Project Overview

--Your goal is to utilize each market index(Dow, S&P, Nasdaq, Russell) for a 20 year period to predict the 10-Year Treasury Constant Maturity Rate - Does the market(s) provide any insight or predictive capabilities on the 10 year rate. -Test your model on data that the model has not seen. How accurate is your model? -Get the appropriate metrics to measure the model accuracy

Project approach

- Get 20 years of data from yahoo using the data reader api
- download the 10 year Maturity rate data from [https://fred.stlouisfed.org/series/DGS10 (https://fred.stlouisfed.org/series/DGS10)](https://fred.stlouisfed.org/series/DGS10) from 2000 to current
- clean any missing data from the data set
- check the correlation of the data set
- Use a neural network to see if the Adj Close of the indices can predict the value of the 10 year maturity rate

In [7]:
```
tickers = ["^GSPC", "^DJI", "^RUT", "^IXIC"]
```

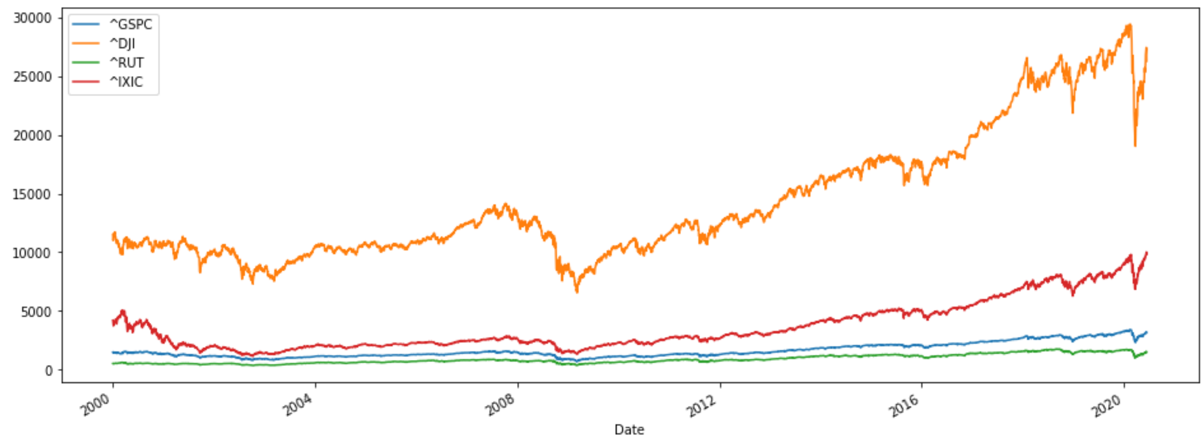
In [8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pandas_datareader.data as web
import seaborn as sns
```

In [9]:
```
index_data = pd.DataFrame()
```

In [10]:
```
for t in tickers:
    index_data[t] = web.DataReader(t, data_source="yahoo", start = "2000-01-03", end = "2020-06-11")["Open"]
```

In [11]:
```
index_data.plot(figsize = (16,6))
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1765a8d0>



Importing 10 year maturity Data

- Date will be from 2000-01-03 to 2020-06-11
- will check for missing data and replace with appropriate values

In [12]:
```
index_data
```

Out[12]:

Date	^GSPC	^DJI	^RUT	^IXIC
2000-01-03	1469.250000	11501.849609	504.760010	4186.189941
2000-01-04	1455.219971	11349.750000	497.049988	4020.000000
2000-01-05	1399.420044	10989.370117	478.380005	3854.350098
2000-01-06	1402.109985	11113.370117	478.829987	3834.439941
2000-01-07	1403.449951	11247.059570	475.339996	3711.090088
...	...	...	...	...
2020-06-05	3163.840088	26836.800781	1460.180054	9703.540039
2020-06-08	3199.919922	27232.929688	1510.589966	9823.440430
2020-06-09	3213.320068	27447.369141	1533.030029	9867.190430
2020-06-10	3213.419922	27251.890625	1506.939941	10012.320312
2020-06-11	3123.530029	26282.509766	1458.339966	9791.240234

5143 rows × 4 columns

In [13]:
```
data = pd.read_csv("resources/10year_treasury.csv")
```

In [14]:
```
data["DATE"] = pd.to_datetime(data["DATE"])
```

In [15]:
```
data.set_index('DATE', inplace=True)
```

In [16]:
```
data.replace(["."], np.nan, inplace = True)
```

In [17]:
```
data["DGS10"] = pd.to_numeric(data["DGS10"])
```

In [18]:
```
data
```

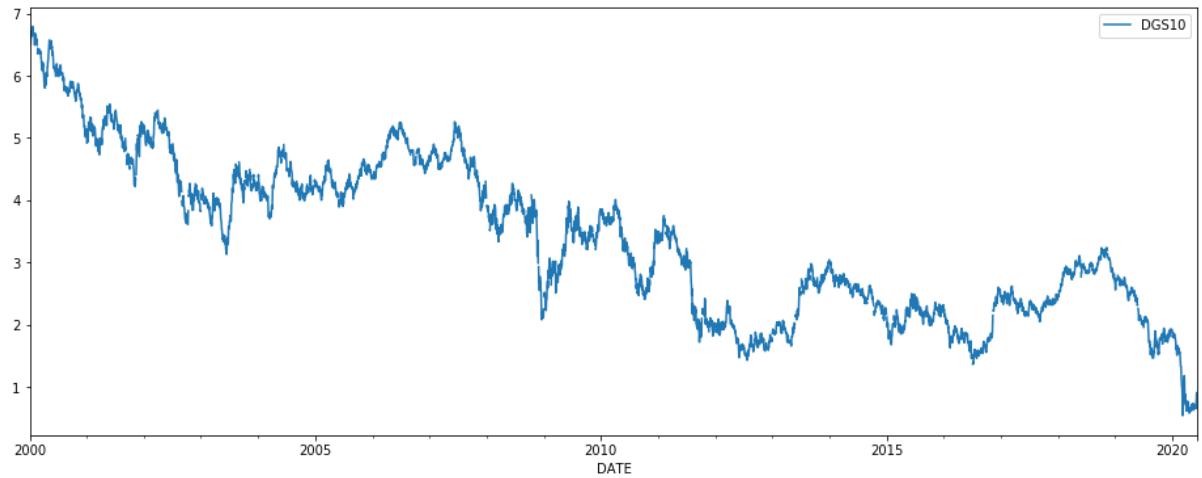
Out[18]:

