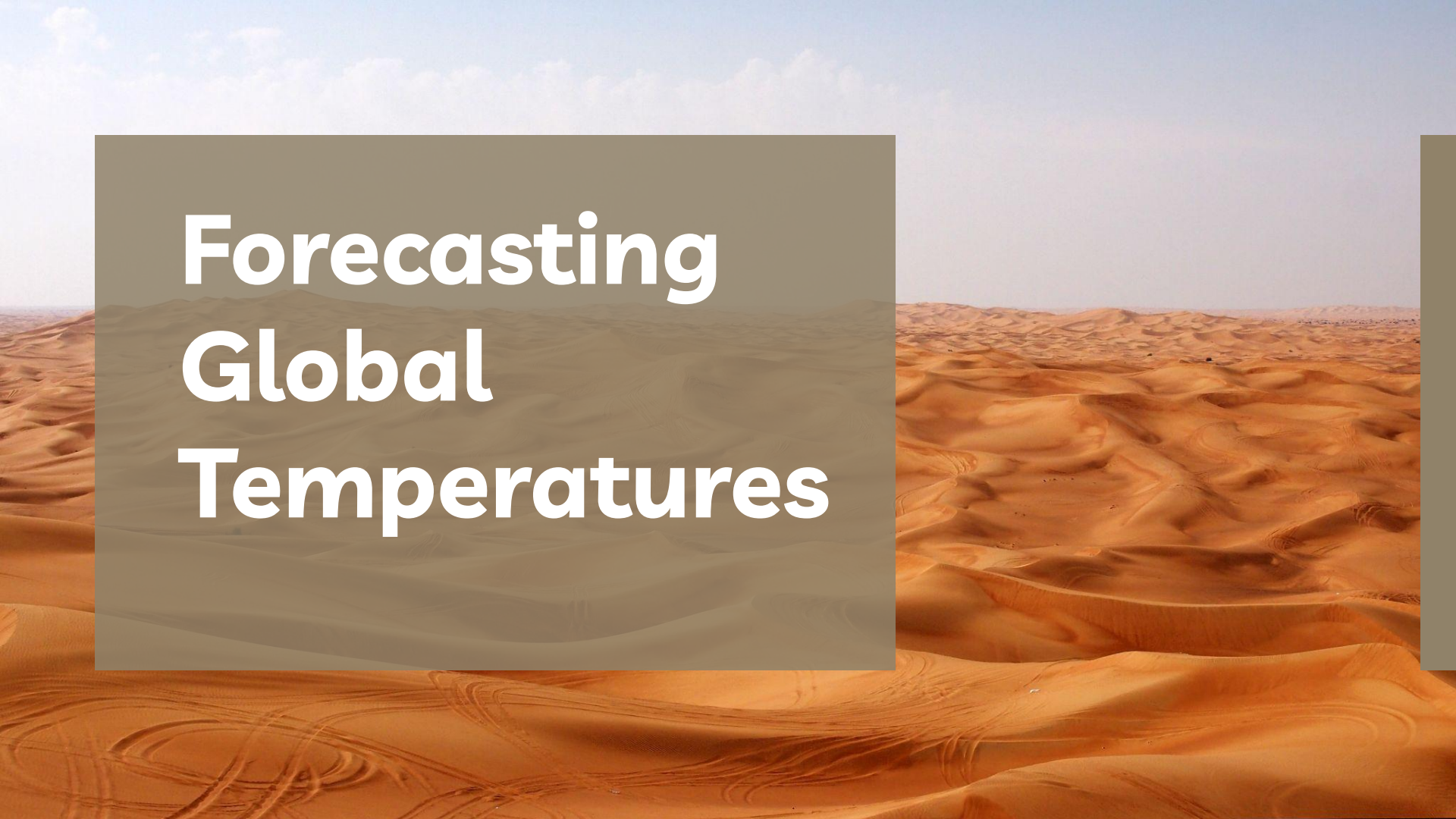

Forecasting
Global
Temperatures

TABLE OF CONTENTS

01 **ABOUT THE PROJECT**
The reason for predicting future global temperatures

02 **Planning**
How we will create our model and where we will get the data

03 **Data Sorting**
How I had to sort the data

04 **Model Creation**
Creation the model, finding best hyperparameters, etc

05 **Conclusions**
What we can learn from this project, what can we improve in the future

What is global warming?

Global warming is the rapid rise in temperature of our planet due to human impact. Global warming has several adverse effects on the living creatures on our planet like droughts, severe weather patterns and extinction of some species that cannot adapt to the higher temperatures.

Why should we forecast global temperatures

Prepare for extreme weather - Predicts heatwaves and droughts to keep people safe.

