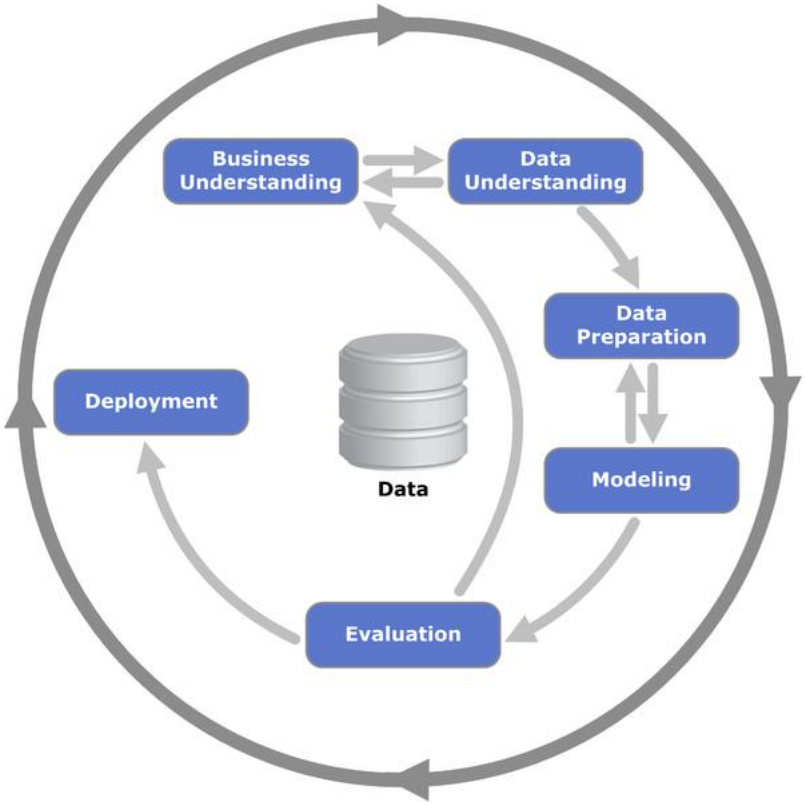
The Lightweight IBM Cloud Garage Method for Data Science

Global Temperature Anomalies - Monthly Time Series

Architectural Decisions Document

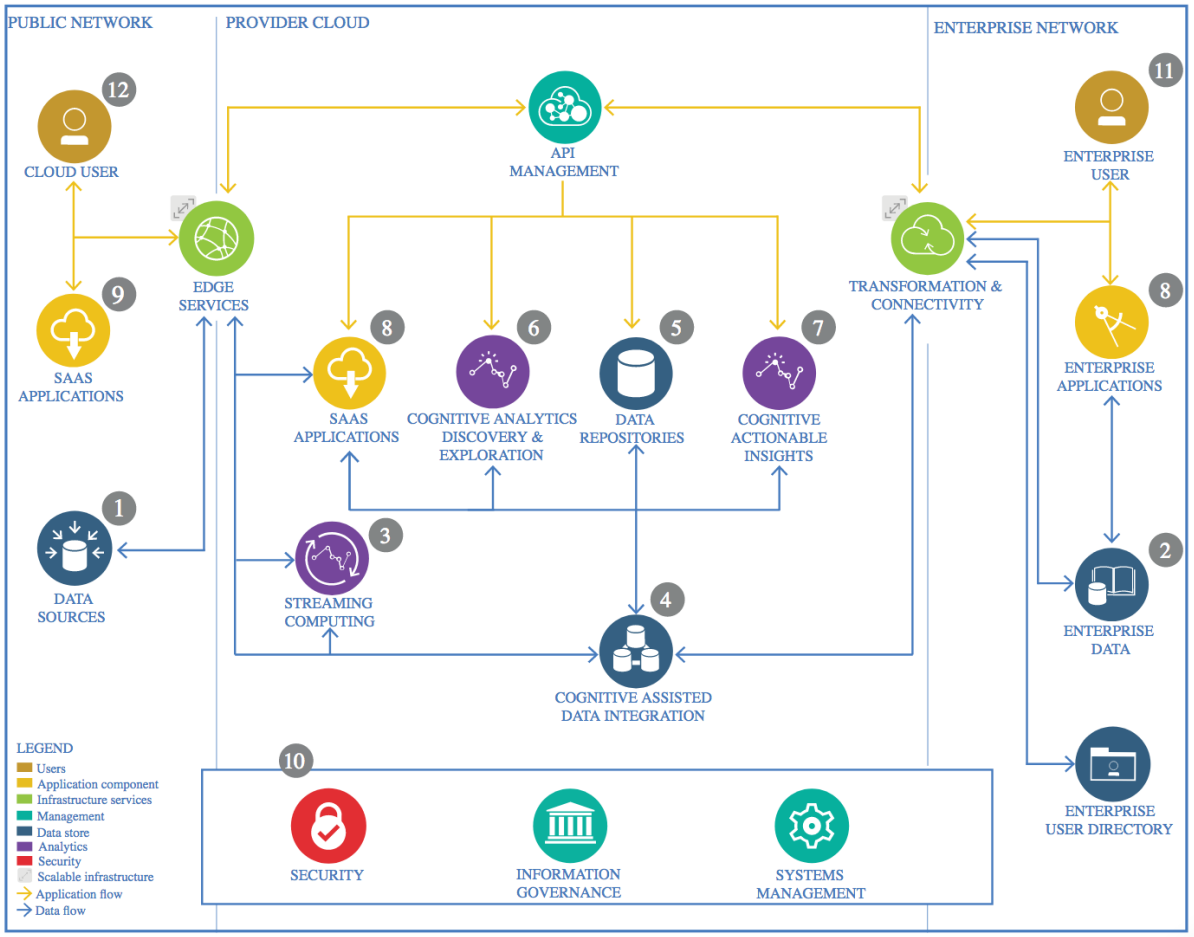
1 An overview on existing process models for Data Science CRISP-DM