	DGS10
DATE	
2000-01-03	6.58
2000-01-04	6.49
2000-01-05	6.62
2000-01-06	6.57
2000-01-07	6.52
...	...
2020-06-05	0.91
2020-06-08	0.88
2020-06-09	0.84
2020-06-10	0.75
2020-06-11	0.66

5334 rows × 1 columns

In [19]:
```
data.plot(figsize = (16,6))
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1720f410>



Concating Data

- Bringing the market data and 10 year maturity together
- We can see that there is a negative correlation betwee nthe markets and the 10 year constant rate

In [20]:
```
working_data = pd.concat([index_data.dropna(), data], axis=1)
```

In [21]:
```
working_data.corr()
```

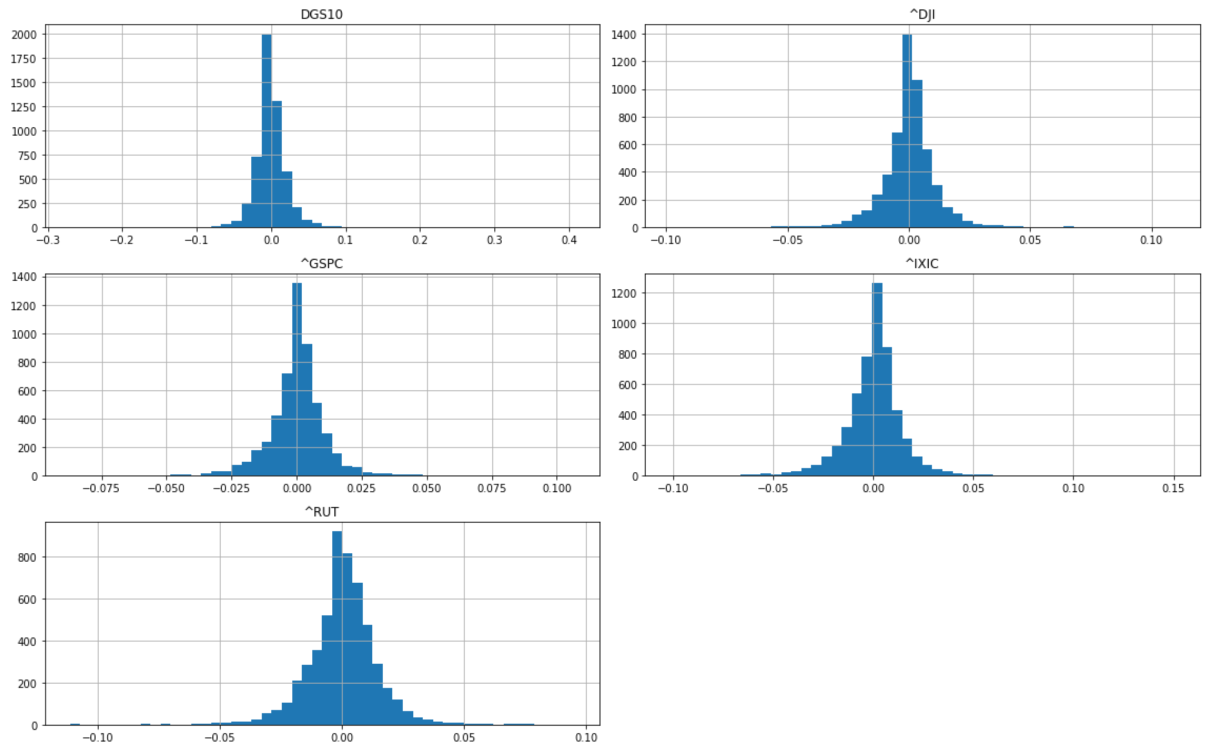
Out[21]:

	^GSPC	^DJI	^RUT	^IXIC	DGS10
^GSPC	1.000000	0.992026	0.961328	0.985083	-0.555849
^DJI	0.992026	1.000000	0.971867	0.972967	-0.596743
^RUT	0.961328	0.971867	1.000000	0.929796	-0.652811
^IXIC	0.985083	0.972967	0.929796	1.000000	-0.551473
DGS10	-0.555849	-0.596743	-0.652811	-0.551473	1.000000

Daily Returns

- Volatility of the market and 10 year maturity rate
- The 10 year rate semes to have had volitility between -10 and 10%
- where as the markets move around 5%+-

In [22]:
```
working_data.pct_change().hist(bins = 50, figsize = (16,10))
plt.tight_layout()
```

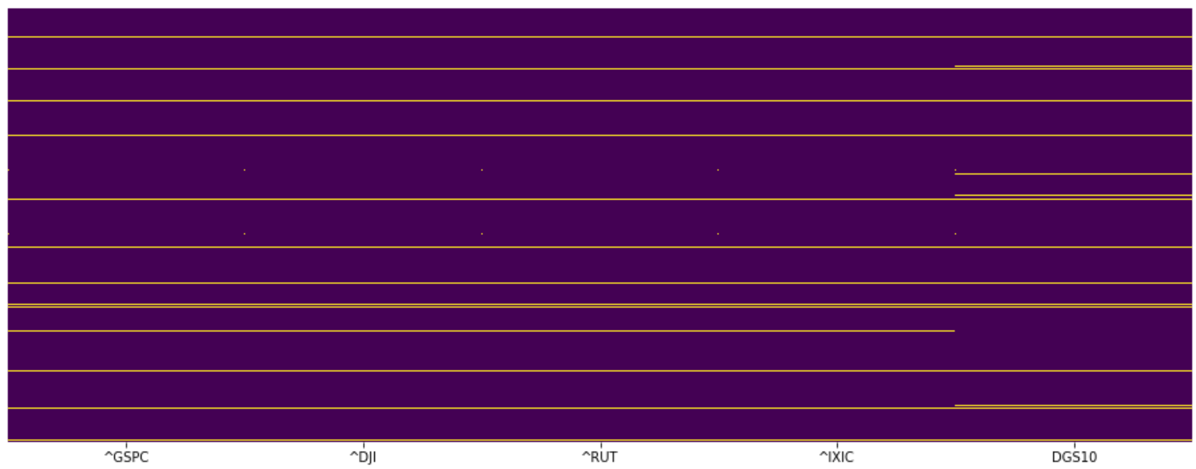


Model Generation

- Will create a Deep Neural Network that will aim to predict he 10 year maturity rate based on the movement of the markets
- We will need to check for missing data and drop the values for the model
- We can see that there is missing data colored in yellow

In [23]:
```
plt.figure(figsize=(16,6))
sns.heatmap(working_data.isnull(), yticklabels=False, cbar=False, cmap=
"viridis")
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1a26fa1b50>



