

Problem Statement

- Create model Cconvert Celsuis to Fahernheit
- The equation follows $T(F) = T(c) * 9/5 + 32$

▼ Import library

```
!pip install tensorflow
```

```
import tensorflow as tf
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
tf.__version__
```

[+ Code](#)[+ Text](#)

Mount Drive

▼ Import DataSets

```
data = pd.read_csv("/content/drive/My Drive/Tensorflow_2.0/Data/Celsius+to+Fahrenh...")
data
```

▼ Visualize Data

```
data.describe()
```

```
data.info()
```

```
sns.scatterplot(data["Celsius"], data["Fahrenheit"], color="green")
```

▼ Training and Test Data

```
x_train = data["Celsius"]  
y_train = data["Fahrenheit"]
```

```
x_train.shape
```

```
y_train.shape
```

```
model = tf.keras.Sequential()  
model.add(tf.keras.layers.Dense(units=1, input_shape=[1]))
```

```
model.summary()
```

```
model.compile(optimizer=tf.keras.optimizers.Adam(0.5), loss="mean_squared_error")
```

```
epochs_history=model.fit(x_train, y_train, epochs=5)
```

▼ Model_2

```
model_2 = tf.keras.Sequential()  
model_2.add(tf.keras.layers.Dense(units=1, input_shape=[1]))
```

```
model_2.summary()
```

```
model_2.compile(optimizer=tf.keras.optimizers.Adam(0.5), loss="mean_squared_error")
```

```
epochs_history_2 = model_2.fit(x_train, y_train, epochs=500)
```

▼ Evaluate Model

```
model_1
```

```
epochs_history.history.keys()
```

```
plt.figure(figsize=(10, 10))  
plt.plot(epochs_history.history['loss'])  
plt.title("Model Loss progress During training Epochs is 100 and Learning Rate 0  
plt.xlabel("Epochs")  
plt.ylabel("Traning Loss")  
plt.legend(["Traning loss"])
```

model_2

```
plt.figure(figsize=(10, 10))
plt.plot(epochs_history_2.history['loss'])
plt.title("Model Loss progress During training Epochs is 100 and Learning Rate 0
plt.xlabel("Epochs")
plt.ylabel("Traning Loss")
plt.legend(["Traning loss"])
```

```
model.get_weights() #  $9/5 = 1.8$ ,
```

```
model_2.get_weights()
```

▼ Using Model

```
temp_c = 0
temp_f = model.predict([temp_c])
print('temperature in degF using trained ANN', temp_f)
```

```
temp_f = 9/5*temp_c +32
print('temperature in degF using trained ANN', temp_f)
```

```
temp_c = -32
temp_f = 9/5*temp_c +32
print('temperature in degF using trained ANN', temp_f)
```

```
temp_c = -32
temp_f = model.predict([temp_c])
print('temperature in degF using trained ANN', temp_f)
```

