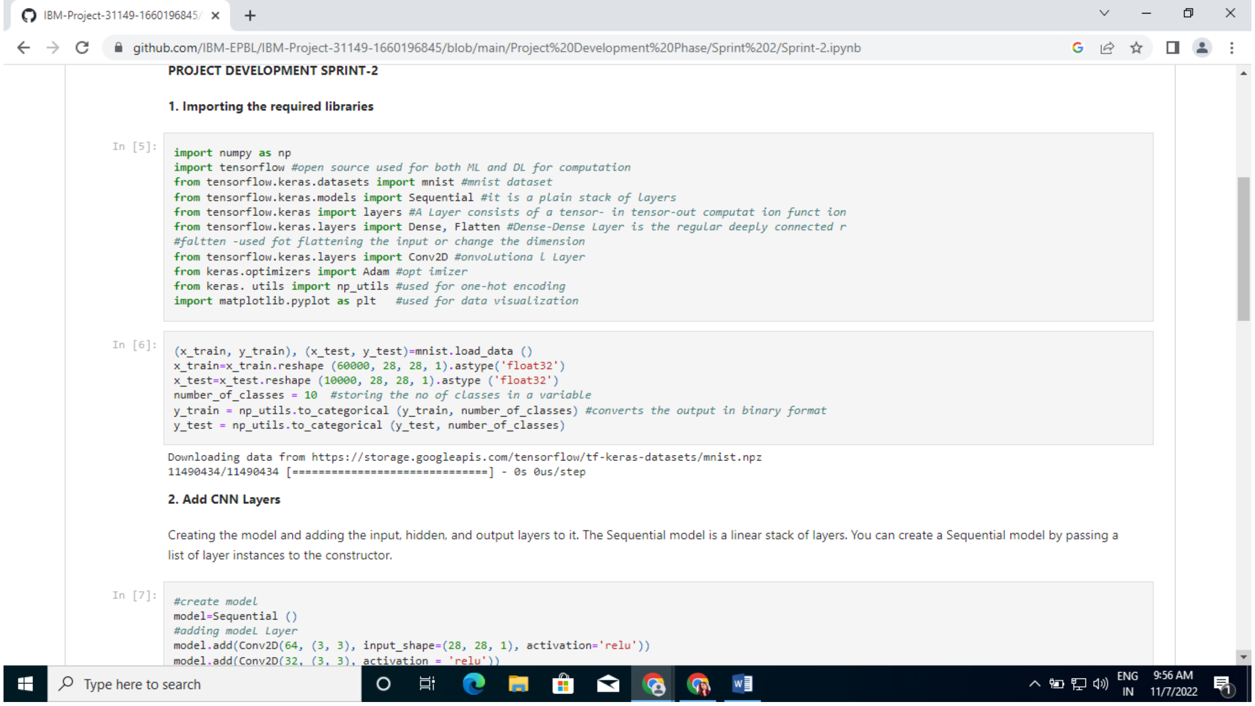
Date	05 NOV 2022
Project Name	A Novel method for Handwritten digit recognition system
Team id	PNT2022TMID46193


IBM-Project-31149-1660196845/
github.com/IBM-EPBL/IBM-Project-31149-1660196845/blob/main/Project%20Development%20Phase/Sprint%202/Sprint-2.ipynb
Type here to search
ENG
IN
9:56 AM
11/7/2022

PROJECT DEVELOPMENT SPRINT-2

1. Importing the required libraries

In [5]:

```
import numpy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt   #used for data visualization
```