In [24]:
```
working_data.dropna().isnull().sum()
```

Out[24]:
```
^GSPC     0
^DJI      0
^RUT      0
^IXIC     0
DGS10     0
dtype: int64
```

In [25]:
```
X = working_data.dropna().drop("DGS10", axis=1).values
y = working_data.dropna()['DGS10'].values
```

Training Testing and Splitting

In [26]:
```
from sklearn.model_selection import train_test_split
```

In [27]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30
, random_state=42)
```

Scaling

- Training the data for a normal distribution

In [28]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [29]:
```
scalar = MinMaxScaler()
```

In [30]:
```
X_train = scalar.fit_transform(X_train)
```

In [31]:
```
X_test = scalar.transform(X_test)
```

In [32]:
```
X_train.shape
```

Out[32]:
```
(3574, 4)
```

In [33]:
```
X_test.shape
```

Out[33]:
```
(1532, 4)
```

Model Prepration

- Will use a sequential model
- Will use a early stopping callback to prevent over fitting

In [34]:
```
from tensorflow.keras.callbacks import EarlyStopping
```

In [35]:
```
stop = EarlyStopping(monitor="val_loss", mode = "min", patience=30)
```

Model Creation

In [36]:
```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Dropout
```

Model Architectire

In [37]:
```
model = Sequential()
model.add(Dense(units = 30, activation = "relu"))
model.add(Dense(units = 30, activation = "relu"))
model.add(Dense(units = 30, activation = "relu"))
model.add(Dense(units = 20, activation = "relu"))
model.add(Dense(units = 20, activation = "relu"))
model.add(Dense(units = 20, activation = "relu"))
model.add(Dense(units = 10, activation = "relu"))
model.add(Dense(units = 10, activation = "relu"))
model.add(Dense(units = 10, activation = "relu"))
model.add(Dense(units = 1))
model.compile(loss = "mse", optimizer = "adam")
```

Fitting model

In [38]:
```
model.fit(X_train,y_train, validation_data=(X_test,y_test), epochs=450,
callbacks=[stop])
```

```
Train on 3574 samples, validate on 1532 samples
Epoch 1/450
3574/3574 [==============================] - 3s 798us/sample - loss: 1
2.0915 - val_loss: 11.1559
Epoch 2/450
3574/3574 [==============================] - 1s 162us/sample - loss: 3.
9628 - val_loss: 0.9686
Epoch 3/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
7704 - val_loss: 0.7546
Epoch 4/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
6886 - val_loss: 0.6741
Epoch 5/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
6309 - val_loss: 0.6624
Epoch 6/450
3574/3574 [==============================] - 1s 166us/sample - loss: 0.
5805 - val_loss: 0.5370
Epoch 7/450
3574/3574 [==============================] - 1s 175us/sample - loss: 0.
4912 - val_loss: 0.4461
Epoch 8/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
3885 - val_loss: 0.3403
Epoch 9/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
2519 - val_loss: 0.1873
Epoch 10/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
1799 - val_loss: 0.1946
Epoch 11/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
1607 - val_loss: 0.1775
Epoch 12/450
3574/3574 [==============================] - 1s 166us/sample - loss: 0.
1509 - val_loss: 0.1344
Epoch 13/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
1476 - val_loss: 0.1281
Epoch 14/450
3574/3574 [==============================] - 1s 171us/sample - loss: 0.
1405 - val_loss: 0.1347
Epoch 15/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
1339 - val_loss: 0.1190
Epoch 16/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
1314 - val_loss: 0.1166
Epoch 17/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
1319 - val_loss: 0.1117
Epoch 18/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1172 - val_loss: 0.1195
```

