been set for 0.5 to convert the probabilities of the testing data into binary prediction. Then the accuracy of the model's prediction is calculated using accuracy function under sklearn framework.

STEP-BY-STEP LEARNING AND ANALYSIS

There are multiple steps that we took in developing a predictive model to predict the status of brake pads. The model can identify whether the brake pad has been worn or not worn. The steps taken are as below:

1. Define the problem

Defining the problem involves stating the predictive model's goal: to anticipate the state of brake pads (used or not worn) based on specified parameters. It is critical to have a thorough knowledge of the problem, to ensure that the model creation and assessment process corresponds with the desired outcome.

2. Acquire and preprocess data

We acquired a dataset that includes relevant features such as kilometers driven, heat, and wear rate. The corresponding status of the brake pads whether it has been worn or not has also been obtained. However, for the wear rate, we generated the sample data using a rough model :

```
z = wear_rate = (0.003 * heat) + (0.004 * kilometers)-78
```

Through this, we plugged the value into the logistic probability function to get the status of the brake pads. If the pr > 50%, then worn is 1 and vice versa.

```
pr = 1 / (1 + e**-z)
```

Based on our coding below, this is everything that we did regarding the data. Firstly, we imported all the necessary libraries that will be used for data preprocessing and model evaluation. Next, we loaded the dataset into our system and printed the first few rows for initial exploration. Then, for data preprocessing, we removed the commas inside the ‘km’ column of our dataset and converted it into float data type. This is to ensure that the values are in a suitable format for analysis. Furthermore, we also split the data into two parts which are the training and testing data sets. The ‘worn’ column is removed from the DataFrame, thus assigning the other columns to be variable ‘x’. The ‘worn’ column will be set as the variable ‘y’. The testing set is set to be 20% of the entire dataset and a random state, 42 is used for reproducibility.

Coding :

```
In [2]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score
        import warnings
        warnings.filterwarnings('ignore')
        import tensorflow as tf
        from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Dense

In [3]: # Load the dataset from the CSV file
        df = pd.read_csv("C:/Users/Asus/OneDrive - Universiti Teknikal Malaysia Melaka/SEM 4/NN/brake_modified1.csv")

In [4]: df.head()

In [5]: df['km'] = df['km'].str.replace(',', '').astype(float)

In [6]: # Split the data into features (X) and target variable (y)
        X = df.drop('worn', axis=1)
        y = df['worn']

In [7]: # Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

3. Choose and train a model.

Then, we split the data into training and testing sets for model training and evaluation. The training set is used to educate the model patterns and correlations between characteristics and the target variable (brake pad condition).

We have selected TensorFlow framework to build the neural network consisting of input layer, dense layer, and output layer. As the classification is binary classification, thus the output layer used sigmoid function as the activation function to compress the output between 1 and 0.

The model optimizes its prediction performance by doing the iterative training process using backpropagation.

Based on our coding below, we have initialized the model as a feedforward neural network model. The normalization used to normalize the input features, thus there's no input feature that will dominate the learning process.The fit() function, is used to train the model using the training data set, to allow the model to learn patterns and relationships between features and target variables. The training and testing data will also be displayed to observe its values and structure such as the features and target variables used. The model.predict() function has been used by us to predict the target variable for testing data. The predicted values will then be stored in y_pred.

Coding :

```
In [8]: # Normalize the input features
        X_train = (X_train - X_train.mean()) / X_train.std()
        X_test = (X_test - X_test.mean()) / X_test.std()
```

```
In [9]: # Build the ANN model
        model = Sequential()
        model.add(Dense(16, activation='relu', input_shape=(X_train.shape[1],)))
        model.add(Dense(1, activation='sigmoid'))
```

```
In [10]: # Compile the model
         model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
In [11]: # Train the model
         model.fit(X_train, y_train, epochs=50, batch_size=32, verbose=0)
```

```
In [12]: # Print training data
         print("Training Data:")
         print(X_train)
         print(y_train)
```

```
In [13]: # Print testing data
         print("Testing Data:")
         print(X_test)
         print(y_test)
```