In [6]:

```
(x_train, y_train), (x_test, y_test)=mnist.load_data ()
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
number_of_classes = 10  #storing the no of classes in a variable
y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in binary format
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

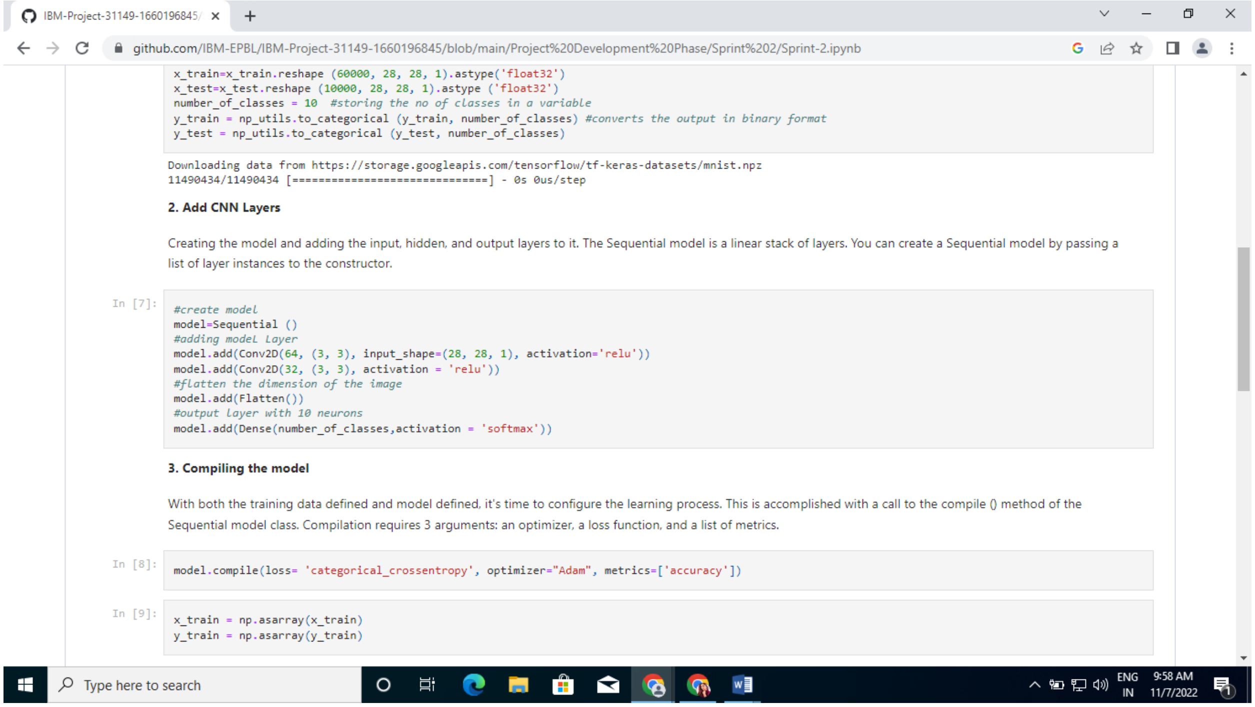
```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

2. Add CNN Layers

Creating the model and adding the input, hidden, and output layers to it. The Sequential model is a linear stack of layers. You can create a Sequential model by passing a list of layer instances to the constructor.

In [7]:

```
#create model
model=Sequential ()
#adding model Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))
```


IBM-Project-31149-1660196845/
github.com/IBM-EPBL/IBM-Project-31149-1660196845/blob/main/Project%20Development%20Phase/Sprint%202/Sprint-2.ipynb
Type here to search
ENG
IN
9:58 AM
11/7/2022

```
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
number_of_classes = 10  #storing the no of classes in a variable
y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in binary format
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

2. Add CNN Layers

Creating the model and adding the input, hidden, and output layers to it. The Sequential model is a linear stack of layers. You can create a Sequential model by passing a list of layer instances to the constructor.

In [7]:

```
#create model
model=Sequential ()
#adding model Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D(32, (3, 3), activation = 'relu'))
#flatten the dimension of the image
model.add(Flatten())
#output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