```
Epoch 19/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1185 - val_loss: 0.1109
Epoch 20/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
1292 - val_loss: 0.1268
Epoch 21/450
3574/3574 [==============================] - 1s 168us/sample - loss: 0.
1216 - val_loss: 0.1162
Epoch 22/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1202 - val_loss: 0.1288
Epoch 23/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1144 - val_loss: 0.1061
Epoch 24/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1153 - val_loss: 0.1001
Epoch 25/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1103 - val_loss: 0.1067
Epoch 26/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1107 - val_loss: 0.1006
Epoch 27/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1097 - val_loss: 0.1080
Epoch 28/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1089 - val_loss: 0.0989
Epoch 29/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1095 - val_loss: 0.1024
Epoch 30/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1098 - val_loss: 0.0973
Epoch 31/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
1039 - val_loss: 0.1156
Epoch 32/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1085 - val_loss: 0.1430
Epoch 33/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1084 - val_loss: 0.1035
Epoch 34/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
1066 - val_loss: 0.1060
Epoch 35/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
1061 - val_loss: 0.1022
Epoch 36/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1064 - val_loss: 0.0992
Epoch 37/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
```

```
1010 - val_loss: 0.0956
Epoch 38/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0996 - val_loss: 0.0945
Epoch 39/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0988 - val_loss: 0.1011
Epoch 40/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
1027 - val_loss: 0.1032
Epoch 41/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1039 - val_loss: 0.0987
Epoch 42/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0969 - val_loss: 0.0915
Epoch 43/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
1030 - val_loss: 0.0989
Epoch 44/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
1026 - val_loss: 0.0912
Epoch 45/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0960 - val_loss: 0.0949
Epoch 46/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0987 - val_loss: 0.0921
Epoch 47/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
1012 - val_loss: 0.1091
Epoch 48/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
1024 - val_loss: 0.1175
Epoch 49/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0938 - val_loss: 0.0948
Epoch 50/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0995 - val_loss: 0.0920
Epoch 51/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0990 - val_loss: 0.0883
Epoch 52/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0954 - val_loss: 0.0914
Epoch 53/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0952 - val_loss: 0.0917
Epoch 54/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0958 - val_loss: 0.0975
Epoch 55/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0963 - val_loss: 0.1002
Epoch 56/450
```

```
3574/3574 [==============================] - 1s 166us/sample - loss: 0.
1026 - val_loss: 0.1100
Epoch 57/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0950 - val_loss: 0.0987
Epoch 58/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0932 - val_loss: 0.0983
Epoch 59/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0933 - val_loss: 0.0896
Epoch 60/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0994 - val_loss: 0.0864
Epoch 61/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0971 - val_loss: 0.0885
Epoch 62/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0935 - val_loss: 0.0864
Epoch 63/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0912 - val_loss: 0.1019
Epoch 64/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0938 - val_loss: 0.0911
Epoch 65/450
3574/3574 [==============================] - 1s 168us/sample - loss: 0.
0930 - val_loss: 0.1021
Epoch 66/450
3574/3574 [==============================] - 1s 175us/sample - loss: 0.
0921 - val_loss: 0.0900
Epoch 67/450
3574/3574 [==============================] - 1s 184us/sample - loss: 0.
0938 - val_loss: 0.0897
Epoch 68/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0931 - val_loss: 0.0918
Epoch 69/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
1004 - val_loss: 0.1349
Epoch 70/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0965 - val_loss: 0.1039
Epoch 71/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0928 - val_loss: 0.0861
Epoch 72/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0890 - val_loss: 0.0858
Epoch 73/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0896 - val_loss: 0.0912
Epoch 74/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0909 - val_loss: 0.1090
```

```
Epoch 75/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0893 - val_loss: 0.0906
Epoch 76/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0905 - val_loss: 0.0897
Epoch 77/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0892 - val_loss: 0.0971
Epoch 78/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0908 - val_loss: 0.0911
Epoch 79/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0875 - val_loss: 0.0800
Epoch 80/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0869 - val_loss: 0.0916
Epoch 81/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0925 - val_loss: 0.0897
Epoch 82/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0891 - val_loss: 0.0838
Epoch 83/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0887 - val_loss: 0.0865
Epoch 84/450
3574/3574 [==============================] - 1s 158us/sample - loss: 0.
1003 - val_loss: 0.1112
Epoch 85/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0870 - val_loss: 0.1181
Epoch 86/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0844 - val_loss: 0.0863
Epoch 87/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0845 - val_loss: 0.0883
Epoch 88/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0805 - val_loss: 0.0834
Epoch 89/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0833 - val_loss: 0.0883
Epoch 90/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0841 - val_loss: 0.0816
Epoch 91/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0909 - val_loss: 0.0893
Epoch 92/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0835 - val_loss: 0.0889
Epoch 93/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
```

```
0920 - val_loss: 0.0833
Epoch 94/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0882 - val_loss: 0.0799
Epoch 95/450
3574/3574 [==============================] - 1s 158us/sample - loss: 0.
0844 - val_loss: 0.0862
Epoch 96/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0819 - val_loss: 0.0776
Epoch 97/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0876 - val_loss: 0.0838
Epoch 98/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0857 - val_loss: 0.0788
Epoch 99/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0811 - val_loss: 0.0905
Epoch 100/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0785 - val_loss: 0.0821
Epoch 101/450
3574/3574 [==============================] - 1s 158us/sample - loss: 0.
0813 - val_loss: 0.0809
Epoch 102/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0858 - val_loss: 0.0833
Epoch 103/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0817 - val_loss: 0.0889
Epoch 104/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0772 - val_loss: 0.0773
Epoch 105/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0878 - val_loss: 0.0841
Epoch 106/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0776 - val_loss: 0.0738
Epoch 107/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0827 - val_loss: 0.0919
Epoch 108/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0833 - val_loss: 0.0842
Epoch 109/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0805 - val_loss: 0.0901
Epoch 110/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0812 - val_loss: 0.0762
Epoch 111/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0932 - val_loss: 0.0853
Epoch 112/450
```