CRISP-DM, which stands for Cross-industry Standard Process for Data Mining, is the most widely used open standard process model.
CRISP-DM defines a set of phases which make up a data science project. Most importantly, transitions between those phases are bi-directional and the whole process is iterative, this means once you've reached the final stage you just start over the whole process and refine your work. The illustration below emphasizes on that nature:



The CRISP-DM process model. By Kenneth Jensen, based on: ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/en/ModelerCRISPDM.pdf - Creative Commons Attribution-Share Alike 3.0 Unported license

2 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

2.1 Data Source

2.1.1 Technology Choice

The data source is a text file from http://berkeleyearth.org/data/ converted to a local .CSV file and read as a pandas data frame.

2.1.2 Justification

Since the plan is to use pandas data frame, it would then be more convenient to read the original file as a .CSV. The decision to use pandas in the first place over parquet for example or any other data frame is related to the size of the original data which is not that massive, in addition that pandas is more user friendly when it comes to format.

2.2 Enterprise Data

2.2.1 Technology Choice

Enterprise Data will not be used in this project.

2.2.2 Justification

As a consumer of the data, the decision to transfer the dataset to the cloud and perform continuous update to the dataset with additional monthly anomaly recordings should come from the data source provider.

2.3 Streaming analytics

2.3.1 Technology Choice

Batch processing approach will be adopted in this project.
Batch size = 32
Epochs = 130

2.3.2 Justification

Reason being that the anomalies are recorded monthly, so we would need to at least wait for a month to get new data to be processed, so real time is not fit for purpose.

Component	Justification
Batch_size: 32	Using a batch size of 32 allows to predict on the test set (X_test_reshaped) without facing a dimensionality error as we are validating on the test set in batches. In addition, as per the market research and analysis, choosing a batch size of 32 for deep learning proved to be performing well and this is because the learning rate and batch size are closely linked
Epochs: 130	The training experiments with different epoch numbers proved that 130 epochs is suitable to have the right normalization of accuracy and loss

2.4 Data Integration

2.4.1 Technology Choice

- The original data contains few data fields that will not be used for training, so these fields will be removed.
- In addition, the target data points contain ‘NaN’ values which will be replaced by 0’s.

- The monthly land average temperature anomalies data have been converted to 'float' which will be more convenient later on during the machine learning process.

2.4.2 Justification

- The fields that will be removed have no correlation with the targeted output that we are trying to predict.
- The 'NaN' values will impact the fitting process of the model, so a reasonable approach is to replace them with 0's where we consider that there were no anomalies recorded in a certain month.

2.5 Data Repository

2.5.1 Technology Choice

The dataset will be stored on GitHub and read from there in pandas data frame. The link to the dataset is: https://github.com/cesarhanna/Data-Science-Projects/blob/master/Global_temperature_anomalies.csv?raw=true

2.5.2 Justification

A public GitHub repo will be easily accessible by public users and will provide a fast and controllable way to update the dataset whenever new records are captured.