```
In [14]: # Test the model and print prediction output
         y_pred_prob = model.predict(X_test)
         y_pred = (y_pred_prob > 0.5).astype(int).flatten()
```

```
In [15]: # Print the predicted output
         print("Prediction Output:")
         print(y_pred)
```

4. Evaluate model

To evaluate the model's performance, we calculated the accuracy of the model. The closer the accuracy to the value of 1, the higher the performance that this model has. The accuracy is calculated by comparing the predicted values to the actual values.

Based on our coding below, we did find the accuracy of the model by comparing the predicted values that we gained using the testing dataset to the actual values. This is done to evaluate the performance of this model.

Coding :

```
In [16]: # Calculate accuracy
         accuracy = accuracy_score(y_test, y_pred)
         print("Accuracy:", accuracy)
```

ANALYSIS AND INTERPRETATION

In this project, we used Python language to develop the predicting model to predict the status of the brake pads whether it is worn or not worn. There are 6 steps of how the analysis was conducted.

1. Data preparation

We read the dataset by using ‘pd.read_csv()’ and thus stored it in a Pandas DataFrame named ‘df’. We removed the commas in the ‘km’ column of the DataFrame and converted the values to float. This is to ensure that the data is in a suitable format for analysis.

2. Data splitting

We split the data into features(X), and target variable (y).The feature for X is km, heat,z and pr while the feature for Y is worn. Next, the data is split into training and testing datasets. 80% for training data and 20% for the testing dataset.

Training Data :

```
Training Data:
           km      heat         z        pr
60  -0.541283 -0.005907 -0.257446  0.627866
227 -0.025667 -1.529785 -0.956495 -1.059136
322  1.732142  1.732497 -1.252211  0.951522
318  0.877414 -0.856615  1.989296  0.951522
17  -0.129118 -0.005907  0.089717  0.951522
..        ...       ...       ...       ...
188 -0.887641 -0.139061 -1.165229 -1.059136
71  -0.295187 -0.242626 -0.202937  0.944396
106 -0.129620  0.216017  0.119258  0.951522
270  0.098475 -0.960180 -0.330665 -1.059136
102  0.111965 -0.301805  0.482051  0.951522

[268 rows x 4 columns]
60     1
227    0
322    1
318    1
17     1
      ..
188    0
71     1
106    1
270    0
102    1
Name: worn, Length: 268, dtype: int64
```

Testing Data :

```
Testing Data:
           km      heat         z        pr
72  -0.218227 -0.076114 -0.081032  0.876205
110  1.613307 -0.814459  2.274494  0.926464
298 -1.164408  2.592166  0.492264 -1.083889
108 -0.218838 -0.083424 -0.079792  0.802218
277 -0.982598 -1.786737  0.962267  0.926464
..        ...       ...       ...       ...
299 -0.849890  0.442921  1.259276  0.926464
295 -0.764023 -1.238460 -1.075627 -1.083889
328  1.566160  1.130095 -1.163767 -1.083889
84  -0.218838 -0.003010 -0.084043  0.802218
245  1.069404  0.925405 -0.991560 -1.083889

[68 rows x 4 columns]
72     1
110    1
298    0
108    1
277    1
      ..
299    1
295    0
328    0
84     1
245    0
Name: worn, Length: 68, dtype: int64
```

3. Model initialization and training

We initialized the feedforward model using “model = Sequential()” and trained the model on training data using “model.fit(X_train, y_train, epochs=50, batch_size=32, verbose=0)”. This step allows the model to learn the patterns and relationships between features and target variables.

4. Data Exploration

Training data consisting (X_train) and corresponding target variable (y_train) is printed to observe values and structure. This is to gain insights into the distribution and relationships between the features and target variables.

```
Training Data:
           km      heat         z        pr
60  -0.541283 -0.005907 -0.257446  0.627866
227 -0.025667 -1.529785 -0.956495 -1.059136
322  1.732142  1.732497 -1.252211  0.951522
318  0.877414 -0.856615  1.989296  0.951522
17  -0.129118 -0.005907  0.089717  0.951522
..        ...       ...       ...       ...
188 -0.887641 -0.139061 -1.165229 -1.059136
71  -0.295187 -0.242626 -0.202937  0.944396
106 -0.129620  0.216017  0.119258  0.951522
270  0.098475 -0.960180 -0.330665 -1.059136
102  0.111965 -0.301805  0.482051  0.951522

[268 rows x 4 columns]
60     1
227    0
322    1
318    1
17     1
      ..
188    0
71     1
106    1
270    0
102    1
Name: worn, Length: 268, dtype: int64
```

5. Testing and prediction

The testing data including the features (X_test) and the target variables (y_test) is printed to examine the values and structure of the testing set. Then we predict the target variable for testing data using 'model.predict(X_test)' via the trained model.

Testing Data :

```
Testing Data:
           km      heat         z        pr
72  -0.218227 -0.076114 -0.081032  0.876205
110  1.613307 -0.814459  2.274494  0.926464
298 -1.164408  2.592166  0.492264 -1.083889
108 -0.218838 -0.083424 -0.079792  0.802218
277 -0.982598 -1.786737  0.962267  0.926464
..        ...       ...       ...       ...
299 -0.849890  0.442921  1.259276  0.926464
295 -0.764023 -1.238460 -1.075627 -1.083889
328  1.566160  1.130095 -1.163767 -1.083889
84  -0.218838 -0.003010 -0.084043  0.802218
245  1.069404  0.925405 -0.991560 -1.083889

[68 rows x 4 columns]
72     1
110    1
298    0
108    1
277    1
      ..
299    1
295    0
328    0
84     1
245    0
Name: worn, Length: 68, dtype: int64
```

Prediction :

```
Prediction Output:
[0 0 0 0 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1 0 1 0 0 0 1 1 0 1 0 1 0 1 1 0 0 0 1
 0 0 0 1 1 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 1 0 1 1 1 1 0 0 0 0 0]
```

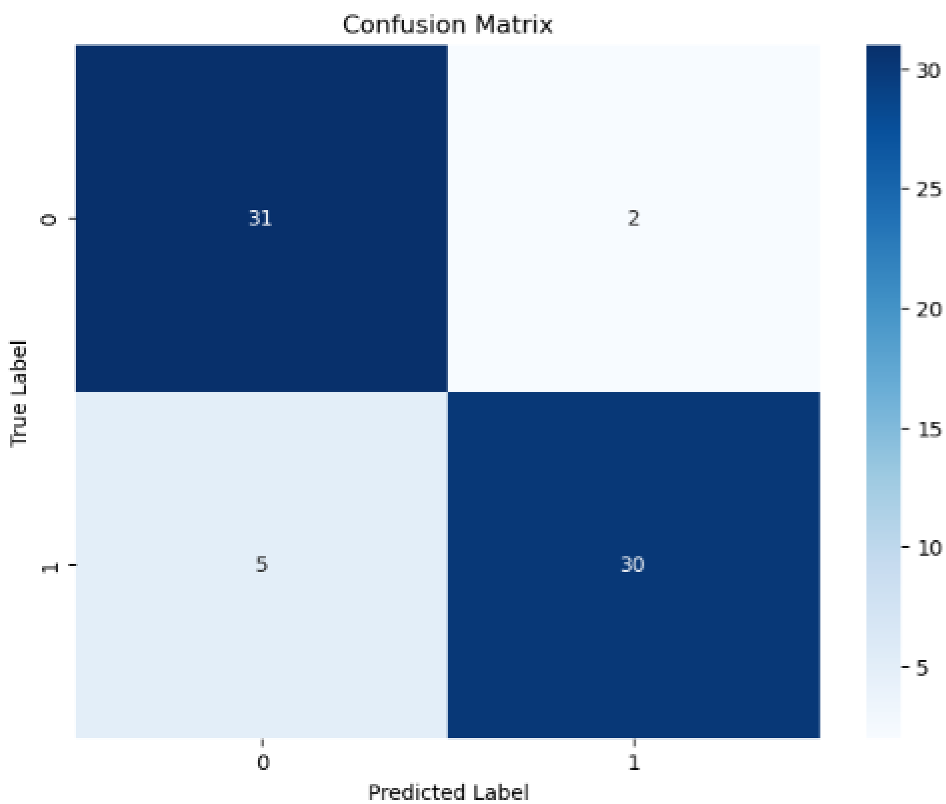
6. Model evaluation

To evaluate our model, we calculated its accuracy using the ‘accuracy_score(y_test,y_pred)’. Based on the output, the accuracy of our model is 0.8970588235294118.
This is by comparing the predicted values of (‘y_pred’) and the actual values (‘y_test’). Then the output has been visualized into confusion matrix to get the summary of the model’s performance. The summary of the performance calculated and summarized in classification report. The report consists of precision, recall(sensitivity), f1-score and support. This report helps us to see the overall model’s performance by giving insights of how the model distinguishes the positive and negative classes.

Output:

```
Accuracy: 0.8970588235294118
```

Confusion matrix:



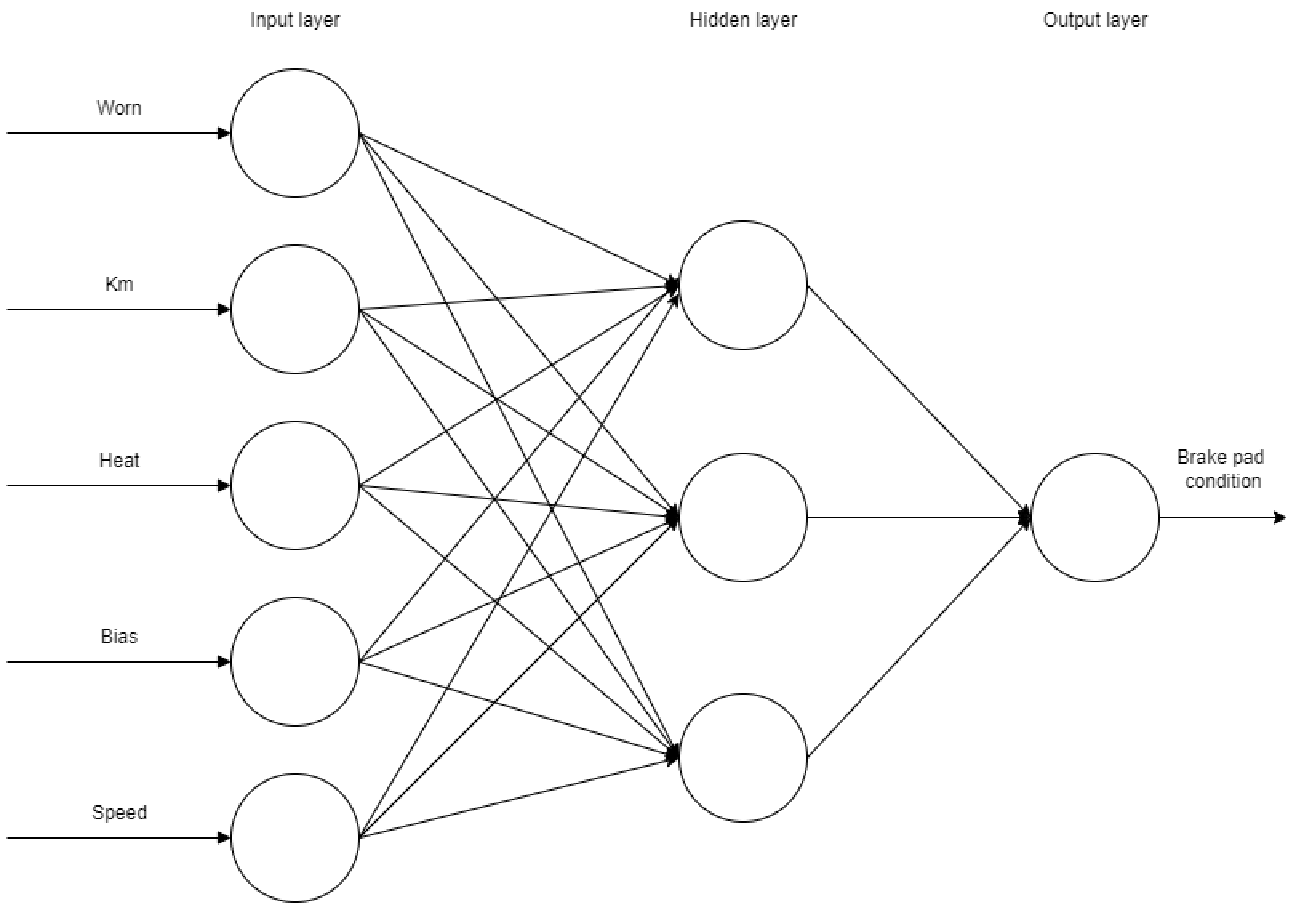
Classification report:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.94      0.90        33
           1       0.94      0.86      0.90        35

    accuracy                           0.90        68
   macro avg       0.90      0.90      0.90        68
weighted avg       0.90      0.90      0.90        68
```

BUSINESS OR INTELLIGENCE MODEL AND FUTURE STUDY



The neural network consists of an input layer on the left, representing the five input features (worn, km, heat, bias, and speed). The inputs are then propagated through one or more hidden layers, where each neuron computes a weighted sum of inputs, applies an activation function, and forwards the output to the subsequent layer. Finally, the information reaches the output layer, which represents the predicted brake pad condition.

During the forward pass, input features are fed into the network, and activations are computed layer by layer. Each neuron's weighted sum of inputs is transformed using an activation function, allowing the network to introduce non-linearity into the model. The output layer generates the predicted brake pad condition based on the learned representations from the hidden layers.

To evaluate the network's performance, we compare the predicted brake pad conditions from the output layer with the actual desired output. The mean squared error (MSE) or other appropriate metrics are employed to quantify the deviation between the predictions and the ground truth, providing insights into the model's accuracy.

After the forward pass, the network calculates the error information by comparing the predicted outputs with the ground truth. This error is then propagated backward through the network, starting with the computation of the error gradients for each neuron in the output layer. The error gradient represents the sensitivity of the output concerning changes in the neuron's activations.

The error gradients are used to calculate the error gradients for neurons in the hidden layers, employing the chain rule. This process accounts for the contribution of subsequent layers and the weights connecting the neurons. Subsequently, the network updates the weights of each layer using an optimization algorithm, such as gradient descent. The weight adjustments are made in a direction that minimizes the error, allowing the network to refine its predictions and improve the accuracy of brake pad condition predictions.

The training process consists of multiple epochs, during which the forward pass, error calculation, backward pass, and weight updates are repeatedly performed. Each epoch refines the network's weights, enabling it to learn from the data and make increasingly accurate predictions.

CONCLUSION

In conclusion, this project shows how predictive analytics and neural network can be applied to the field of preventive maintenance for vehicles, with a focus on maintaining brake pads in large trucks. The project successfully created a model that can forecast when brake pads should be replaced based on variables like heat, distance traveled, and wear rates by leveraging data from IoT sensors and utilizing the power of TensorFlow. By proactively addressing maintenance needs, the implementation of such a predictive maintenance strategy can significantly improve safety, lower the risk of brake failures, and minimize downtime. This project demonstrates how data-driven decision-making can be used to optimize preventive maintenance plans, resulting in increased operational effectiveness and cost savings for businesses that manage vehicles.

REFERENCES

1) Werowe. (n.d.). *GitHub - werowe/mist_preventive_maintenance_ml: This uses HydroSphere to expose a Python machine learning preventive maintenance model for truck brake maintenance*. GitHub. https://github.com/werowe/mist_preventive_maintenance_ml/tree/master
2) Wickersham, H. (2021, October 18). *Brake Maintenance & Safety Tips for Commercial Motor Vehicles - Fleetworthy Solutions*. Fleetworthy Solutions - Beyond Compliant. https://www.fleetworthy.com/compliance-blog/1283/
3) Gailis, M., & Berjoza, D. (2012). On prediction of motor vehicle brake pad wearout. *ResearchGate*. https://www.researchgate.net/publication/288626955_On_prediction_of_motor_vehicle_brake_pad_wearout

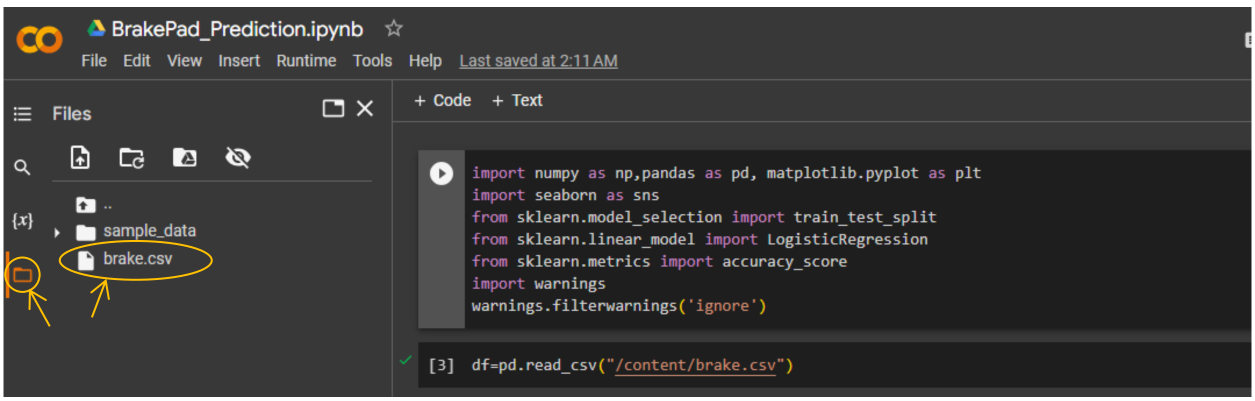
APPENDIX

a) User Manual – Detail step by step to run the program.

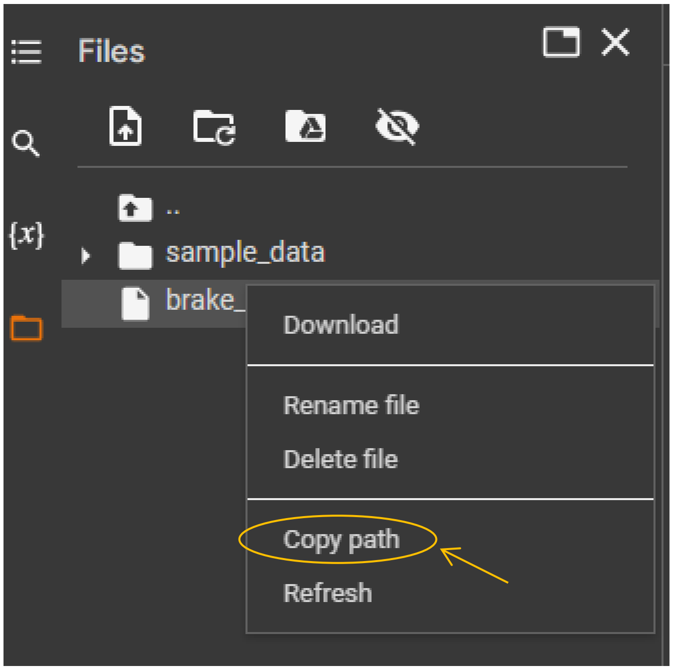
Step 1: click this link to get the codes:

https://colab.research.google.com/drive/1FemWOzVIbsVPu0lgfyy9SJWQxzPqU9oN?usp=sharing

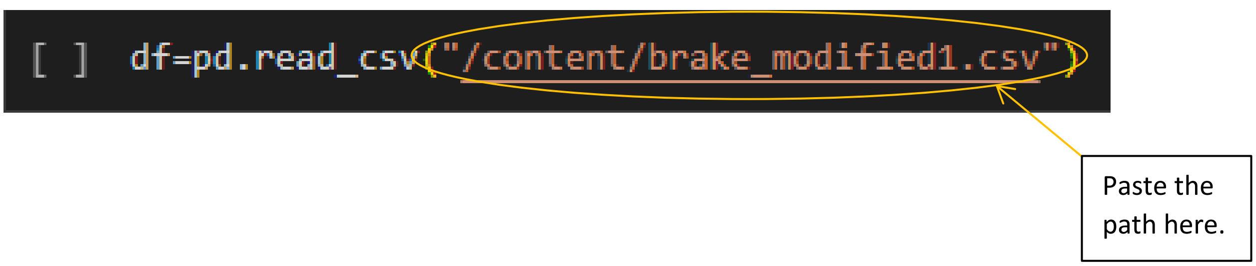
Step 2: Add the dataset file(brake.csv) by clicking on the folder button.



Step 3: Copy the dataset file path.



Step 4: Paste the path in the code below.



Step 5: Run all code by clicking on the run button.

```python
import numpy as np,pandas as pd, matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')
```

The output (if applicable for the segment) will appear after running the code segment.

For example,

```python
df.head()
```

	worn	km	heat	z	pr
0	1	20,000	240	2.720	0.938197
1	0	5,000	98	-57.706	0.000000
2	1	50,000	140	122.420	1.000000
3	0	8,000	260	-45.220	0.000000
4	1	23,790	225	17.835	1.000000

output

Step 6: Run all 17 code segments to get the prediction and the accuracy.

b) Others material

ON PREDICTION OF MOTOR VEHICLE BRAKE PAD WEAROUT

Maris Gailis, Dainis Berjoza
Latvia University of Agriculture
maris@electromobile.lv

Abstract. During vehicle maintenance some components, such as motor oil and filters are replaced when a certain period of time has passed or mileage reached. The braking system components which are subject to friction and wearout have to be replaced when certain minimal thickness is reached. Manufacturers of vehicles provide information about the initial and minimal size of braking components. It is a rare situation if the brake pads or d iscs are found at minimal thickness during inspection at scheduled maintenance. In the real aftersales workshop conditions mechanics have to evaluate the condition of the brake pads and discs and determinate whenever to propose replacement of either component. Dynamics of wear of brake components on vehicles differ from the fro nt or rear axle and inner or outer pad. Cor rect pred iction of the service life can reduce the maintenance costs and probability of unexpected vehicle breakdown. The service life of the brake components depends on many conditions, such as the vehicle model, type o f road, style of driving etc. The road conditions in Latvia differ from other European countries with high ratio of unpaved roads. Some of the vehicles sold in Latvia have maximum maintenance periodicity up to 40 000 km but brake pads are wearing out at 30 000 km. Therefore, a correct model of prediction o f brake component wear for the local conditions can only be b ased on the data collected from the local data sources. The pap er presents a model of the service life of b rake components for 3 different types of vehicles manufactured by the French automobile producer Renault. The model is based on regression analysis of the data collected at workshops of the manufacturer's official dealerships. The samples were selected from the vehicles belonging to randomly chosen owners to include most of population.

Keywords: automobile, periodicity, maintenance, brake systems.

Above is the abstract of a journal about the prediction of motor vehicle brake pad wear out. The sample of the research is selected randomly chosen vehicle owners in France.

c) Sample of dataset

1	worn	km	heat	z	pr
2	1	20,000	240	2.72	0.938197
3	0	5,000	98	-57.706	0
4	1	50,000	140	122.42	1
5	0	8,000	260	-45.22	0
6	1	23,790	225	17.835	1
7	1	24,644	245	21.311	1
8	1	29,934	195	42.321	1
9	0	14,045	153	-21.361	0
10	0	8,000	222	-45.334	0

d) Video presentation link
https://youtu.be/81bLkSeGqsk