```
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0815 - val_loss: 0.0883
Epoch 113/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0845 - val_loss: 0.0860
Epoch 114/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0796 - val_loss: 0.0777
Epoch 115/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0785 - val_loss: 0.0784
Epoch 116/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0751 - val_loss: 0.0741
Epoch 117/450
3574/3574 [==============================] - 1s 167us/sample - loss: 0.
0766 - val_loss: 0.0789
Epoch 118/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0729 - val_loss: 0.1165
Epoch 119/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0781 - val_loss: 0.0778
Epoch 120/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0725 - val_loss: 0.0707
Epoch 121/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0722 - val_loss: 0.0848
Epoch 122/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0718 - val_loss: 0.0683
Epoch 123/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0782 - val_loss: 0.0739
Epoch 124/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0740 - val_loss: 0.0726
Epoch 125/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0768 - val_loss: 0.0742
Epoch 126/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0744 - val_loss: 0.0688
Epoch 127/450
3574/3574 [==============================] - 1s 175us/sample - loss: 0.
0710 - val_loss: 0.0716
Epoch 128/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0769 - val_loss: 0.0819
Epoch 129/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0750 - val_loss: 0.0992
Epoch 130/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0726 - val_loss: 0.0648
```

```
Epoch 131/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0688 - val_loss: 0.0670
Epoch 132/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0693 - val_loss: 0.0694
Epoch 133/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0669 - val_loss: 0.0658
Epoch 134/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0720 - val_loss: 0.0715
Epoch 135/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0718 - val_loss: 0.0806
Epoch 136/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0731 - val_loss: 0.0667
Epoch 137/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0696 - val_loss: 0.1060
Epoch 138/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0713 - val_loss: 0.0664
Epoch 139/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0717 - val_loss: 0.0636
Epoch 140/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0665 - val_loss: 0.0990
Epoch 141/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0719 - val_loss: 0.0674
Epoch 142/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0690 - val_loss: 0.0675
Epoch 143/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0669 - val_loss: 0.0646
Epoch 144/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0627 - val_loss: 0.0663
Epoch 145/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0660 - val_loss: 0.0693
Epoch 146/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0700 - val_loss: 0.0684
Epoch 147/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0734 - val_loss: 0.0700
Epoch 148/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0725 - val_loss: 0.0716
Epoch 149/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
```

```
0646 - val_loss: 0.0662
Epoch 150/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0687 - val_loss: 0.0677
Epoch 151/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0658 - val_loss: 0.0720
Epoch 152/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0729 - val_loss: 0.0662
Epoch 153/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0613 - val_loss: 0.0629
Epoch 154/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0669 - val_loss: 0.0664
Epoch 155/450
3574/3574 [==============================] - 1s 168us/sample - loss: 0.
0707 - val_loss: 0.0666
Epoch 156/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0593 - val_loss: 0.0711
Epoch 157/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0636 - val_loss: 0.0634
Epoch 158/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0673 - val_loss: 0.0630
Epoch 159/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0639 - val_loss: 0.0872
Epoch 160/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0637 - val_loss: 0.0692
Epoch 161/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0615 - val_loss: 0.0719
Epoch 162/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0645 - val_loss: 0.0731
Epoch 163/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0628 - val_loss: 0.0651
Epoch 164/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0688 - val_loss: 0.0603
Epoch 165/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0646 - val_loss: 0.0616
Epoch 166/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0589 - val_loss: 0.0666
Epoch 167/450
3574/3574 [==============================] - 1s 169us/sample - loss: 0.
0610 - val_loss: 0.0687
Epoch 168/450
```

```
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0634 - val_loss: 0.0630
Epoch 169/450
3574/3574 [==============================] - 1s 168us/sample - loss: 0.
0622 - val_loss: 0.0601
Epoch 170/450
3574/3574 [==============================] - 1s 173us/sample - loss: 0.
0699 - val_loss: 0.0876
Epoch 171/450
3574/3574 [==============================] - 1s 171us/sample - loss: 0.
0623 - val_loss: 0.0581
Epoch 172/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0579 - val_loss: 0.0806
Epoch 173/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0621 - val_loss: 0.0588
Epoch 174/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0593 - val_loss: 0.0710
Epoch 175/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0660 - val_loss: 0.0716
Epoch 176/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0609 - val_loss: 0.0597
Epoch 177/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0604 - val_loss: 0.0758
Epoch 178/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0585 - val_loss: 0.0547
Epoch 179/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0613 - val_loss: 0.0645
Epoch 180/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0635 - val_loss: 0.0603
Epoch 181/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0596 - val_loss: 0.0542
Epoch 182/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0577 - val_loss: 0.0621
Epoch 183/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0671 - val_loss: 0.0551
Epoch 184/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0558 - val_loss: 0.0548
Epoch 185/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0611 - val_loss: 0.0527
Epoch 186/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0556 - val_loss: 0.0579
```

```
Epoch 187/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0534 - val_loss: 0.0593
Epoch 188/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0578 - val_loss: 0.0643
Epoch 189/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0630 - val_loss: 0.0687
Epoch 190/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0555 - val_loss: 0.0557
Epoch 191/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0651 - val_loss: 0.0775
Epoch 192/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0574 - val_loss: 0.0533
Epoch 193/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0527 - val_loss: 0.0547
Epoch 194/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0584 - val_loss: 0.0758
Epoch 195/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0624 - val_loss: 0.0646
Epoch 196/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0578 - val_loss: 0.0550
Epoch 197/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0576 - val_loss: 0.0595
Epoch 198/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0657 - val_loss: 0.0567
Epoch 199/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0603 - val_loss: 0.0705
Epoch 200/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0626 - val_loss: 0.0564
Epoch 201/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0550 - val_loss: 0.0578
Epoch 202/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0527 - val_loss: 0.0549
Epoch 203/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0524 - val_loss: 0.0524
Epoch 204/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0550 - val_loss: 0.0577
Epoch 205/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
```

```
0511 - val_loss: 0.0555
Epoch 206/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0658 - val_loss: 0.0511
Epoch 207/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0509 - val_loss: 0.0573
Epoch 208/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0514 - val_loss: 0.0587
Epoch 209/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0511 - val_loss: 0.0590
Epoch 210/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0504 - val_loss: 0.0584
Epoch 211/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0572 - val_loss: 0.0571
Epoch 212/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0522 - val_loss: 0.0625
Epoch 213/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0511 - val_loss: 0.0544
Epoch 214/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0526 - val_loss: 0.0485
Epoch 215/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0547 - val_loss: 0.0611
Epoch 216/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0515 - val_loss: 0.0538
Epoch 217/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0533 - val_loss: 0.0713
Epoch 218/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0529 - val_loss: 0.0482
Epoch 219/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0526 - val_loss: 0.0544
Epoch 220/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0530 - val_loss: 0.0513
Epoch 221/450
3574/3574 [==============================] - 1s 167us/sample - loss: 0.
0495 - val_loss: 0.0606
Epoch 222/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0477 - val_loss: 0.0484
Epoch 223/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0521 - val_loss: 0.0557
Epoch 224/450
```

```
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0517 - val_loss: 0.0663
Epoch 225/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0539 - val_loss: 0.0494
Epoch 226/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0506 - val_loss: 0.0726
Epoch 227/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0488 - val_loss: 0.0510
Epoch 228/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0496 - val_loss: 0.0588
Epoch 229/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0505 - val_loss: 0.0459
Epoch 230/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0522 - val_loss: 0.0488
Epoch 231/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0525 - val_loss: 0.0498
Epoch 232/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0499 - val_loss: 0.0496
Epoch 233/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0496 - val_loss: 0.0594
Epoch 234/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0456 - val_loss: 0.0481
Epoch 235/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0518 - val_loss: 0.0530
Epoch 236/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0473 - val_loss: 0.0450
Epoch 237/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0488 - val_loss: 0.0576
Epoch 238/450
3574/3574 [==============================] - 1s 166us/sample - loss: 0.
0468 - val_loss: 0.0450
Epoch 239/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0498 - val_loss: 0.0505
Epoch 240/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0512 - val_loss: 0.0450
Epoch 241/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0477 - val_loss: 0.0519
Epoch 242/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0479 - val_loss: 0.0443
```

```
Epoch 243/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0483 - val_loss: 0.0483
Epoch 244/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0485 - val_loss: 0.0608
Epoch 245/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0460 - val_loss: 0.0440
Epoch 246/450
3574/3574 [==============================] - 1s 166us/sample - loss: 0.
0489 - val_loss: 0.0500
Epoch 247/450
3574/3574 [==============================] - 1s 165us/sample - loss: 0.
0487 - val_loss: 0.0496
Epoch 248/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0480 - val_loss: 0.0557
Epoch 249/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0483 - val_loss: 0.0442
Epoch 250/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0465 - val_loss: 0.0468
Epoch 251/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0493 - val_loss: 0.0518
Epoch 252/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0471 - val_loss: 0.0551
Epoch 253/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0452 - val_loss: 0.0711
Epoch 254/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0525 - val_loss: 0.0459
Epoch 255/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0465 - val_loss: 0.0474
Epoch 256/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0462 - val_loss: 0.0445
Epoch 257/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0482 - val_loss: 0.0764
Epoch 258/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0551 - val_loss: 0.0454
Epoch 259/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0469 - val_loss: 0.0455
Epoch 260/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0503 - val_loss: 0.0583
Epoch 261/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
```

```
0459 - val_loss: 0.0442
Epoch 262/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0464 - val_loss: 0.0498
Epoch 263/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0458 - val_loss: 0.0461
Epoch 264/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0478 - val_loss: 0.0498
Epoch 265/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0507 - val_loss: 0.0526
Epoch 266/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0497 - val_loss: 0.0463
Epoch 267/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0465 - val_loss: 0.0544
Epoch 268/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0464 - val_loss: 0.0456
Epoch 269/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0443 - val_loss: 0.0460
Epoch 270/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0454 - val_loss: 0.0478
Epoch 271/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0490 - val_loss: 0.0522
Epoch 272/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0487 - val_loss: 0.0484
Epoch 273/450
3574/3574 [==============================] - 1s 166us/sample - loss: 0.
0485 - val_loss: 0.0610
Epoch 274/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0434 - val_loss: 0.0417
Epoch 275/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0515 - val_loss: 0.0444
Epoch 276/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0439 - val_loss: 0.0434
Epoch 277/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0431 - val_loss: 0.0431
Epoch 278/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0432 - val_loss: 0.0421
Epoch 279/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0452 - val_loss: 0.0443
Epoch 280/450
```

```
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0531 - val_loss: 0.0489
Epoch 281/450
3574/3574 [==============================] - 1s 171us/sample - loss: 0.
0545 - val_loss: 0.0448
Epoch 282/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0469 - val_loss: 0.0425
Epoch 283/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0451 - val_loss: 0.0435
Epoch 284/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0477 - val_loss: 0.0442
Epoch 285/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0441 - val_loss: 0.0442
Epoch 286/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0487 - val_loss: 0.0434
Epoch 287/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0463 - val_loss: 0.0474
Epoch 288/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0460 - val_loss: 0.0465
Epoch 289/450
3574/3574 [==============================] - 1s 164us/sample - loss: 0.
0433 - val_loss: 0.0463
Epoch 290/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0427 - val_loss: 0.0491
Epoch 291/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0431 - val_loss: 0.0444
Epoch 292/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0476 - val_loss: 0.0446
Epoch 293/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0441 - val_loss: 0.0497
Epoch 294/450
3574/3574 [==============================] - 1s 163us/sample - loss: 0.
0449 - val_loss: 0.0716
Epoch 295/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0538 - val_loss: 0.0452
Epoch 296/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0455 - val_loss: 0.0449
Epoch 297/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0419 - val_loss: 0.0428
Epoch 298/450
3574/3574 [==============================] - 1s 162us/sample - loss: 0.
0474 - val_loss: 0.0477
```