2.6 Discovery and Exploration

2.6.1 Technology Choice

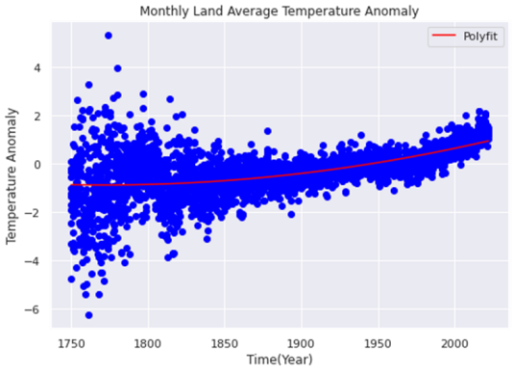
The dataset decided upon for the time series prediction model entails the time per year per month and the monthly land average anomaly which is our target.
The training range of data will be from year 1850 until present day.
A quick information overview about the data:

	Size	Features
Old dataset	3272 rows × 12 columns	• Year • Month • Monthly Land Average Temperature Anomaly • Monthly Land Average Temperature Anomaly Uncertainty • Annual Land Average Temperature Anomaly • Annual Land Average Temperature Anomaly Uncertainty • Five-year Land Average Temperature Anomaly

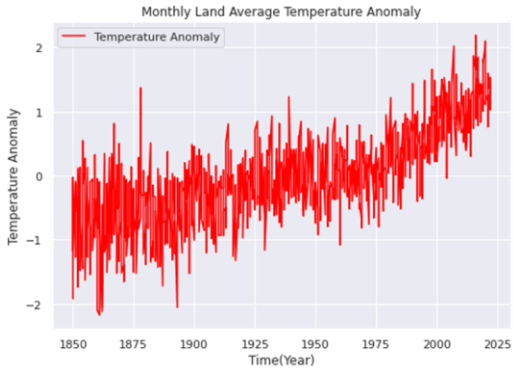
		• Five-year Land Average Temperature Anomaly Uncertainty • Ten-year Land Average Temperature Anomaly • Ten-year Land Average Temperature Anomaly Uncertainty • Twenty-year Land Average Temperature Anomaly • Twenty-year Land Average Temperature Anomaly Uncertainty
New/Final dataset	2072 rows × 3 columns	• Year • Month • Monthly Land Average Temperature Anomaly

2.6.2 Justification

For this particular business case, we only need to have one field that will be used for training and prediction using the sliding window approach.
The reason data from 1750 until 1849 was discarded is since this data shows inconsistency in frequency and a lot of outliers which was eventually normalizing as time went by.
The assumption here could be that during those 100 years from 1750 until 1850 they didn't have the right technology to accurately record the anomalies.



Old dataset



New dataset

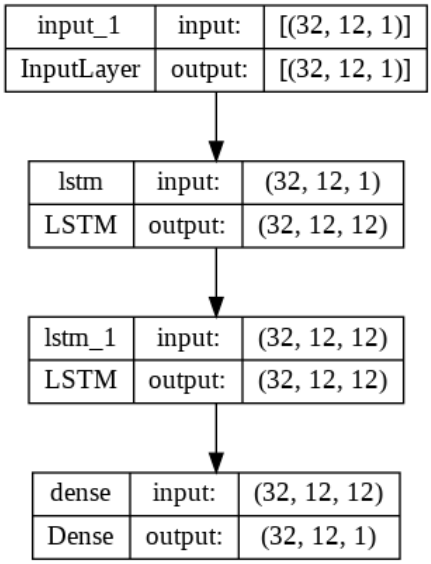
The regression line shown in the old dataset is created using **Linear Regression** with **Polynomial Features** which fits the cluster of data points better. In addition, the trajectory of this line indicates that the temperature anomalies from 1850 until present day are going up.

2.7 Actionable Insights

2.7.1 Technology Choice

The complete approach to create the model and train it will be illustrated as follows:

- Analysis of the Dataset
- Decision on the Final Dataset
- Creating the Final Dataset
- Feature Engineering – constructing the input and output using a sliding window
- Feature Scaling using 'MinMaxScaler', which is scaling the data to be in a range from 0 to 1
- Designing an LSTM model in Keras
    - Functional API will be used instead of Sequential LSTM
    - Optimizer = 'adam'
    - The model will have the following design:



- Training the LSTM model and evaluating
- Evaluation of the non-deep learning algorithms using multi-output regression will be done using:
    - R2 Score
    - Mean Squared Error (MSE)
- Evaluation of the deep learning algorithm, LSTM, will using the following:
    - Loss = 'mae' (Mean Absolute Error)
    - Metrics = 'accuracy'
    - Val_loss
    - Val_accuracy

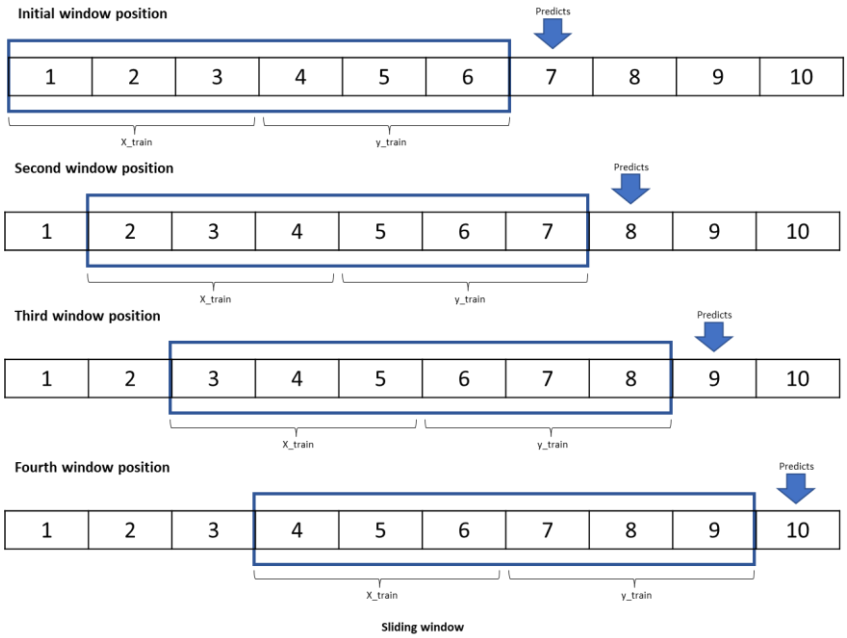
2.7.2 Justification

- The following table shows the justification why each of the parameters and components were used in this project:

Component - Parameter	Justification
Feature engineering: Sliding window	Sliding window is the way to restructure a univariate time series dataset as a supervised learning problem
Feature scaling: MinMaxScaler	MinMaxScaler preserves the shape of the original distribution. It doesn't meaningfully change the information embedded in the original data. MinMaxScaler doesn't reduce the importance of outliers as well
Evaluation of the non-deep learning algorithms: R2 Score and Mean Squared Error	• R2: R-squared evaluates the scatter of the data points around the fitted regression line. It is also called the coefficient of determination, or the coefficient of multiple determination for multiple regression. For the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values. • MSE: The two biggest advantages of MSE are that they provide a quadratic loss function and that they are also measures of the uncertainty in forecasting; it fits very in our use case for time series prediction.
Evaluation for deep learning algorithm: • Loss = 'mae' (Mean Absolute Error) • Metrics = 'accuracy' • Val_loss • Val_accuracy	• MAE: findings indicate that MAE is a more natural measure of average error, and unlike RMSE, is unambiguous. Dimensioned evaluations and inter-comparisons of average model-performance error, therefore, should be based on MAE • Accuracy: this metric is most useful when we are working with balanced datasets, which is our case in this project • Val_loss: validation loss is a metric used to assess the performance of a deep learning model on the validation set; in our case the model validates using the features created with the sliding window. It validates the loss between training data (X_train) and the data used for

	prediction acting as the test data (y_train) • Val_accuracy: validating the model outputs is important to ensure its accuracy. When a machine learning model is trained, a huge amount of training data is used and the main aim of checking the model validation provides an opportunity to improve the data quality and quantity
Functional API instead of Sequential	Functional API provides more flexibility in designing the input, hidden and output layers
Optimizer: adam	There are different reasons of using adam as optimizer, to name few: • Easy to implement • Quite computationally efficient • Requires little memory space • Good for non-stationary objectives (time series is a good example of that)

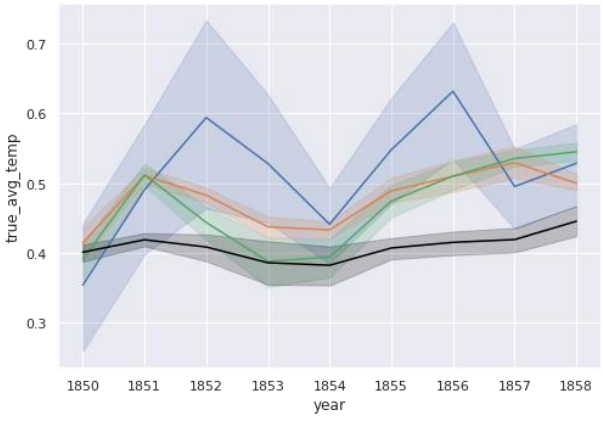
- In today's environment, demand forecasting is complex, and the data needed for accurately forecasting at scale isn't always straightforward. Using LSTM, time series forecasting models can predict future values based on previous, sequential data. This provides greater accuracy for demand forecasters which results in better decision making for the business. The following are some LSTM learnings:
    - Demand data sets are featured with different seasonality. The LSTM is capable of capturing the patterns of both long-term seasonality's such as a yearly pattern and short-term seasonality's such as weekly or monthly patterns.
    - The LSTM could take inputs with different lengths. This feature is especially useful when LSTM is used to build general forecasting models for specific customers or industries.
    - The different gates inside LSTM boost its capability for capturing non-linear relationships for forecasting. Causal factors generally have non-linear impact on demand. When these factors are used as part of the input variable, the LSTM could learn the nonlinear relationship for forecasting.
- Sliding window is the way to restructure a time series dataset as a supervised learning problem.
- Univariate forecasting time series in our case here will be framed as supervised learning using the sliding window method as shown in the following figure:



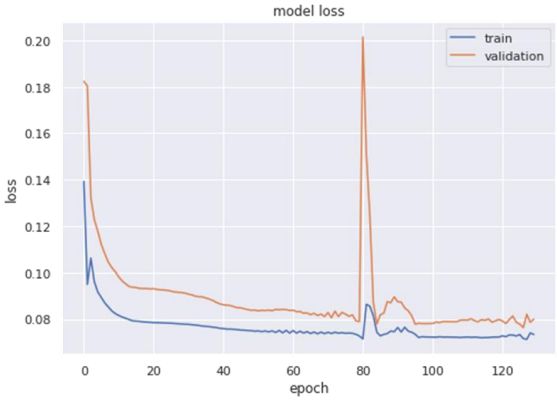
This is how the sliding window in this project works:

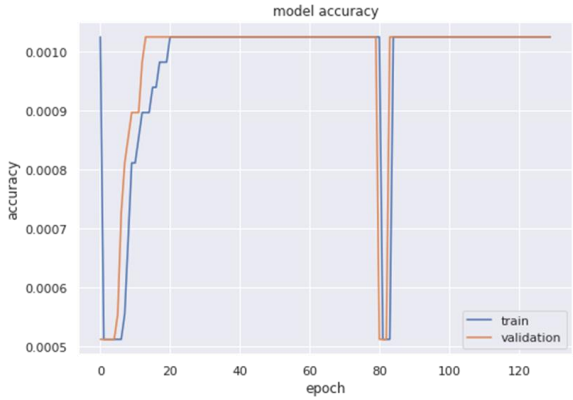
X_train	y_train	yhat (target)
Left side of the window with 12 records – Training input	Right side of the window with 12 records – Used as input to predict the target after the model has been trained using the X_train	Predicted target – The next monthly anomaly

- Another fundamental reason why LSTM is the algorithm of choice is that different non-deep learning regression models – **Ridge Regression**, **Random Forest Regression** and **Support Vector Machine** all encapsulated in a **Multi-output Regressor**, have been tested in this project and none of them performed well against the test set; here's a graph summarizing those performance compared to the true values represented by the blue line:



- Finally, the LSTM model proves to be performing very good with the hyperparameters used; hers are the loss and accuracy graphs using this algorithm:





2.8 Applications / Data Products

2.8.1 Technology Choice

The final product of this project will be a notebook which could be used by both the business users and developers to create, for example, a REST API that could be integrated with a web application. Access to the notebook is via GitHub following this link: Global Temperature Anomalies - Monthly Time Series Prediction Model

2.8.2 Justification

- The business users: to have visibility on the data used and most importantly the prediction accuracy of the model.
- Developers: to understand the approach and the code base used to create the prediction model. In addition, the code base could be used to create a REST API to integrate the machine learning model with the business application(s).

2.9 Security, Information Governance and Systems Management

2.9.1 Technology Choice

The access to resources of this project will be role based. This can be illustrated as follows:

- Business Users
- Developers
- Data Engineers
- Data Scientists

2.9.2 Justification

- Business users: they have read access to the following:
    - The dataset; this could be retrieved as mentioned before from GitHub - https://github.com/cesarhanna/Data-Science-Projects/blob/master/Global_temperature_anomalies.csv?raw=true
    - The notebook in PDF format

- Developers: they have read access to the dataset and the notebook in PDF format
- Data Engineers: they have read-write access to the notebook for further ETL activities.
- Data Scientists: they have read-write access to the notebook in .ipynb format in order to adjust and maintain the model through its lifecycle.