Plan economically - Guides farmers and energy companies to make better decisions.

Protect health and environment - Warns about health risks and helps conserve nature.

Combat climate change - Informs policies to reduce emissions and adapt to changes.

Data Preparation

In this section we will prepare the data for the model

Model Building

Here we will find hyperparameters and build the model.

Model Evaluation

We will look at our model and how well it did

ABOUT THE PROJECT

This is a time series problem where I will have to predict future values based on the values I am given

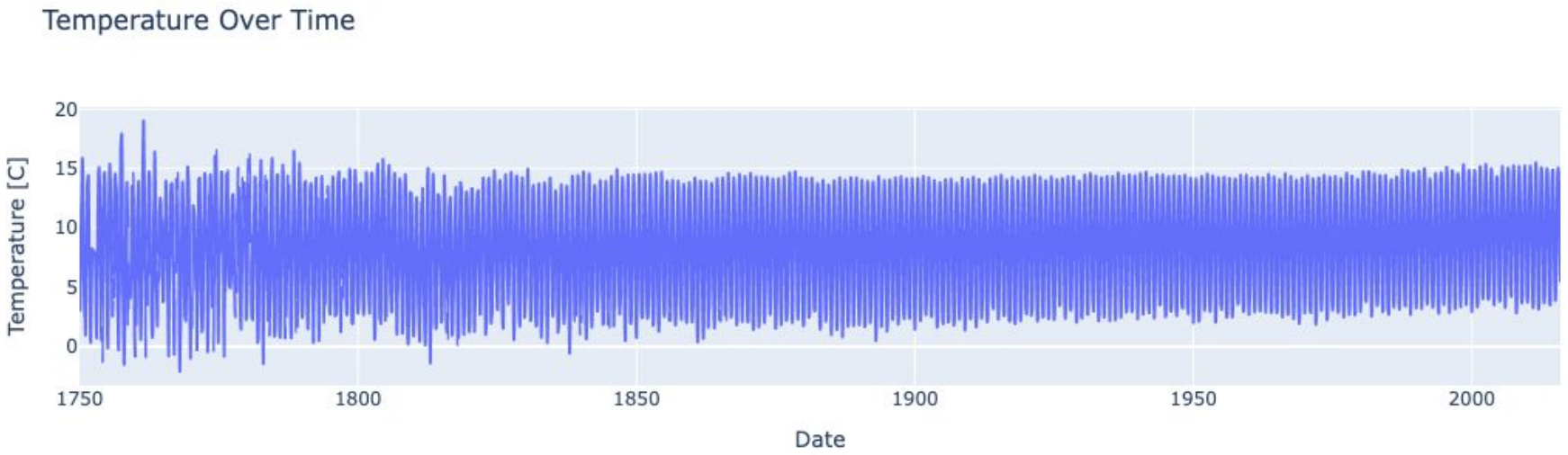
Data Processing

The Dataset I am using: Kaggle, Berkeley Earth, "Climate Change: Earth Surface Temperature Data,"

Handling Missing Values: First we dropped all the rows with missing temperatures using “data = data['LandAverageTemperature'].dropna()”

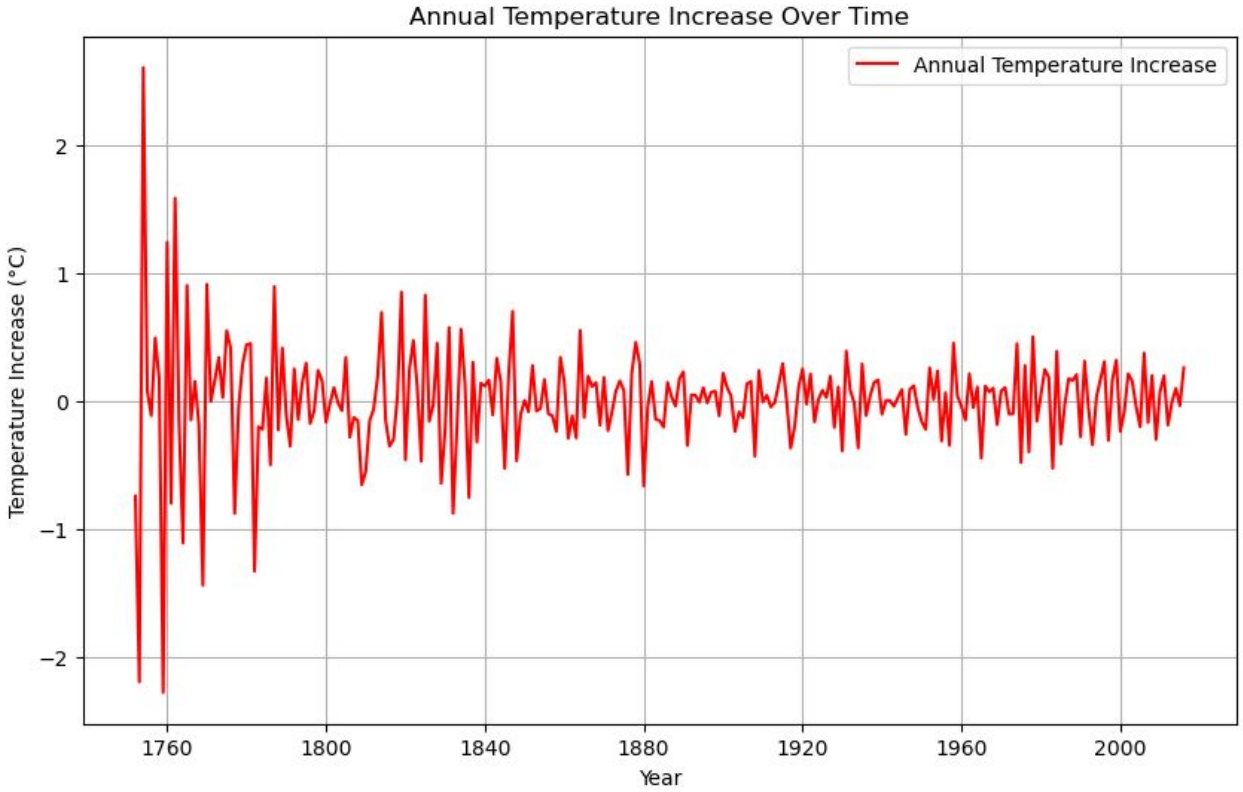
Normalization:
The dataset was normalized using the MinMaxScaler from the sklearn library. Normalization scales the data to a range of 0 to 1, making it suitable for training neural networks.

The Data


Temperature Over Time
Temperature [C]
20
15
10
5
0
1750
1800
1850
1900
1950
2000
Date

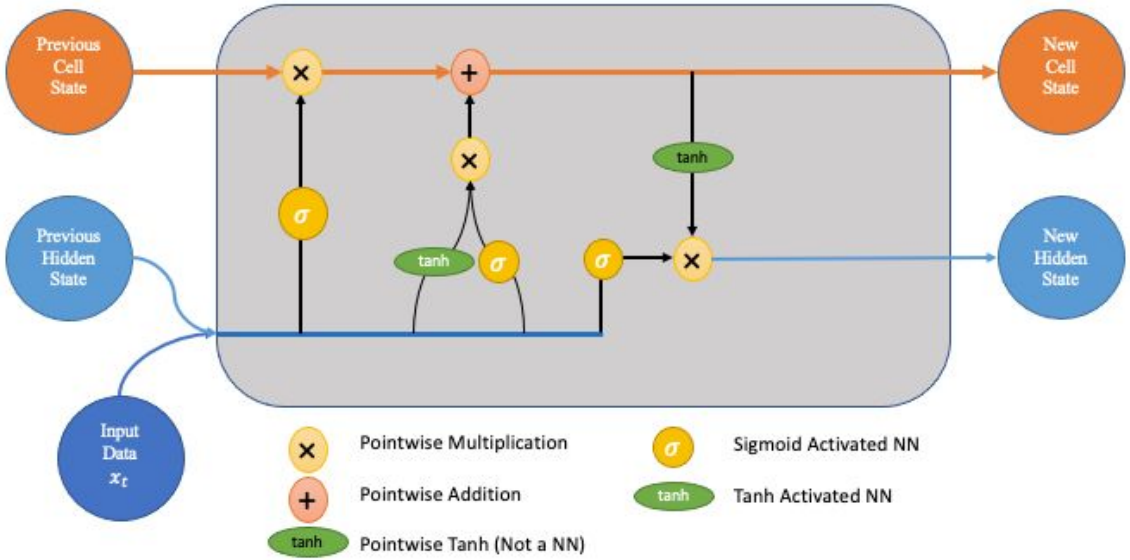
The Annual Temperature Over Time

The Annual Temperature increase Over Time



What are LSTMs?

LSTMs (Long Short-Term Memory) are a type of recurrent neural network (RNN) capable of learning long-term dependencies. Unlike standard feedforward neural networks, LSTMs have feedback connections, making them suitable for time series prediction tasks.



Hyperparameter tuning (Bayes Search)

To make our model the best it can be we usually paly around with the hyperparameters of it. However playing with the hyperparameters manually usually takes a lot of time and doesn't give any benefits. For that reason we often use things like GridSearch. GridSearch takes a really long time since it brute forces every single hyperparameter. A more effective solution is the Bayes Search which instead of brute forcing through every hyperparameter it makes evaluations and only tries the hyperparametes most probable to help our model.

```
keras_regressor = KerasRegressor(model=create_model, verbose=0)


param_space = {
    'model__learning_rate': Real(1e-4, 1e-2, prior='log-uniform'),
    'model__units': Integer(1, 50),
    'fit__epochs': Integer(50, 200),
    'fit__batch_size': Integer(1, 32)
}

# Setup BayesSearchCV
opt = BayesSearchCV(
    estimator=keras_regressor,
    search_spaces=param_space,
    n_iter=32,
    cv=KFold(n_splits=3),
    scoring="neg_mean_squared_error",
    verbose=2,
    return_train_score=True,
    n_jobs=-1
)


opt.fit(trainX, trainY)


best_params = opt.best_params_
print("Best parameters found: ", best_params)
```

R2 Score for Train Set: 0.7126
R2 Score for Test Set: 0.7439

My Model

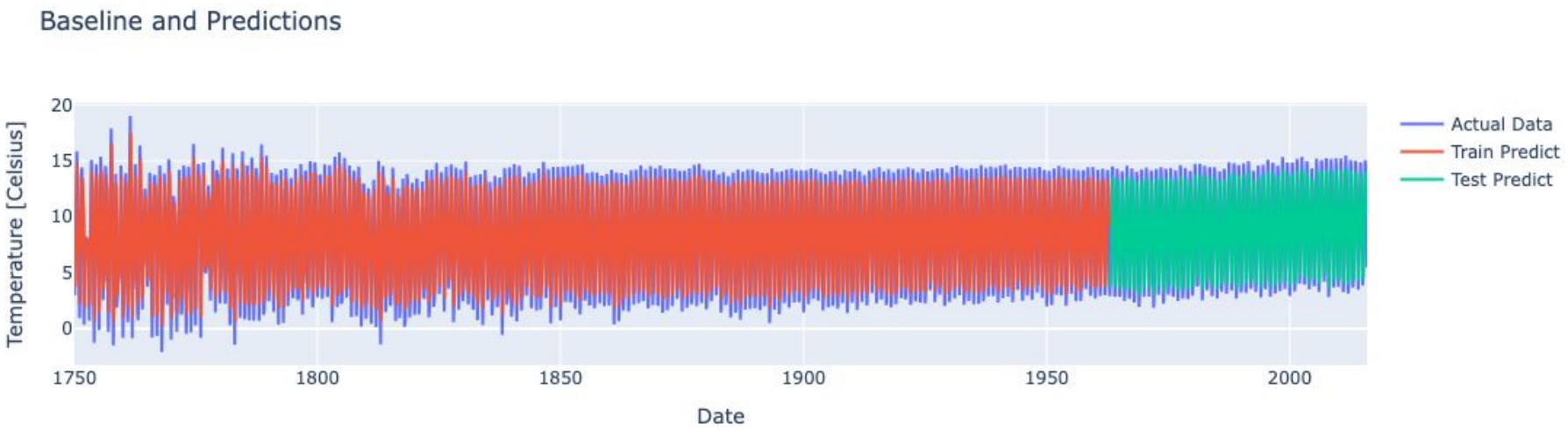
```
80/80 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 311us/step
Train Score: 0.11 RMSE
Test Score: 0.10 RMSE
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 1, 9)	396
dropout_1 (Dropout)	(None, 1, 9)	0
lstm_3 (LSTM)	(None, 9)	684
dense_1 (Dense)	(None, 1)	10

Total params: 3,272 (12.79 KB)
Trainable params: 1,090 (4.26 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 2,182 (8.53 KB)

My Model


Baseline and Predictions
Temperature [Celsius]
20
15
10
5
0
1750
1800
1850
1900
1950
2000
Date
Actual Data
Train Predict
Test Predict

Conclusion

Evaluation: The results of the model were satisfactory and we had a very small RMSE(Root Mean Squared Error). However we can always improve this accuracy with newer models and more data.

What I could have done better: Maybe I could have predicted future temperatures as well. I tried to do this but ran into many issues.

For the Future: Use this model to predict temperatures in specific countries our regions since our database has all the values for that. We can even measure the temperature of the water and much more.

Bibliography

1. ChatGPT. (2024). OpenAI. Retrieved from https://www.openai.com/chatgpt

2. Brownlee, J. (2018, August 14). Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

3. Berkeley Earth. (2020). Climate Change: Earth Surface Temperature Data. Kaggle. Retrieved from https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data

THANKS

Does anyone have any questions?