3. Compiling the model

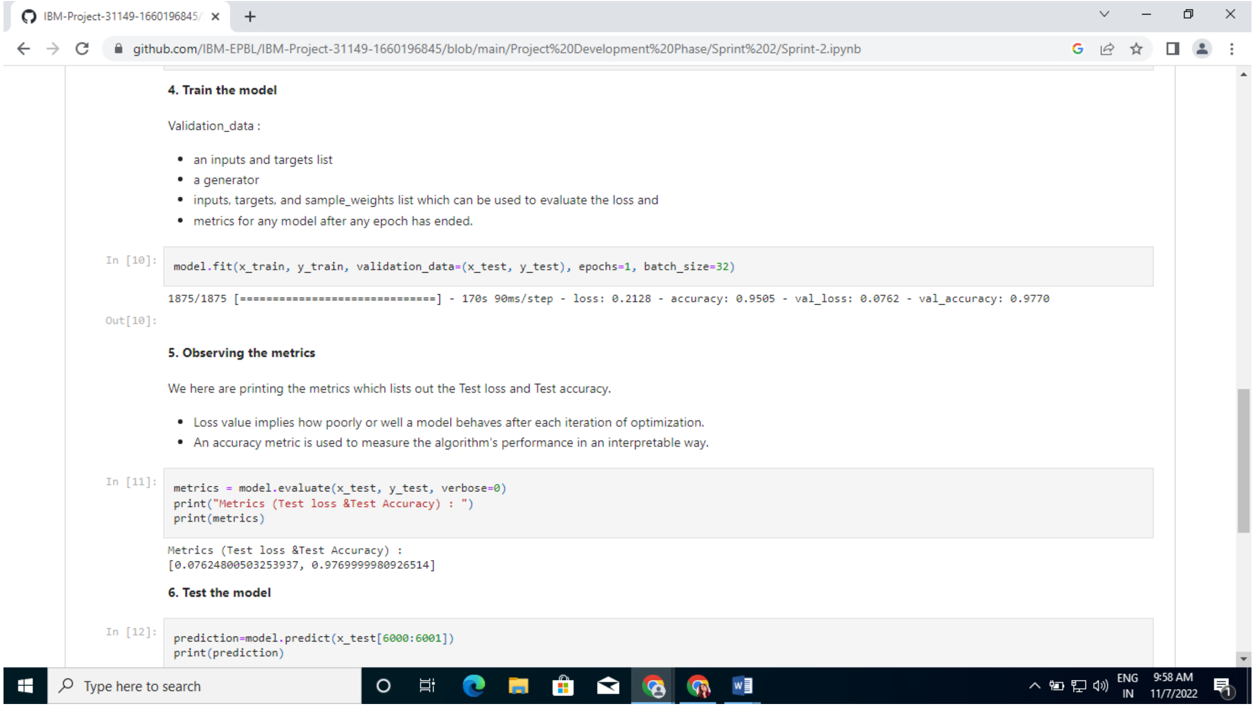
With both the training data defined and model defined, it's time to configure the learning process. This is accomplished with a call to the compile () method of the Sequential model class. Compilation requires 3 arguments: an optimizer, a loss function, and a list of metrics.

In [8]:

```
model.compile(loss= 'categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

In [9]:

```
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```


IBM-Project-31149-1660196845/
github.com/IBM-EPBL/IBM-Project-31149-1660196845/blob/main/Project%20Development%20Phase/Sprint%202/Sprint-2.ipynb
Type here to search
ENG
IN
9:58 AM
11/7/2022

4. Train the model

Validation_data :

- an inputs and targets list
- a generator
- inputs, targets, and sample_weights list which can be used to evaluate the loss and
- metrics for any model after any epoch has ended.

```
In [10]: model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=1, batch_size=32)

         1875/1875 [==============================] - 170s 90ms/step - loss: 0.2128 - accuracy: 0.9505 - val_loss: 0.0762 - val_accuracy: 0.9770
Out[10]:
```

5. Observing the metrics

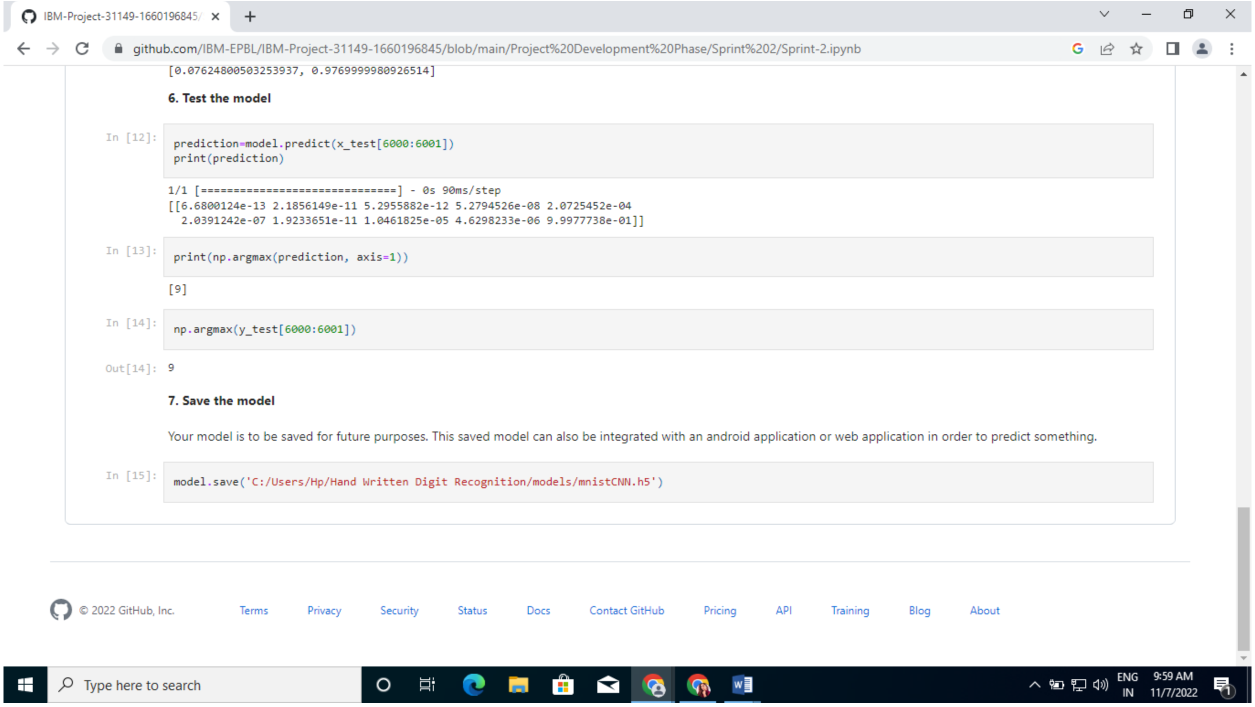
We here are printing the metrics which lists out the Test loss and Test accuracy.

- Loss value implies how poorly or well a model behaves after each iteration of optimization.
- An accuracy metric is used to measure the algorithm's performance in an interpretable way.

```
In [11]: metrics = model.evaluate(x_test, y_test, verbose=0)
         print("Metrics (Test loss &Test Accuracy) : ")
         print(metrics)

         Metrics (Test loss &Test Accuracy) :
         [0.07624800503253937, 0.9769999980926514]
```

6. Test the model

```
In [12]: prediction=model.predict(x_test[6000:6001])
         print(prediction)
```


IBM-Project-31149-1660196845/
github.com/IBM-EPBL/IBM-Project-31149-1660196845/blob/main/Project%20Development%20Phase/Sprint%202/Sprint-2.ipynb
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About
Type here to search
ENG
IN
9:59 AM
11/7/2022

[0.07624800503253937, 0.9769999980926514]

6. Test the model

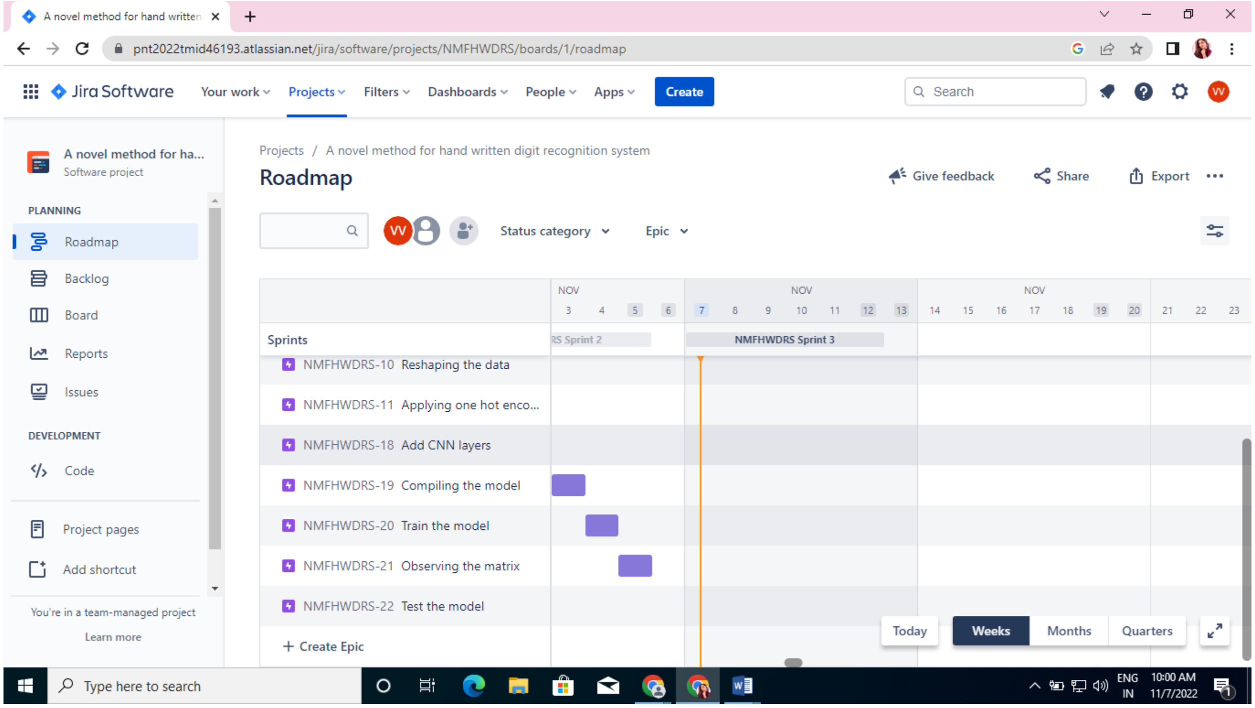
```
In [12]: prediction=model.predict(x_test[6000:6001])
         print(prediction)

         1/1 [==============================] - 0s 90ms/step
         [[6.6800124e-13 2.1856149e-11 5.2955882e-12 5.2794526e-08 2.0725452e-04
           2.0391242e-07 1.9233651e-11 1.0461825e-05 4.6298233e-06 9.9977738e-01]]

In [13]: print(np.argmax(prediction, axis=1))

         [9]

In [14]: np.argmax(y_test[6000:6001])

Out[14]: 9
```

7. Save the model

Your model is to be saved for future purposes. This saved model can also be integrated with an android application or web application in order to predict something.

```
In [15]: model.save('C:/Users/Hp/Hand Written Digit Recognition/models/mnistCNN.h5')
```

Road Map:



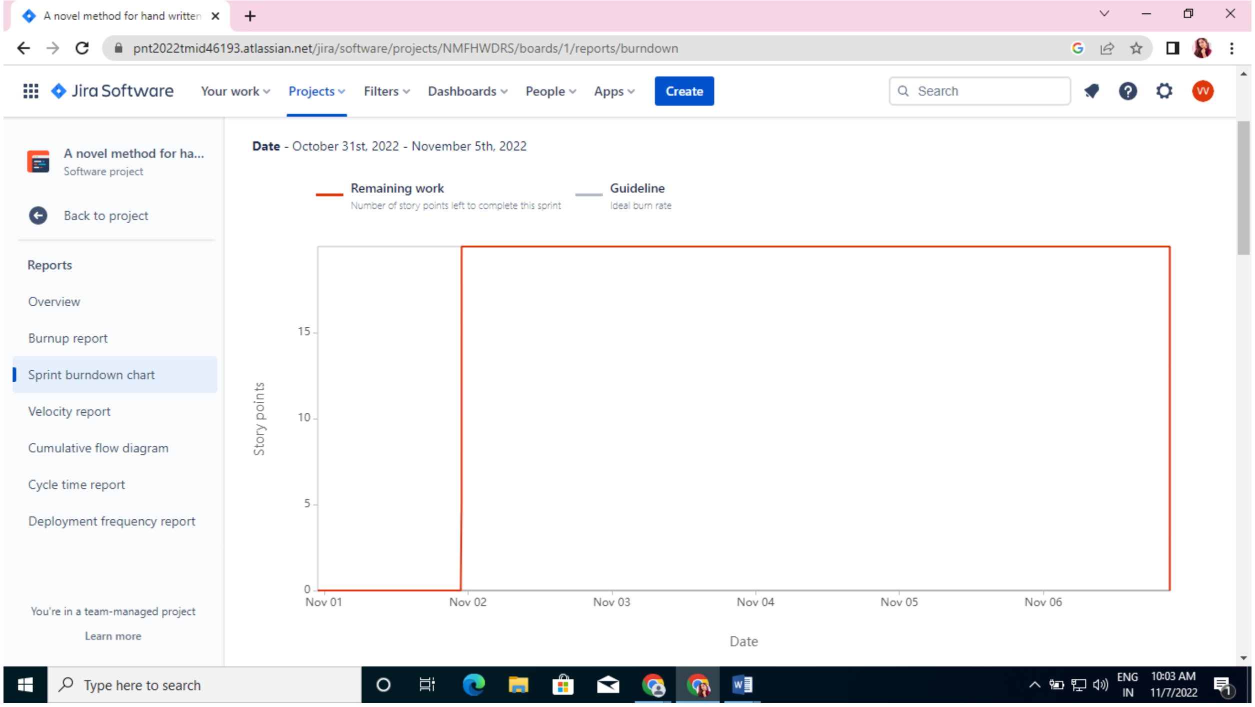
Projects / A novel method for hand written digit recognition system

Roadmap

- NMFHWDRS-10 Reshaping the data
- NMFHWDRS-11 Applying one hot enco...
- NMFHWDRS-18 Add CNN layers
- NMFHWDRS-19 Compiling the model
- NMFHWDRS-20 Train the model
- NMFHWDRS-21 Observing the matrix
- NMFHWDRS-22 Test the model

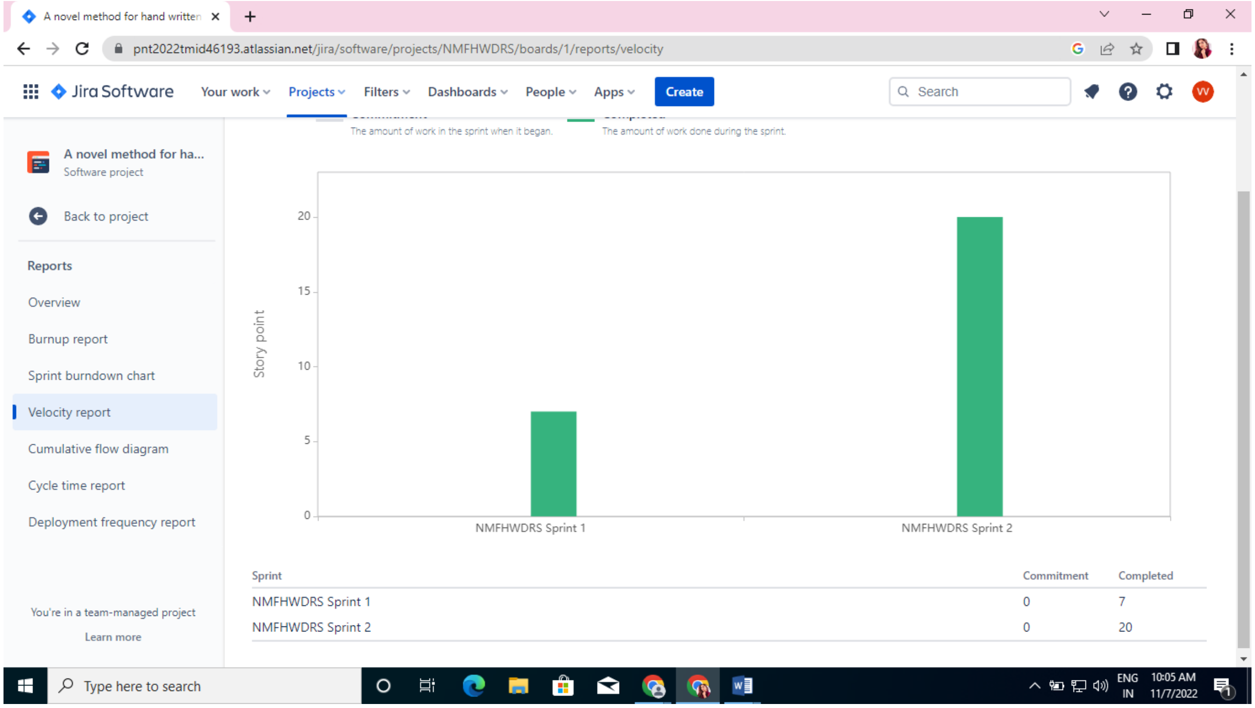
+ Create Epic

Sprint Burndown chart:



Date - October 31st, 2022 - November 5th, 2022

Velocity chart:



Sprint	Commitment	Completed
NMFHWDRS Sprint 1	0	7
NMFHWDRS Sprint 2	0	20