

```
In [4]: import yfinance as yf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense
```

```
In [5]: # Download Google stock data (GOOG) for the last 5 years
print("Downloading stock data...")
data = yf.download('GOOG', start='2018-01-01', end='2024-01-01')
data_close = data[['Close']]
```

Downloading stock data...

YF.download() has changed argument auto_adjust default to True

[*****100%*****] 1 of 1 completed

```
In [6]: # Normalize the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data_close)
```

```
In [9]: # Create sequences
def create_sequences(data, seq_length):
    x, y = [], []
    for i in range(seq_length, len(data)):
        x.append(data[i-seq_length:i, 0])
        y.append(data[i, 0])
    return np.array(x), np.array(y)
```

```
In [10]: sequence_length = 60
X, y = create_sequences(scaled_data, sequence_length)
```

```
In [11]: # Reshape for RNN input
X = np.reshape(X, (X.shape[0], X.shape[1], 1))
```

```
In [12]: # Train-Test Split
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
y_train, y_test = y[:train_size], y[train_size:]
```

```
In [13]: # Build the RNN model
model = Sequential([
    SimpleRNN(units=50, return_sequences=False, input_shape=(X_train.shape[1], 1)),
    Dense(1) ])
```

WARNING:tensorflow:From C:\Users\SOFT LAB11\anaconda3\Lib\site-packages\keras\src\layers\rnn\simple_rnn.py:130: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

```
In [14]: model.compile(optimizer='adam', loss='mean_squared_error')
model.summary()
```

WARNING:tensorflow:From C:\Users\SOFT LAB11\anaconda3\Lib\site-packages\keras\src\optimizers__init__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
simple_rnn (SimpleRNN)	(None, 50)	2600
dense (Dense)	(None, 1)	51
=====		
Total params: 2651 (10.36 KB)		
Trainable params: 2651 (10.36 KB)		
Non-trainable params: 0 (0.00 Byte)		
=====		

```
In [15]: # Train the model
print("Training the model...")
history = model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_
```

Training the model...

Epoch 1/20

WARNING:tensorflow:From C:\Users\SOFT LAB11\anaconda3\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

37/37 [=====] - 2s 12ms/step - loss: 0.0112 - val_loss: 0.0019

Epoch 2/20

37/37 [=====] - 0s 5ms/step - loss: 0.0011 - val_loss: 0.0016

Epoch 3/20

37/37 [=====] - 0s 5ms/step - loss: 9.3004e-04 - val_loss: 0.0016

Epoch 4/20

37/37 [=====] - 0s 5ms/step - loss: 8.5241e-04 - val_loss: 0.0015

Epoch 5/20

37/37 [=====] - 0s 5ms/step - loss: 8.6069e-04 - val_loss: 0.0012

Epoch 6/20

37/37 [=====] - 0s 6ms/step - loss: 7.1656e-04 - val_loss: 0.0012

Epoch 7/20

37/37 [=====] - 0s 6ms/step - loss: 6.8128e-04 - val_loss: 0.0013

Epoch 8/20

37/37 [=====] - 0s 6ms/step - loss: 6.6665e-04 - val_loss: 0.0011

Epoch 9/20

37/37 [=====] - 0s 6ms/step - loss: 6.1728e-04 - val_loss: 0.0013

Epoch 10/20

37/37 [=====] - 0s 5ms/step - loss: 5.8468e-04 - val_loss: 9.7133e-04

Epoch 11/20

37/37 [=====] - 0s 6ms/step - loss: 5.6347e-04 - val_loss: 9.6573e-04

Epoch 12/20

37/37 [=====] - 0s 6ms/step - loss: 6.0508e-04 - val_loss: 0.0017

Epoch 13/20

37/37 [=====] - 0s 6ms/step - loss: 6.7146e-04 - val_loss: 9.2999e-04

Epoch 14/20

37/37 [=====] - 0s 5ms/step - loss: 6.2422e-04 - val_loss: 9.4788e-04

Epoch 15/20

37/37 [=====] - 0s 5ms/step - loss: 5.1212e-04 - val_loss: 9.0562e-04

Epoch 16/20

37/37 [=====] - 0s 6ms/step - loss: 5.3505e-04 - val_loss: 8.2649e-04

Epoch 17/20

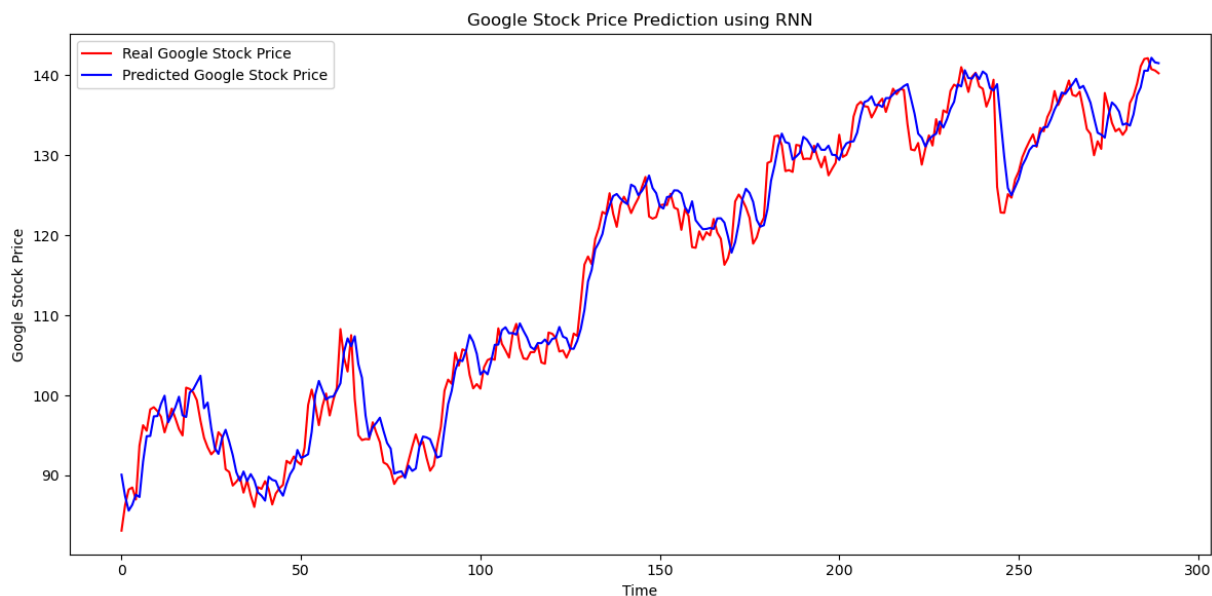
37/37 [=====] - 0s 6ms/step - loss: 4.7620e-04 - val_loss: 8.1822e-04

```
Epoch 18/20
37/37 [=====] - 0s 6ms/step - loss: 4.7720e-04 - val_loss: 9.2428e-04
Epoch 19/20
37/37 [=====] - 0s 6ms/step - loss: 5.1031e-04 - val_loss: 7.6162e-04
Epoch 20/20
37/37 [=====] - 0s 6ms/step - loss: 4.7932e-04 - val_loss: 7.9718e-04
```

```
In [16]: # Predict
print("Predicting stock prices...")
predicted_stock_price = model.predict(X_test)
predicted_stock_price = scaler.inverse_transform(predicted_stock_price.reshape(-1,
real_stock_price = scaler.inverse_transform(y_test.reshape(-1, 1)))
```

```
Predicting stock prices...
10/10 [=====] - 0s 2ms/step
```

```
In [17]: # Plotting results
plt.figure(figsize=(12,6))
plt.plot(real_stock_price, color='red', label='Real Google Stock Price')
plt.plot(predicted_stock_price, color='blue', label='Predicted Google Stock Price')
plt.title('Google Stock Price Prediction using RNN')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
plt.tight_layout()
plt.show()
```



```
In [ ]:
```