```
Epoch 299/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0419 - val_loss: 0.0476
Epoch 300/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0440 - val_loss: 0.0464
Epoch 301/450
3574/3574 [==============================] - 1s 161us/sample - loss: 0.
0428 - val_loss: 0.0418
Epoch 302/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0463 - val_loss: 0.0431
Epoch 303/450
3574/3574 [==============================] - 1s 159us/sample - loss: 0.
0428 - val_loss: 0.0457
Epoch 304/450
3574/3574 [==============================] - 1s 160us/sample - loss: 0.
0454 - val_loss: 0.0505
```

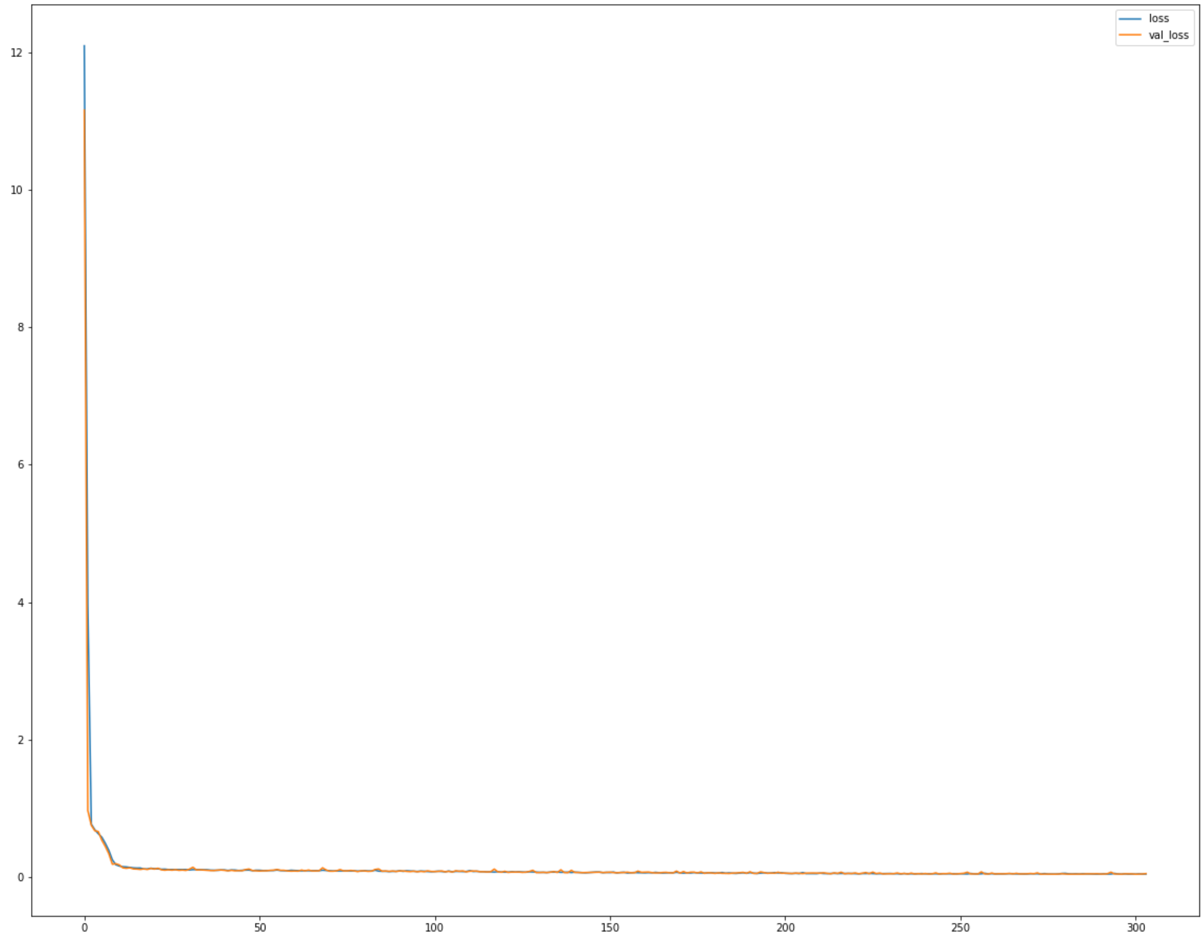
Out[38]:
```
<tensorflow.python.keras.callbacks.History at 0x1a4da266d0>
```

In [ ]:

Model Performnce

- Will take a look at the trainign performance of the model
- Model trained well and did not stop with our early stopping callback. There is still room for training here bur we will see how them model predicts

In [41]:
```
pd.DataFrame(model.history.history).plot(figsize = (20,16))
plt.savefig("resources/model_training_plot.jpg")
```



In [39]:
```
model.save("indexto_10year_yield_Deployed.h5")
```

In [40]:
```
pd.DataFrame(model.history.history).to_csv("resources/model_deployed_his
tory.csv")
```

Model Predictions

In [42]:
```
predictions = model.predict(X_test)
```

Will take a look at the metrics

In [43]:
```
from sklearn.metrics import mean_squared_error, mean_absolute_error, exp
lained_variance_score
```

In [44]:
```
mean_absolute_error(y_test, predictions)
```
Out[44]:
```
0.1722995728892384
```

In [45]:
```
mean_squared_error(y_test,predictions)
```
Out[45]:
```
0.05054592102327381
```

In [46]:
```
np.sqrt(mean_squared_error(y_test, predictions))
```
Out[46]:
```
0.22482420026161287
```

In [47]:
```
explained_variance_score(y_test,predictions)
```
Out[47]:
```
0.970760409723183
```

Predictions on Random Data

In [89]:
```
from random import randint
random_index = randint(1, len(working_data.dropna()))
random_day = working_data.dropna().drop(["DGS10"], axis = 1).iloc[random
_index]
```

Random Data Prepration

- will need to scale and get the values for the data
- Random day will also need to reshape the new data

In [90]:
```
X_train.shape
```
Out[90]:
```
(3574, 4)
```

In [91]:
```
new_day = scalar.transform(random_day.values.reshape(1,4))
```

In [92]:
```
new_day
```
Out[92]:
```
array([[0.2755664 , 0.25611734, 0.3319722 , 0.15378328]])
```

Prediction on Random Data

- The 10 year rate is predicted below

In [93]:
```
model.predict(new_day)
```

Out[93]:
```
array([[4.7466507]], dtype=float32)
```

Actual Value

In [94]:
```
working_data.iloc[random_index]
```

Out[94]:
```
^GSPC      1304.250000
^DJI      11383.469727
^RUT        720.570007
^IXIC      2189.169922
DGS10         4.740000
Name: 2006-08-31 00:00:00, dtype: float64
```

In [95]:
```
working_data.iloc[random_index]["DGS10"]
```

Out[95]:
```
4.74
```

In [ ]:
```

```

Lets check todays Prediction got the 10 year constant Rate

- We will will drop the target which is the 10 Year constant Rate

In [55]:
```
week_data = pd.DataFrame()
```

In [56]:
```
for t in tickers:
    week_data[t] = web.DataReader(t, data_source= "yahoo", start = "2020
-6-12")["Adj Close"]
```

In [57]:
```
week_data
```

Out[57]:

Date	^GSPC	^DJI	^RUT	^IXIC
2020-06-12	3041.310059	25605.539062	1387.680054	9588.809570
2020-06-15	3066.590088	25763.160156	1419.609985	9726.019531
2020-06-16	3124.739990	26289.980469	1452.260010	9895.870117
2020-06-17	3113.489990	26119.609375	1426.530029	9910.530273
2020-06-18	3115.340088	26080.099609	1427.329956	9943.049805
2020-06-19	3097.739990	25871.460938	1418.630005	9946.120117
2020-06-22	3117.860107	26024.960938	1433.530029	10056.480469
2020-06-23	3131.290039	26156.099609	1439.339966	10131.370117
2020-06-24	3050.330078	25445.939453	1389.739990	9909.169922
2020-06-25	3083.760010	25745.599609	1413.310059	10017.000000
2020-06-26	3009.050049	25015.550781	1378.780029	9757.219727
2020-06-29	3053.239990	25595.800781	1421.209961	9874.150391
2020-06-30	3100.290039	25812.880859	1441.369995	10058.769531
2020-07-01	3115.860107	25734.970703	1427.310059	10154.629883
2020-07-02	3130.010010	25827.359375	1431.859985	10207.629883
2020-07-06	3179.719971	26287.029297	1442.880005	10433.650391
2020-07-07	3145.320068	25890.179688	1416.000000	10343.889648
2020-07-08	3169.939941	26067.279297	1427.400024	10492.500000
2020-07-09	3152.050049	25706.089844	1398.920044	10547.750000
2020-07-10	3185.040039	26075.300781	1422.680054	10617.440430
2020-07-13	3155.219971	26085.800781	1403.569946	10390.839844
2020-07-14	3197.520020	26642.589844	1428.260010	10488.580078
2020-07-15	3226.560059	26870.099609	1478.270020	10550.490234
2020-07-16	3215.570068	26734.710938	1467.560059	10473.830078
2020-07-17	3224.729980	26671.949219	1473.319946	10503.190430
2020-07-20	3251.840088	26680.869141	1467.949951	10767.089844
2020-07-21	3257.300049	26840.400391	1487.510010	10680.360352
2020-07-22	3276.020020	27005.839844	1490.140015	10706.129883
2020-07-23	3235.659912	26652.330078	1490.199951	10461.419922
2020-07-24	3215.629883	26469.890625	1467.550049	10363.179688
2020-07-27	3239.409912	26584.769531	1484.650024	10536.269531
2020-07-28	3218.439941	26379.279297	1469.760010	10402.089844

Date	^GSPC	^DJI	^RUT	^IXIC
2020-07-29	3258.439941	26539.570312	1500.630005	10542.940430
2020-07-30	3246.219971	26313.650391	1495.099976	10587.809570
2020-07-31	3271.120117	26428.320312	1480.430054	10745.269531
2020-08-03	3294.610107	26664.400391	1506.800049	10902.799805
2020-08-04	3306.510010	26828.470703	1517.209961	10941.169922
2020-08-05	3327.770020	27201.519531	1546.239990	10998.400391
2020-08-06	3349.159912	27386.980469	1544.619995	11108.070312

In [58]:
```
X_train.shape
```
Out[58]:
```
(3574, 4)
```

In [59]:
```
todays_data = scalar.transform(week_data[-1:].values.reshape(1,4))
```

In [60]:
```
model.predict(todays_data)
```
Out[60]:
```
array([[0.68280876]], dtype=float32)
```

In [ ]:

In [ ]: