agile-project-estimation

March 23, 2024

[10]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from sklearn.metrics import mean_squared_error
```

[11]:
```
data = pd.read_csv('issue_data.csv')
data.head()
```

```
/tmp/ipykernel_2316/214401849.py:1: DtypeWarning: Columns (24) have mixed types.
Specify dtype option on import or set low_memory=False.
  data = pd.read_csv('issue_data.csv')
```

[11]:

	ID	Jira_ID	Issue_Key	URL	Title	Description	Description_Text
0	65	77638	XD-3768	https://jira.spring.io/rest/api/2/issue/77638	"How do I make a job restartable in spring xd"	"The jobs that appear under Executions section…	"""The jobs that appear under Executions secti…
1	66	77511	XD-3767	https://jira.spring.io/rest/api/2/issue/77511	"admin config timezone command does not work"	"Working with Spring-XD version 1.3.2.RELEASE …	
2	67	77130	XD-3766	https://jira.spring.io/rest/api/2/issue/77130	"Module Upload command not pushing jar to all …	"My project 7 node cluster and in that 2 node …	
3	68	71950	XD-3765	https://jira.spring.io/rest/api/2/issue/71950	"Fix stream failover "	"See https://github.com/spring-projects/spring…	
4	69	71805	XD-3764	https://jira.spring.io/rest/api/2/issue/71805	"SpringXD Job is still executing even after fo…	"I'm trying to run a Job on SpringXD and the j…	

```
1  """Working with Spring-XD version 1.3.2.RELEAS…
2  """My project 7 node cluster and in that 2 nod…
3  """See https://github.com/spring-projects/spri…
4  """I'm trying to run a Job on SpringXD and the…

                                  Description_Code   Type Priority  …  \
0                                              NaN    Bug    Major  …
1  " xd:>admin config admin config info        ad…    Bug  Trivial  …
2                                              NaN    Bug    Major  …
3                                              NaN  Story    Minor  …
4                                              NaN    Bug    Major  …

  Resolution_Time_Minutes Title_Changed_After_Estimation  \
0                     0.0                              0
1                     0.0                              0
2                     0.0                              0
3                     0.0                              1
4                     0.0                              0

  Description_Changed_After_Estimation Story_Point_Changed_After_Estimation  \
0                                    0                                    0
1                                    0                                    0
2                                    0                                    0
3                                    0                                    0
4                                    0                                    0

  Pull_Request_URL Creator_ID  Reporter_ID  Assignee_ID  Project_ID  Sprint_ID
0              NaN       68.0         68.0          NaN           1        NaN
1              NaN       69.0         69.0          NaN           1        NaN
2              NaN       70.0         70.0          NaN           1        NaN
3              NaN       72.0         72.0         71.0           1        NaN
4              NaN       73.0         73.0          NaN           1        NaN

[5 rows x 30 columns]
```

[12]:
```
cols = pd.DataFrame(data.isnull().mean().round(4) * 100, columns␣
 ↪=['precentage_missing_value']).sort_values(by=['precentage_missing_value'])
print(cols)
```

```
                                  precentage_missing_value
ID                                                    0.00
In_Progress_Minutes                                   0.00
Title_Changed_After_Estimation                        0.00
Description_Changed_After_Estimation                  0.00
Last_Updated                                          0.00
Project_ID                                            0.00
Story_Point_Changed_After_Estimation                  0.00
```

```
Creation_Date                                 0.00
Total_Effort_Minutes                          0.00
Status                                        0.00
Type                                          0.00
Title                                         0.00
URL                                           0.00
Issue_Key                                     0.00
Jira_ID                                       0.00
Resolution_Time_Minutes                       0.00
Creator_ID                                    0.35
Reporter_ID                                   0.54
Description_Text                              6.36
Description                                   6.36
Resolution                                   16.91
Resolution_Date                              23.05
Priority                                     27.64
Assignee_ID                                  42.86
Story_Point                                  85.72
Estimation_Date                              85.72
Description_Code                             87.12
Sprint_ID                                    90.51
Timespent                                    97.76
Pull_Request_URL                             99.86
```

[13]:
```
column_types = data.dtypes
print(column_types)
```

```
ID                                int64
Jira_ID                           int64
Issue_Key                        object
URL                              object
Title                            object
Description                      object
Description_Text                 object
Description_Code                 object
Type                             object
Priority                         object
Status                           object
Resolution                       object
Creation_Date                    object
Estimation_Date                  object
Resolution_Date                  object
Last_Updated                     object
Story_Point                     float64
Timespent                       float64
In_Progress_Minutes             float64
Total_Effort_Minutes            float64
Resolution_Time_Minutes         float64
```

```
Title_Changed_After_Estimation            int64
Description_Changed_After_Estimation      int64
Story_Point_Changed_After_Estimation      int64
Pull_Request_URL                         object
Creator_ID                              float64
Reporter_ID                             float64
Assignee_ID                             float64
Project_ID                                int64
Sprint_ID                               float64
dtype: object
```

[14]:
```
df = data[[ 'Type','Priority','Title_Changed_After_Estimation' ,␣
 ↪'Description_Changed_After_Estimation',␣
 ↪'Story_Point_Changed_After_Estimation',  'Story_Point',␣
 ↪'Resolution_Time_Minutes']]
```

[15]:
```
df = df.dropna()
df = df[df['Resolution_Time_Minutes'] != 0]
```

[16]:
```
df.head()
```

[16]:

	Type	Priority	Title_Changed_After_Estimation	Description_Changed_After_Estimation	Story_Point_Changed_After_Estimation	Story_Point	Resolution_Time_Minutes
20	Bug	Minor	0	0	0	1.0	436558.0
21	Improvement	Minor	1	0	0	1.0	437537.0
24	Improvement	Minor	0	0	0	1.0	194.0
25	Bug	Minor	0	0	0	1.0	5039.0
26	Improvement	Minor	1	0	0	3.0	4625.0

[17]:
```
df.describe()
df.shape
```

[17]:
```
(38239, 7)
```

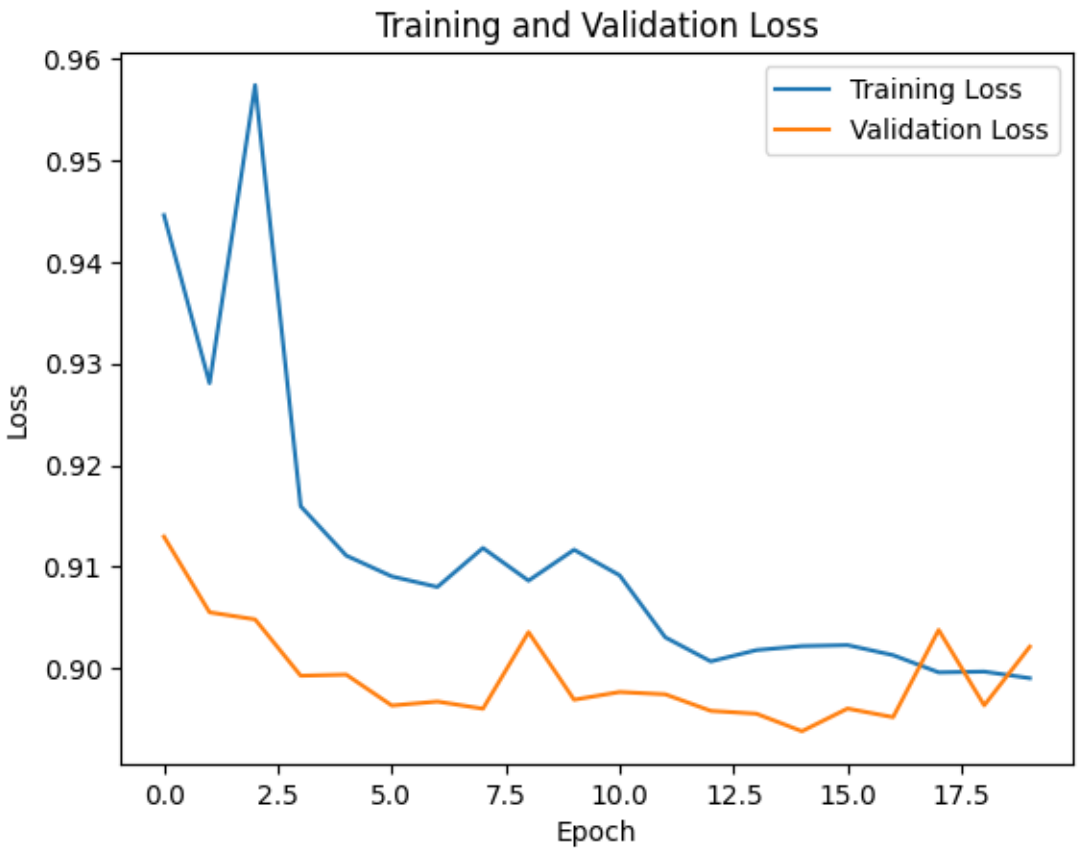
[23]:
```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer

X = df.drop('Resolution_Time_Minutes', axis=1)
y = df['Resolution_Time_Minutes']
categorical_columns = ['Type','Priority']
numerical_columns = ['Title_Changed_After_Estimation',␣
 ↪'Description_Changed_After_Estimation',␣
 ↪'Story_Point_Changed_After_Estimation', 'Story_Point']
# Split the data into training, validation, and testing sets
X_train_val, X_test, y_train_val, y_test = train_test_split(X, y, test_size=0.
 ↪2, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train_val, y_train_val,␣
 ↪test_size=0.25, random_state=42)  # 0.25 x 0.8 = 0.2
# Preprocess categorical columns using one-hot encoding
preprocessor = ColumnTransformer([
    ('encoder', OneHotEncoder(handle_unknown='ignore'), categorical_columns),
    ('scaler', StandardScaler(), numerical_columns)
])

X_train_processed = preprocessor.fit_transform(X_train)
X_val_processed = preprocessor.transform(X_val)
X_test_processed = preprocessor.transform(X_test)


# Normalize the target variable
scaler_y = StandardScaler()
y_train_normalized = scaler_y.fit_transform(y_train.values.reshape(-1, 1))
y_val_normalized = scaler_y.transform(y_val.values.reshape(-1, 1))
y_test_normalized = scaler_y.transform(y_test.values.reshape(-1, 1))

# Define the neural network model
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(X_train_processed.
 ↪shape[1],)),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(1)
])

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model with early stopping
```

```
early_stopping = tf.keras.callbacks.EarlyStopping(patience=5,␣
 ↪restore_best_weights=True)
history = model.fit(X_train_processed, y_train_normalized, epochs=100,␣
 ↪batch_size=32, validation_data=(X_val_processed, y_val_normalized),␣
 ↪callbacks=[early_stopping])
```

```
Epoch 1/100
717/717 [==============================] - 7s 8ms/step - loss: 0.9446 -
val_loss: 0.9130
Epoch 2/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9281 -
val_loss: 0.9055
Epoch 3/100
717/717 [==============================] - 5s 8ms/step - loss: 0.9574 -
val_loss: 0.9048
Epoch 4/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9160 -
val_loss: 0.8993
Epoch 5/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9111 -
val_loss: 0.8994
Epoch 6/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9091 -
val_loss: 0.8964
Epoch 7/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9080 -
val_loss: 0.8968
Epoch 8/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9119 -
val_loss: 0.8961
Epoch 9/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9086 -
val_loss: 0.9036
Epoch 10/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9117 -
val_loss: 0.8969
Epoch 11/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9092 -
val_loss: 0.8977
Epoch 12/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9031 -
val_loss: 0.8975
Epoch 13/100
717/717 [==============================] - 6s 8ms/step - loss: 0.9007 -
val_loss: 0.8958
Epoch 14/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9018 -
```

```
val_loss: 0.8956
Epoch 15/100
717/717 [==============================] - 6s 8ms/step - loss: 0.9022 -
val_loss: 0.8938
Epoch 16/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9023 -
val_loss: 0.8961
Epoch 17/100
717/717 [==============================] - 5s 7ms/step - loss: 0.9013 -
val_loss: 0.8952
Epoch 18/100
717/717 [==============================] - 5s 7ms/step - loss: 0.8996 -
val_loss: 0.9038
Epoch 19/100
717/717 [==============================] - 6s 8ms/step - loss: 0.8997 -
val_loss: 0.8964
Epoch 20/100
717/717 [==============================] - 6s 8ms/step - loss: 0.8991 -
val_loss: 0.9022
```

[24]:
```python
import matplotlib.pyplot as plt
# Plotting the Training and Validationi Loss
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Training and Validation Loss')
plt.legend()
plt.show()
```



[25]:
```
# Evaluate the model
loss = model.evaluate(X_test_processed, y_test_normalized)
print(f'Loss: {loss}')
# Evaluate the model
loss = model.evaluate(X_test_processed, y_test_normalized)
print(f'Test Loss: {loss}')
```

```
239/239 [==============================] - 1s 4ms/step - loss: 0.9456
Loss: 0.9455568790435791
239/239 [==============================] - 1s 4ms/step - loss: 0.9456
Test Loss: 0.9455568790435791
```

[ ]: