

## Praktikum 4: Recurrent Neural Networks

### Datenanalyse

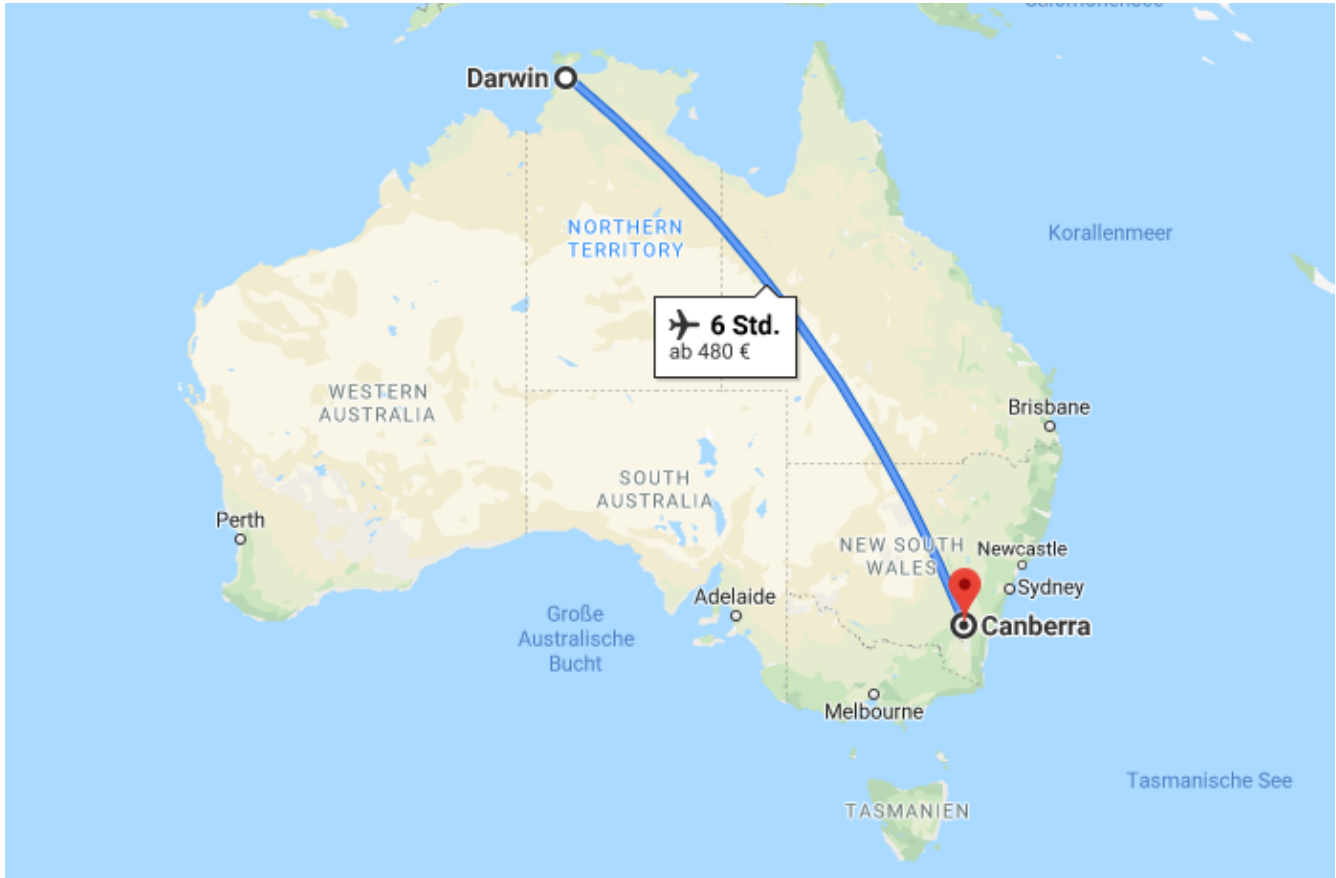


Abbildung 1: Google Maps: Geographische Position von Canberra und Darwin

Wir beginnen zunächst damit uns einen Überblick über die Daten zu machen, um weiteres Vorgehen zu evaluieren. In Abbildung 1 werden beide Städte geographisch dargestellt. Beide Datensätze beinhalten die selben Merkmale, weshalb wir diese nun zunächst allgemein erläutern.

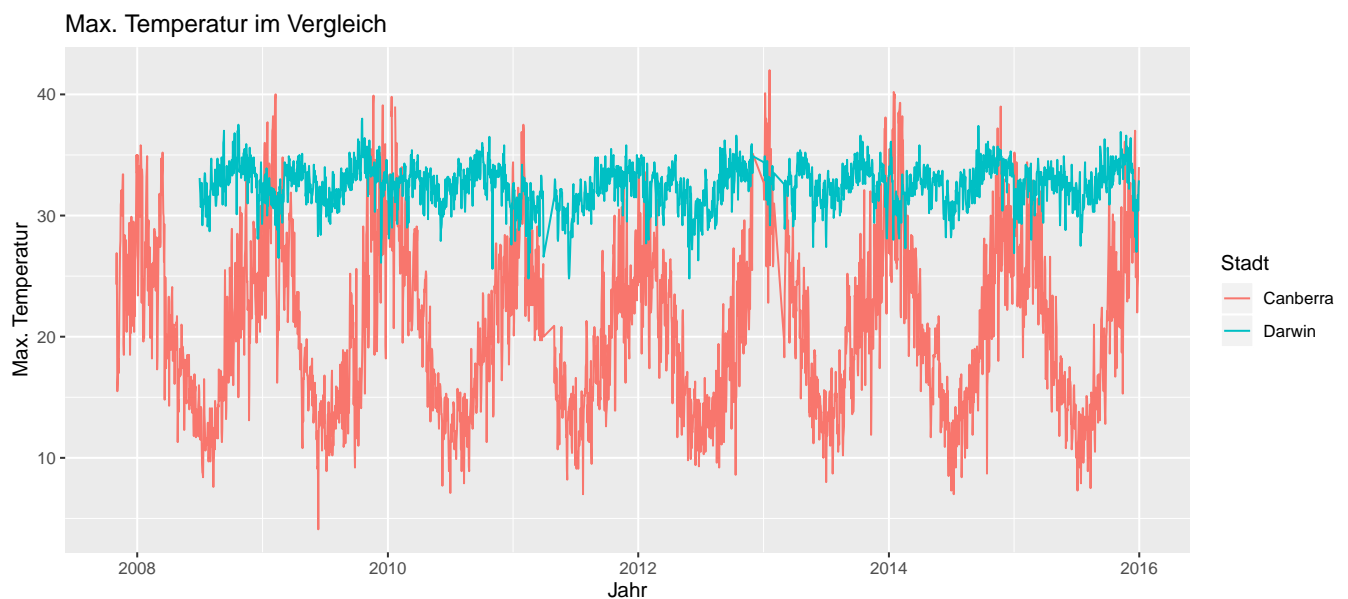
In Tabelle 1 sehen wir welche Merkmale zu welchen Skalenniveau vorliegen. In gelb markiert sind all diejenigen Merkmale, welche wir noch kodieren müssen, da das rnn nur numerische Werte akzeptiert. Deshalb verwenden wir One-Hot-Encoding und nehmen die verschiedenen Ausprägungen als neue Features.

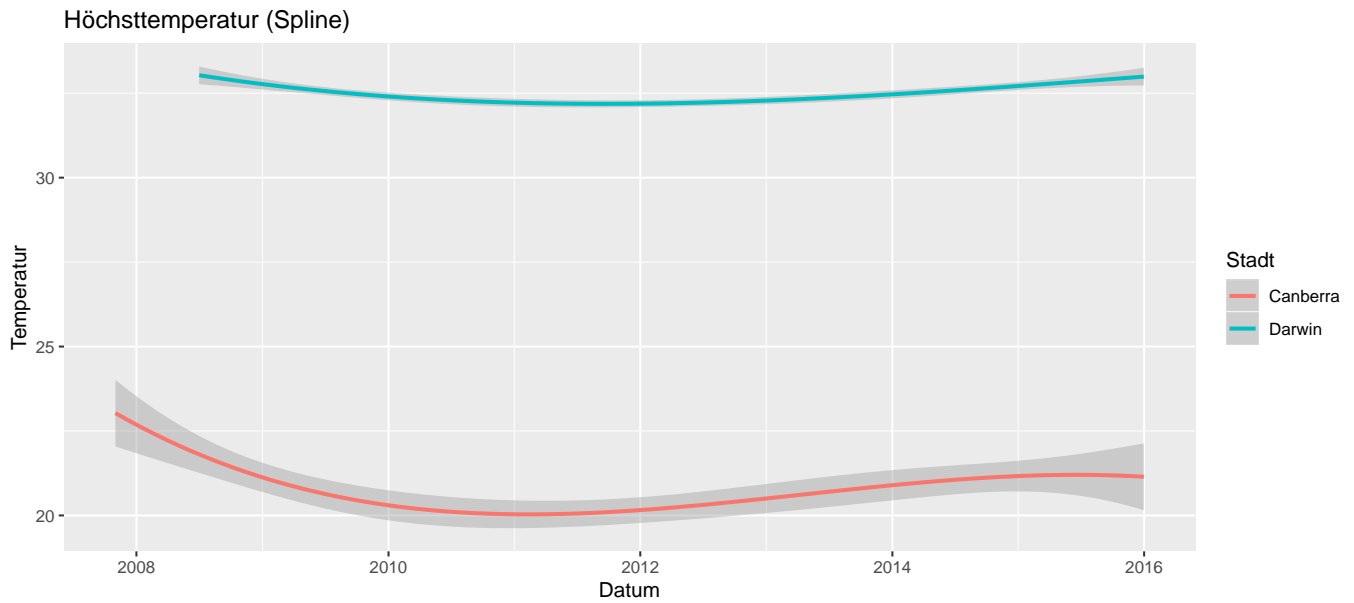
### Überblick Temperatur

Wir sehen, dass es über das Jahr deutlich stärkere Temperaturschwankungen in Canberra als in Darwin gibt und, dass die durchschnittliche Temperatur in Darwin höher ist.

Merkmal	Beschreibung	Skalenniveau	Beispiel
<b>X</b>	<b>ID</b>	<b>nominal</b>	<b>49232</b>
<b>Date</b>	Datum	ordinal	2007-11-01
<b>MinTemp</b>	Minimum Temperatur des Tages	metrisch	8.0
<b>MaxTemp</b>	Maximum Temperatur des Tages	metrisch	24.3
<b>Evaporation</b>	Verdunstung	metrisch	3.4
<b>Sunshine</b>	Sonnenschein ? STUNDEN ?	metrisch	6.3
<b>WindGustDir</b>	Wind Richtung ? VOM TAG ?	nominal	NW
<b>WindGustSpeed</b>	Wind Geschwindigkeit	metrisch	30
<b>WindGustDir9am</b>	Wind Richtung morgens um 9	nominal	SW
<b>WindDir3pm</b>	Wind Richtung mittags um 15	nominal	NW
<b>WindSpeed9am</b>	Wind Geschwindigkeit morgens um 9	metrisch	6
<b>WindSpeed3pm</b>	Wind Geschwindigkeit mittags um 15	metrisch	20
<b>Humidity9am</b>	Luftfeuchtigkeit morgens um 9	metrisch	68
<b>Humidity3pm</b>	Luftfeuchtigkeit mittags um 15	metrisch	29
<b>Pressure9am</b>	Luftdruck morgens um 9	metrisch	1019.7
<b>Pressure3pm</b>	Luftdruck mittags um 15	metrisch	1015.0
<b>Cloud9am</b>	Bewölkung morgens um 9	ordinal	7
<b>Cloud3pm</b>	Bewölkung mittags um 15	ordinal	7
<b>Temp9am</b>	Temperatur morgens um 9	metrisch	14.4
<b>Temp3pm</b>	Temperatur mittags um 15	metrisch	23.6
<b>RainToday</b>	Regentag ?	nominal	No
<b>RainTomorrow</b>	War der nächste Tag ein Regentag?	nominal	Yes

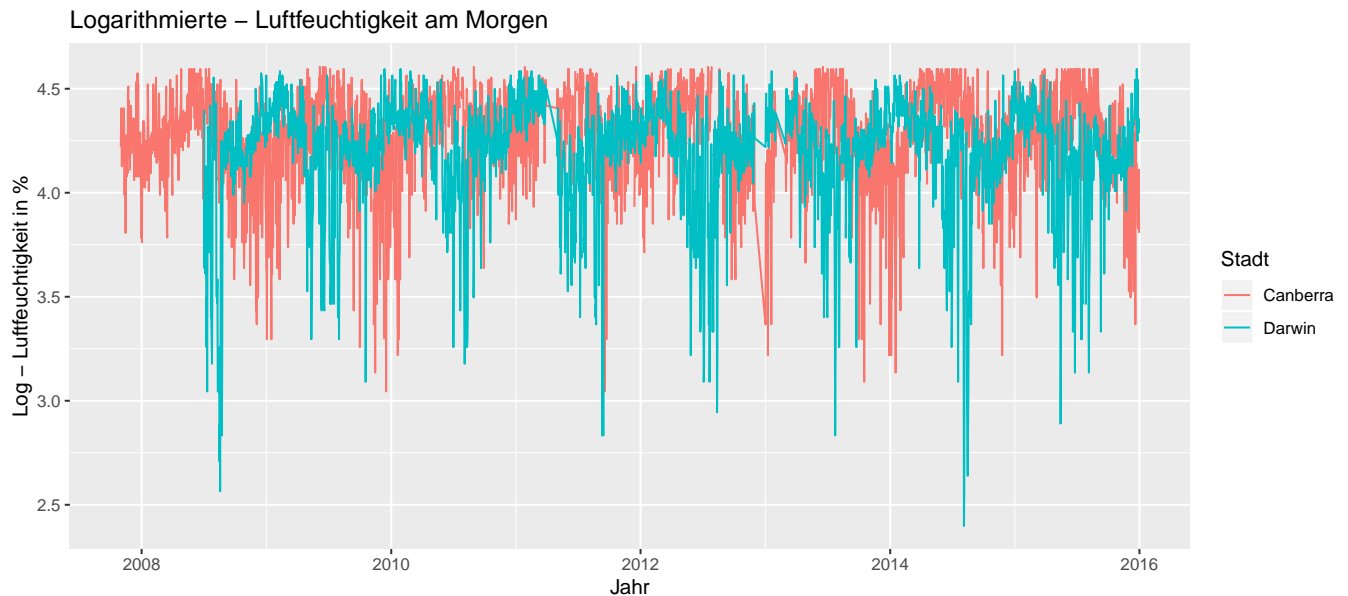
Tabelle 1: Überblick der vorhanden Metrik

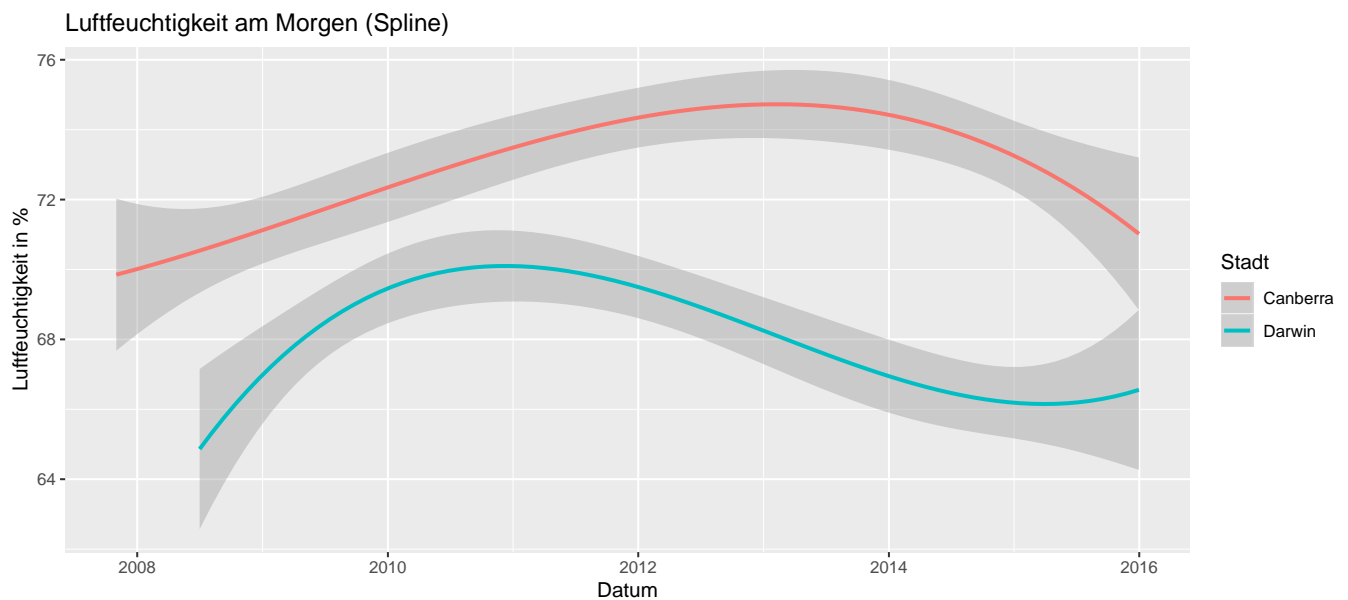
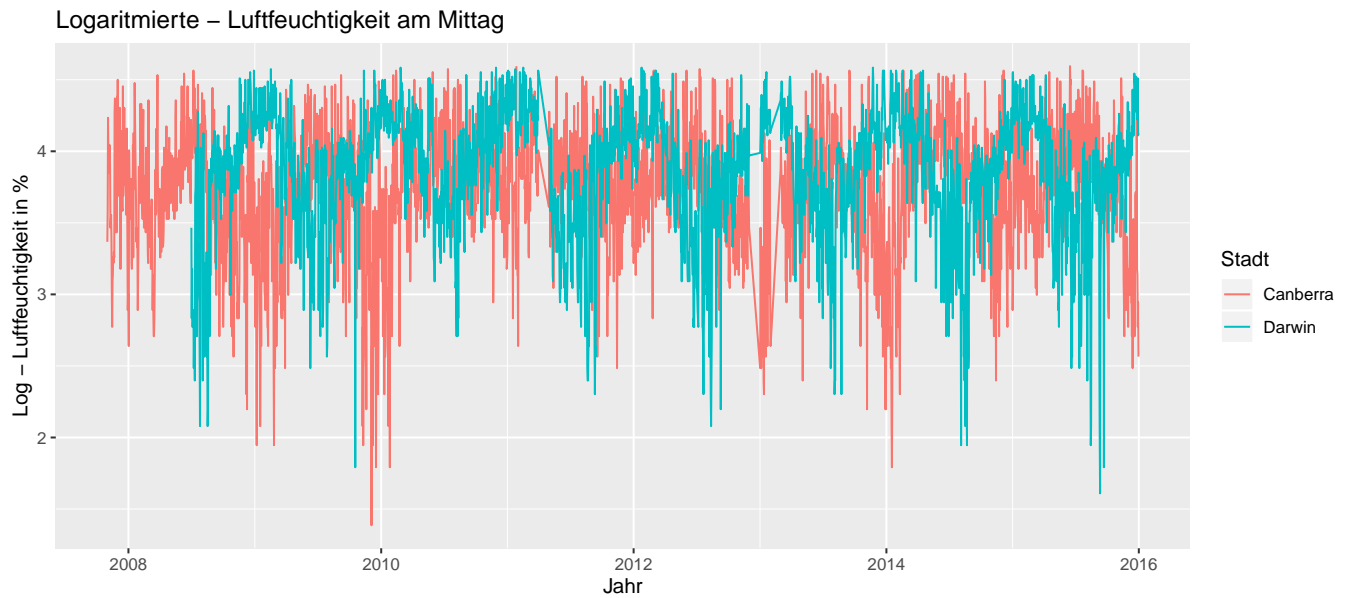




## Überblick Luftfeuchtigkeit

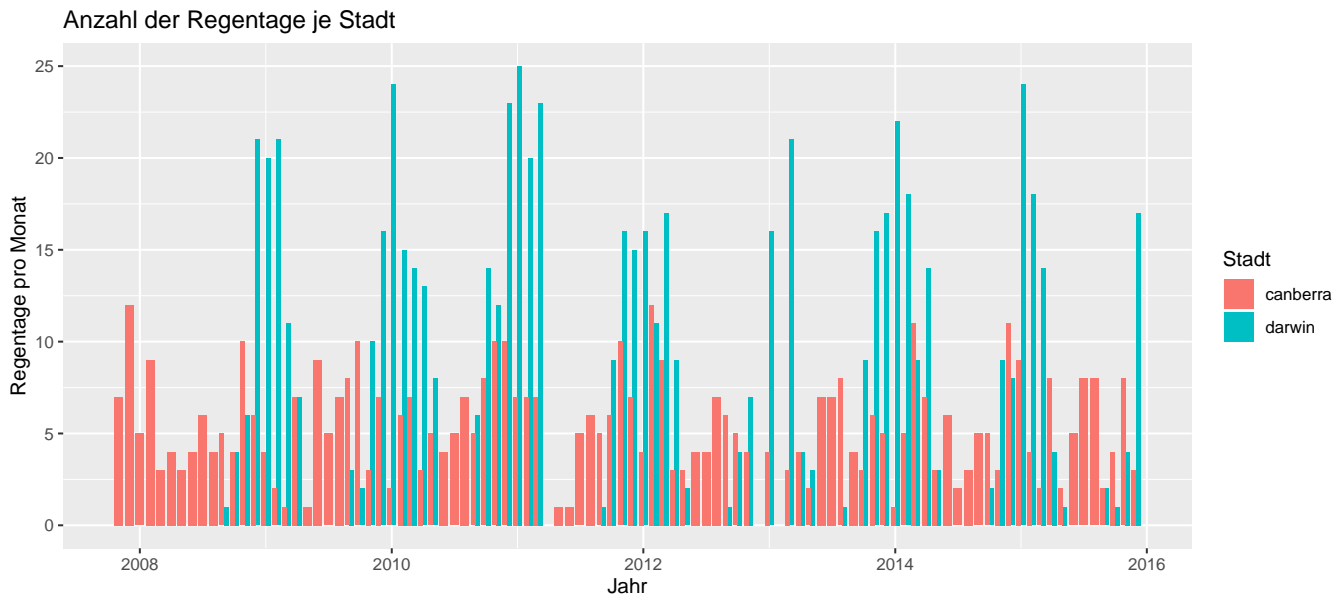
Verschaffen wir uns nun einen Überblick über die Luftfeuchtigkeit. Hierzu haben wir die Daten logarithmiert, wodurch das Verhalten der Luftfeuchtigkeit besser ersichtlich ist. Es fällt besonders auf, dass in den Hochphasen von Canberra geringere Luftfeuchtigkeiten in Darwin gemessen werden und umgekehrt. Man erkennt somit ein anti-zyklisches Muster und, dass es in Canberra im Mittel über das Jahr feuchter als in Darwin ist.





### Übersicht der Regentage

Schauen wir uns die aufsummierte Anzahl an Regentagen pro Monat an, so erkennen wir deutlich, dass es in Darwin offensichtlich Regenzeiten und Trockenzeiten gibt, wohin gegen der Regen in Canberra über das Jahr hin leichten, aber nicht auffälligen Schwankungen unterliegt.

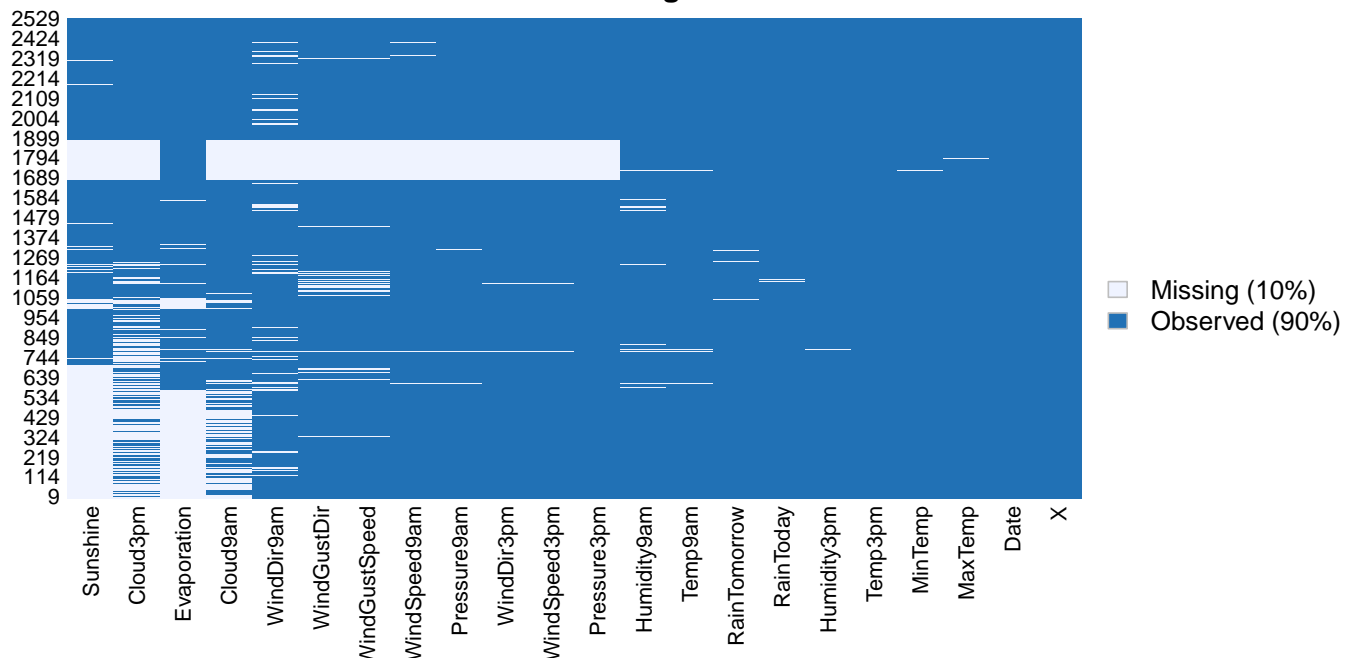


## Feature Engineering

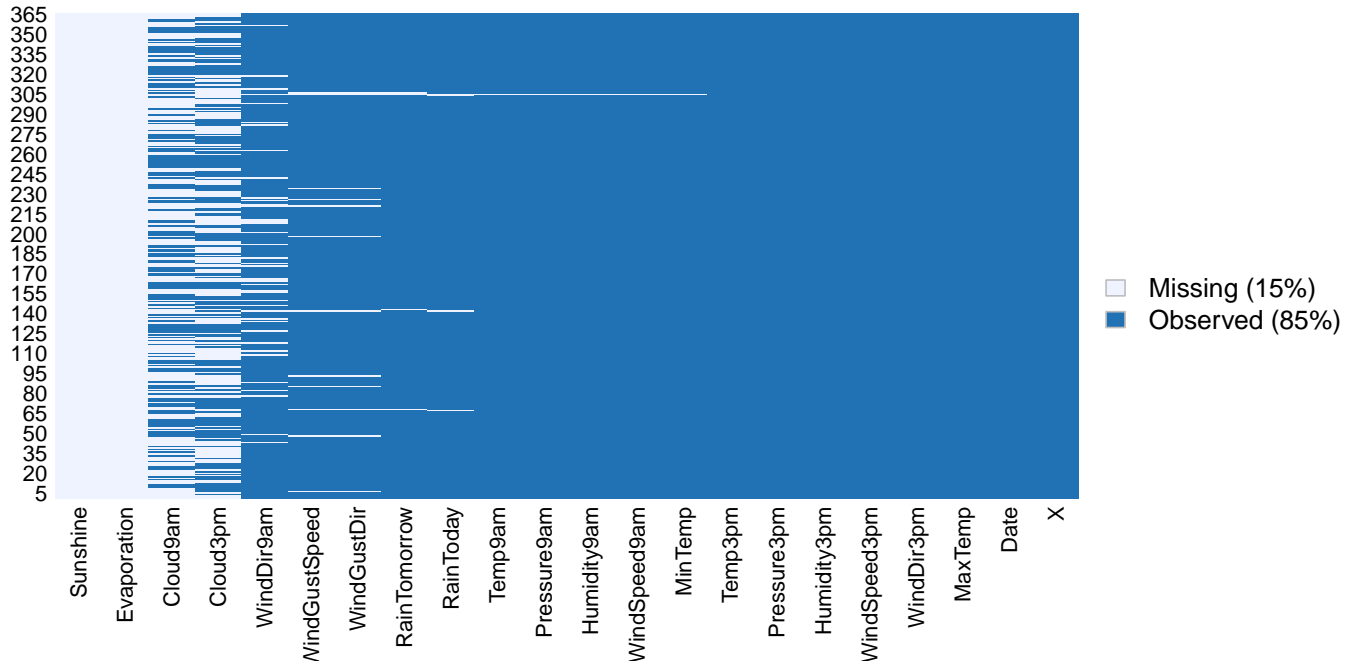
### Erstellung der Merkmale

Wie zuvor erwähnt müssen einige Spalten in numerische Werte kodiert werden, damit diese vom neuronalen Netz genutzt werden können. Hierzu wenden wir nun One-Hot-Encoding an, was jedoch zum Nachteil hat, dass sich die Dimension unserer Trainingsdaten massiv erweitert. Zunächst schauen wir uns jedoch einmal die Reinheit des Datensatz an, um gegebenenfalls fehlende Werte herauszunehmen oder zu ersetzen.

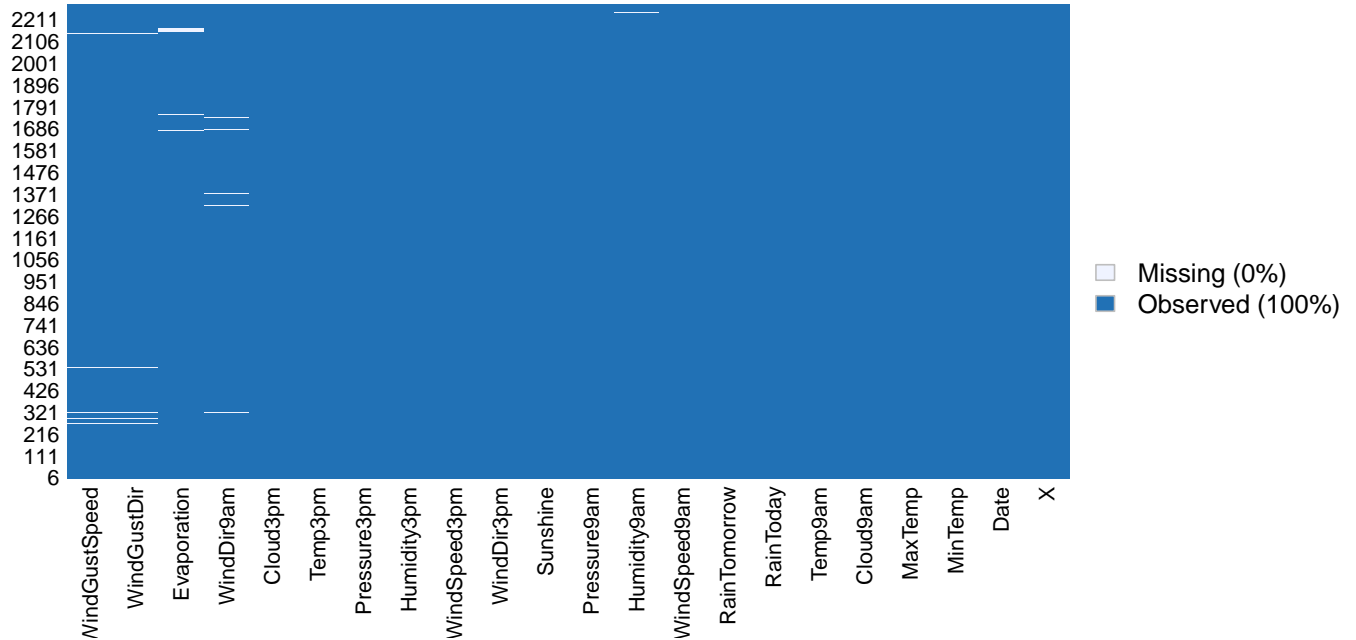
### Canberra – Training Data

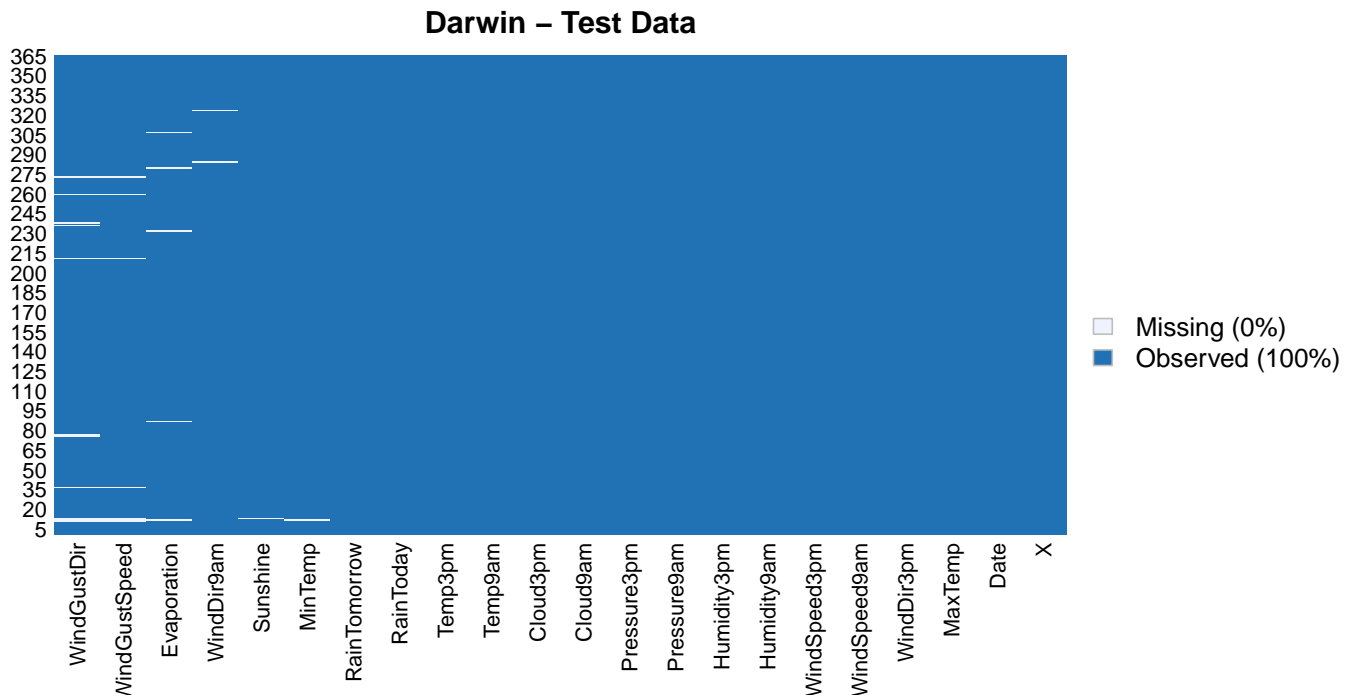


Canberra – Test Data

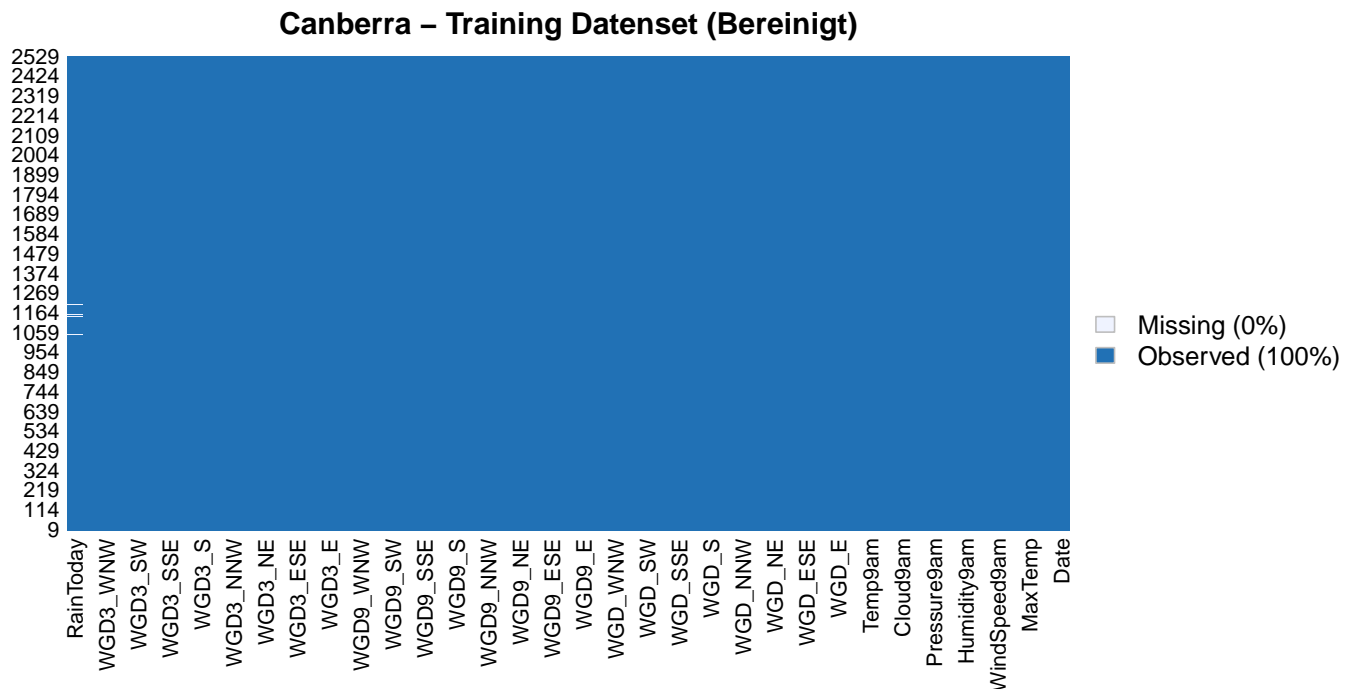


Darwin – Training Data





Der Canberra Trainingsdatensatz hat beinhaltet ca. 10% fehlende Werte, welche es zu ersetzen gilt. Der Testdatensatz beinhaltet sogar noch mehr fehlende Werte (15%). Insbesondere fehlen Werte in den Spalten Sunshine und Evaporation komplett, weshalb wir diese Spalten komplett entfernen werden und die restlichen fehlenden Werte berechnen wird mittels Random Forest Regression. Hierzu nutzen wir das Package *missForest*.



Die Anzahl an Features ist durch das One-Hot-Encoding von 22 auf 63 gewachsen. Das kann man soweit für das Netz auch erst einmal lassen ohne Dimensionsreduktion durchzuführen.

### Long Short-Term Memory (LSTM)

Das Long Short-Term Memory (LSTM) ist ein rekurrentes neuronales Netz. Wir haben zuvor die Wetterdaten betrachtet, welche als Sequenz interpretiert werden können. Wir wollen nun ein Modell erstellen, welches die Regenwahrscheinlichkeit für den Folgetag prognostizieren kann. Unser Ziel ist es den Folgetag als Regentag oder als Nicht-Regentag zu klassifizieren.

Zum Verständnis schauen wir uns im Detail an, wie die LSTM Zelle aufgebaut ist.

$$f^{(t)} = (W_f h^{(t-1)} + U_f x^{(t)} + V_f c^{(t-1)}) \quad (1)$$

$$i^{(t)} = (W_i h^{(t-1)} + U_i x^{(t)} + V_i c^{(t-1)}) \quad (2)$$

$$c^{(t)} = \tanh(W_c h^{(t-1)} + U_c x^{(t)}) \quad (3)$$

$$c^{(t)} = f^{(t)} c^{(t-1)} + i^{(t)} c^{(t)} \quad (4)$$

$$o^{(t)} = (W_o h^{(t-1)} + U_o x^{(t)} + V_o c^{(t)}) \quad (5)$$

$$h^{(t)} = o^{(t)} \tanh(c^{(t)}) \quad (6)$$

Rekurrente Netze sind in der Lage Sequenzen zu erlernen, da die Eingabe der Beobachtungen jeweils zu einem gegebenen Zeitpunkt  $t$  geschehen. Wichtig ist hierbei, dass das Modell weiß, welche Beobachtungen bisher erfolgt sind, wodurch eine Art Gedächtnis benötigt wird. Im Gegensatz zum herkömmlichen neuronalen Netz haben rekurrente Netze einen inneren Zustand, welche als Gedächtnis gesehen werden kann. Dieses Gedächtnis wird über die Gewichte  $U, W$  und  $V$  gewährleistet und über das gesamte Netz geteilt. Während  $U$  die Gewichte des Input repräsentieren, stellen die Gewichte  $W$  die Gewichte des Hidden-Layer dar.  $V$  hingegen bildet die Gewichte zum Ausgangs-layer ab.

Gl.1-6 zeigen die Berechnungen der LSTM Zelle. Dabei stellt Gl.1 das Forget-Gate dar. Gate im Sinne von Schleusen oder Toren steuern den Wissenszustand des Netzes. Das Forget-Gate (zu deutsch vergessen) entfernt Informationen aus dem Gedächtnis. Der Werte Bereich der Sigmoid Funktion beläuft sich auf  $[0,1]$  wodurch Werte nahe 0 als vergessen interpretiert werden können. Weiterhin fällt aber auch auf, dass das Forget-Gate eigene Gewichtsmatrizen  $U_f, W_f$  und  $V_f$  besitzt, welche es zu optimieren gilt. Das Pendant zum Vergessen ist das Erlernen, welches durch das Input-Gate in Gl.2 dargestellt wird. Neue Beobachtungen werden hierüber gewichtet, wie stark deren Einfluss Faktor ist und berücksichtigt dabei aber auch erlerntes. Wir erkennen das auch hier wieder eigene Gewichtsmatrizen  $U_i, W_i$  und  $V_i$  vorliegen.



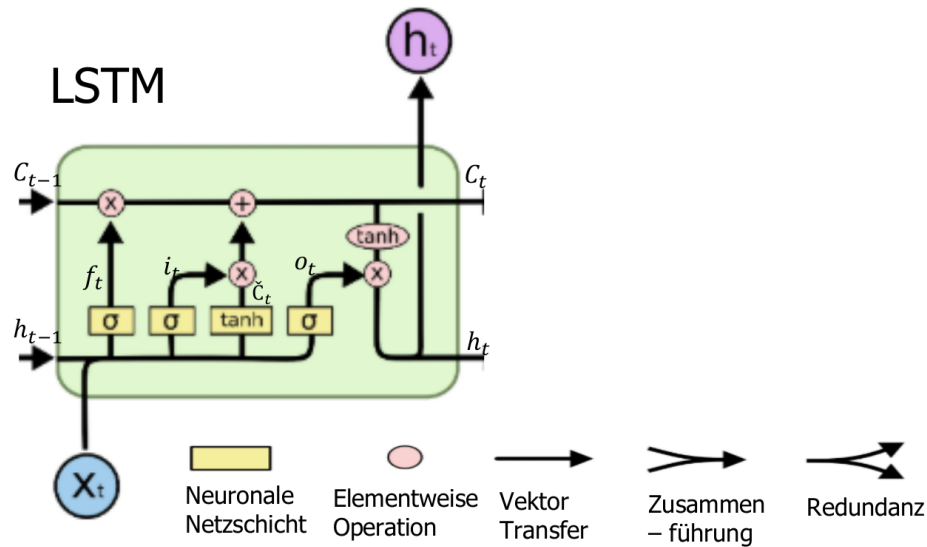


Abbildung 2: LSTM - Zell Aufbau

Quelle: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

In Abb.2 sehen wir den Aufbau der LSTM Zelle. Die Gewichtsmatrizen sind an den jeweiligen neuronalen Schichten platziert. Gl.4 zeigt den Kontrollfluss des Gedächtnis auf, sowohl das Forget-Gate als auch das Input-Gate beeinflussen es. Ein weiteren Einfluss nimmt das Output-Gate auf den Kontroll-Fluss. Die Besonderheit hierbei ist, dass  $c^t$  der Zelle einen aktuellen Zustand des Gedächtnis wiedergibt, während  $h^t$  die tatsächliche Ausprägung repräsentiert (vgl. Gl.6). Übertragen wir dies einmal auf einen Menschen, speziell einen Student. Während der Klausurphase wird, dass Input-Gate massiv mit neuen Informationen belastet, gleichzeitig aber auch das Forget-Gate, welches gerade irrelevantes Wissen verdrängt, um Platz für Neues zu schaffen. Heißt jedoch nicht, dass dieses Wissen endgültig verloren ist, es wird einfach temporär verdrängt (bspw. der Sommerurlaub des letzten Jahres). Die schönen Erinnerungen sind jedoch weiterhin beständig in  $c$ , jedoch nicht in  $h$ . Nach der Klausurphase lässt sich jedoch oft beobachten, dass das Forget-Gate gegen 0 konvergiert und so alles neu erlernte erstmal verdrängt, um Platz für neue Eindrücke und Erfahrungen für die Semesterferien zu schaffen. Diese neuen Eindrücke werden dann als  $c^t$  (vgl. Gl.3) erfasst und rufen zudem Erinnerungen aus vergangenen Semesterferien in  $c^t$  hoch.

Wir werden nachfolgend jeweils ein LSTM für Canberra und Darwin erzeugen. Hierbei ziehen wir jeweils ein Klassifikationsproblem heran, als auch ein Regressionsproblem. Bei dem Klassifikationsproblem handelt es sich um binäres Problem, weshalb wir als finale Aktivierungsfunktion die Sigmoid-Funktion nutzen werden. Das Regressionsproblem werden wir jeweils mit der ReLU-Funktion ausstatten. Für das binäre Klassifikationsproblem nehmen wir die Kreuzentropie als Fehlermaß, wobei das MSE auf unser Regressionsproblem angewendet wird. Es sei noch angemerkt, dass rekurrente neuronale ähnlich zu normalen neuronalen Netzen lernen. Der Lernprozess (Backpropagation) wird ebenfalls bestimmt durch das zurück propagieren der Gradienten der Fehlerfunktion, welches über ein Minimierungsproblem quantifiziert wird. Wichtig ist jedoch hervorzuheben, dass das zurückpropagieren im zeitlichen Verlauf rückwärts geschieht, wodurch dieses Verfahren auch den Namen *Backpropagation Through Time* (BPTT) erhält.

Ziel dieses Praktikum wird insbesondere sein, in wie weit sich die Modellgüte dieser Modell unterscheidet. Wir haben gesehen, dass die Daten unterschiedliche Varianzen im zeitlichen Verlauf aufzeigen. Insbesondere die Stadt Darwin, welche durch Regen - und Trockenzeit geprägt ist.