Name: Arnab Ghosh

Group No: 058

Group Members: Arnab Ghosh, Sarbajit Roy, RISHABH RAKESH

Dataset Name: weatherHistory.csv

Import Libraries & Data

Predict the Humidity

In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
##Preprocessing Imports

from sklearn import preprocessing

## Visualization Imports
import matplotlib.pyplot as plt
from matplotlib.dates import DateFormatter
import seaborn as sns

##Model Building Imports
from sklearn.linear_model import LinearRegression,Ridge
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split

## Metrics Import
from sklearn.metrics import mean_squared_error,max_error,confusion_matrix

## Feature Selection Imports
from sklearn.feature_selection import SelectKBest, f_regression,mutual_info_regression

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directo

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you cre
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

Load Dataset

> Use the Date Column 'Formatted Date' as the index column

In [2]:
```
weather_hourly = pd.read_csv('weatherHistory.csv',
                             index_col=['Formatted Date'],
                             na_values=['9999.99'])
weather_hourly.index = weather_hourly.index.str.replace('\+0200','')
weather_hourly.index=pd.to_datetime(weather_hourly.index,format="%Y-%m-%d %H:%M:%S", utc= True)
weather_hourly.head(5)
```

Out[2]:

Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)	Daily Summary
2006-04-01 00:00:00+00:00	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251.0	15.8263	0.0	1015.13	Partly cloudy throughout the day.
2006-04-01 01:00:00+00:00	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259.0	15.8263	0.0	1015.63	Partly cloudy throughout the day.
2006-04-01	Mostly										Partly cloudy

02:00:00+00:00	Cloudy	rain	9.377778	9.377778	0.89	3.9284	204.0	14.9569	0.0	1015.94	throughout the day.
2006-04-01 03:00:00+00:00	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	269.0	15.8263	0.0	1016.41	Partly cloudy throughout the day.
2006-04-01 04:00:00+00:00	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	259.0	15.8263	0.0	1016.51	Partly cloudy throughout the day.

Data Exploration & Visualization

Description of the dataset

In [3]:
```
weather_hourly.describe()
```

Out[3]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
count	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.0	96453.000000
mean	11.932678	10.855029	0.734899	10.810640	187.509232	10.347325	0.0	1003.235956
std	9.551546	10.696847	0.195473	6.913571	107.383428	4.192123	0.0	116.969906
min	-21.822222	-27.716667	0.000000	0.000000	0.000000	0.000000	0.0	0.000000
25%	4.688889	2.311111	0.600000	5.828200	116.000000	8.339800	0.0	1011.900000
50%	12.000000	12.000000	0.780000	9.965900	180.000000	10.046400	0.0	1016.450000
75%	18.838889	18.838889	0.890000	14.135800	290.000000	14.812000	0.0	1021.090000
max	39.905556	39.344444	1.000000	63.852600	359.000000	16.100000	0.0	1046.380000

Shape of the dataset

In [4]:
```
weather_hourly.shape
```

Out[4]:
```
(96453, 11)
```

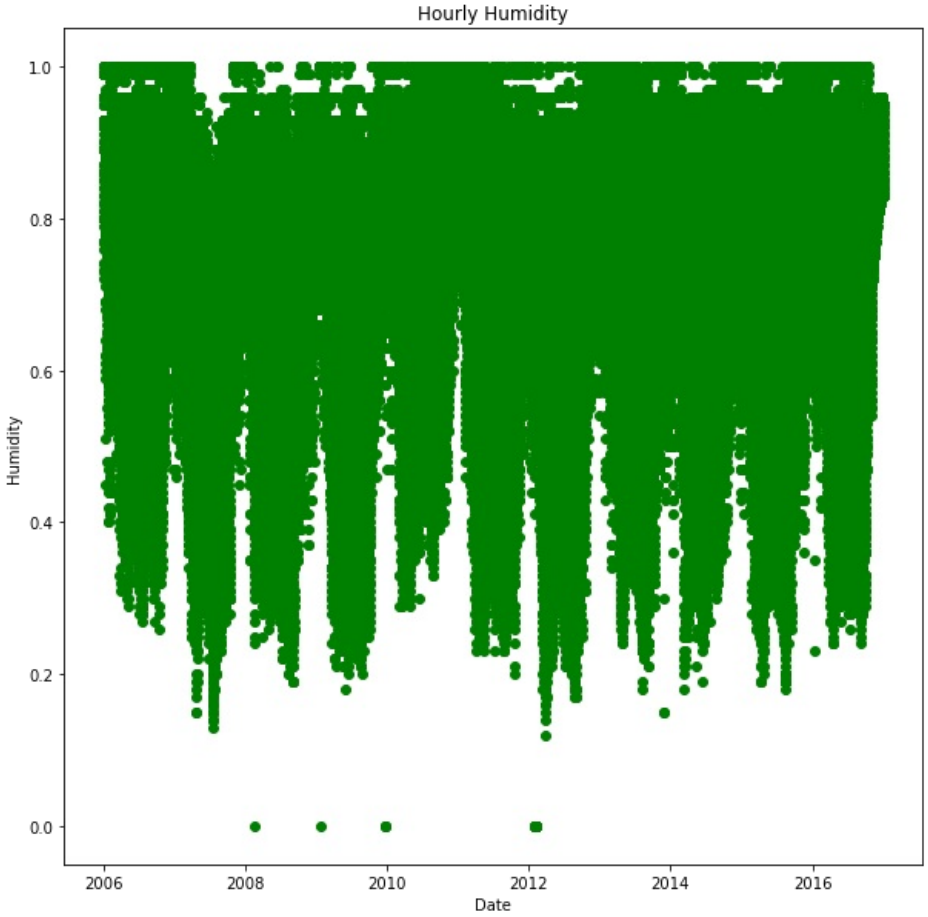
Features of the Dataset

In [5]:
```
class color:
   PURPLE = '\033[95m'
   CYAN = '\033[96m'
   DARKCYAN = '\033[36m'
   BLUE = '\033[94m'
   GREEN = '\033[92m'
   YELLOW = '\033[93m'
   RED = '\033[91m'
   BOLD = '\033[1m'
   UNDERLINE = '\033[4m'
   END = '\033[0m'

print(f'{color.BOLD} Features of the dataset{color.END}')
print(f'{color.BLUE} {weather_hourly.columns} {color.END}')

print(f'{color.BOLD} Categorical features of the dataset {color.END}')
categorical_features = weather_hourly.select_dtypes(include='object').columns
print(f'{color.BLUE} {categorical_features} {color.END}')
print(f'{color.BOLD} Continous features of the dataset {color.END}')
continous_features = weather_hourly.select_dtypes(exclude='object').columns
print(f' {color.BLUE} {continous_features} {color.END}')
```

```
 Features of the dataset
Index(['Summary', 'Precip Type', 'Temperature (C)', 'Apparent Temperature (C)',
       'Humidity', 'Wind Speed (km/h)', 'Wind Bearing (degrees)',
       'Visibility (km)', 'Loud Cover', 'Pressure (millibars)',
       'Daily Summary'],
      dtype='object')
 Categorical features of the dataset
Index(['Summary', 'Precip Type', 'Daily Summary'], dtype='object')
 Continous features of the dataset
 Index(['Temperature (C)', 'Apparent Temperature (C)', 'Humidity',
       'Wind Speed (km/h)', 'Wind Bearing (degrees)', 'Visibility (km)',
       'Loud Cover', 'Pressure (millibars)'],
      dtype='object')
```

Plot Hourly Humidity

In [6]:
```
fig, ax = plt.subplots(figsize=(10,10))
ax.scatter(weather_hourly.index.values,
           weather_hourly['Humidity'],
           color='green')
ax.set(xlabel='Date',ylabel='Humidity',title='Hourly Humidity')
plt.show()
```



Correlation matrix for hourly data

In [7]:
```
#drop the `Loud Cover` column has it has only 0s
weather_hourly_1 = None
weather_hourly_1 = weather_hourly.drop(labels='Loud Cover',axis=1)
weather_hourly_1.corr('spearman').style.background_gradient(cmap='RdYlGn')
```

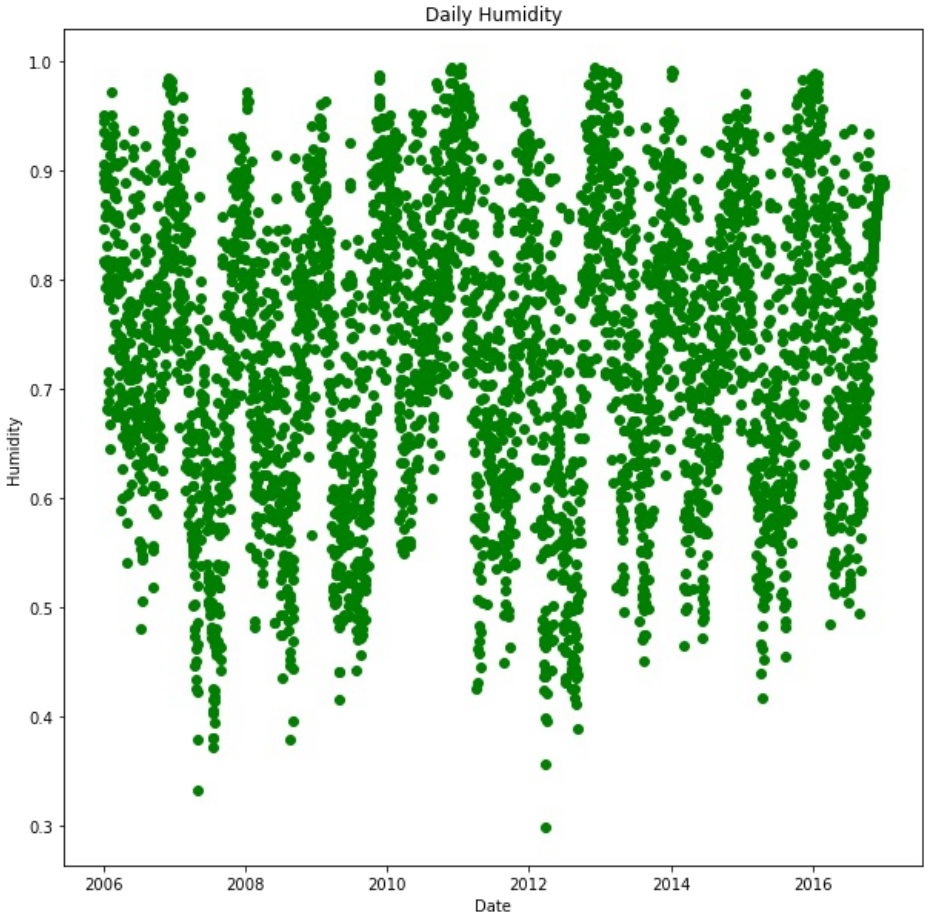
Out[7]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)
Temperature (C)	1.000000	0.995974	-0.588106	0.016890	0.030243	0.394442	-0.303346
Apparent Temperature (C)	0.995974	1.000000	-0.572551	-0.039162	0.027446	0.378676	-0.282621
Humidity	-0.588106	-0.572551	1.000000	-0.261980	-0.001967	-0.432075	0.042993
Wind Speed (km/h)	0.016890	-0.039162	-0.261980	1.000000	0.086397	0.100856	-0.227787
Wind Bearing (degrees)	0.030243	0.027446	-0.001967	0.086397	1.000000	0.051804	-0.072843
Visibility (km)	0.394442	0.378676	-0.432075	0.100856	0.051804	1.000000	-0.130756
Pressure (millibars)	-0.303346	-0.282621	0.042993	-0.227787	-0.072843	-0.130756	1.000000

- Resample Hourly Humidity to Daily Humidity
- Plot Daily Humidity

In [8]:
```
weather_daily = weather_hourly.resample('D').aggregate('mean')
fig, ax = plt.subplots(figsize=(10,10))

ax.scatter(weather_daily.index.values,
```

```
          weather_daily['Humidity'], color='green')
ax.set(xlabel='Date',ylabel='Humidity',title='Daily Humidity')
plt.show()
```



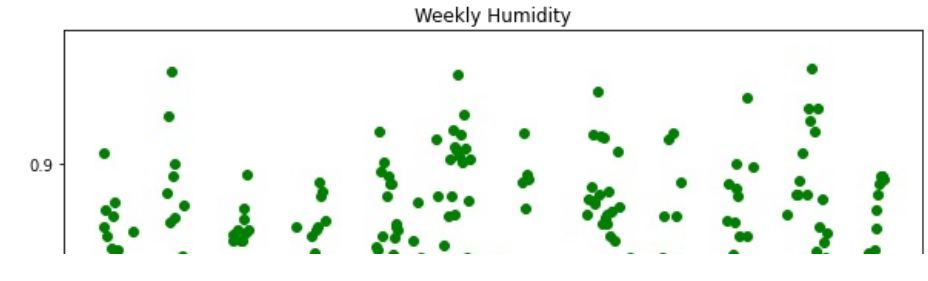
In [9]:
```
weather_daily_1 = weather_daily.drop(labels='Loud Cover',axis=1)
weather_daily_1.corr('spearman').style.background_gradient(cmap='RdYlGn')
```

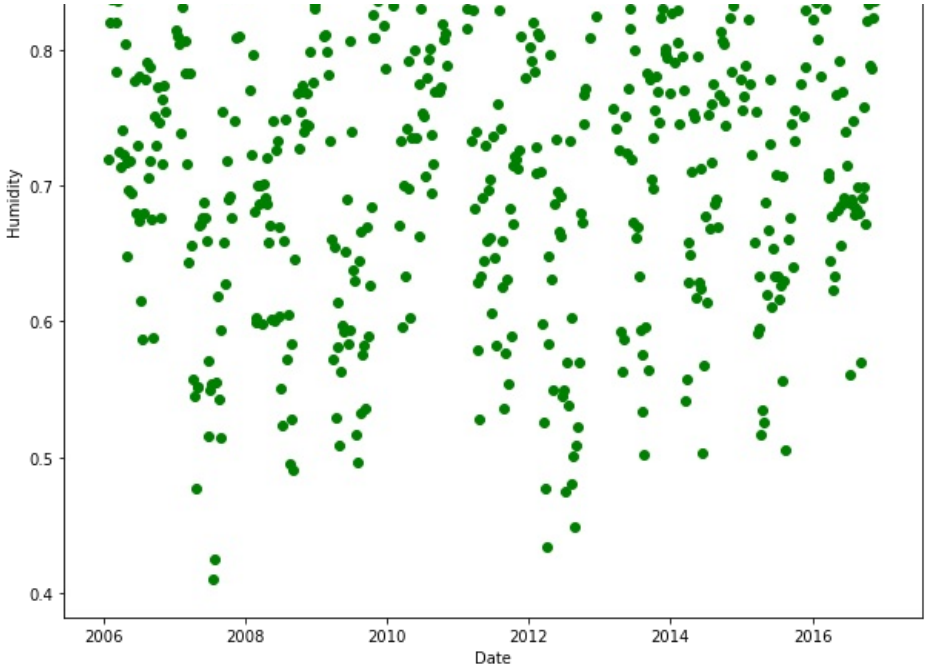
Out[9]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)
Temperature (C)	1.000000	0.998103	-0.605610	-0.130891	0.036182	0.550581	-0.275456
Apparent Temperature (C)	0.998103	1.000000	-0.594838	-0.169512	0.031809	0.534301	-0.258583
Humidity	-0.605610	-0.594838	1.000000	-0.055953	-0.012508	-0.708880	0.024981
Wind Speed (km/h)	-0.130891	-0.169512	-0.055953	1.000000	0.166392	0.190475	-0.281050
Wind Bearing (degrees)	0.036182	0.031809	-0.012508	0.166392	1.000000	0.101916	-0.114356
Visibility (km)	0.550581	0.534301	-0.708880	0.190475	0.101916	1.000000	-0.114600
Pressure (millibars)	-0.275456	-0.258583	0.024981	-0.281050	-0.114356	-0.114600	1.000000

- Resample Daily Humidity to Weekly Humidity
- Plot Weekly Humidity

In [10]:
```
weather_weekly = weather_daily.resample('W').aggregate('mean')
fig, ax = plt.subplots(figsize=(10,10))

ax.scatter(weather_weekly.index.values,
          weather_weekly['Humidity'], color='green')
ax.set(xlabel='Date',ylabel='Humidity',title='Weekly Humidity')
plt.show()
```





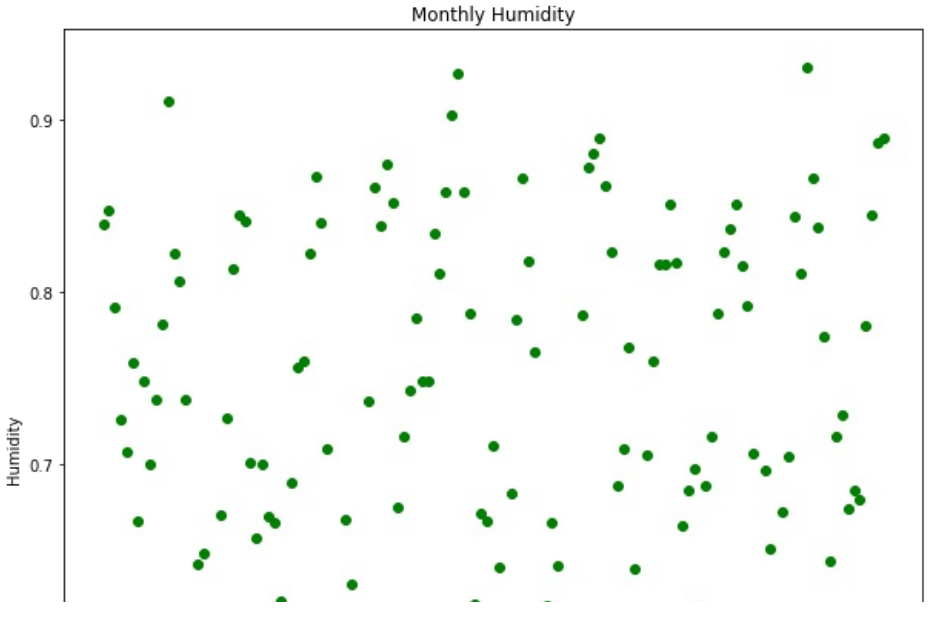
In [11]:
```
weather_weekly_1 = weather_weekly.drop(labels='Loud Cover',axis=1)
weather_weekly_1.corr().style.background_gradient(cmap='RdYlGn')
```

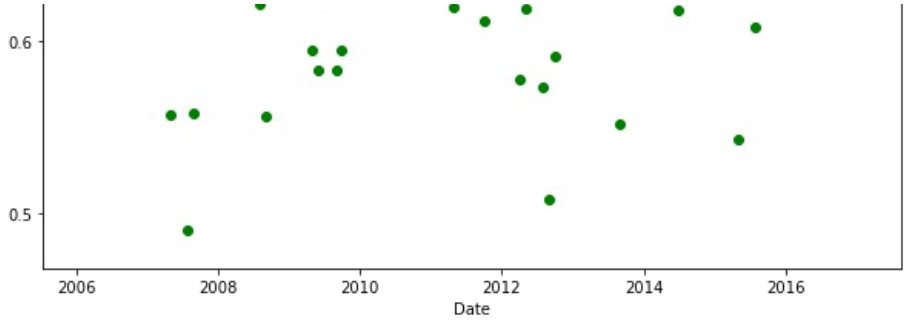
Out[11]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)
Temperature (C)	1.000000	0.997925	-0.672089	-0.281170	0.146814	0.695543	0.012994
Apparent Temperature (C)	0.997925	1.000000	-0.662691	-0.312781	0.140636	0.683497	0.022341
Humidity	-0.672089	-0.662691	1.000000	0.008988	-0.151060	-0.743851	-0.001728
Wind Speed (km/h)	-0.281170	-0.312781	0.008988	1.000000	0.207445	0.136795	-0.134813
Wind Bearing (degrees)	0.146814	0.140636	-0.151060	0.207445	1.000000	0.210366	-0.051255
Visibility (km)	0.695543	0.683497	-0.743851	0.136795	0.210366	1.000000	0.057441
Pressure (millibars)	0.012994	0.022341	-0.001728	-0.134813	-0.051255	0.057441	1.000000

- Resampe Weekly Humidity to Monthly Humidity
- Plot Monthly Humidity

In [12]:
```
weather_monthly = weather_weekly.resample('M').aggregate('mean')
fig, ax = plt.subplots(figsize=(10,10))

ax.scatter(weather_monthly.index.values,
           weather_monthly['Humidity'], color='green')
ax.set(xlabel='Date',ylabel='Humidity',title='Monthly Humidity')
plt.show()
```





Correlation Analysis

In [13]:
```
weather_monthly.corr().style.background_gradient(cmap="RdYlGn")
```

```
C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\formats\style.py:1126: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(s.to_numpy()) if vmin is None else vmin
C:\ProgramData\Anaconda3\lib\site-packages\pandas\io\formats\style.py:1127: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(s.to_numpy()) if vmax is None else vmax
```

Out[13]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
Temperature (C)	1.000000	0.998835	-0.747787	-0.452034	0.318978	0.799300	nan	0.027001
Apparent Temperature (C)	0.998835	1.000000	-0.743199	-0.469835	0.306240	0.794878	nan	0.036615
Humidity	-0.747787	-0.743199	1.000000	0.144515	-0.301536	-0.797538	nan	-0.050663
Wind Speed (km/h)	-0.452034	-0.469835	0.144515	1.000000	0.142272	-0.090679	nan	-0.176954
Wind Bearing (degrees)	0.318978	0.306240	-0.301536	0.142272	1.000000	0.325334	nan	-0.042210
Visibility (km)	0.799300	0.794878	-0.797538	-0.090679	0.325334	1.000000	nan	0.083049
Loud Cover	nan	nan	nan	nan	nan	nan	nan	nan
Pressure (millibars)	0.027001	0.036615	-0.050663	-0.176954	-0.042210	0.083049	nan	1.000000

As we can see the column `Loud Cover` has all zeros , lets drop the columan and then visualize the correlation matrix

In [14]:
```
weather_monthly_1 = weather_monthly.drop(labels='Loud Cover',axis=1)
weather_monthly_1.corr().style.background_gradient(cmap='RdYlGn')
```

Out[14]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)
Temperature (C)	1.000000	0.998835	-0.747787	-0.452034	0.318978	0.799300	0.027001
Apparent Temperature (C)	0.998835	1.000000	-0.743199	-0.469835	0.306240	0.794878	0.036615
Humidity	-0.747787	-0.743199	1.000000	0.144515	-0.301536	-0.797538	-0.050663
Wind Speed (km/h)	-0.452034	-0.469835	0.144515	1.000000	0.142272	-0.090679	-0.176954
Wind Bearing (degrees)	0.318978	0.306240	-0.301536	0.142272	1.000000	0.325334	-0.042210
Visibility (km)	0.799300	0.794878	-0.797538	-0.090679	0.325334	1.000000	0.083049
Pressure (millibars)	0.027001	0.036615	-0.050663	-0.176954	-0.042210	0.083049	1.000000

Data PreProcessing & Cleanup

Hourly Dataset

Check data quality

In [15]:
```
weather_hourly.isnull().sum()
```

Out[15]:
```
Summary                   0
```

```
Precip Type                 517
Temperature (C)               0
Apparent Temperature (C)      0
Humidity                      0
Wind Speed (km/h)             0
Wind Bearing (degrees)        0
Visibility (km)               0
Loud Cover                    0
Pressure (millibars)          0
Daily Summary                 0
dtype: int64
```

> Dataset is pretty clean with only 1 column 'Precip Type' having null values. As part of the 1st iteration, lets drop this feature and proceed. In the later part we will include this feature and check if it helps in giving a better prediction of Humidity

In [16]:
```
def min_max(input_df):
    max_series = input_df.max(numeric_only=True)
    min_series = input_df.min(numeric_only=True)
    min_max = pd.DataFrame(max_series).transpose().append(pd.DataFrame(min_series).transpose())
    return min_max

min_max(weather_hourly)
```

Out[16]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
0	39.905556	39.344444	1.0	63.8526	359.0	16.1	0.0	1046.38
0	-21.822222	-27.716667	0.0	0.0000	0.0	0.0	0.0	0.00

Steps Performed

- Drop the Precipe Type column
- Drop the target column **Humidity**
- The numerical (continous) features have values in different scale. e.g **Apparent Temperature** values ranges between *39.34444* and *-27.7166* while **Wind Speed** values ranges between *359* and *0*. We use `StandardScalar` to perform normalization
- The categorical columns are label encoded using `LabelEncoder`

In [17]:
```
class DataWrangler_Approach_A():
    def __init__(self,df):
        self.df = df

    def _get_numerical_columns(self):
        return self.df.select_dtypes(include=['float64','int64']).columns

    def _get_categorical_columns(self):
        return self.df.select_dtypes(include=['object']).columns

    def _remove_columns(self,labels):
        self.df.drop(labels=labels,axis=1,inplace=True)


    def _drop_rows_with_null(self):
        self.df.dropna(inplace=True,axis=1)

    def _scale_numerical_features(self):
        scaler = preprocessing.StandardScaler()
        numerical_cols = self._get_numerical_columns()
        numerical_df = self.df.loc[:,numerical_cols]
        self.df.drop(labels=numerical_cols, axis=1, inplace=True)
        scaler.fit(numerical_df)
        scaled_cols = scaler.transform(numerical_df)
        self.df[numerical_cols] = scaled_cols

    def _label_encode_categories(self):
        le = preprocessing.LabelEncoder()
        categorical_cols = self._get_categorical_columns()
        categorical_df = self.df.loc[:,categorical_cols]
        categorical_df = categorical_df.apply(le.fit_transform)
        self.df[categorical_cols] = categorical_df


    def perform_wrangling(self):
        #self._remove_columns(['Humidity'])
        self._drop_rows_with_null()
        self._scale_numerical_features()
        self._label_encode_categories()
        dropna=True
```

```
        return (self.df,dropna)
```

In [18]:
```
raw_df = pd.read_csv('weatherHistory.csv',
                             index_col=['Formatted Date'],
                             na_values=['9999.99'])
raw_df.index = raw_df.index.str.replace('\+0200','')
raw_df.index=pd.to_datetime(raw_df.index,format="%Y-%m-%d %H:%M:%S", utc= True)
raw_df.head(5)
y = raw_df['Humidity']
raw_df.drop(labels=['Loud Cover'],axis=1,inplace=True)
(pre_processed_df, dropna) = DataWrangler_Approach_A(raw_df).perform_wrangling()
pre_processed_df.head()
```

Out[18]:

Formatted Date	Summary	Daily Summary	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)
2006-04-01 00:00:00+00:00	19	197	-0.257599	-0.324035	0.793470	0.478635	0.591256	1.306976	0.101685
2006-04-01 01:00:00+00:00	19	197	-0.269814	-0.339097	0.639996	0.499594	0.665756	1.306976	0.105960
2006-04-01 02:00:00+00:00	17	197	-0.267487	-0.138102	0.793470	-0.995473	0.153570	1.099586	0.108610
2006-04-01 03:00:00+00:00	19	197	-0.381489	-0.459071	0.486521	0.476306	0.758881	1.306976	0.112628
2006-04-01 04:00:00+00:00	17	197	-0.332631	-0.362469	0.486521	0.033841	0.665756	1.306976	0.113483

In [19]:
```
pre_processed_df.corr().style.background_gradient(cmap='RdYlGn')
```

Out[19]:

	Summary	Daily Summary	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Pressure (millibars)
Summary	1.000000	0.157569	0.146658	0.142760	-0.113119	0.004258	0.022789	0.140138	0.143405
Daily Summary	0.157569	1.000000	0.438879	0.433791	-0.368180	-0.004115	0.053709	0.457913	0.054305
Temperature (C)	0.146658	0.438879	1.000000	0.992629	-0.632255	0.008957	0.029988	0.392847	-0.005447
Apparent Temperature (C)	0.142760	0.433791	0.992629	1.000000	-0.602571	-0.056650	0.029031	0.381718	-0.000219
Humidity	-0.113119	-0.368180	-0.632255	-0.602571	1.000000	-0.224951	0.000735	-0.369173	0.005454
Wind Speed (km/h)	0.004258	-0.004115	0.008957	-0.056650	-0.224951	1.000000	0.103822	0.100749	-0.049263
Wind Bearing (degrees)	0.022789	0.053709	0.029988	0.029031	0.000735	0.103822	1.000000	0.047594	-0.011651
Visibility (km)	0.140138	0.457913	0.392847	0.381718	-0.369173	0.100749	0.047594	1.000000	0.059818
Pressure (millibars)	0.143405	0.054305	-0.005447	-0.000219	0.005454	-0.049263	-0.011651	0.059818	1.000000

In [20]:
```
# Drop the target column
pre_processed_df.drop(labels=['Humidity'],axis=1,inplace=True)
```

In [21]:
```
def select_feature(feat_sel_func,k,**kwargs):
    fs = SelectKBest(score_func=feat_sel_func,k=k)
    fs.fit(X_train,y_train)
    X_train_fs = fs.transform(X_train)
    X_test_fs = fs.transform(X_test)
    return X_train_fs,X_test_fs,fs
```

In [22]:
```
def select_feature_freg(**kwargs):
    feat_count = kwargs['feature_count']
    X_train_fs, X_test_fs, fs = select_feature(f_regression,feat_count,
                                               X_test=X_test,
                                               X_train=X_train,y_train=y_train)
    mask = fs.get_support(indices=True)
    features = [pre_processed_df.columns[index] for index in mask]
    scores = [fs.scores_[index] for index in mask]
    selected_feature_df = pd.DataFrame(X_train,columns=features)
    if print_score:
        for i in range(len(scores)):
            print(f'Feature {selected_feature_df.columns[i]} score : {fs.scores_[i]}')

    plt.barh([selected_feature_df.columns[i] for i in range(len(features))], scores)
    plt.tight_layout()
    plt.show()
    return X_train_fs, X_test_fs
```

In [23]:
```
def select_feature_mutualinfo(**kwargs):
```

```
    feat_count=kwargs['feature_count']
    X_train_fs, X_test_fs, fs = select_feature(mutual_info_regression,feat_count,
                                               X_test=X_test,
                                               X_train=X_train,y_train=y_train)
    mask = fs.get_support(indices=True)
    features = [pre_processed_df.columns[index] for index in mask]
    scores = [fs.scores_[index] for index in mask]
    selected_feature_df = pd.DataFrame(X_train,columns=features)
    if print_score:
        for i in range(len(scores)):
            print(f'Feature {selected_feature_df.columns[i]} score : {fs.scores_[i]}')

    plt.barh([selected_feature_df.columns[i] for i in range(len(features))], scores)
    plt.tight_layout()
    plt.show()
    return X_train_fs,X_test_fs
```

In [24]:
```
def perform_linearreg(**kwargs):
    X_train = kwargs['X_train']
    y_train = kwargs['y_train']
    X_test = kwargs['X_test']
    y_test = kwargs['y_test']
    reg = LinearRegression().fit(X_train, y_train)
    score = reg.score(X_test, y_test)
    y_pred = reg.predict(X_test)
    print(f'Training dataset shape: {X_train.shape}')
    print(f'Test dataset shape: {X_test.shape}')
    print(f'{color.BOLD}Accuracy Score {color.END} {score:.4f}')
    print(f'{color.BOLD}Mean squared error:{color.END} {mean_squared_error(y_test, y_pred):.4f}')
    print(f'{color.BOLD}Root Mean squared error:{color.END} {mean_squared_error(y_test, y_pred,squared=False):.4f

    fig, ax = plt.subplots(figsize=(10,10))
    ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
    ax.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=4)
    ax.set_xlabel('Measured')
    ax.set_ylabel('Predicted')
    plt.show()
```

In [25]:
```
def perform_ridgereg(**kwargs):
    X_train = kwargs['X_train']
    y_train = kwargs['y_train']
    X_test = kwargs['X_test']
    y_test = kwargs['y_test']
    r_reg = Ridge(alpha=1.0,solver='svd').fit(X_train, y_train)
    score = r_reg.score(X_test, y_test)
    y_pred = r_reg.predict(X_test)
    print(f'Training dataset shape: {X_train.shape}')
    print(f'Test dataset shape: {X_test.shape}')
    print(f'{color.BOLD}Accuracy Score {color.END} {score:.4f}')
    print(f'{color.BOLD}Mean squared error:{color.END} {mean_squared_error(y_test, y_pred):.4f}')
    print(f'{color.BOLD}Root Mean squared error:{color.END} {mean_squared_error(y_test, y_pred,squared=False):.4f

    fig, ax = plt.subplots(figsize=(10,10))
    ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
    ax.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=4)
    ax.set_xlabel('Measured')
    ax.set_ylabel('Predicted')
    plt.show()
```

In [26]:
```
def perform_decisiontreereg(**kwargs):
    X_train = kwargs['X_train']
    y_train = kwargs['y_train']
    X_test = kwargs['X_test']
    y_test = kwargs['y_test']
    dtree_reg = DecisionTreeRegressor(random_state=0).fit(X_train, y_train)
    score = dtree_reg.score(X_test, y_test)
    y_pred = dtree_reg.predict(X_test)
    print(f'Training dataset shape: {X_train.shape}')
    print(f'Test dataset shape: {X_test.shape}')
    print(f'{color.BOLD}Accuracy Score {color.END} {score:.4f}')
    print(f'{color.BOLD}Mean squared error:{color.END} {mean_squared_error(y_test, y_pred):.4f}')
    print(f'{color.BOLD}Root Mean squared error:{color.END} {mean_squared_error(y_test, y_pred,squared=False):.4f

    fig, ax = plt.subplots(figsize=(10,10))
    ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
    ax.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=4)
    ax.set_xlabel('Measured')
    ax.set_ylabel('Predicted')
    plt.show()
```

In [27]:
```
def perform_randomforestreg(**kwargs):
    X_train = kwargs['X_train']
    y_train = kwargs['y_train']
    X_test = kwargs['X_test']
    y_test = kwargs['y_test']
    random_reg = RandomForestRegressor(n_estimators = 100, random_state=0).fit(X_train, y_train)
    score = random_reg.score(X_test, y_test)
```

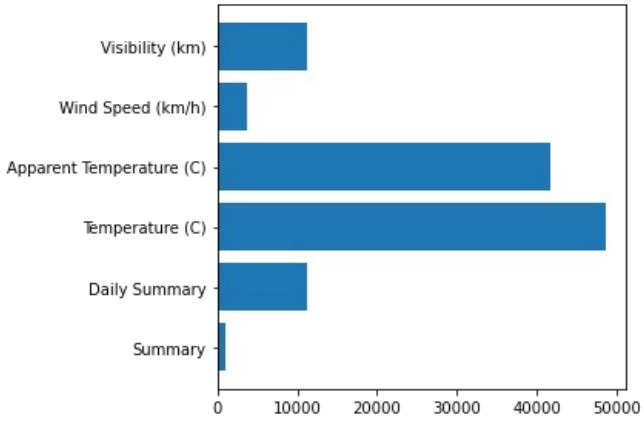
```
    y_pred = random_reg.predict(X_test)
    print(f'Training dataset shape: {X_train.shape}')
    print(f'Test dataset shape: {X_test.shape}')
    print(f'{color.BOLD}Accuracy Score {color.END} {score:.4f}')
    print(f'{color.BOLD}Mean squared error:{color.END} {mean_squared_error(y_test, y_pred):.4f}')
    print(f'{color.BOLD}Root Mean squared error:{color.END} {mean_squared_error(y_test, y_pred,squared=False):.4f
    fig, ax = plt.subplots(figsize=(10,10))
    ax.scatter(y_test, y_pred, edgecolors=(0, 0, 0))
    ax.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=4)
    ax.set_xlabel('Measured')
    ax.set_ylabel('Predicted')
    plt.show()
```

Feature Selection

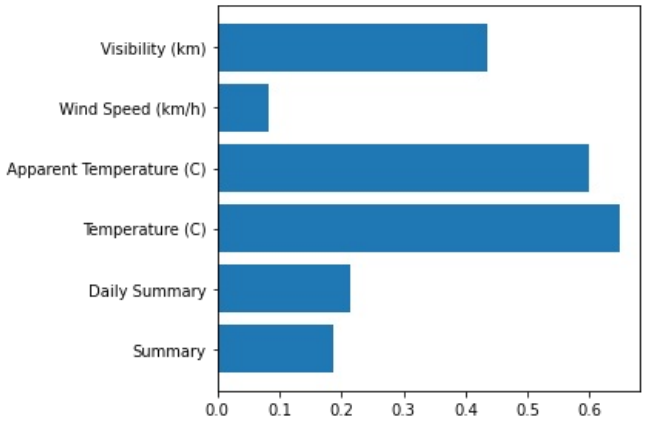
- Train Test Split
- Linear Reg with all features
- f_regression feature selection
- Linear regression with K best features
- mutual_info_regression feature selection

In [28]:
```
X_train, X_test, y_train, y_test = train_test_split(pre_processed_df,y,random_state=42)

freg_feature_count = 6
print_score = False
print(f'{color.BLUE}Select {freg_feature_count} Best features using f_regression{color.END}')
X_train_freg, X_test_freg = select_feature_freg(X_train=X_train,y_train=y_train,
                                                X_test=X_test,feature_count=freg_feature_count,
                                                print_score=print_score)
minfo_feature_count = 6
print_score = False
print(f'{color.BLUE}Select {minfo_feature_count} Best features using mutual_info_regression{color.END}')
X_train_mreg, X_test_mreg = select_feature_mutualinfo(X_train=X_train,
                                                X_test=X_test,feature_count=minfo_feature_count,
                                                print_score=print_score)
```

```
Select 6 Best features using f_regression
```



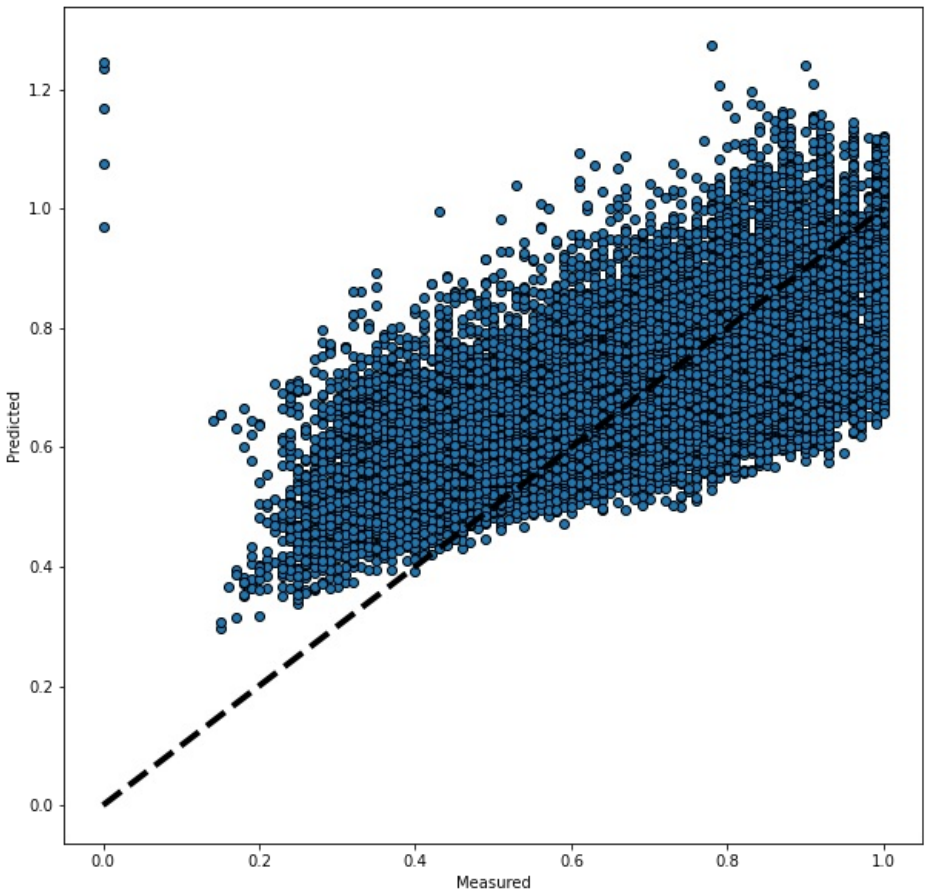

```
Select 6 Best features using mutual_info_regression
```



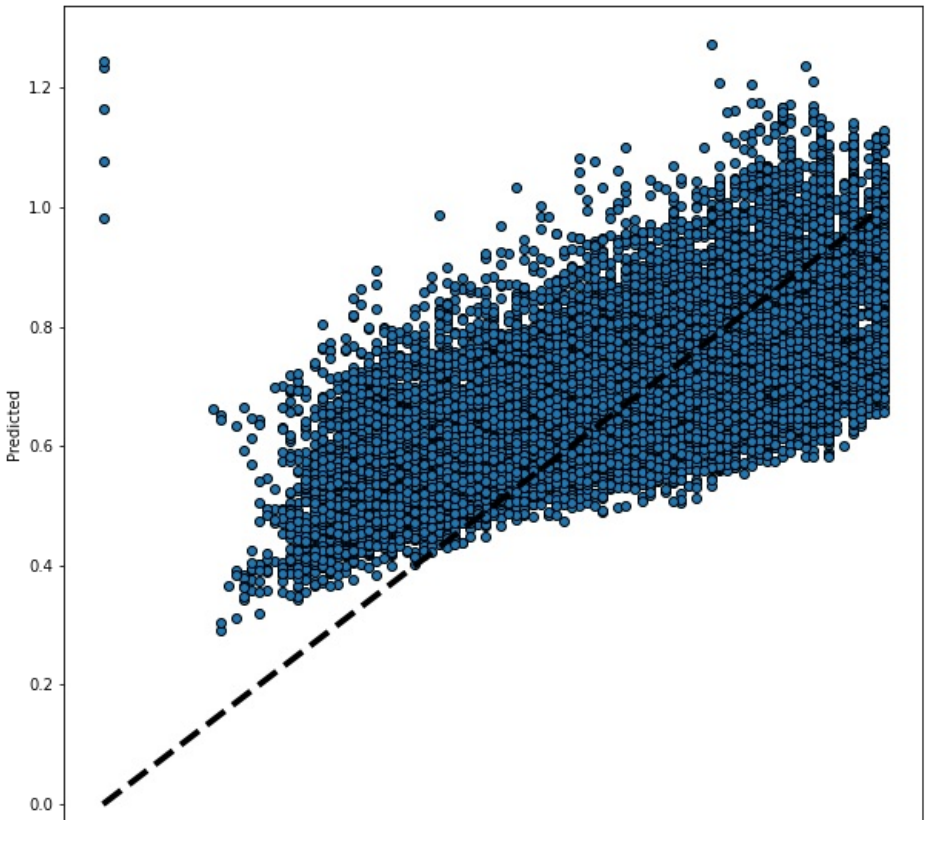
Linear Regression

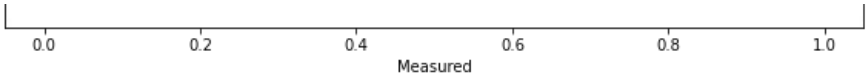
In [29]:
```
l_reg_allfeat = perform_linearreg(X_train=X_train,X_test=X_test,y_train=y_train,y_test=y_test)
l_reg_freg = perform_linearreg(X_train=X_train_freg,X_test=X_test_freg,y_train=y_train,y_test=y_test)
l_reg_minfo = perform_linearreg(X_train=X_train_mreg,X_test=X_test_mreg,y_train=y_train,y_test=y_test)
```

```
Training dataset shape: (72339, 8)
Test dataset shape: (24114, 8)
```
Accuracy Score 0.4741
Mean squared error: 0.0201
Root Mean squared error: 0.1417

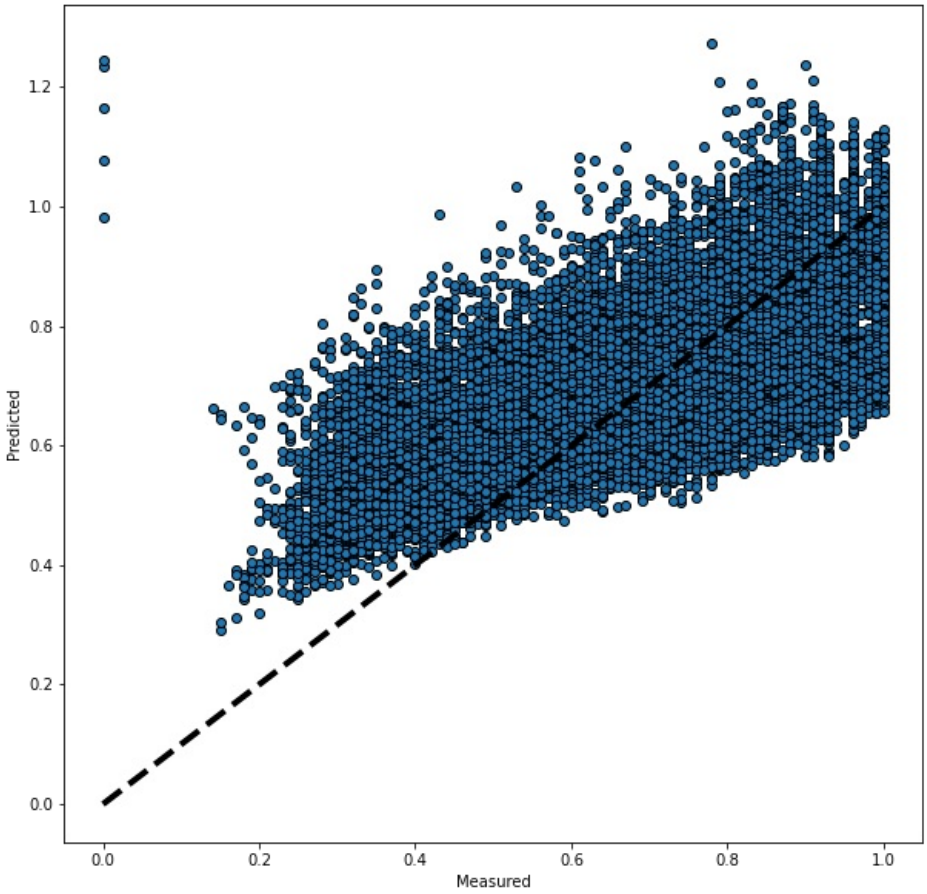



```
Training dataset shape: (72339, 6)
Test dataset shape: (24114, 6)
```
Accuracy Score 0.4727
Mean squared error: 0.0201
Root Mean squared error: 0.1419





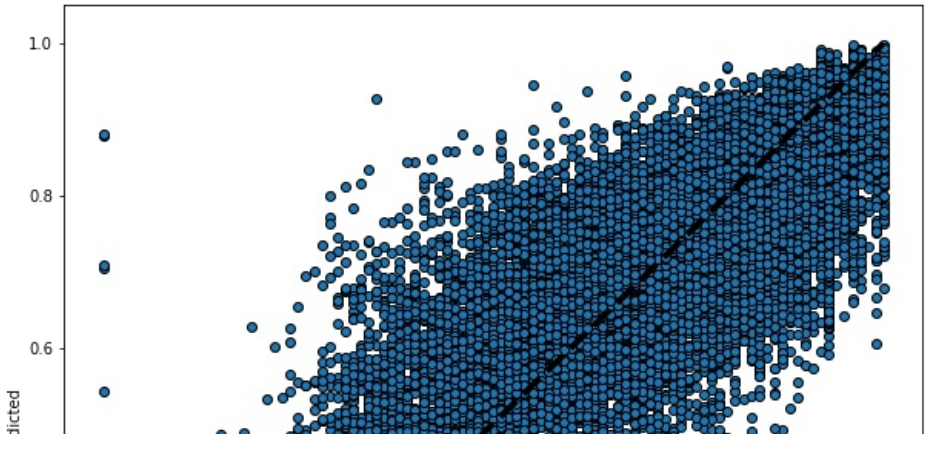

```
Training dataset shape: (72339, 6)
Test dataset shape: (24114, 6)
Accuracy Score  0.4727
Mean squared error: 0.0201
Root Mean squared error: 0.1419
```

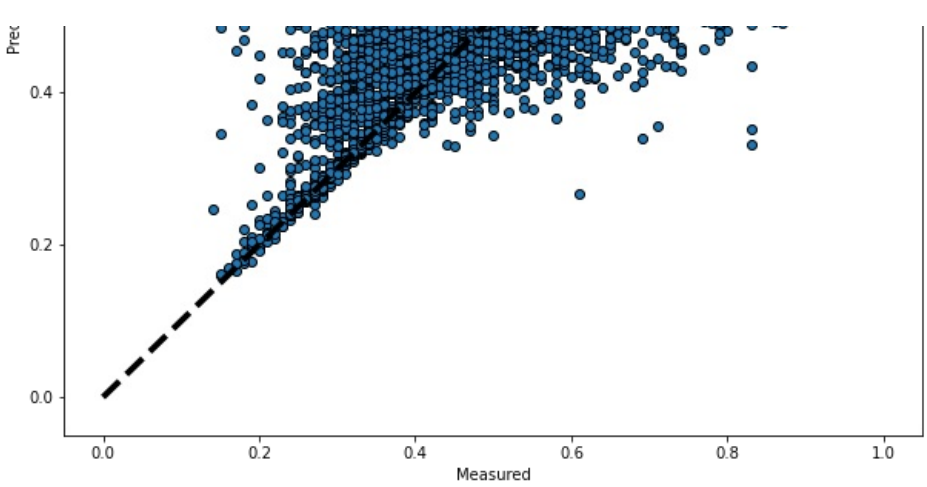


Random Forest Regression

In [30]:
```
print(f'{color.BLUE} RandomForest Regression using all features {color.END}')
random_reg_allfeat = perform_randomforestreg(X_train=X_train,X_test=X_test,
                                             y_train=y_train,y_test=y_test)
print(f'{color.BLUE} RandomForest Regression using 6 top features selected using `f_regression` {color.END}')
random_reg_freg = perform_randomforestreg(X_train=X_train_freg,X_test=X_test_freg,
                                y_train=y_train,y_test=y_test)
print(f'{color.BLUE} RandomForest Regression using 6 top features selected using `mutual_info_regression` {color.
random_reg_minfo = perform_randomforestreg(X_train=X_train_mreg,X_test=X_test_mreg,
                                 y_train=y_train,y_test=y_test)
```

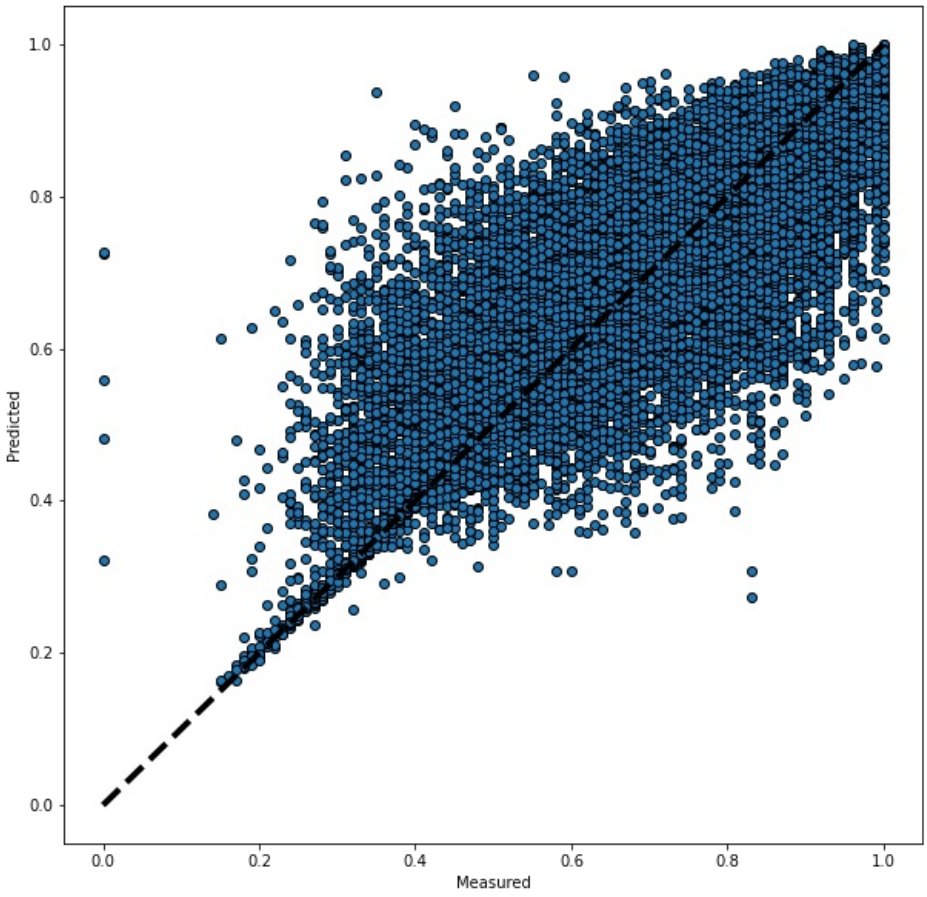
```
 RandomForest Regression using all features
Training dataset shape: (72339, 8)
Test dataset shape: (24114, 8)
Accuracy Score  0.7653
Mean squared error: 0.0090
Root Mean squared error: 0.0947
```





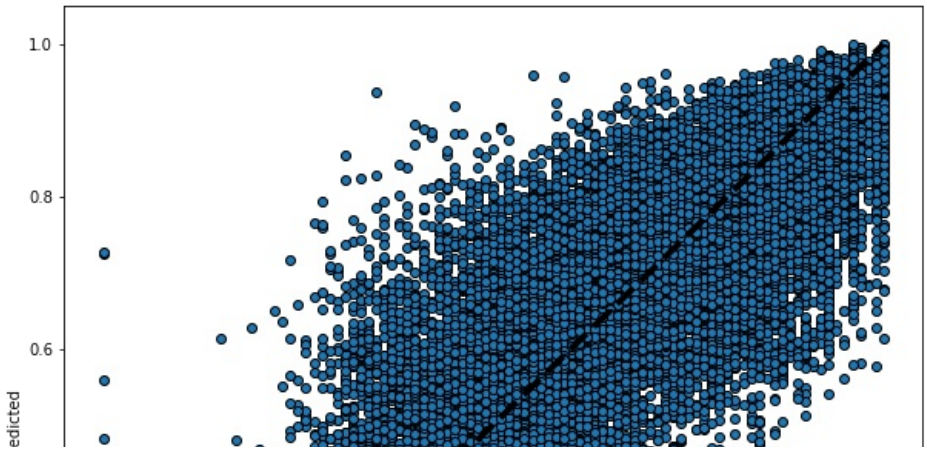
RandomForest Regression using 6 top features selected using `f_regression`

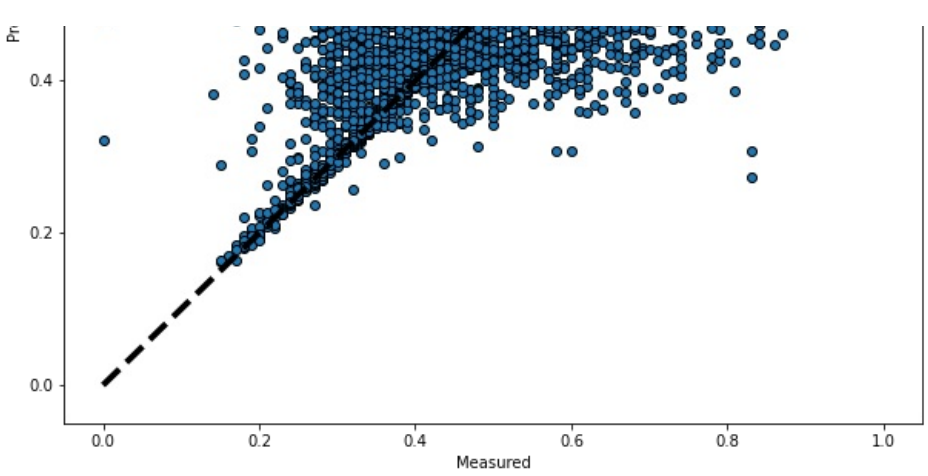
Training dataset shape: (72339, 6)
Test dataset shape: (24114, 6)
Accuracy Score 0.7183
Mean squared error: 0.0108
Root Mean squared error: 0.1037



RandomForest Regression using 6 top features selected using `mutual_info_regression`

Training dataset shape: (72339, 6)
Test dataset shape: (24114, 6)
Accuracy Score 0.7183
Mean squared error: 0.0108
Root Mean squared error: 0.1037

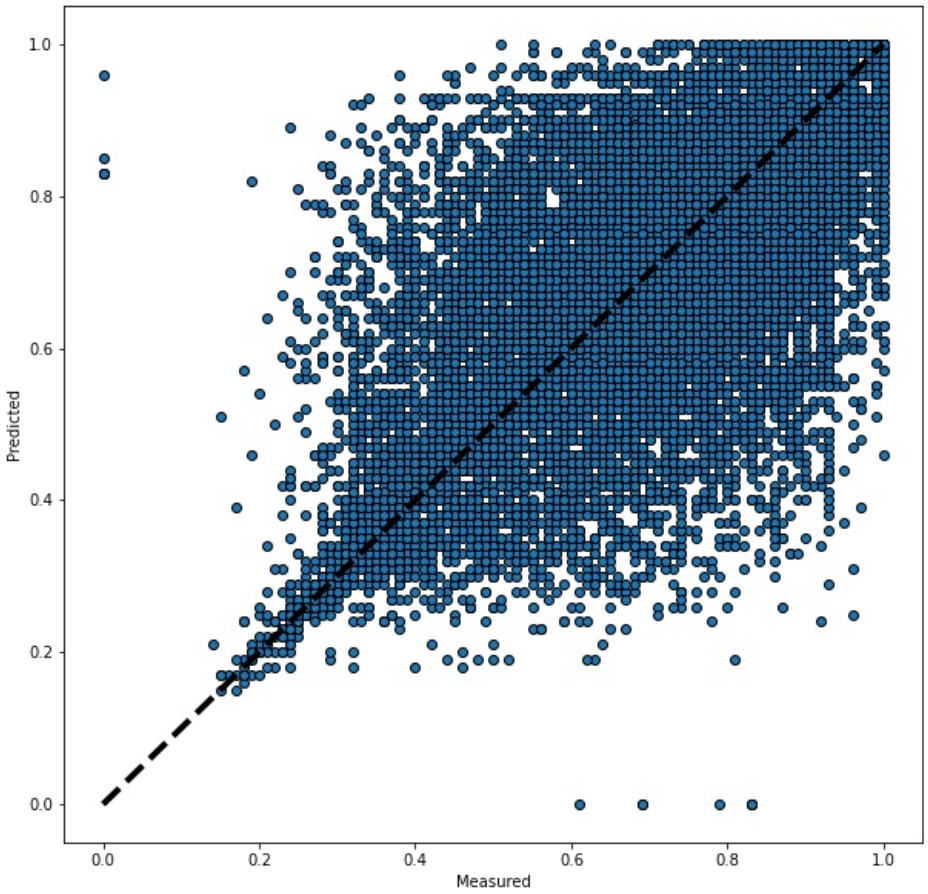




DecisionTree Regression

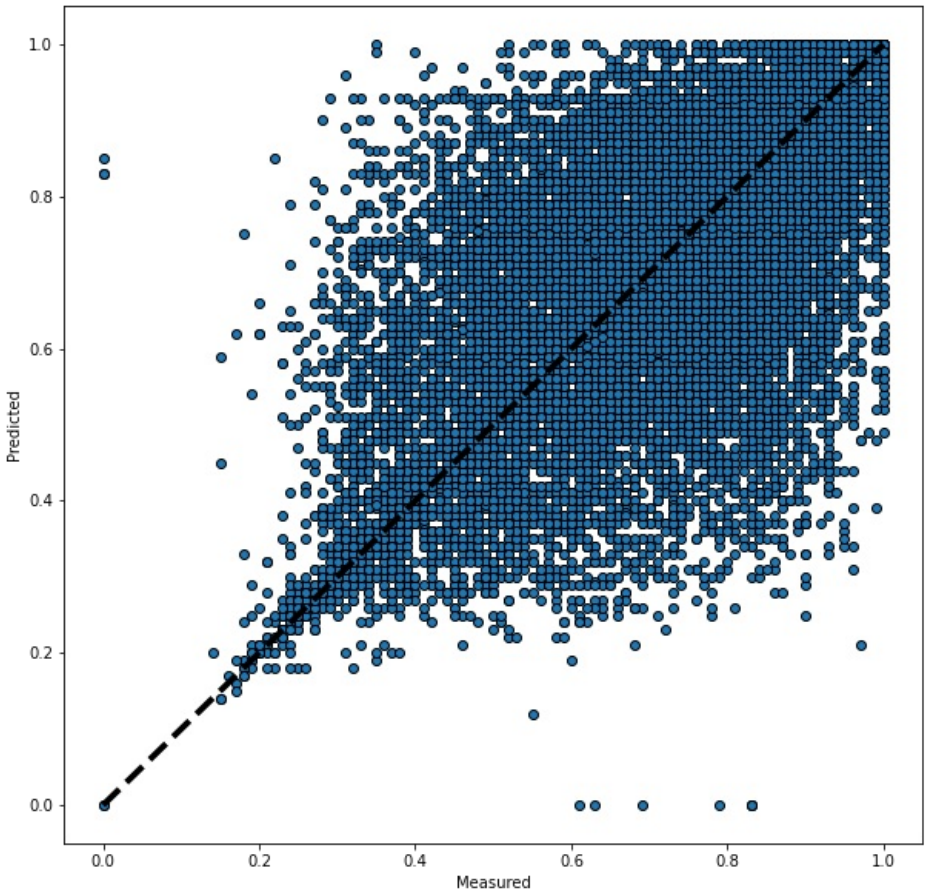
In [31]:
```
print(f'{color.BLUE} DecisionTree Regression using all features {color.END}')
dtree_reg_allfeat = perform_decisiontreereg(X_train=X_train,X_test=X_test,
                                            y_train=y_train,y_test=y_test)
print(f'{color.BLUE} DecisionTree Regression using 6 top features selected using `f_regression` {color.END}')
dtree_reg_freg = perform_decisiontreereg(X_train=X_train_freg,X_test=X_test_freg,
                                 y_train=y_train,y_test=y_test)
print(f'{color.BLUE} DecisionTree Regression using 6 top features selected using `mutual_info_regression` {color.
dtree_reg_minfo = perform_decisiontreereg(X_train=X_train_mreg,X_test=X_test_mreg,
                                  y_train=y_train,y_test=y_test)
```

DecisionTree Regression using all features
Training dataset shape: (72339, 8)
Test dataset shape: (24114, 8)
Accuracy Score 0.5287
Mean squared error: 0.0180
Root Mean squared error: 0.1341



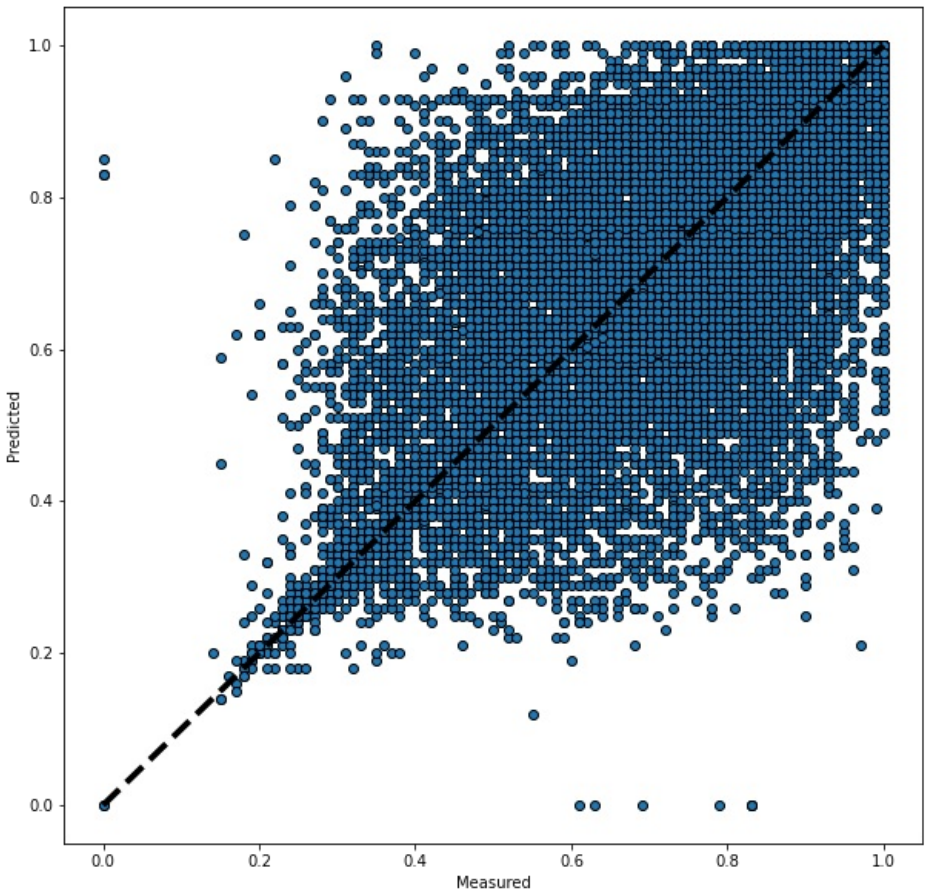
DecisionTree Regression using 6 top features selected using `f_regression`
Training dataset shape: (72339, 6)
Test dataset shape: (24114, 6)
Accuracy Score 0.4674
Mean squared error: 0.0203

Root Mean squared error: 0.1426



DecisionTree Regression using 6 top features selected using `mutual_info_regression`

Training dataset shape: (72339, 6)
Test dataset shape: (24114, 6)
Accuracy Score 0.4674
Mean squared error: 0.0203
Root Mean squared error: 0.1426



Since the target is not categorical in nature, Confusion mattrix can not be applied here.

In [ ]:

Machine Learning Assignment Problem Statement II

Group No 58

Done By Rishabh Rakesh, Arnab Ghosh, Sarbojit Roy

Prepare a jupyter notebook (recommended - Google Colab) to build, train and evaluate a Machine Learning model on the given dataset. Please read the instructions carefully.

Part A (13 marks)
Dataset - https://www.kaggle.com/muthuj7/weather-dataset
This is a dataset for a larger project and the idea is to analyze and compare real historical weather with weather folklore. Predict the humidity and display the following results MSE,RMSE

1. **Import Libraries/Dataset**
   a. Download the dataset
   b. Import the required libraries

In [59]:
```
import pandas as pd
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
import numpy as np
```

In [351…
```
data = pd.read_csv('weatherHistory.csv')
```

2. **Data Visualization and Exploration**
   a. Print at least 5 rows for sanity check to identify all the features present in the dataset and if the target matches with them.
   b. Print the description and shape of the dataset.
   c. Provide appropriate visualization to get an insight about the dataset.
   d. Try exploring the data and see what insights can be drawn from the dataset.

In [3]:
```
data.head(5)
```

Out[3]:

	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)	Daily Summary
0	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222	7.388889	0.89	14.1197	251.0	15.8263	0.0	1015.13	Partly cloudy throughout the day.
1	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355556	7.227778	0.86	14.2646	259.0	15.8263	0.0	1015.63	Partly cloudy throughout the day.
2	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377778	9.377778	0.89	3.9284	204.0	14.9569	0.0	1015.94	Partly cloudy throughout the day.
3	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288889	5.944444	0.83	14.1036	269.0	15.8263	0.0	1016.41	Partly cloudy throughout the day.
4	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755556	6.977778	0.83	11.0446	259.0	15.8263	0.0	1016.51	Partly cloudy throughout the day.

In [4]:
```
data.describe()
```

Out[4]:

	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	Visibility (km)	Loud Cover	Pressure (millibars)
count	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.000000	96453.0	96453.000000
mean	11.932678	10.855029	0.734899	10.810640	187.509232	10.347325	0.0	1003.235956
std	9.551546	10.696847	0.195473	6.913571	107.383428	4.192123	0.0	116.969906
min	-21.822222	-27.716667	0.000000	0.000000	0.000000	0.000000	0.0	0.000000

25%	4.688889	2.311111	0.600000	5.828200	116.000000	8.339800	0.0	1011.900000
50%	12.000000	12.000000	0.780000	9.965900	180.000000	10.046400	0.0	1016.450000
75%	18.838889	18.838889	0.890000	14.135800	290.000000	14.812000	0.0	1021.090000
max	39.905556	39.344444	1.000000	63.852600	359.000000	16.100000	0.0	1046.380000

In [5]:
```
data.shape
```

Out[5]:
```
(96453, 12)
```

In [350…
```
data.dtypes
```

Out[350…
```
Temperature (C)           float64
Humidity                  float64
Wind Speed (km/h)         float64
Wind Bearing (degrees)    float64
Visibility (km)           float64
Pressure (millibars)      float64
Summary                    object
Precip Type                object
dtype: object
```

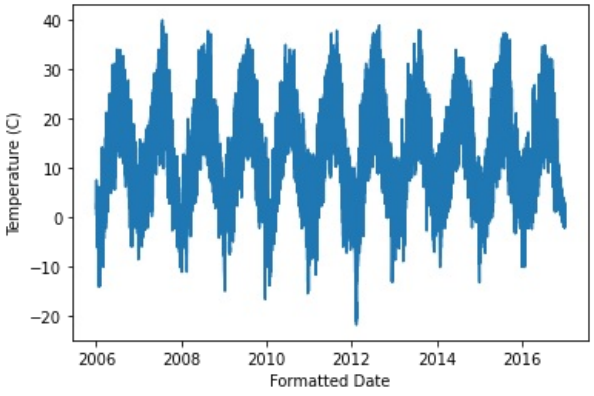
In [352…
```
data['Formatted Date'] = pd.to_datetime(data['Formatted Date'],utc=True)
```

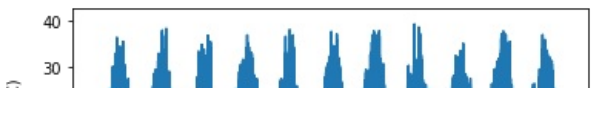
In [16]:
```
attFeatures = []
for i in data.columns:
    attFeatures.append([i, data[i].nunique(), data[i].drop_duplicates().values])
pd.DataFrame(attFeatures, columns = ['Features', 'Unique Number', 'Values'])
```

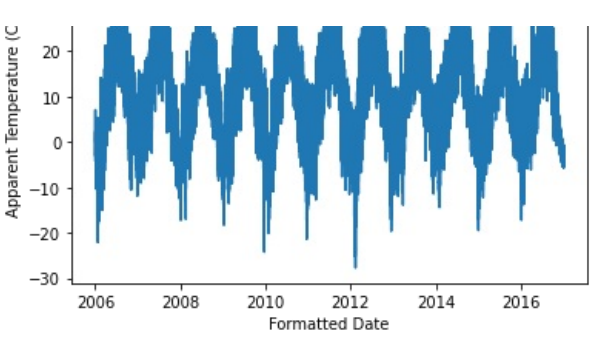
Out[16]:

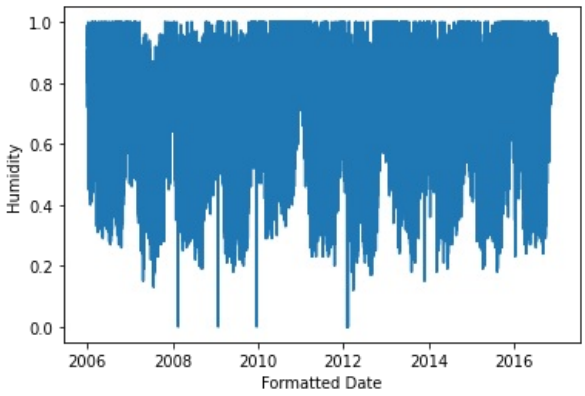
	Features	Unique Number	Values
0	Formatted Date	96429	[2006-04-01 00:00:00+02:00, 2006-04-01 01:00:0...
1	Summary	27	[Partly Cloudy, Mostly Cloudy, Overcast, Foggy...
2	Precip Type	2	[rain, snow, nan]
3	Temperature (C)	7574	[9.472222222222221, 9.355555555555558, 9.37777...
4	Apparent Temperature (C)	8984	[7.388888888888887, 7.227777777777777, 9.37777...
5	Humidity	90	[0.89, 0.86, 0.83, 0.85, 0.95, 0.82, 0.72, 0.6...
6	Wind Speed (km/h)	2484	[14.1197, 14.2646, 3.9284, 14.1036, 11.0446, 1...
7	Wind Bearing (degrees)	360	[251.0, 259.0, 204.0, 269.0, 258.0, 260.0, 279...
8	Visibility (km)	949	[15.826300000000002, 14.9569, 9.982, 11.2056, ...
9	Loud Cover	1	[0.0]
10	Pressure (millibars)	4979	[1015.13, 1015.63, 1015.94, 1016.41, 1016.51, ...
11	Daily Summary	214	[Partly cloudy throughout the day., Mostly clo...

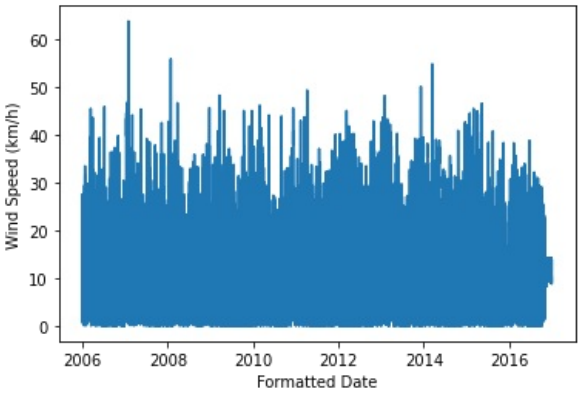
In [30]:
```
numerical_feat = [col for col in data.columns if data[col].dtypes ==float]
for col in numerical_feat:
    sns.lineplot(data=data,x='Formatted Date', y= col)
    plt.show()
```

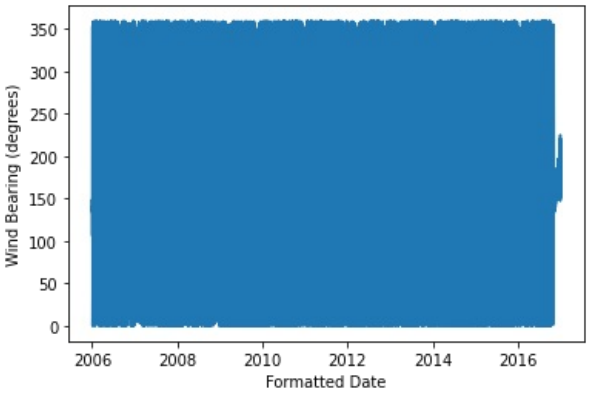


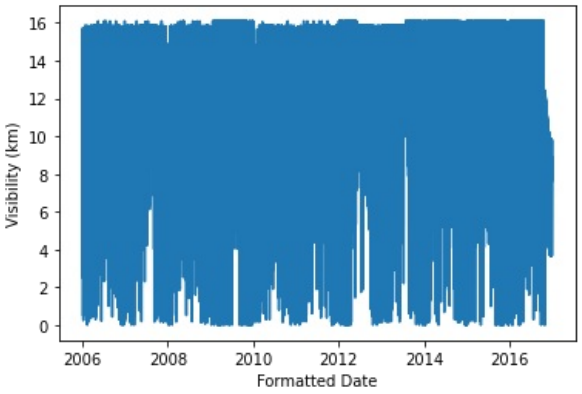



Apparent Temperature (C
20
10
0
−10
−20
−30
2006
2008
2010
2012
2014
2016
Formatted Date

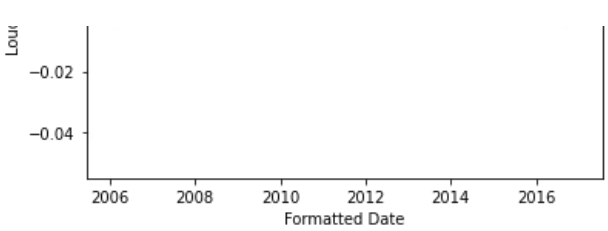

Humidity
1.0
0.8
0.6
0.4
0.2
0.0
2006
2008
2010
2012
2014
2016
Formatted Date

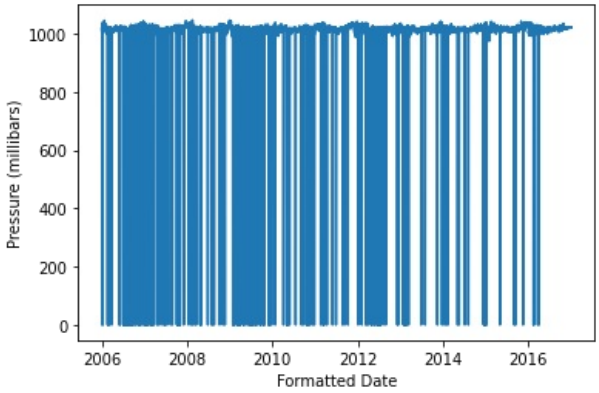

Wind Speed (km/h)
60
50
40
30
20
10
0
2006
2008
2010
2012
2014
2016
Formatted Date


Wind Bearing (degrees)
350
300
250
200
150
100
50
0
2006
2008
2010
2012
2014
2016
Formatted Date


Visibility (km)
16
14
12
10
8
6
4
2
0
2006
2008
2010
2012
2014
2016
Formatted Date


0.04
0.02
0.00





In [ ]:

```
for col in numerical_feat:
    fig =px.area(data,x='Formatted Date', y= col,)
    fig.show()
```

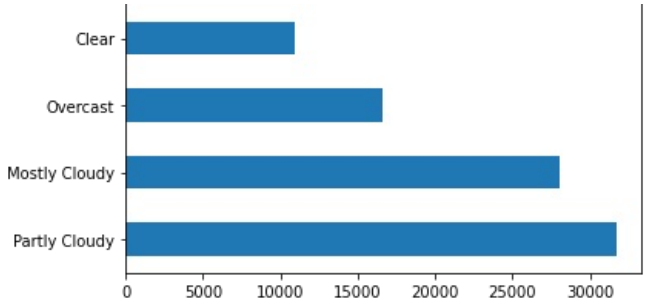
```
ERROR:root:Internal Python error in the inspect module.
Below is the traceback from this internal error.

ERROR:root:Internal Python error in the inspect module.
Below is the traceback from this internal error.

ERROR:root:Internal Python error in the inspect module.
Below is the traceback from this internal error.

Error in sys.excepthook:
Traceback (most recent call last):
  File "C:\Users\Rishabh\anaconda3\lib\traceback.py", line 481, in __init__
    cause = TracebackException(
  File "C:\Users\Rishabh\anaconda3\lib\traceback.py", line 481, in __init__
    cause = TracebackException(
  File "C:\Users\Rishabh\anaconda3\lib\traceback.py", line 481, in __init__
    cause = TracebackException(
  [Previous line repeated 996 more times]
  File "C:\Users\Rishabh\anaconda3\lib\traceback.py", line 476, in __init__
    _seen.add(id(exc_value))
RecursionError: maximum recursion depth exceeded while calling a Python object
```

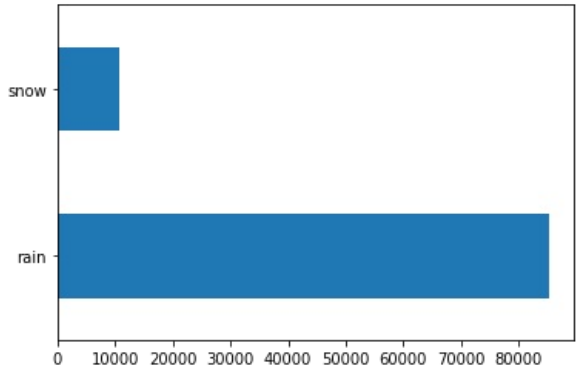
In [36]:
```
categorical_feat = [col for col in data.columns if data[col].dtype=='O' and col!='Daily Summary']
```

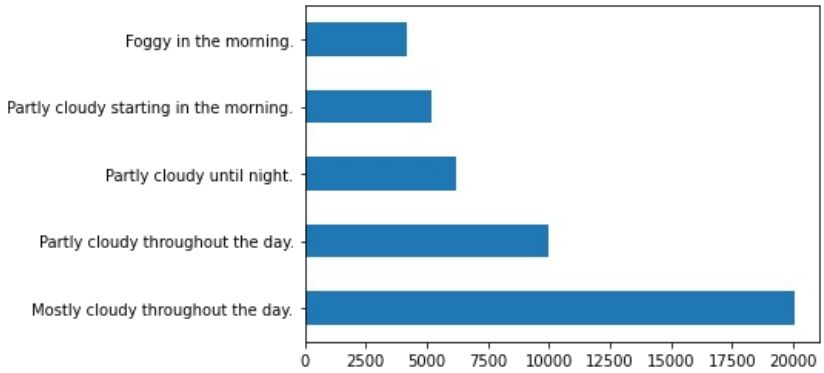
In [52]:
```
[col for col in data.columns if data[col].dtype=='O']
```

Out[52]:
```
['Formatted Date', 'Summary', 'Precip Type', 'Daily Summary']
```

In [53]:
```
for col in ['Summary', 'Precip Type', 'Daily Summary']:
    data[col].value_counts().head().plot(kind='barh')
    plt.show()
```

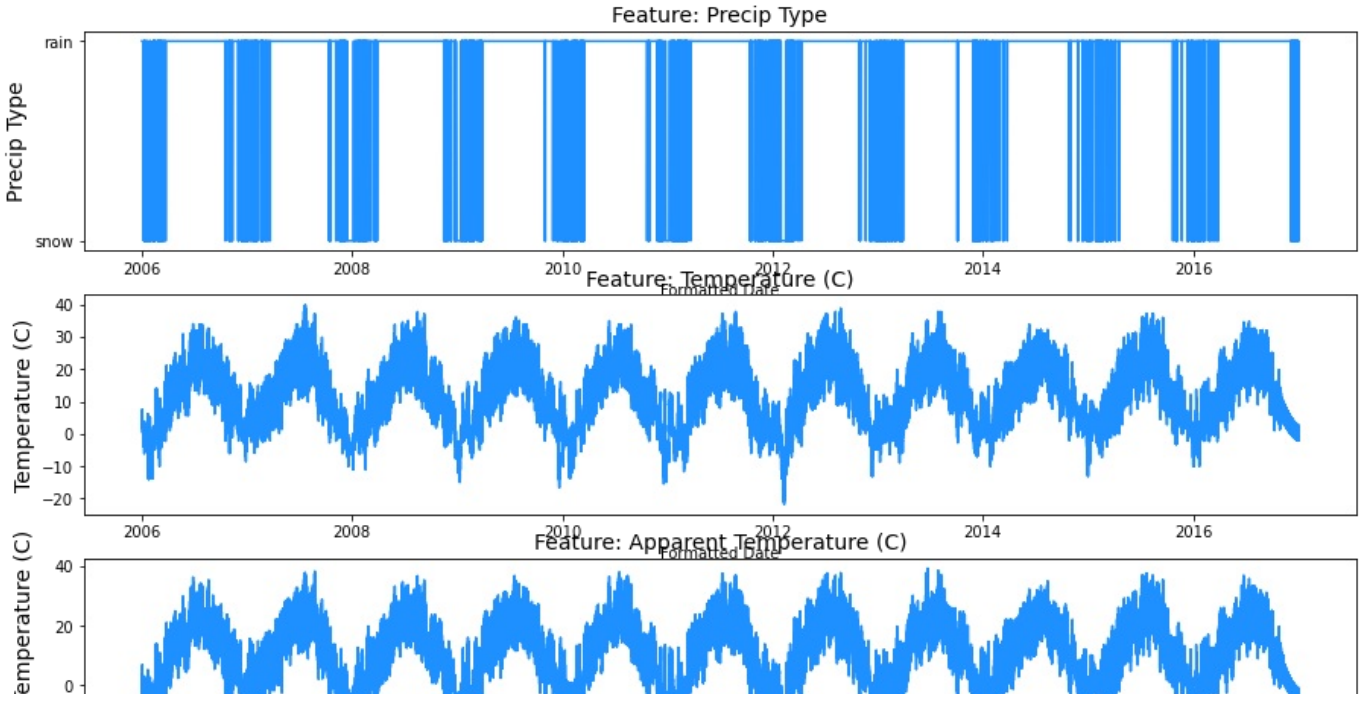


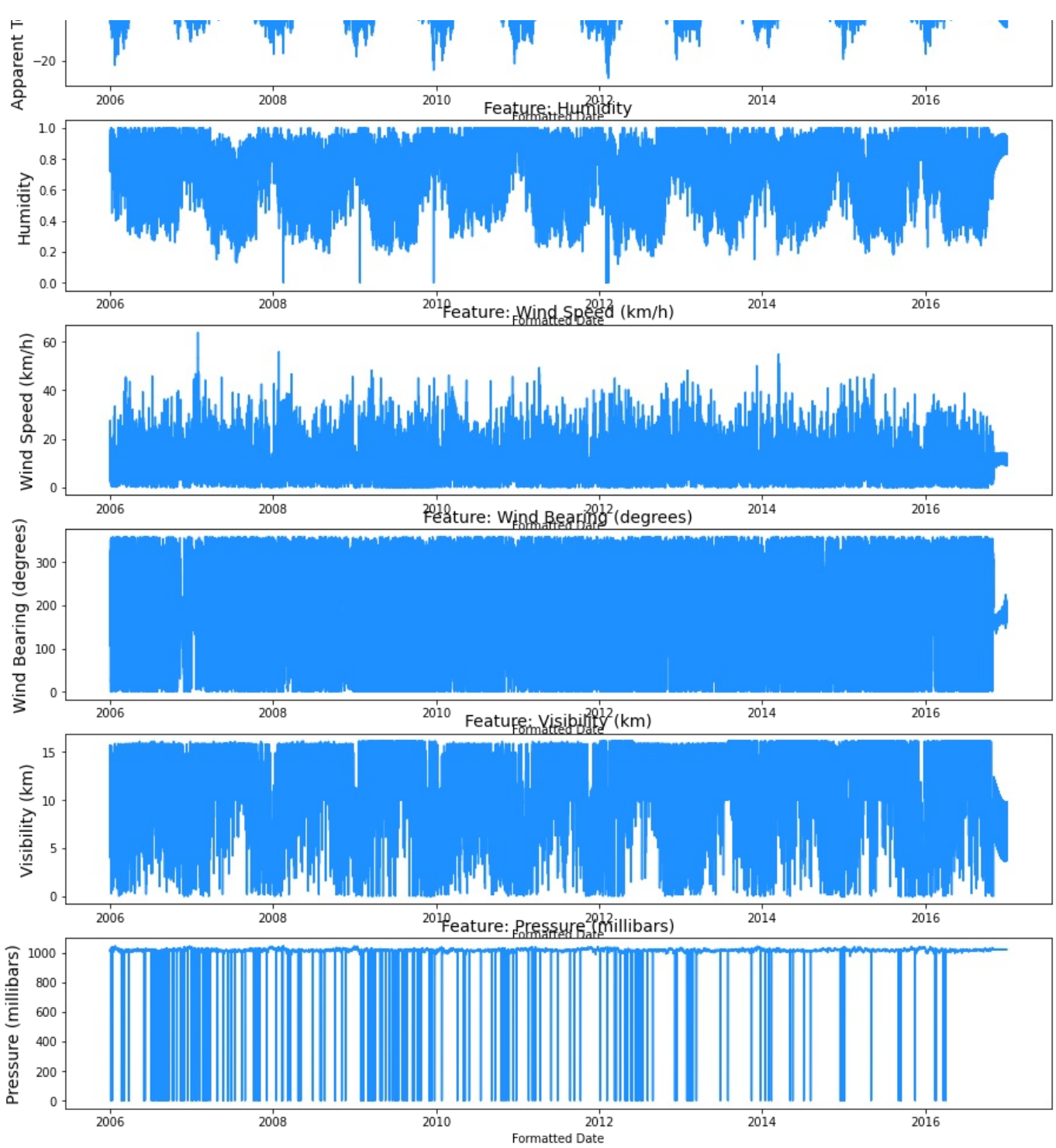




In [ ]:
```
for color in  ['Summary', 'Precip Type', 'Daily Summary']:
    for col in numerical_feat:
        fig =px.area(data,x='Formatted Date', y= col,color= color)
        fig.show()
```

In [45]:
```
f, ax = plt.subplots(nrows=8, ncols=1, figsize=(15, 25))

for i, column in enumerate(data.drop(['Formatted Date','Summary', 'Loud Cover', 'Daily Summary'], axis=1).columns
    sns.lineplot(x=data['Formatted Date'], y=data[column].fillna(method='ffill'), ax=ax[i], color='dodgerblue')
    ax[i].set_title('Feature: {}'.format(column), fontsize=14)
    ax[i].set_ylabel(ylabel=column, fontsize=14)
```



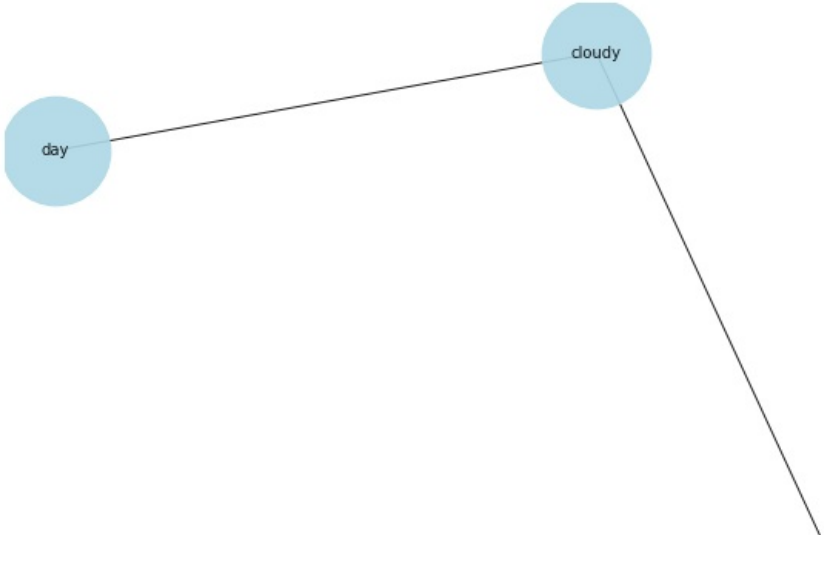


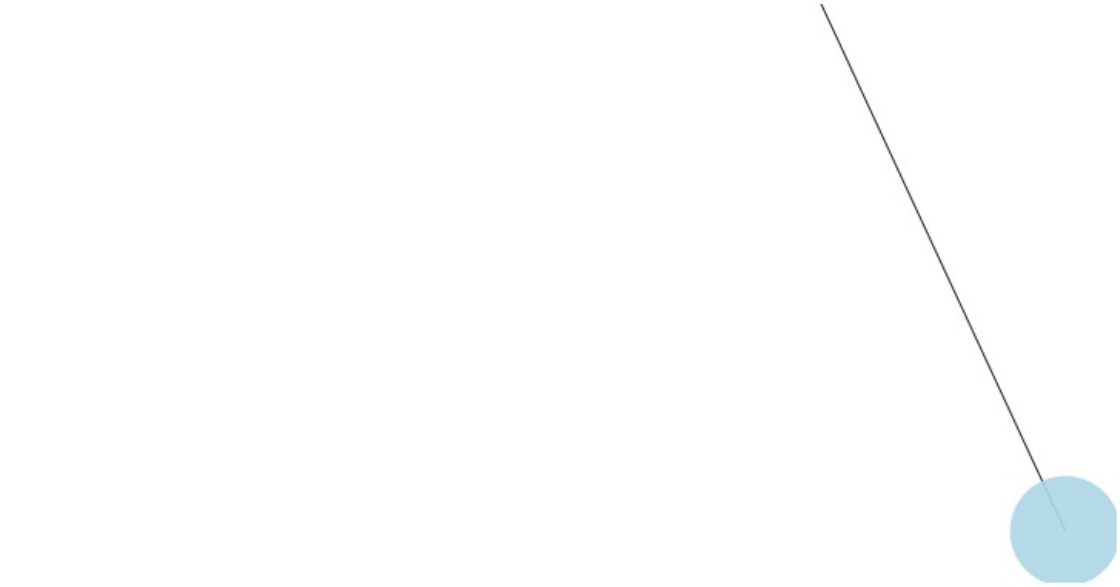
In [42]:

```
import spacy
import re
import string
from spacy.lang.en import English
import networkx as nx
import matplotlib.pyplot as plt
import nltk
def getSentences(text):
    nlp = English()
    nlp.add_pipe(nlp.create_pipe('sentencizer'))
    document = nlp(text)
    return [sent.string.strip() for sent in document.sents]

def printToken(token):
    print(token.text, "->", token.dep_)

def appendChunk(original, chunk):
    return original + ' ' + chunk
def isRelationCandidate(token):
    deps = ["ROOT", "adj", "attr", "agent", "amod"]
    return any(subs in token.dep_ for subs in deps)
def isConstructionCandidate(token):
    deps = ["compound", "prep", "conj", "mod"]
    return any(subs in token.dep_ for subs in deps)
def processSubjectObjectPairs(tokens):
    subject = ''
```

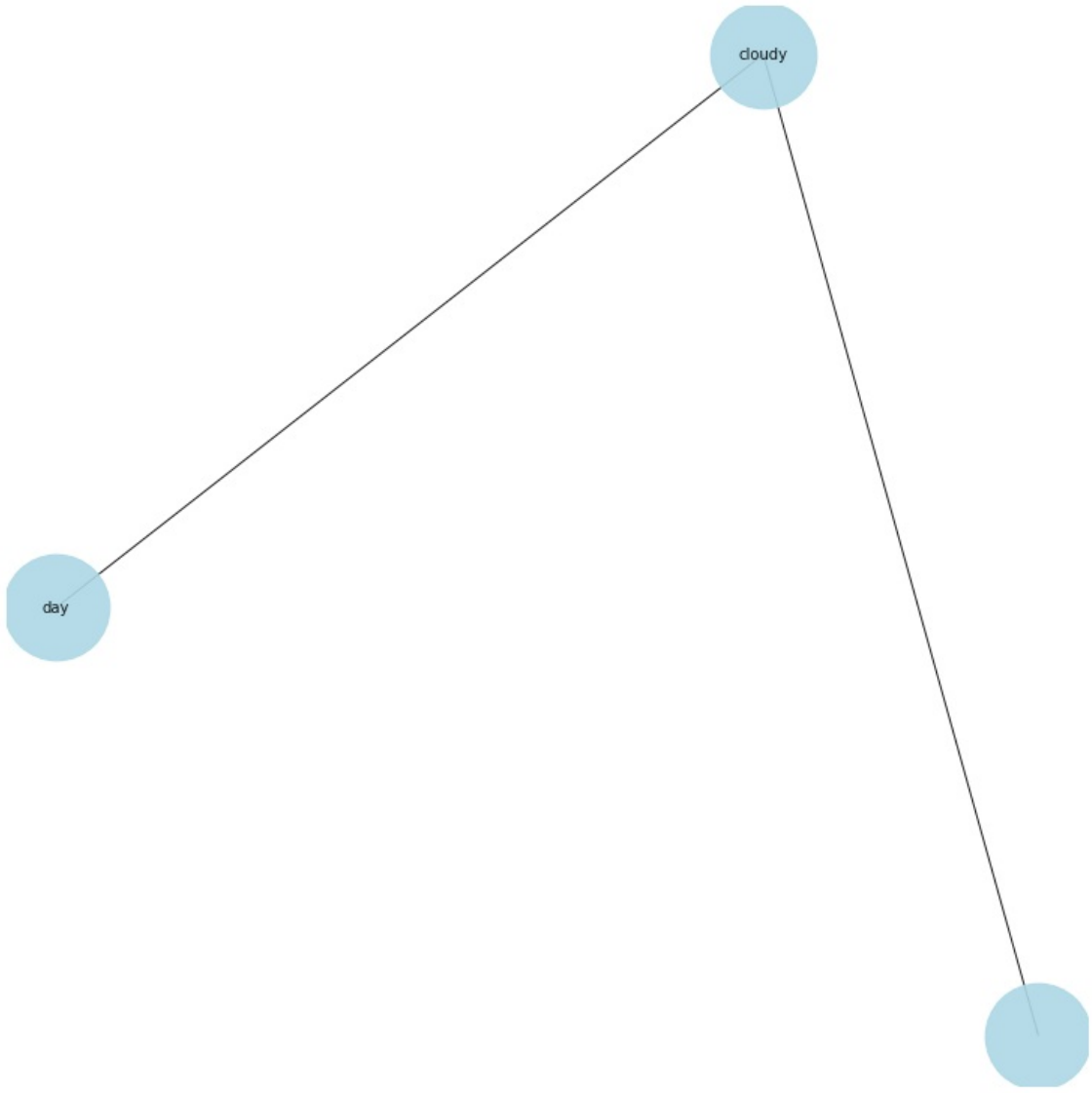
```
    object = ''
    relation = ''
    subjectConstruction = ''
    objectConstruction = ''
    for token in tokens:
        printToken(token)
        if "punct" in token.dep_:
            continue
        if isRelationCandidate(token):
            relation = appendChunk(relation, token.lemma_)
        if isConstructionCandidate(token):
            if subjectConstruction:
                subjectConstruction = appendChunk(subjectConstruction, token.text)
            if objectConstruction:
                objectConstruction = appendChunk(objectConstruction, token.text)
        if "subj" in token.dep_:
            subject = appendChunk(subject, token.text)
            subject = appendChunk(subjectConstruction, subject)
            subjectConstruction = ''
        if "obj" in token.dep_:
            object = appendChunk(object, token.text)
            object = appendChunk(objectConstruction, object)
            objectConstruction = ''

    print (subject.strip(), ",", relation.strip(), ",", object.strip())
    return (subject.strip(), relation.strip(), object.strip())


def processSentence(sentence):
    tokens = nlp_model(sentence)
    return processSubjectObjectPairs(tokens)

def printGraph(triples):
    G = nx.Graph()
    for triple in triples:
        G.add_node(triple[0])
        G.add_node(triple[1])
        G.add_node(triple[2])
        G.add_edge(triple[0], triple[1])
        G.add_edge(triple[1], triple[2])

    pos = nx.spring_layout(G,k=2, iterations=50)
    plt.figure(figsize=(10, 10))
    nx.draw(G, pos, edge_color='black', width=1, linewidths=1,
            node_size=5000, node_color='lightblue', alpha=0.9,font_size=10,
            labels={node: node for node in G.nodes()})
    plt.axis('off')
    plt.show()
for text in  data['Daily Summary'].value_counts().index.tolist():
    sentences = getSentences(text)
    nlp_model = spacy.load('en_core_web_sm')

    triples = []
    print (text)
    for sentence in sentences:
        triples.append(processSentence(sentence))
    printGraph(triples)
```

```
Mostly cloudy throughout the day.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
. -> punct
 , cloudy , day
```

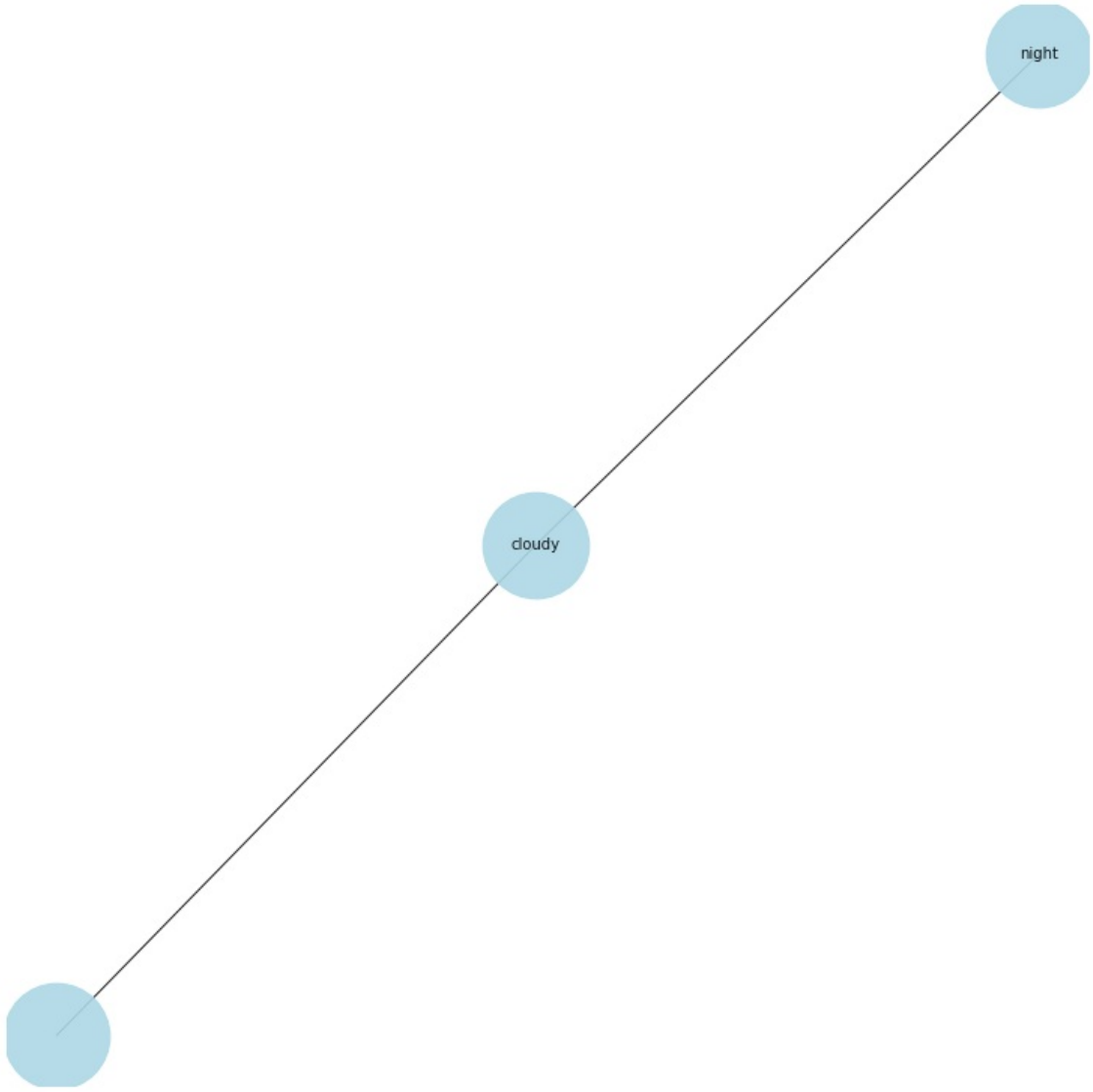




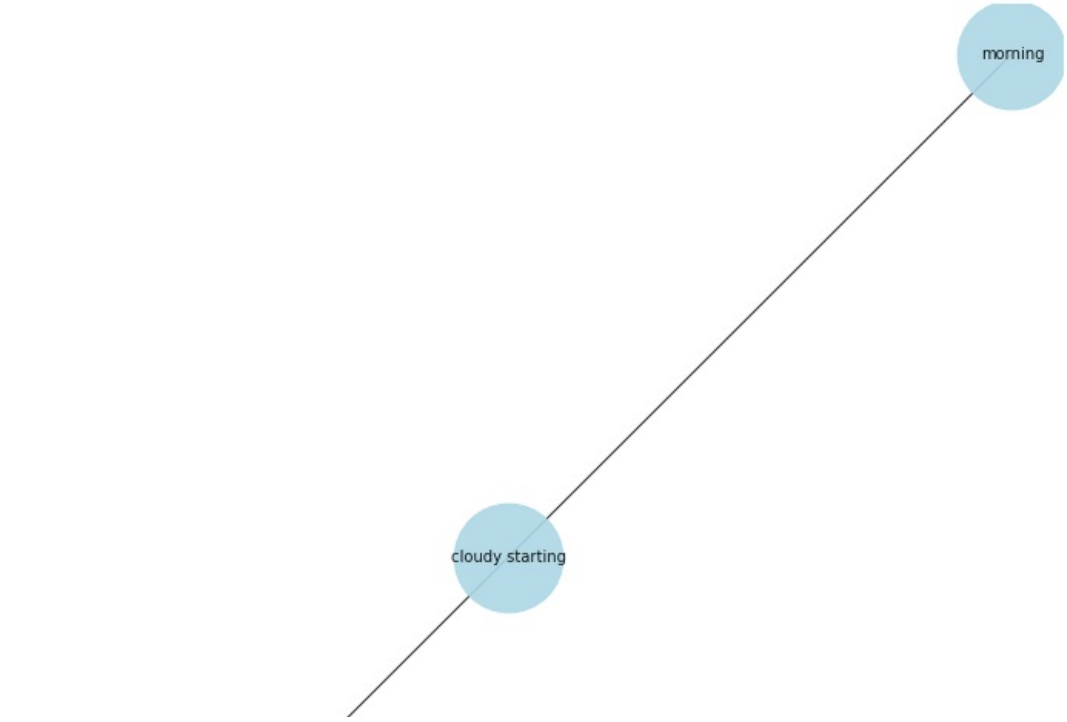
```
Partly cloudy throughout the day.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
. -> punct
 , cloudy , day
```

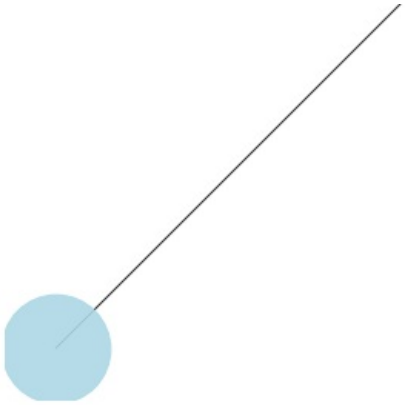



```
Partly cloudy until night.
Partly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
. -> punct
 , cloudy , night
```

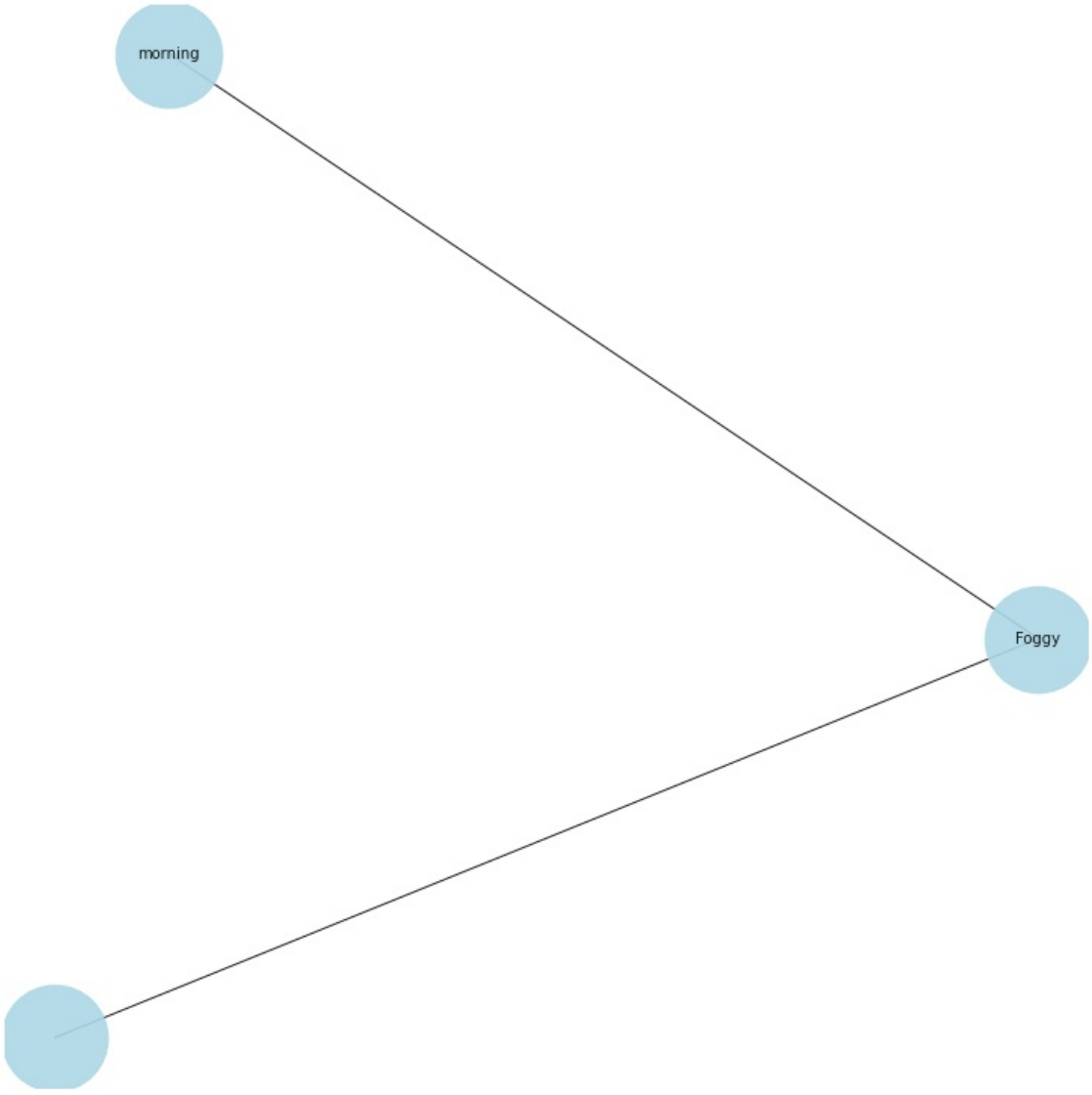



```
Partly cloudy starting in the morning.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
. -> punct
 , cloudy starting , morning
```

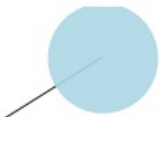


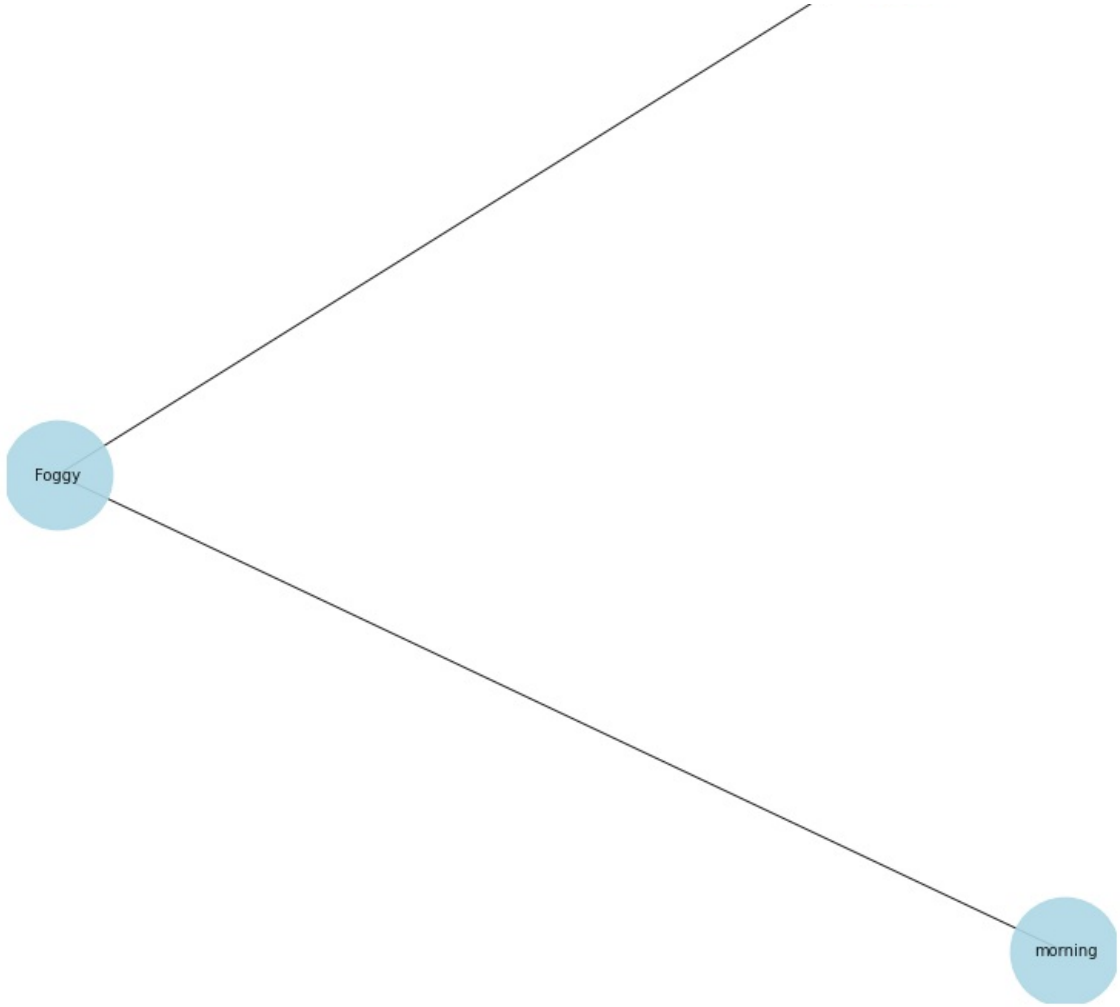


```
Foggy in the morning.
Foggy -> ROOT
in -> prep
the -> det
morning -> pobj
. -> punct
 , Foggy , morning
```

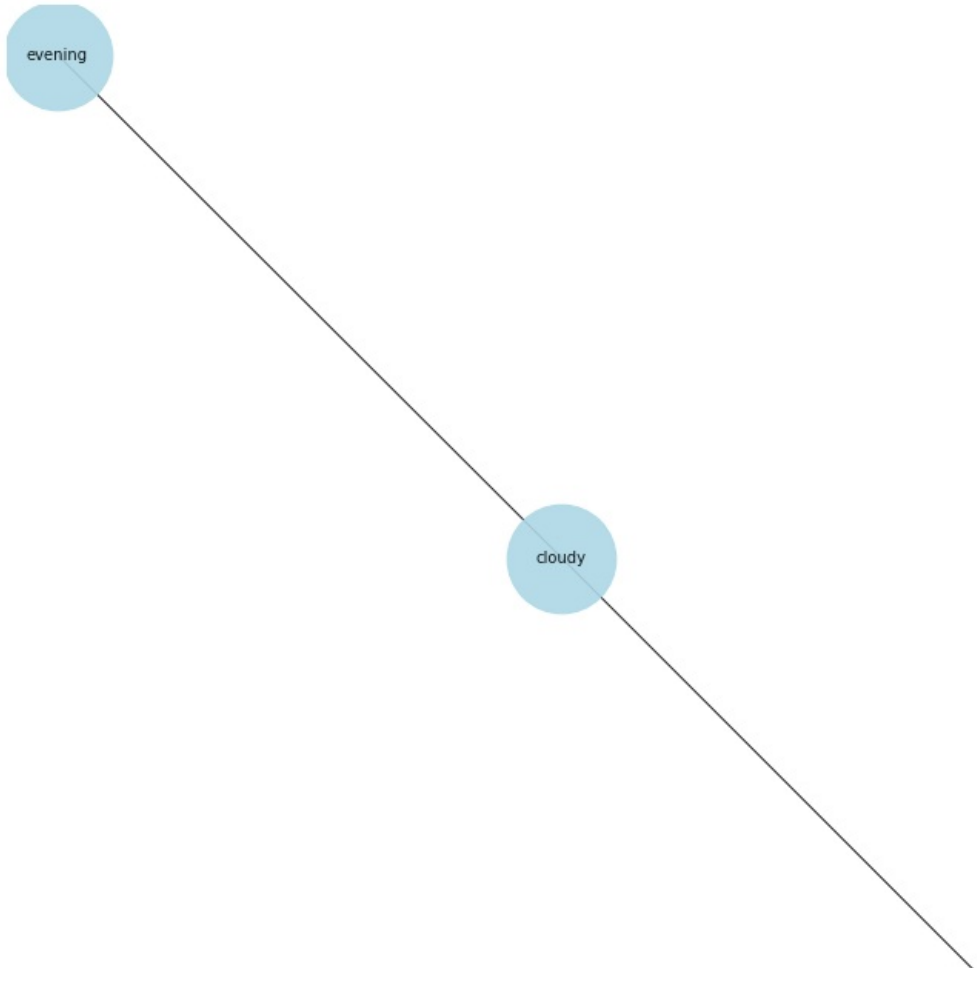



```
Foggy starting overnight continuing until morning.
Foggy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
. -> punct
 , Foggy , morning
```

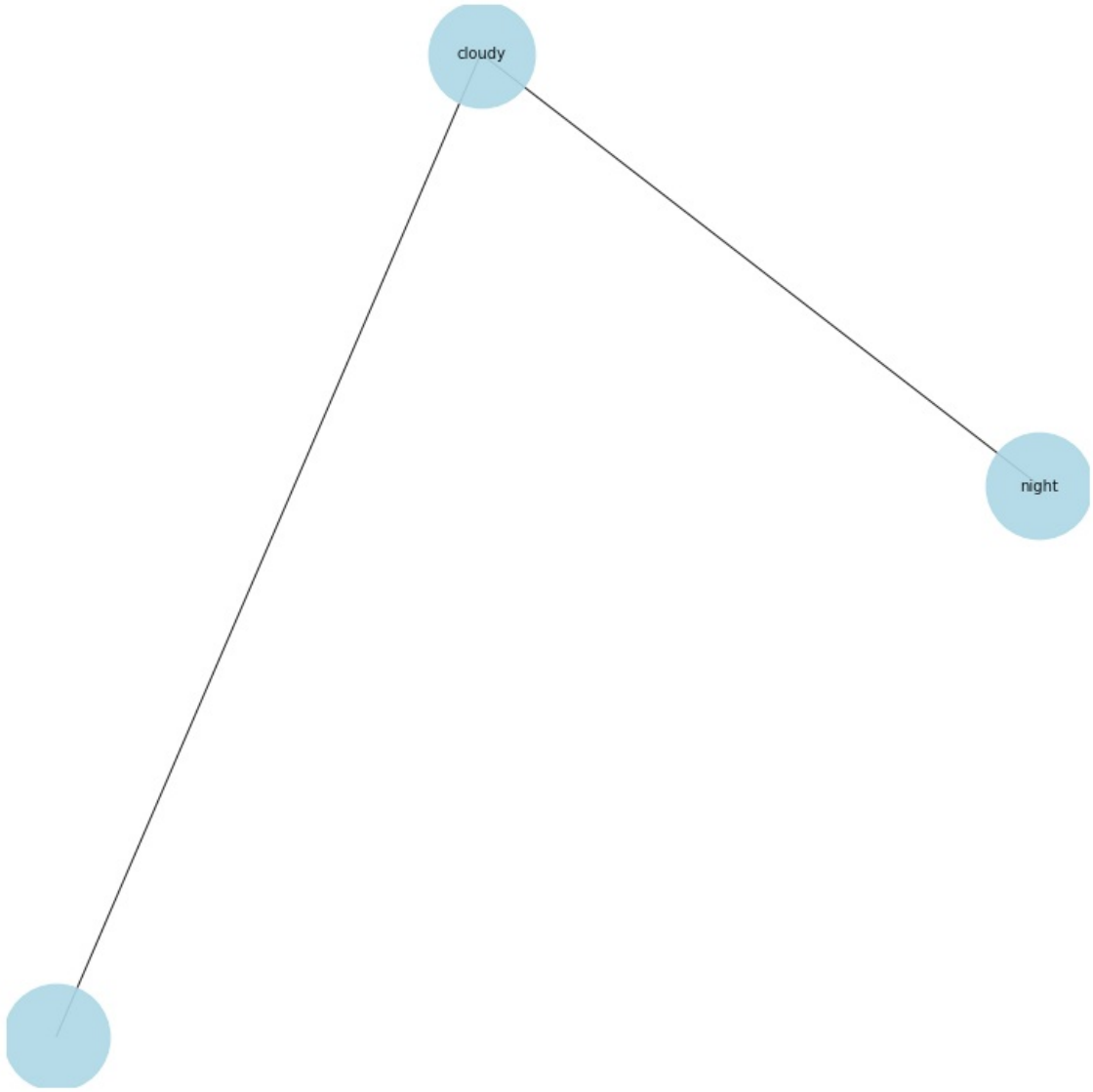




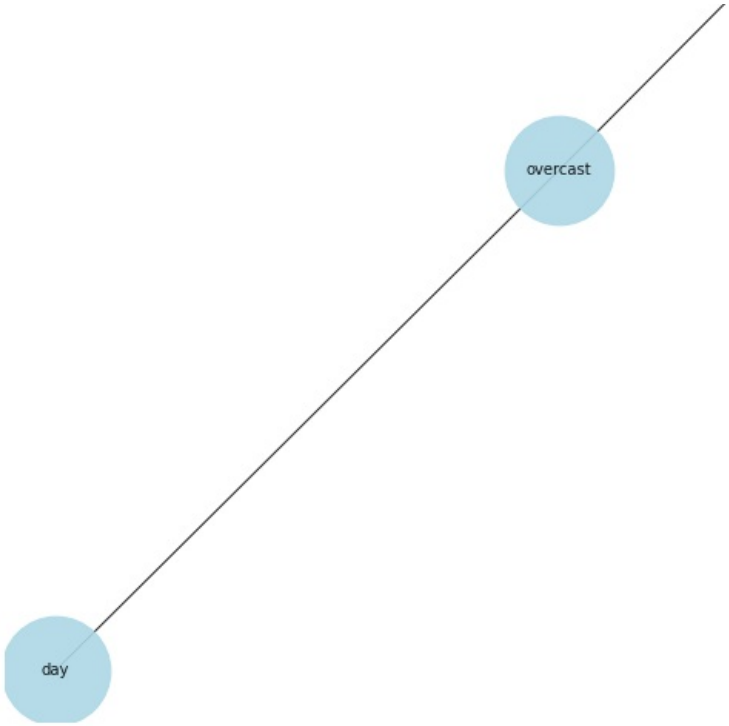

```
Partly cloudy until evening.
Partly -> advmod
cloudy -> ROOT
until -> prep
evening -> pobj
. -> punct
 , cloudy , evening
```



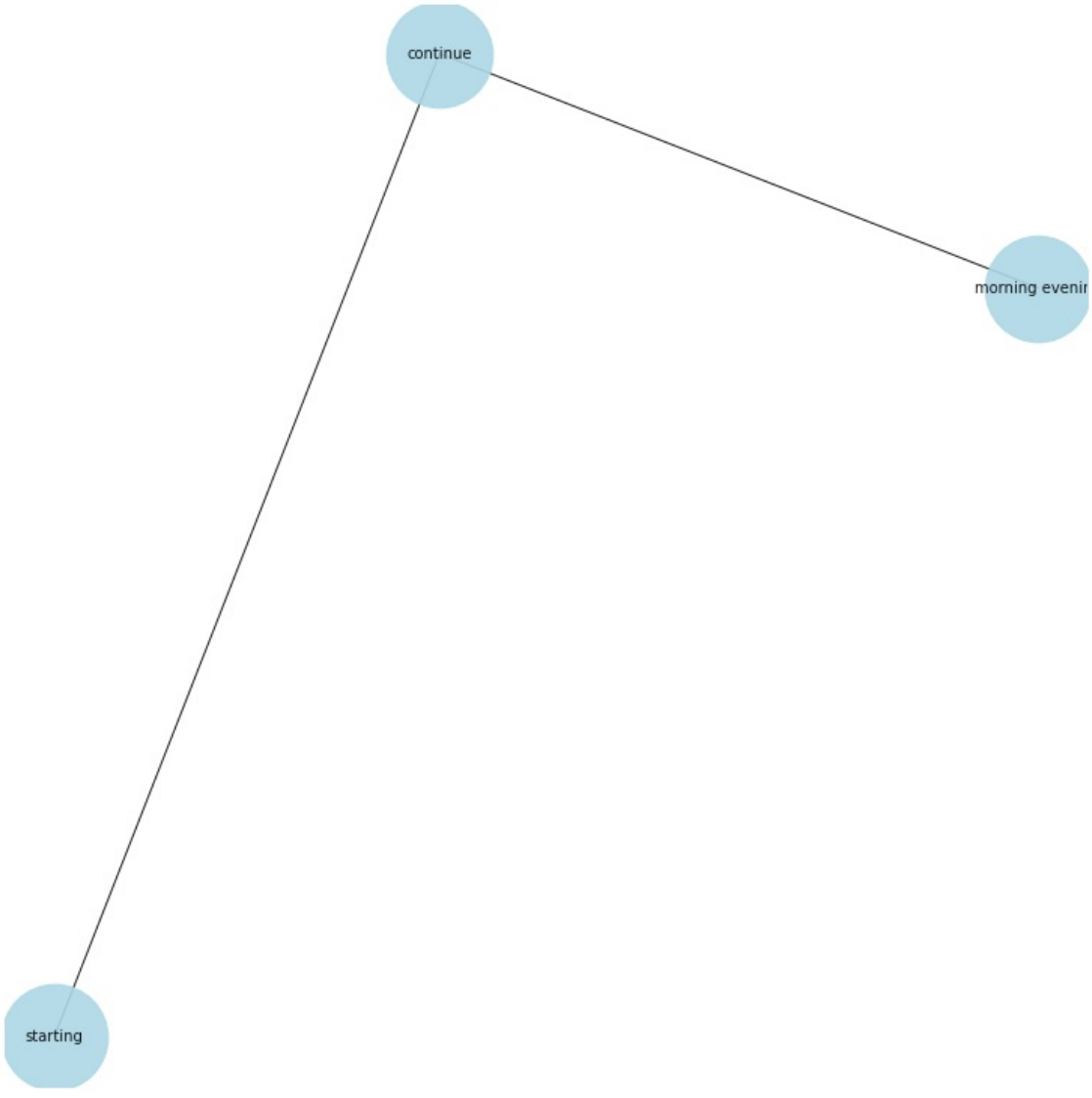
```
Mostly cloudy until night.
Mostly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
. -> punct
 , cloudy , night
```



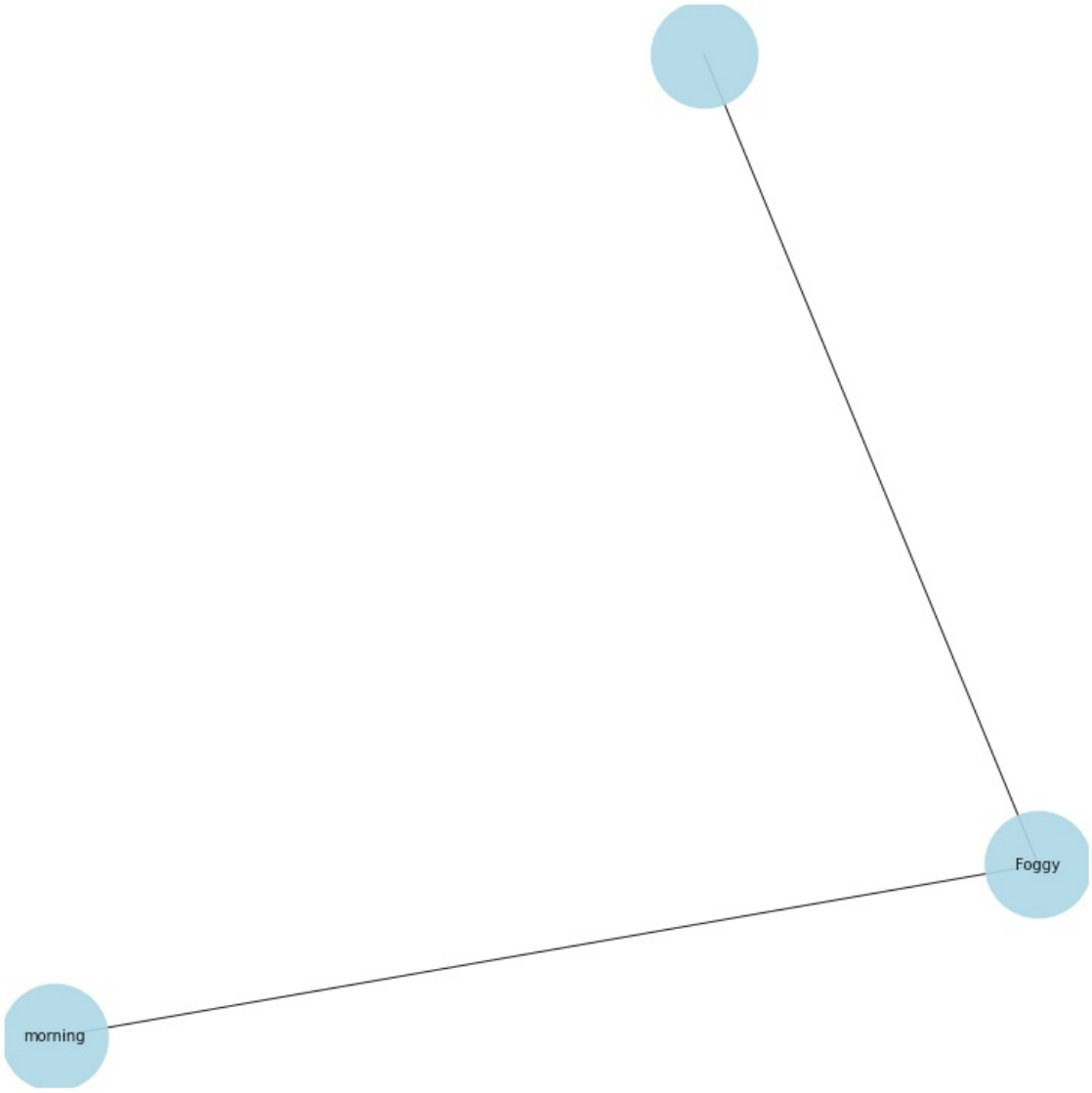

```
Overcast throughout the day.
Overcast -> ROOT
throughout -> prep
the -> det
day -> pobj
. -> punct
 , overcast , day
```



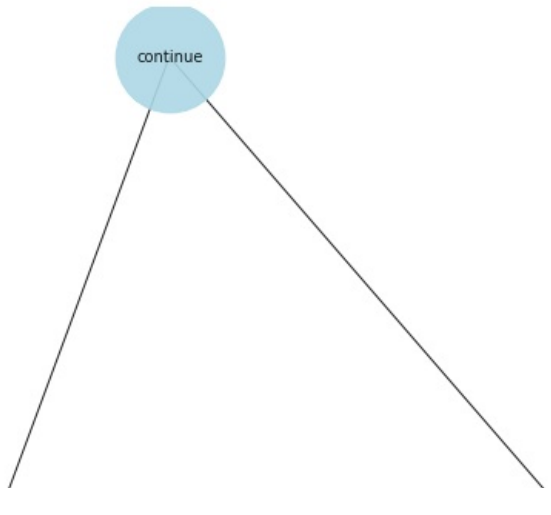

```
Partly cloudy starting in the morning continuing until evening.
Partly -> advmod
cloudy -> advmod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
starting , continue , morning evening
```

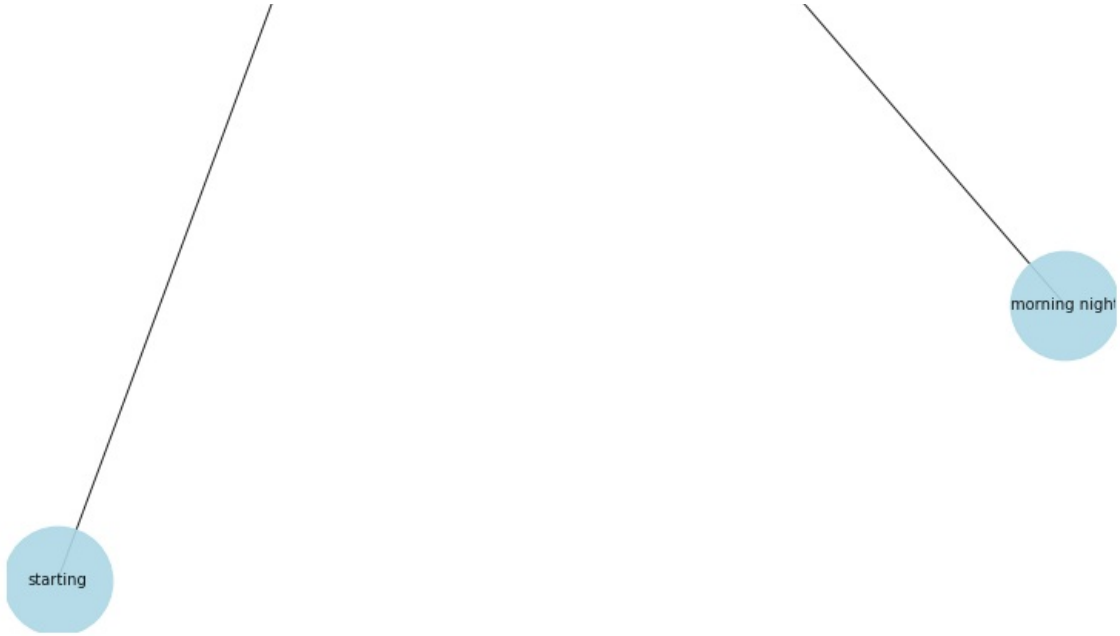


```
Foggy until morning.
Foggy -> ROOT
until -> prep
morning -> pobj
. -> punct
 , Foggy , morning
```

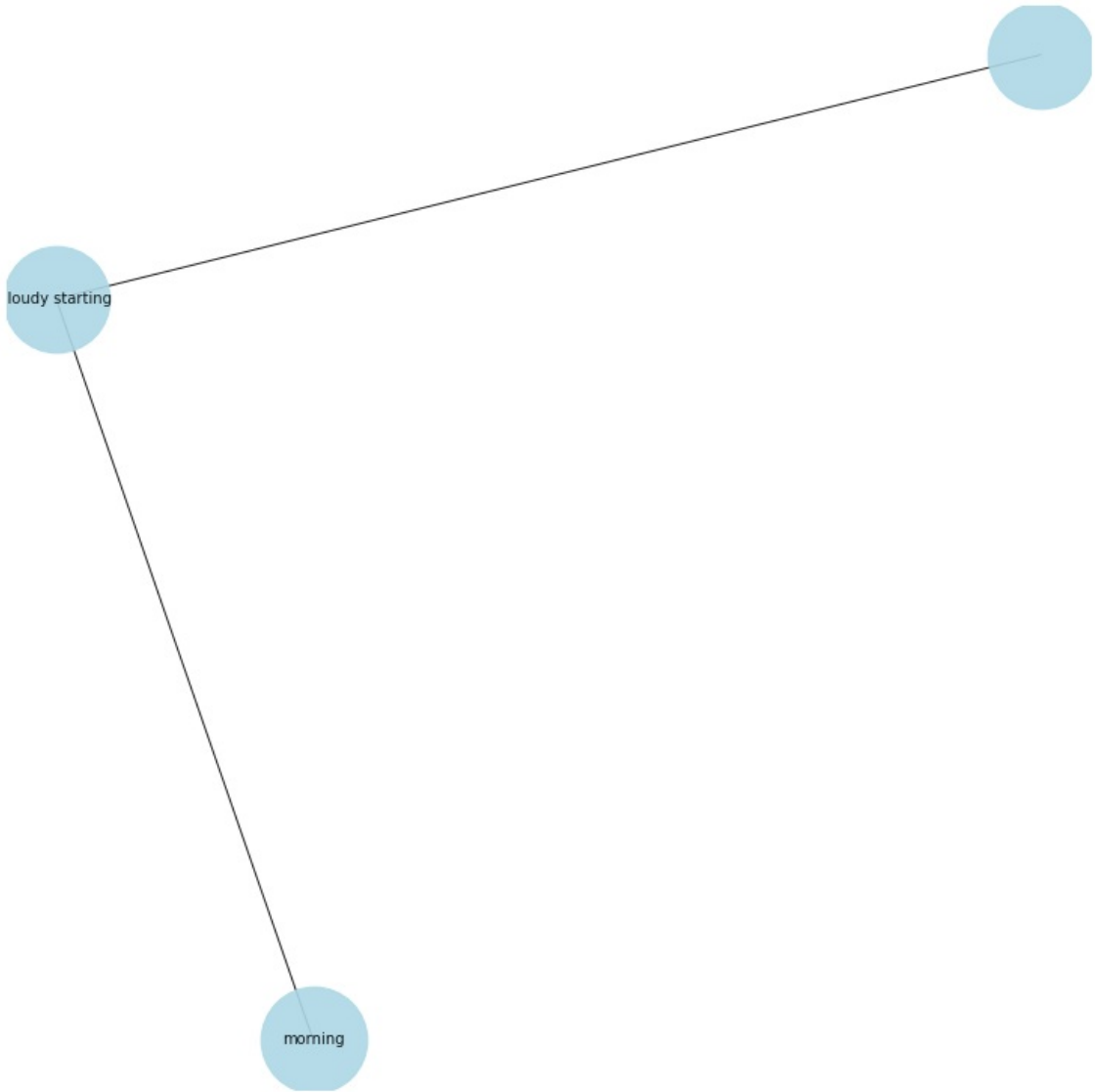



```
Partly cloudy starting in the morning continuing until night.
Partly -> advmod
cloudy -> advmod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
starting , continue , morning night
```



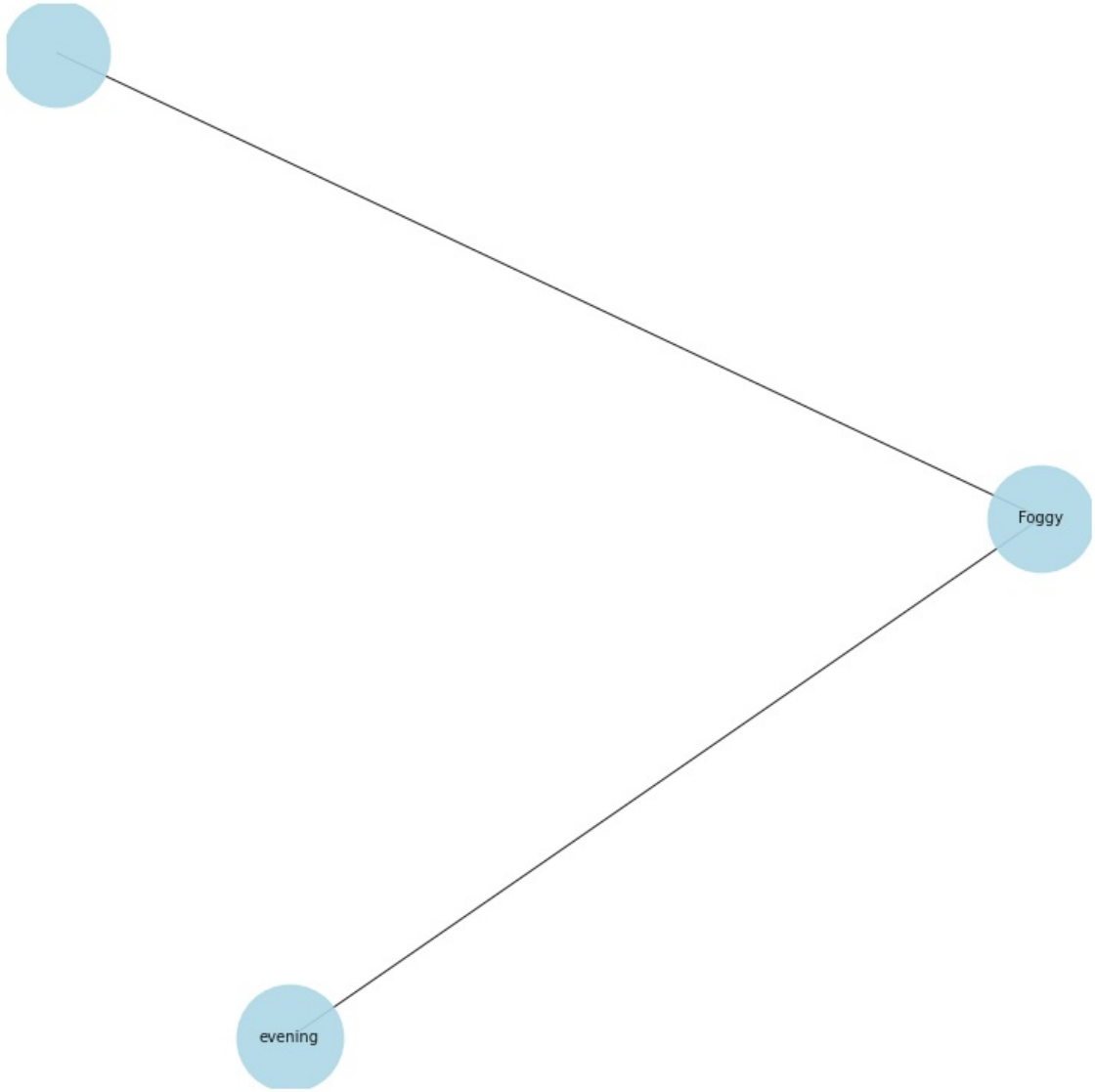



```
Mostly cloudy starting in the morning.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
. -> punct
 , cloudy starting , morning
```

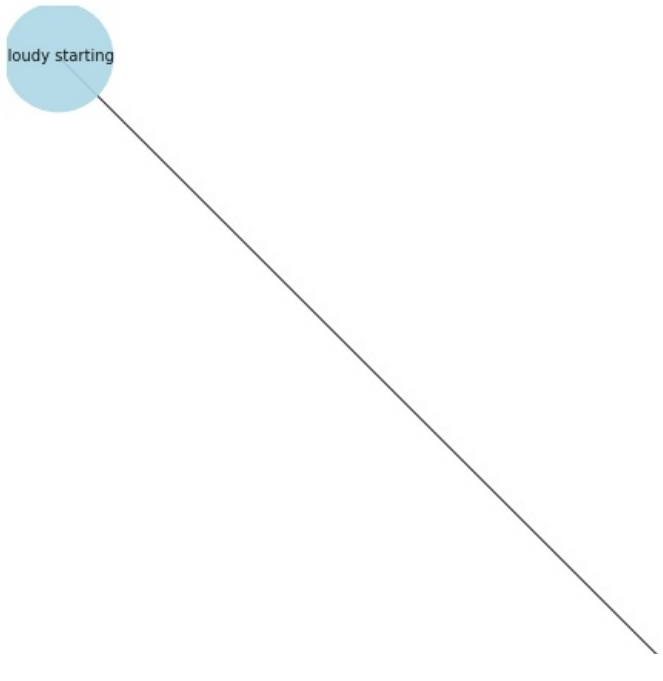


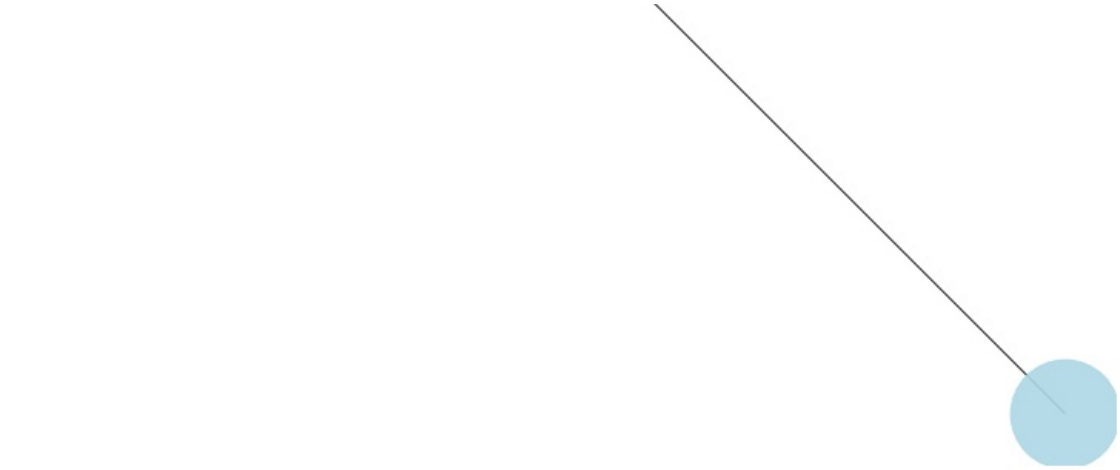

```
Foggy starting in the evening.
Foggy -> ROOT
starting -> acl
```

```
in -> prep
the -> det
evening -> pobj
. -> punct
 , Foggy , evening
```

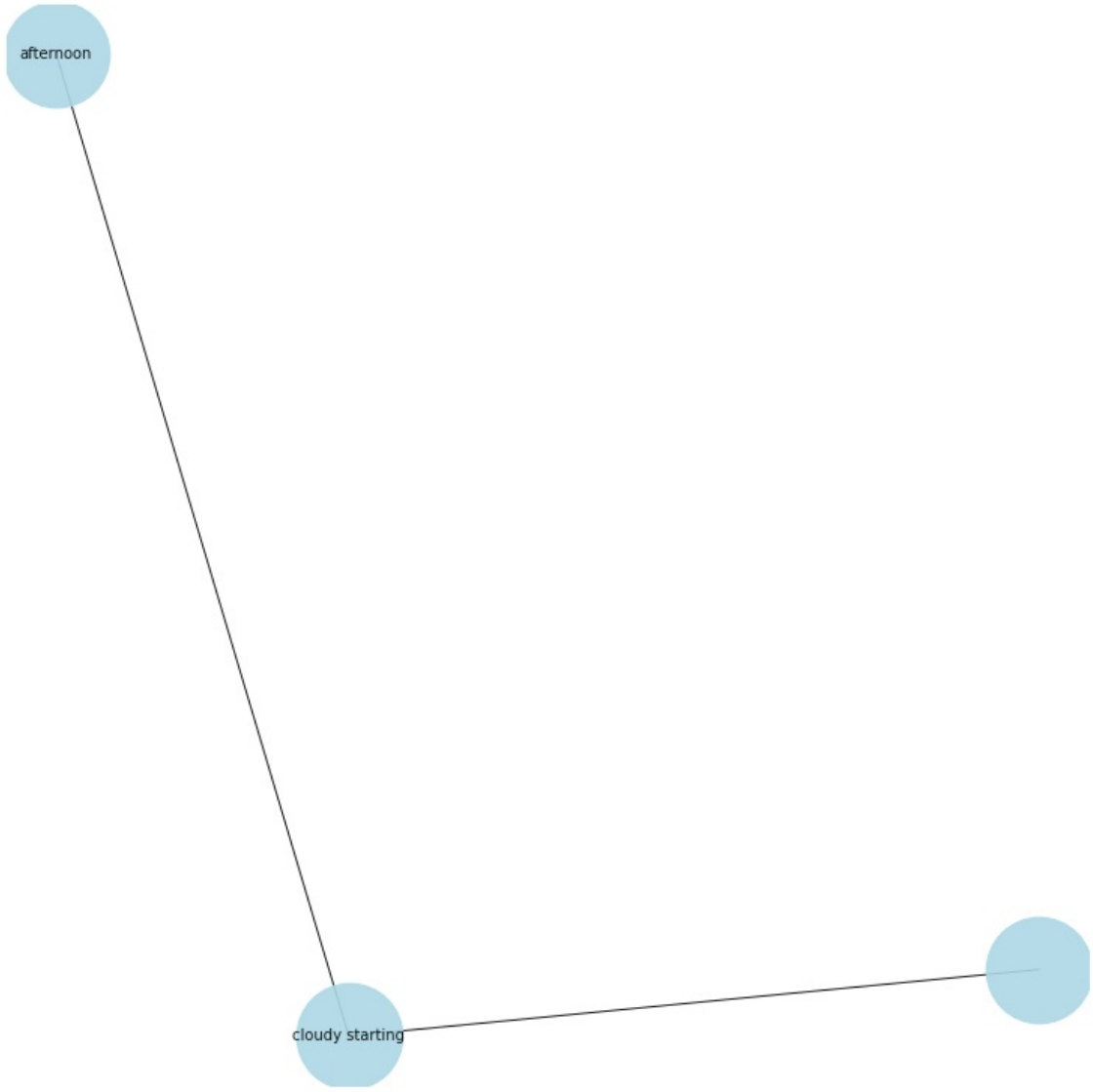



```
Partly cloudy starting overnight.
Partly -> advmod
cloudy -> amod
starting -> ROOT
overnight -> advmod
. -> punct
 , cloudy starting ,
```



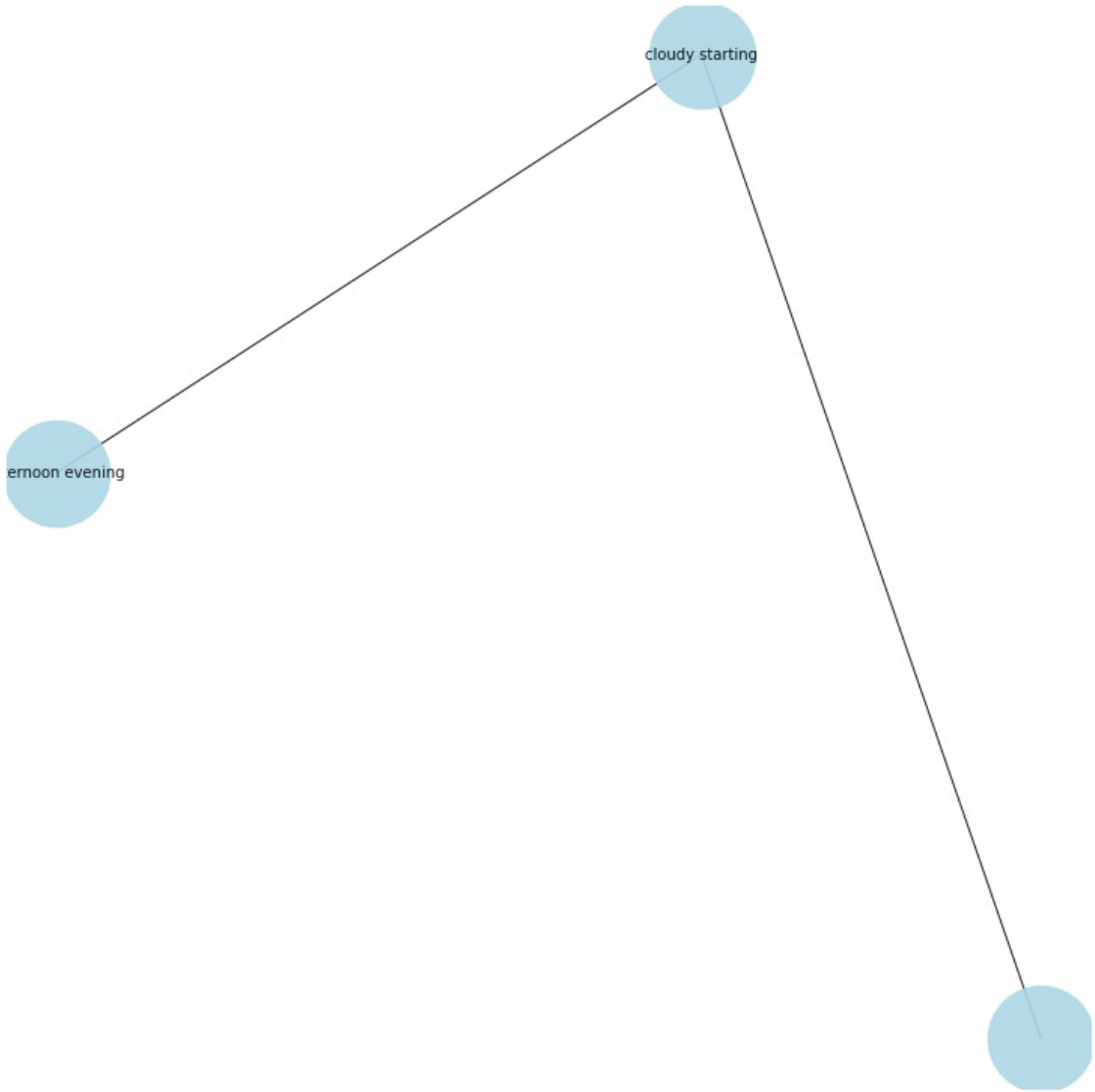


```
Partly cloudy starting in the afternoon.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy starting , afternoon
```

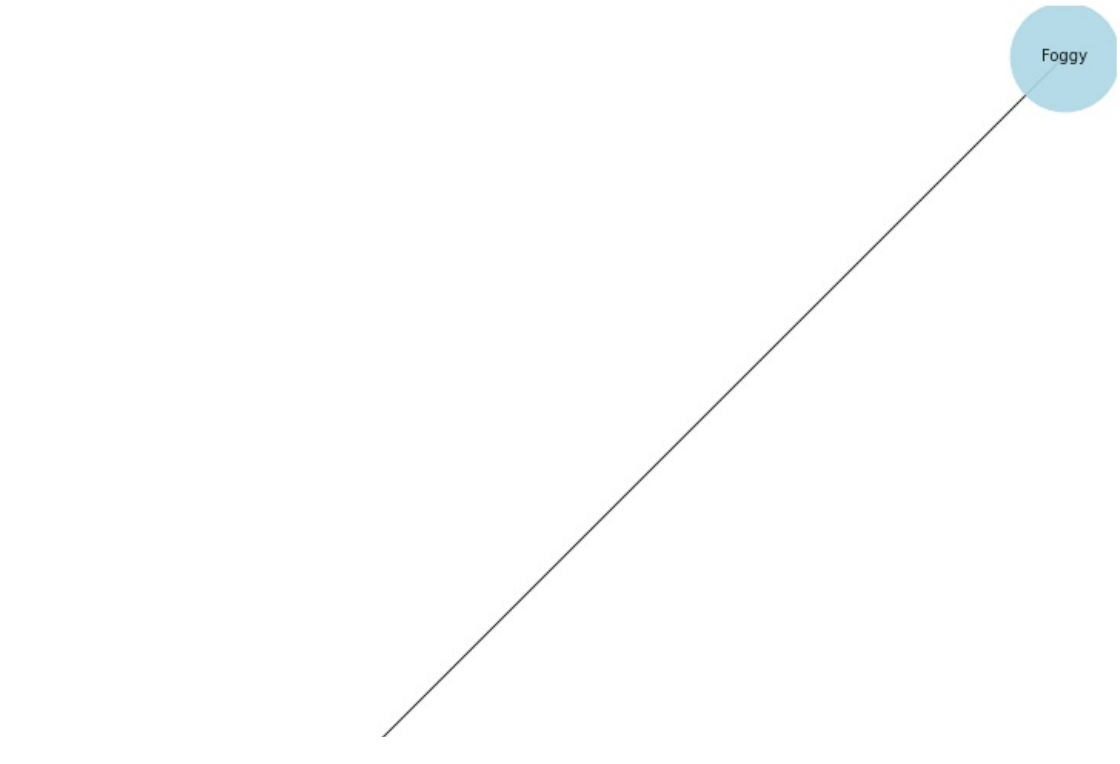


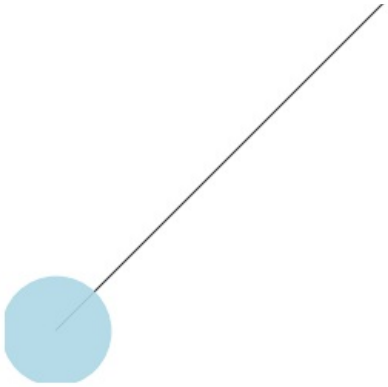

```
Partly cloudy starting in the afternoon continuing until evening.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
afternoon -> pobj
continuing -> xcomp
until -> prep
```

```
evening -> pobj
. -> punct
 , cloudy starting , afternoon evening
```

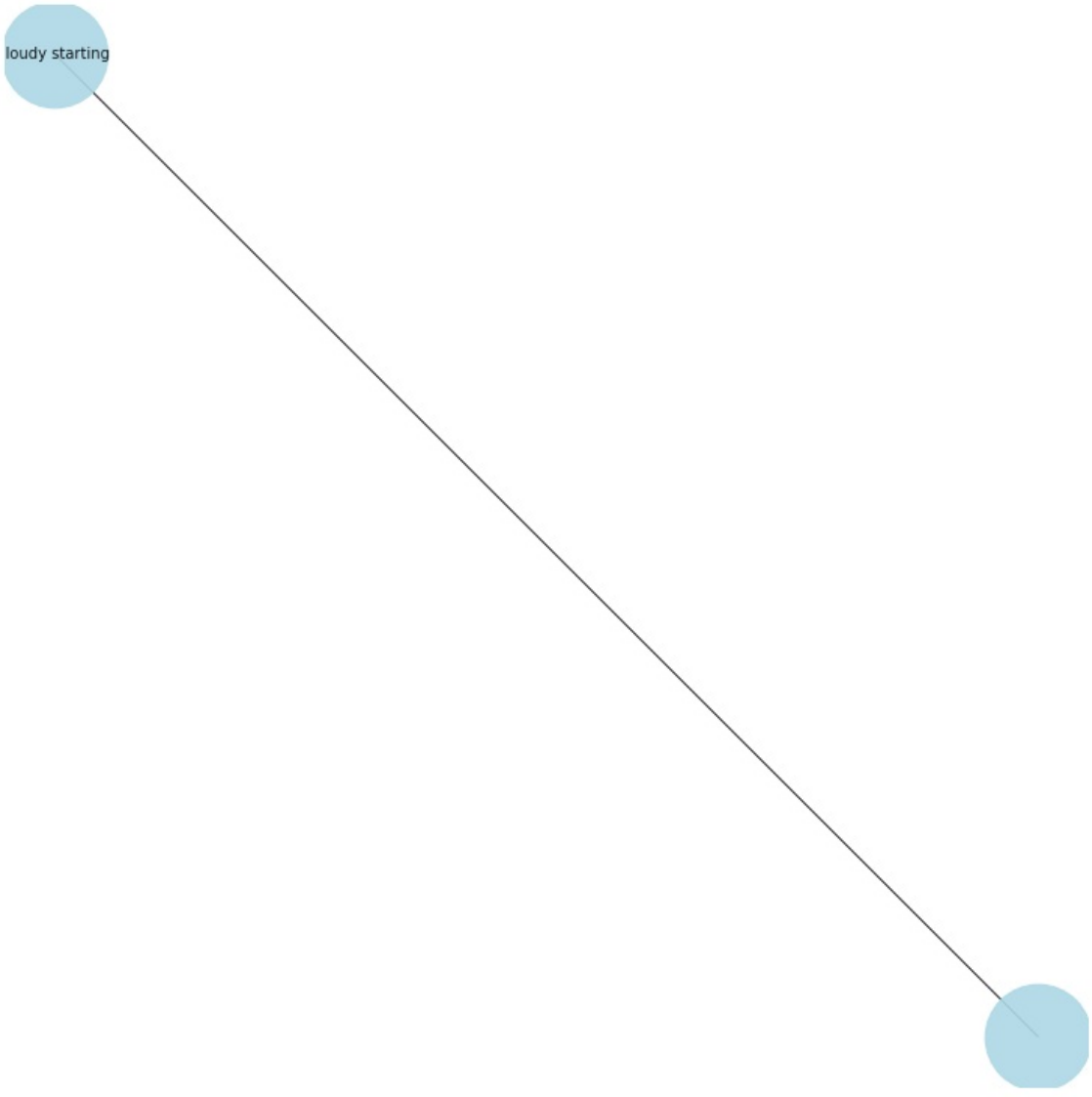



```
Foggy overnight.
Foggy -> ROOT
overnight -> advmod
. -> punct
 , Foggy ,
```

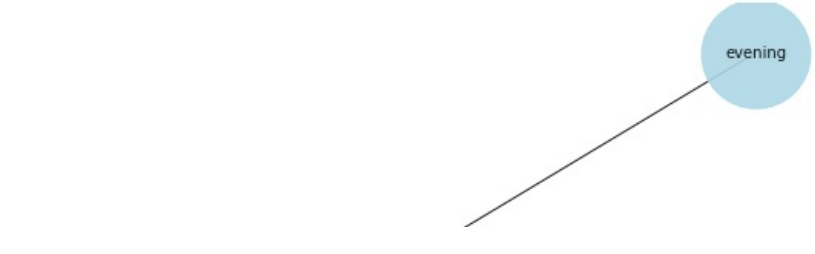


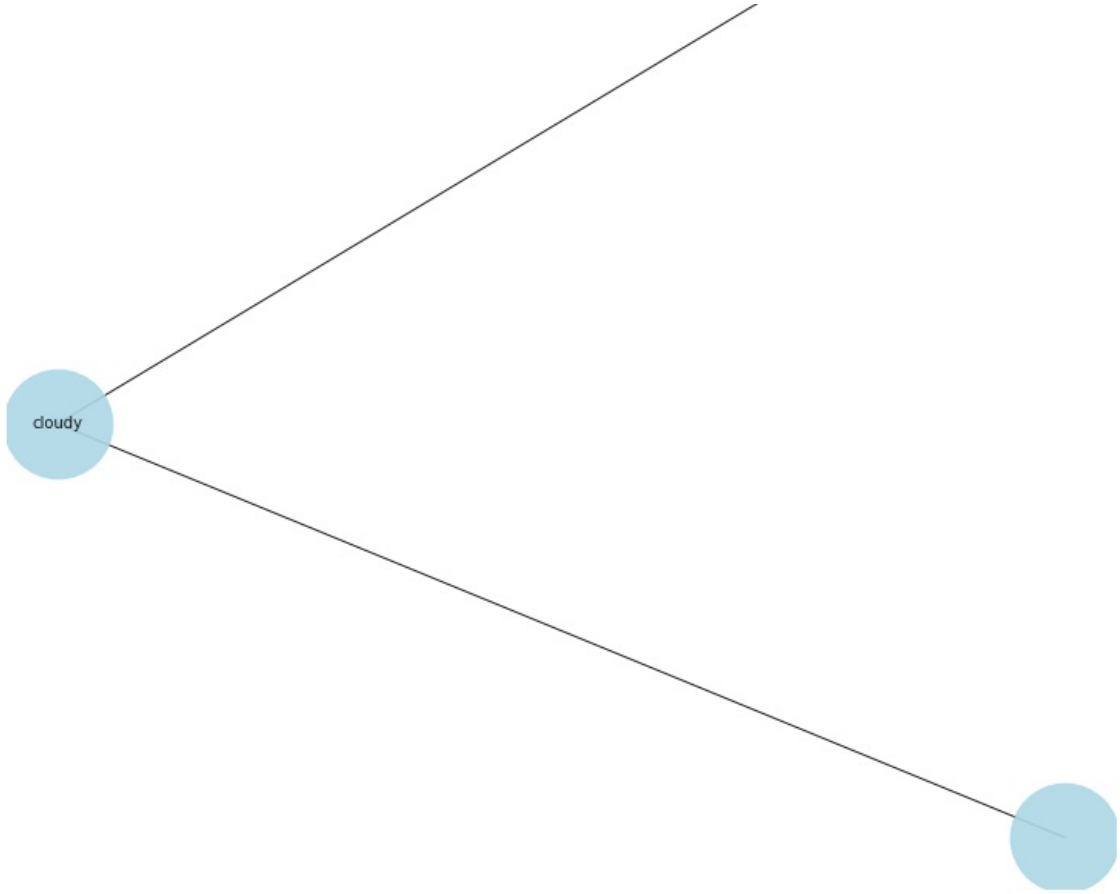


```
Mostly cloudy starting overnight.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
overnight -> advmod
. -> punct
 , cloudy starting ,
```

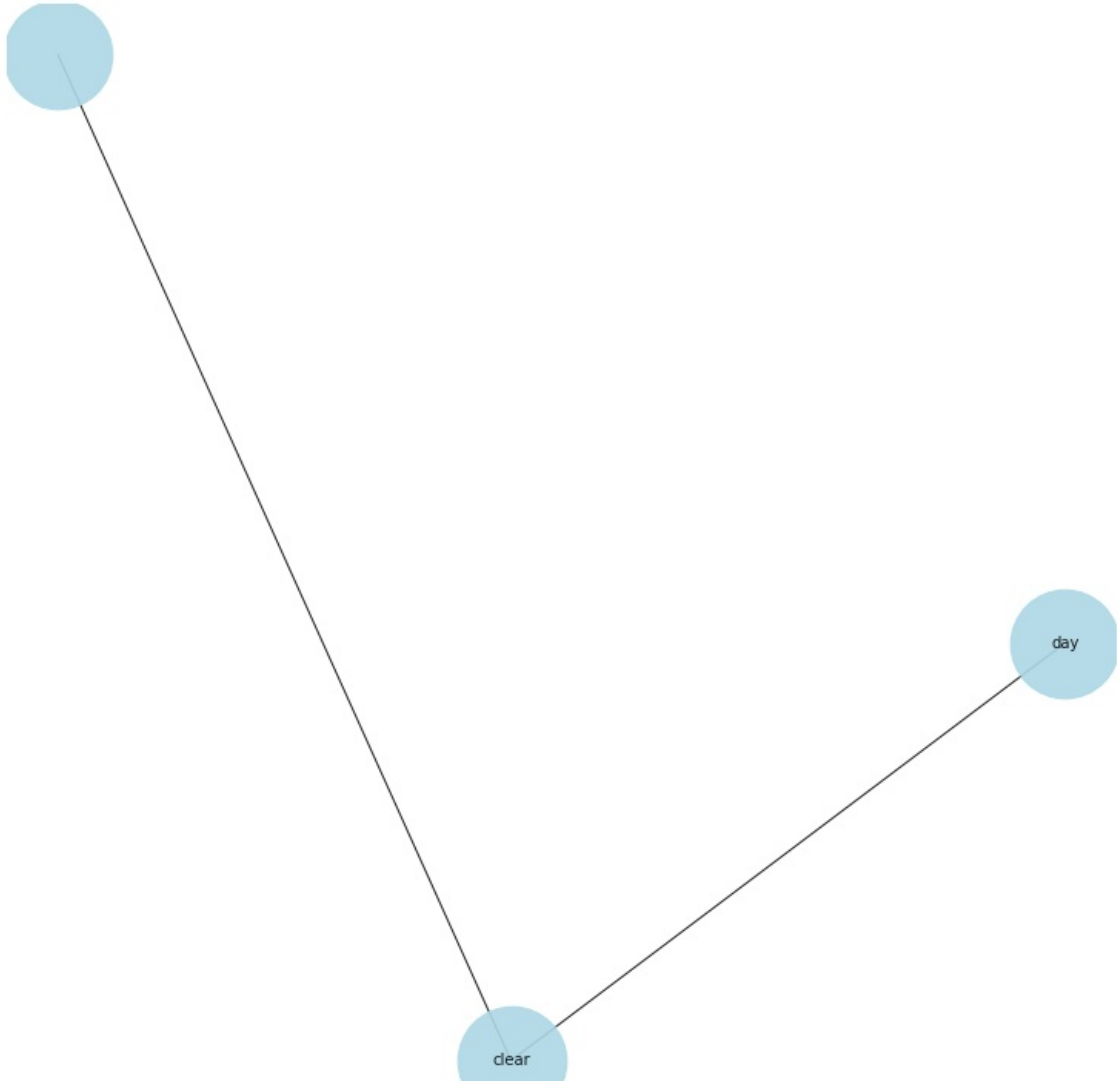



```
Mostly cloudy until evening.
Mostly -> advmod
cloudy -> ROOT
until -> prep
evening -> pobj
. -> punct
 , cloudy , evening
```

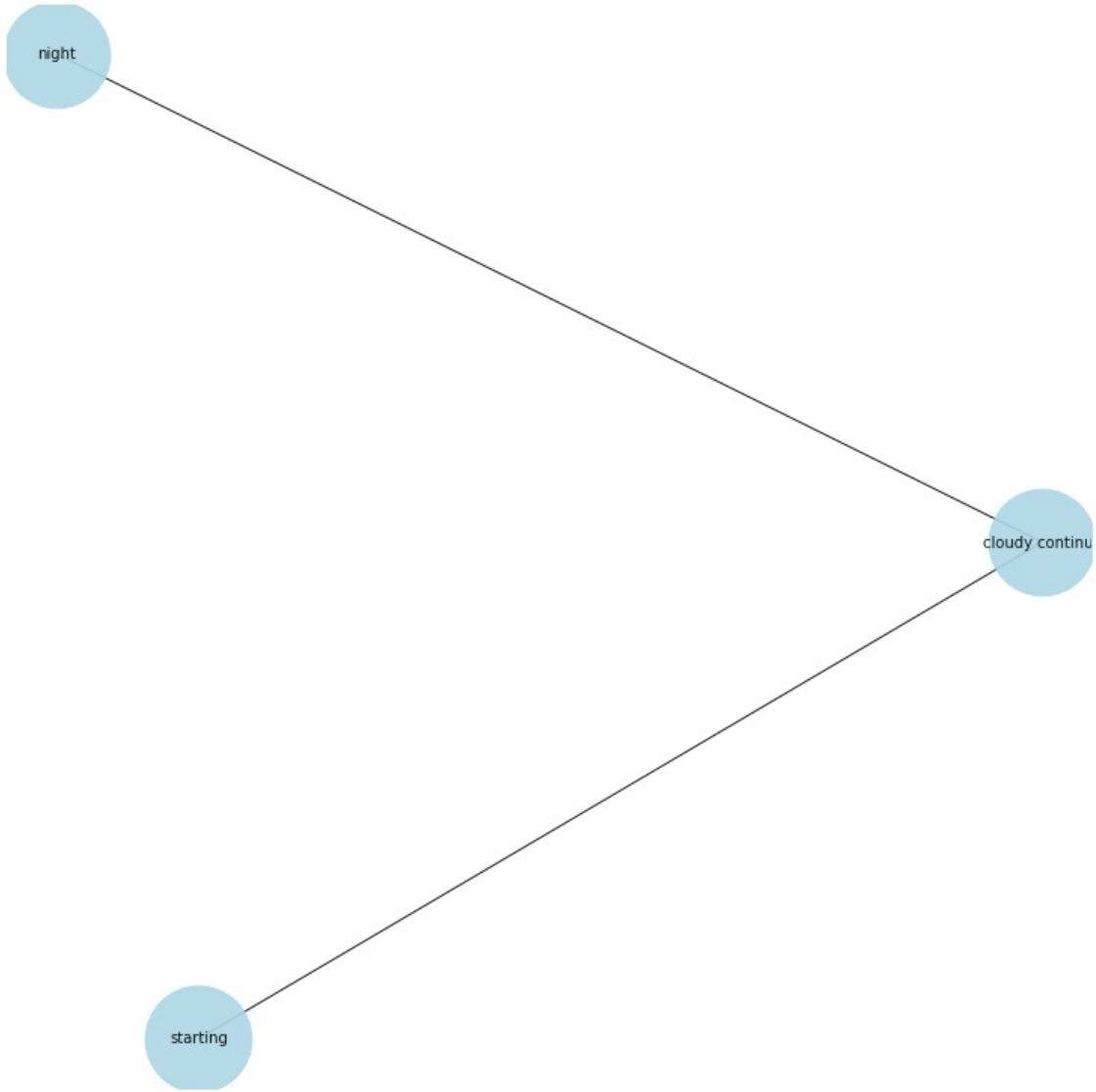




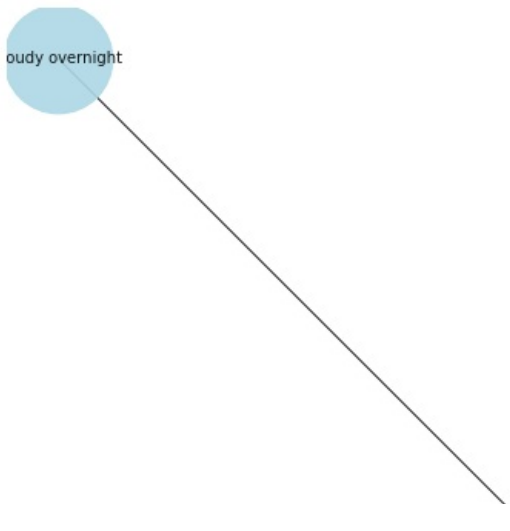

```
Clear throughout the day.
Clear -> ROOT
throughout -> prep
the -> det
day -> pobj
. -> punct
 , clear , day
```

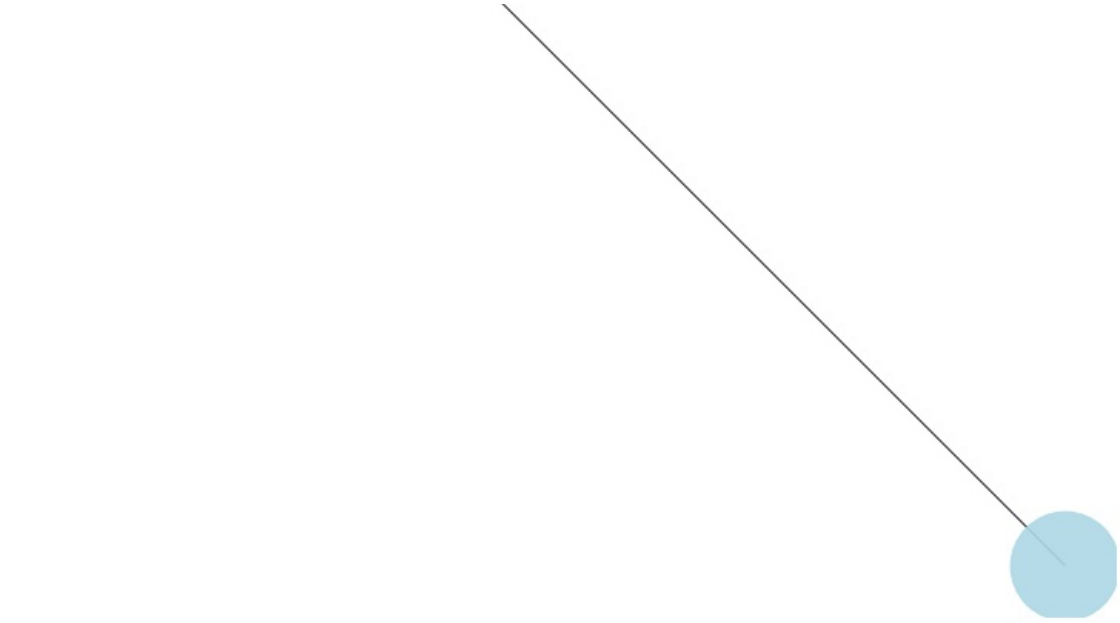


```
Partly cloudy starting overnight continuing until night.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
starting , cloudy continue , night
```

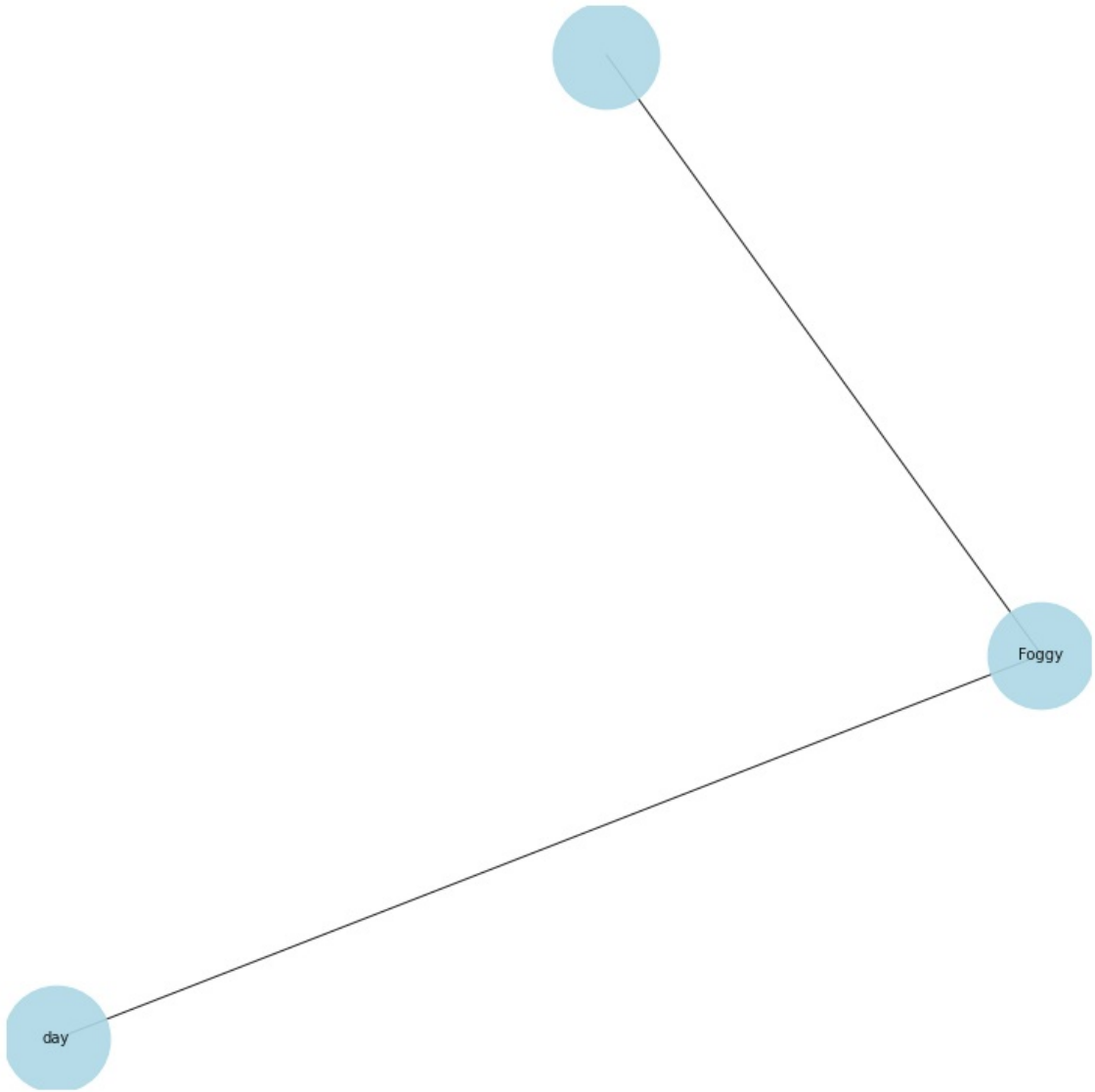



```
Partly cloudy overnight.
Partly -> advmod
cloudy -> amod
overnight -> ROOT
. -> punct
 , cloudy overnight ,
```



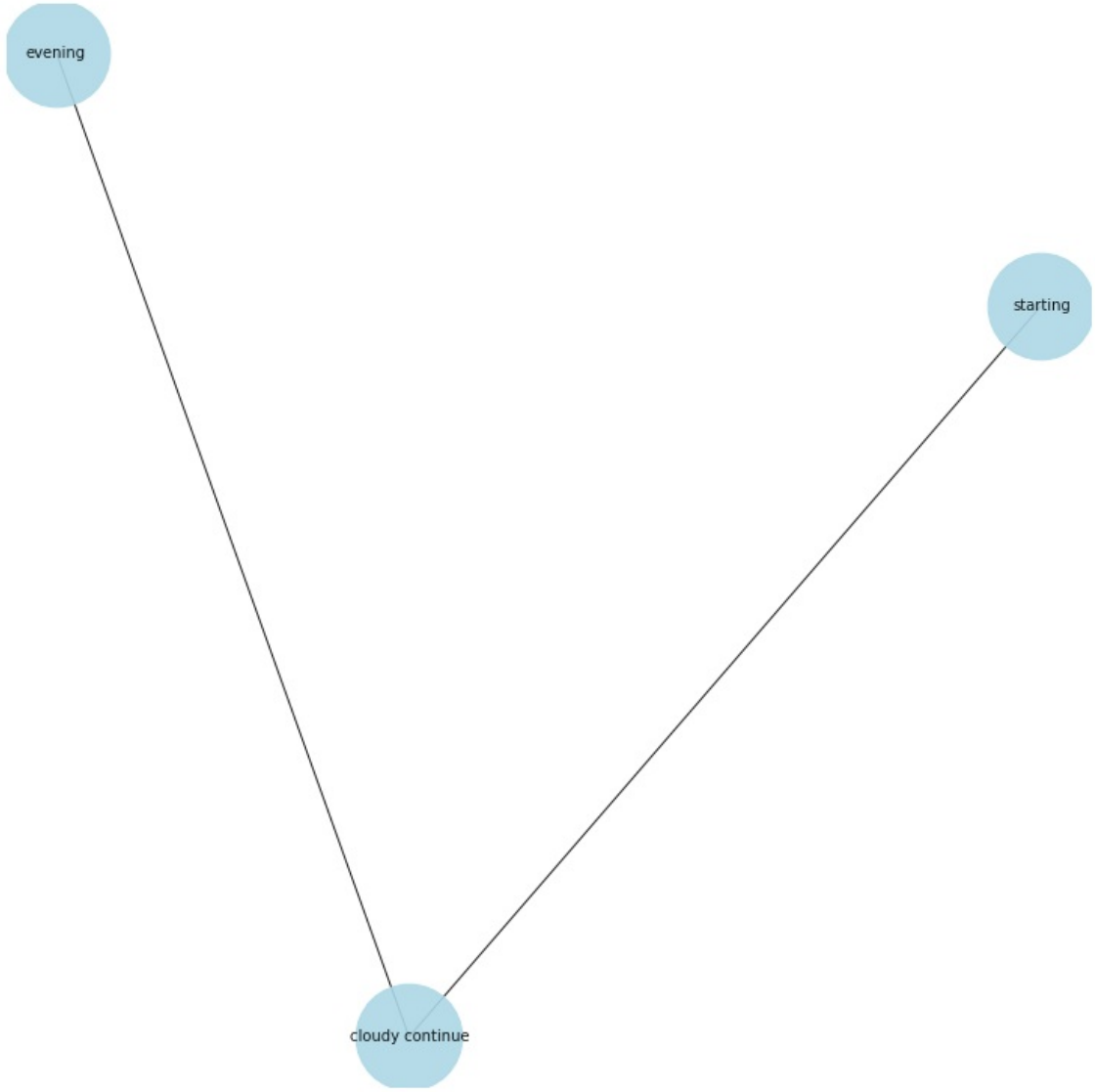


```
Foggy throughout the day.
Foggy -> ROOT
throughout -> prep
the -> det
day -> pobj
. -> punct
 , Foggy , day
```

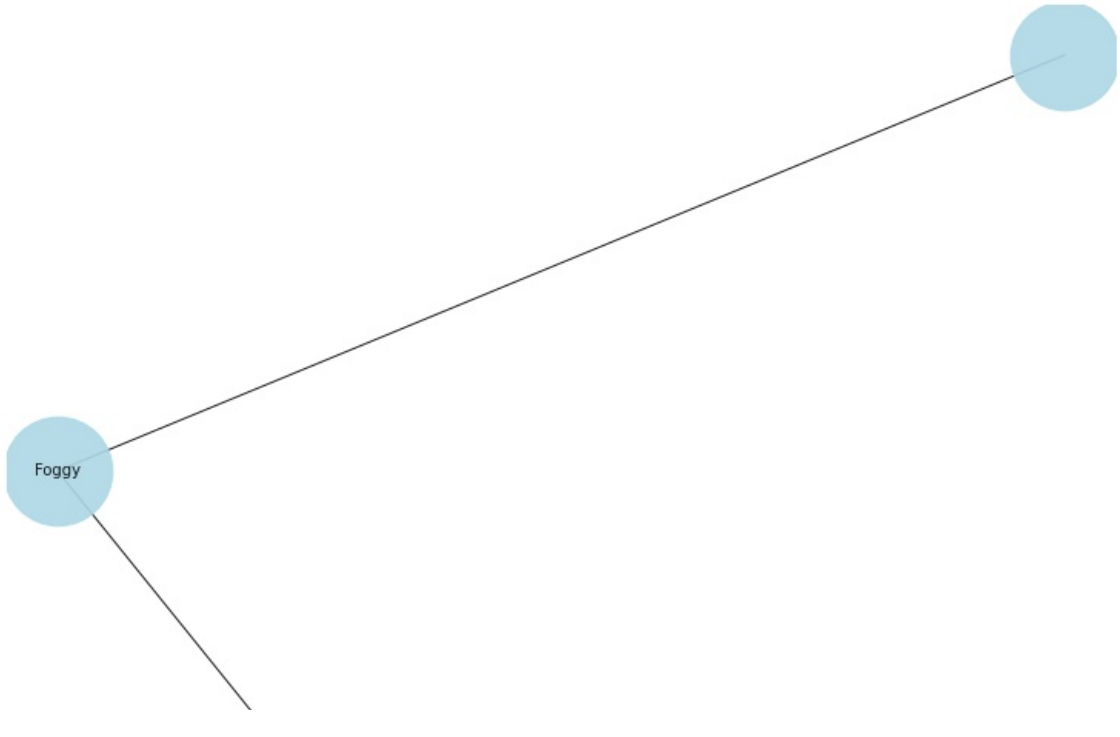


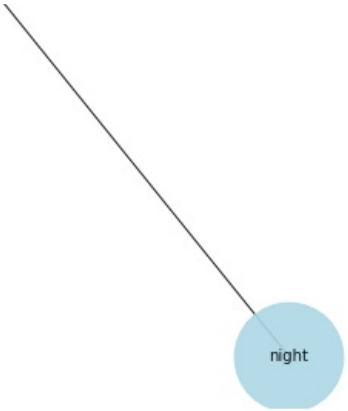

```
Partly cloudy starting overnight continuing until evening.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
```

```
evening -> pobj
. -> punct
starting , cloudy continue , evening
```

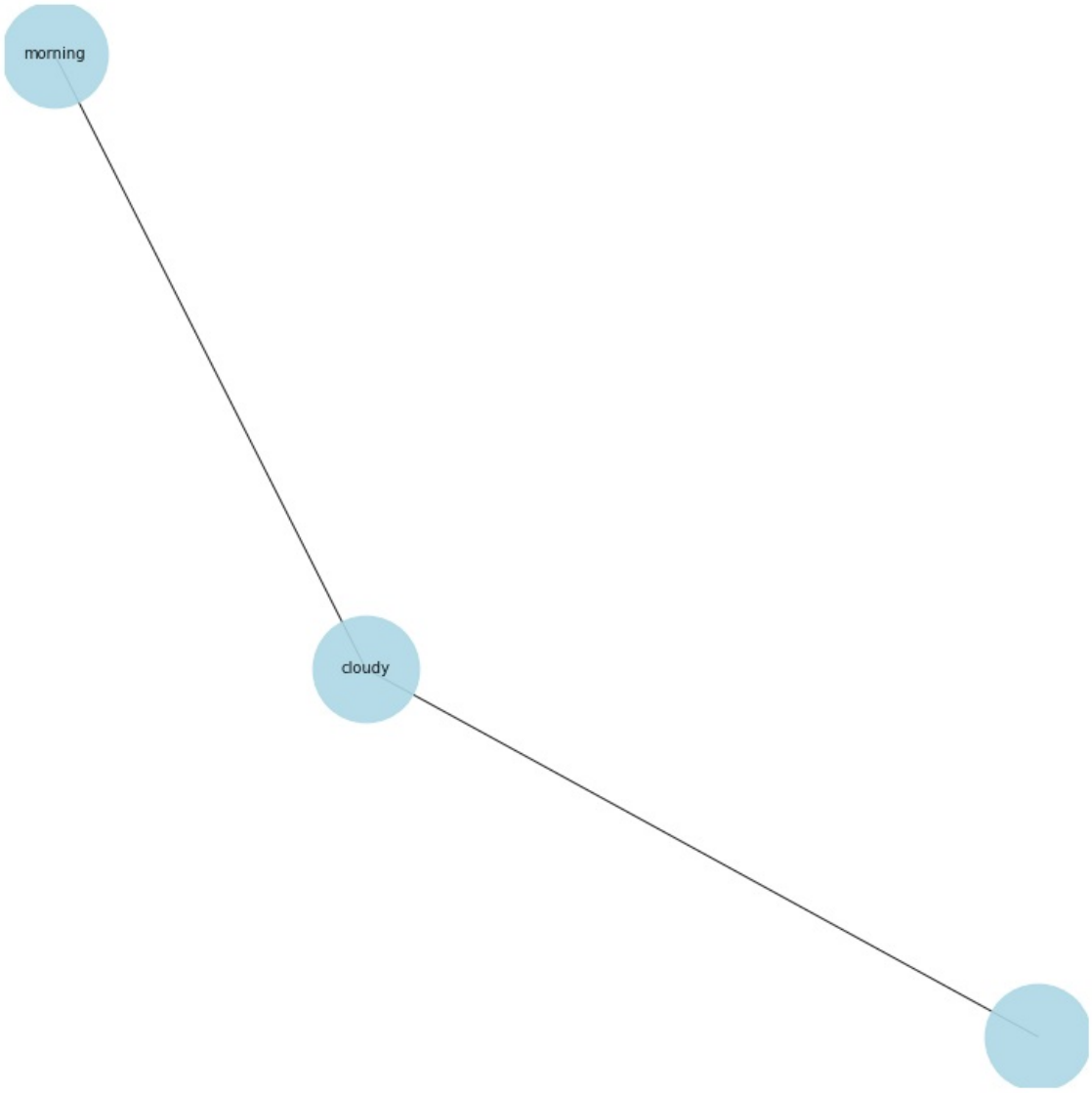



```
Foggy until night.
Foggy -> ROOT
until -> prep
night -> pobj
. -> punct
 , Foggy , night
```

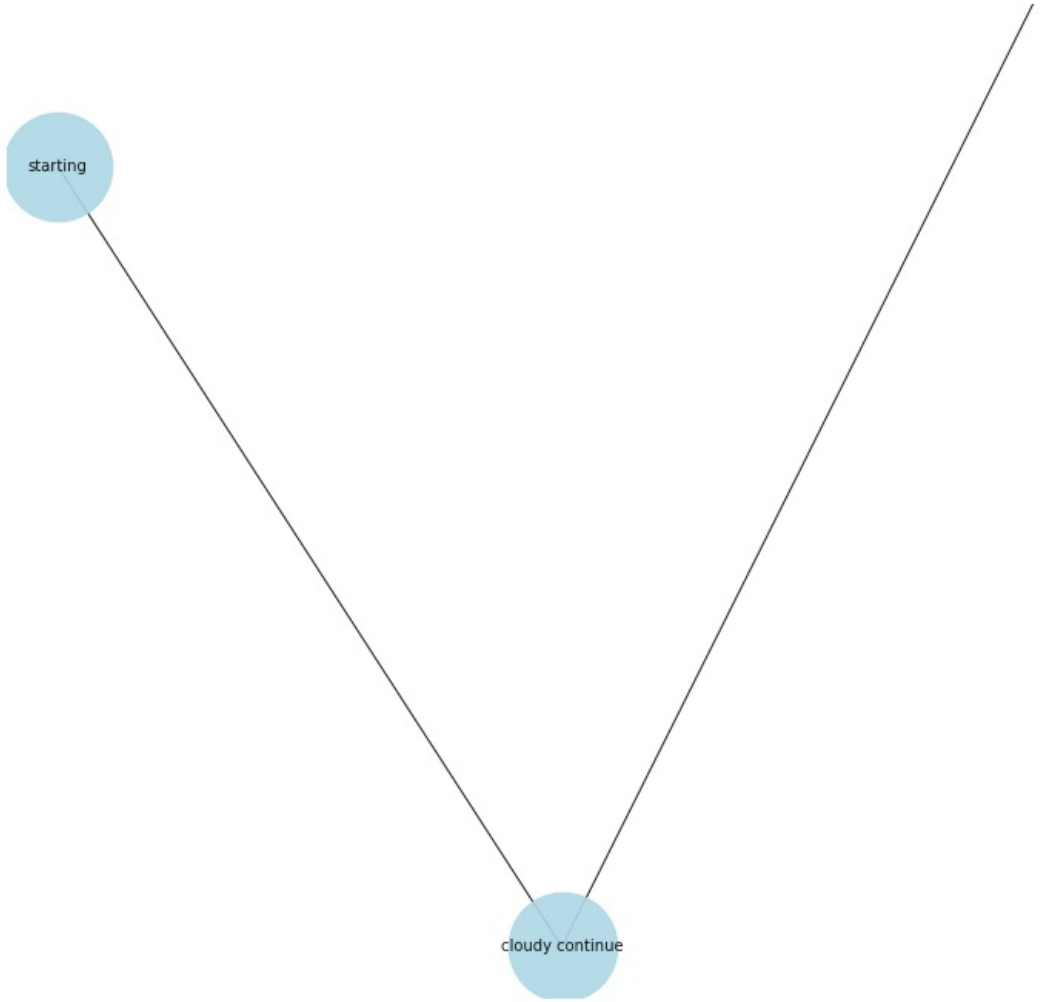




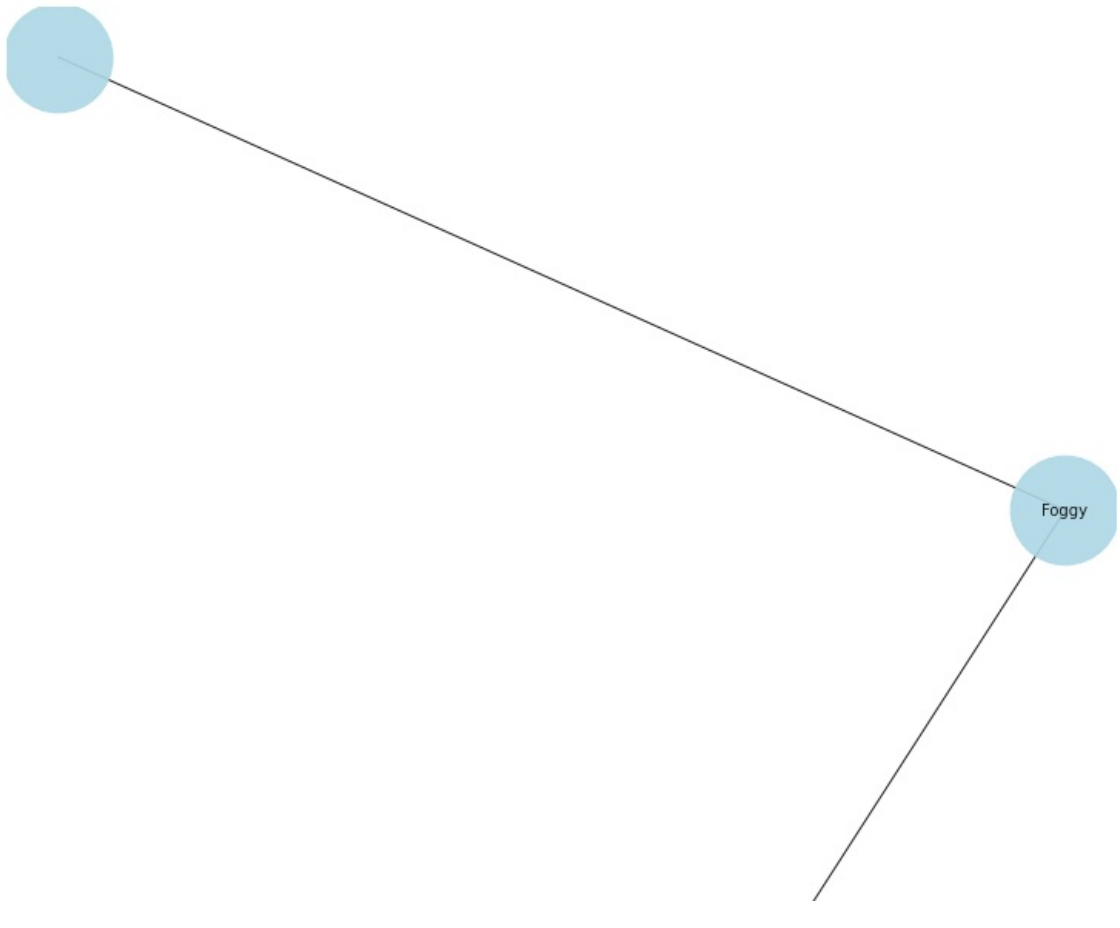

```
Partly cloudy in the morning.
Partly -> advmod
cloudy -> ROOT
in -> prep
the -> det
morning -> pobj
. -> punct
 , cloudy , morning
```

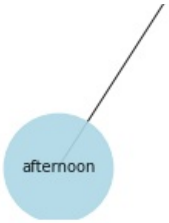



```
Mostly cloudy starting overnight continuing until night.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
starting , cloudy continue , night
```

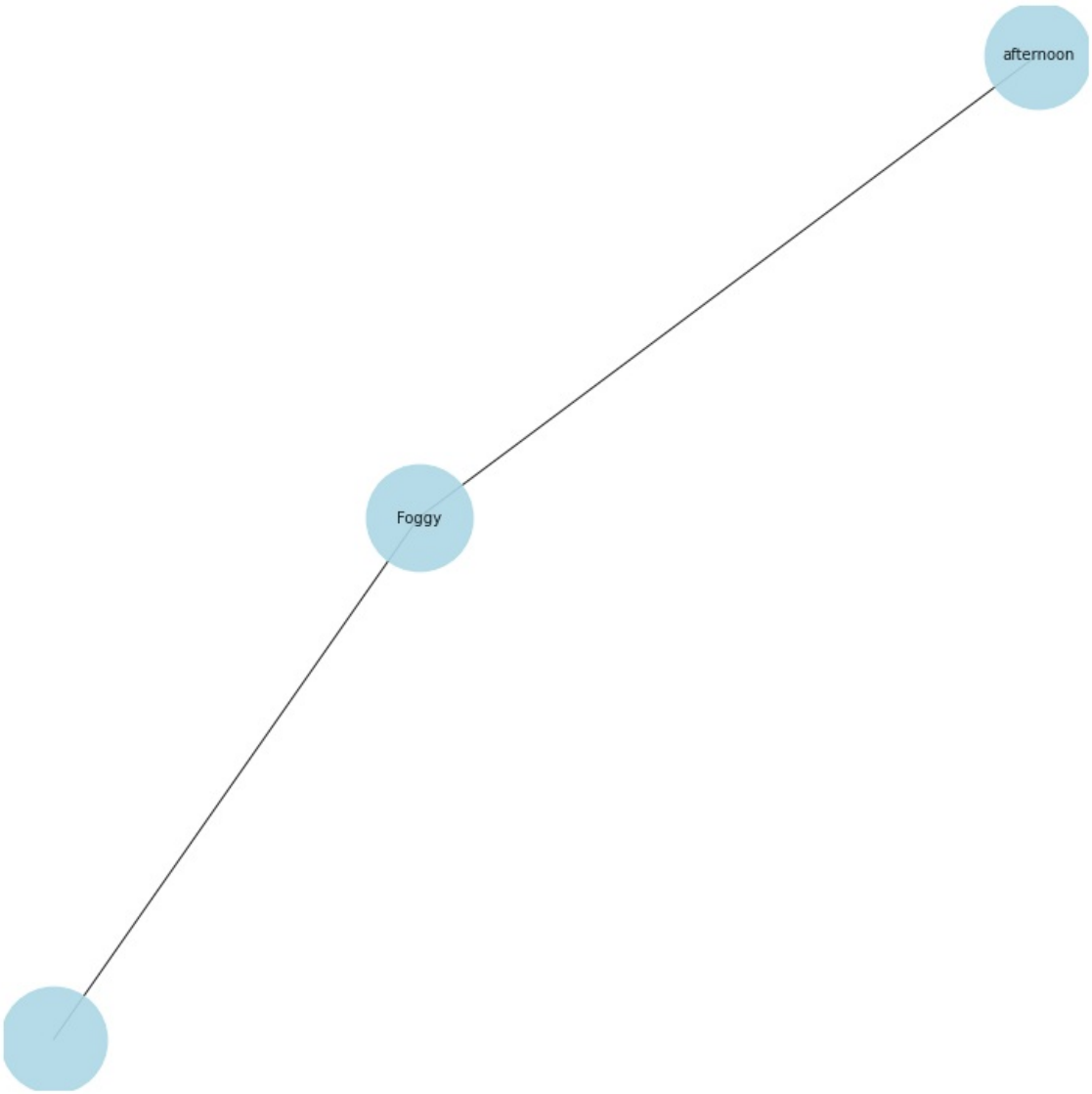



```
Foggy starting overnight continuing until afternoon.
Foggy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
afternoon -> pobj
. -> punct
 , Foggy , afternoon
```

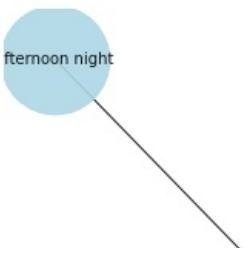


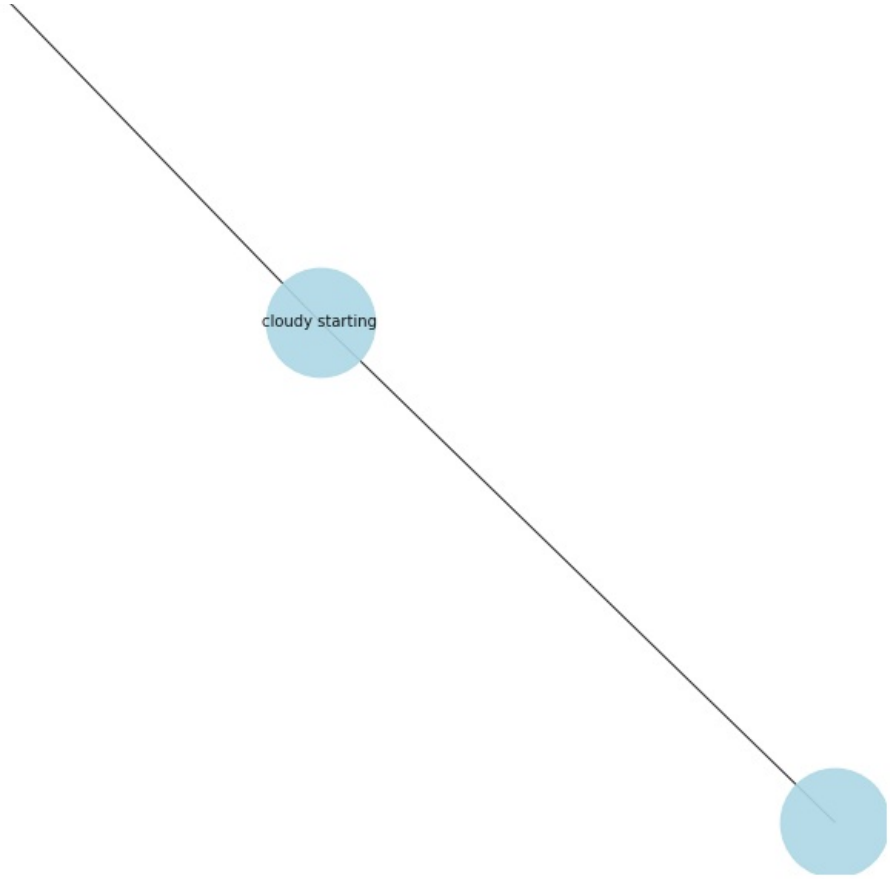



```
Foggy until afternoon.
Foggy -> ROOT
until -> prep
afternoon -> pobj
. -> punct
 , Foggy , afternoon
```

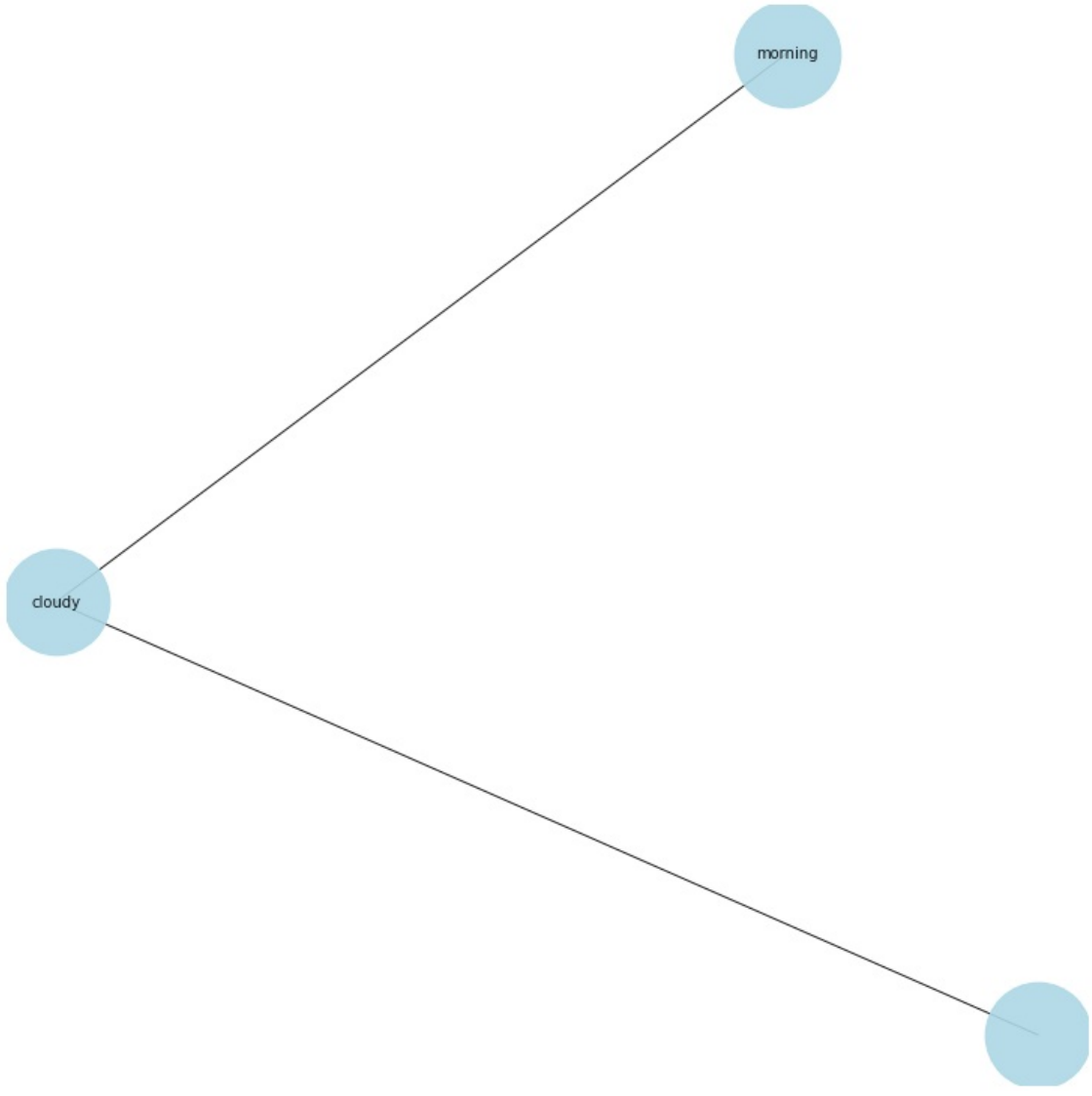



```
Partly cloudy starting in the afternoon continuing until night.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
afternoon -> pobj
continuing -> xcomp
until -> prep
night -> pobj
. -> punct
 , cloudy starting , afternoon night
```

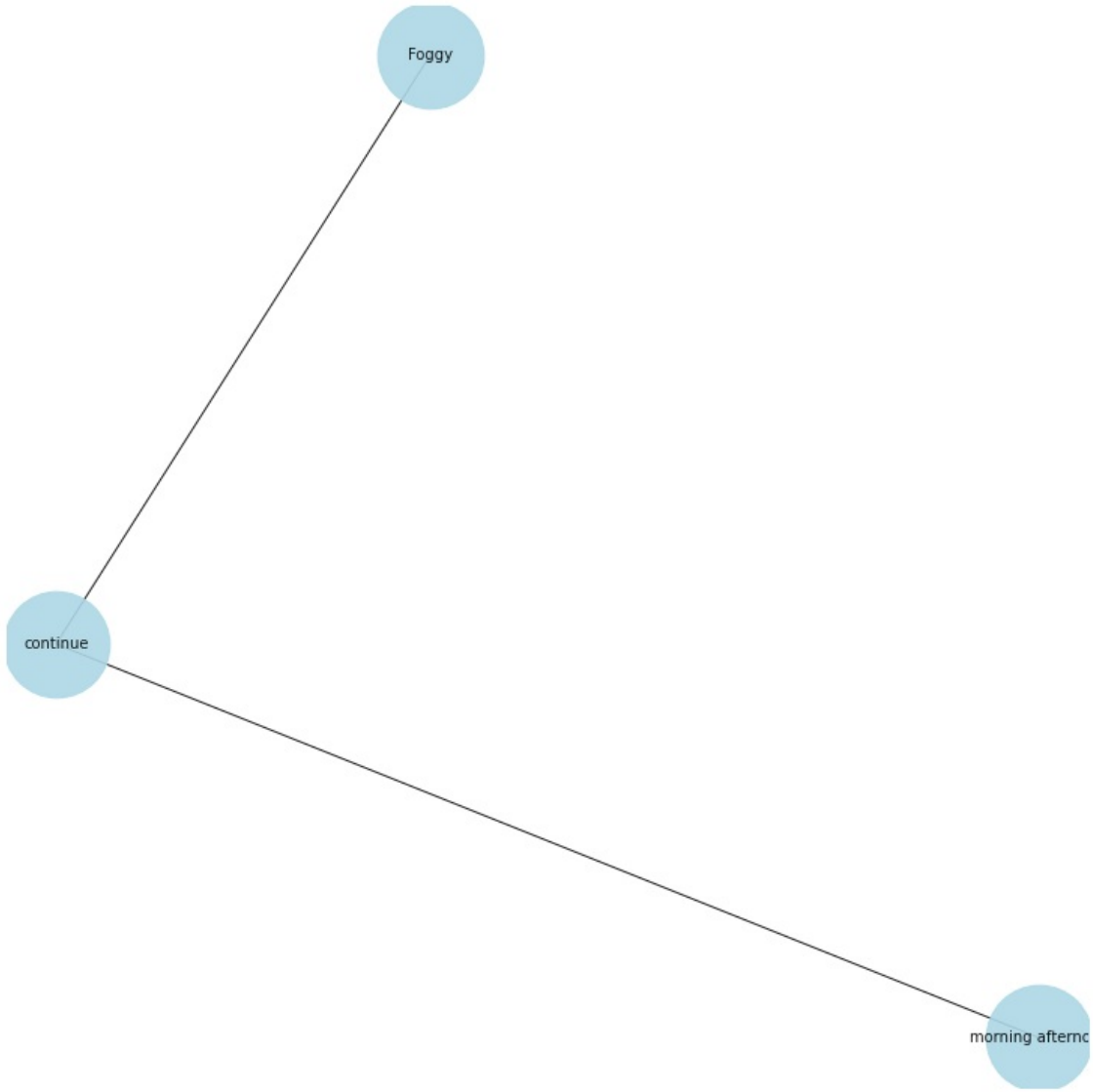




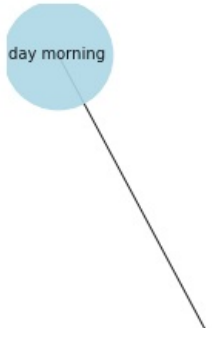

```
Partly cloudy until morning.
Partly -> advmod
cloudy -> ROOT
until -> prep
morning -> pobj
. -> punct
 , cloudy , morning
```

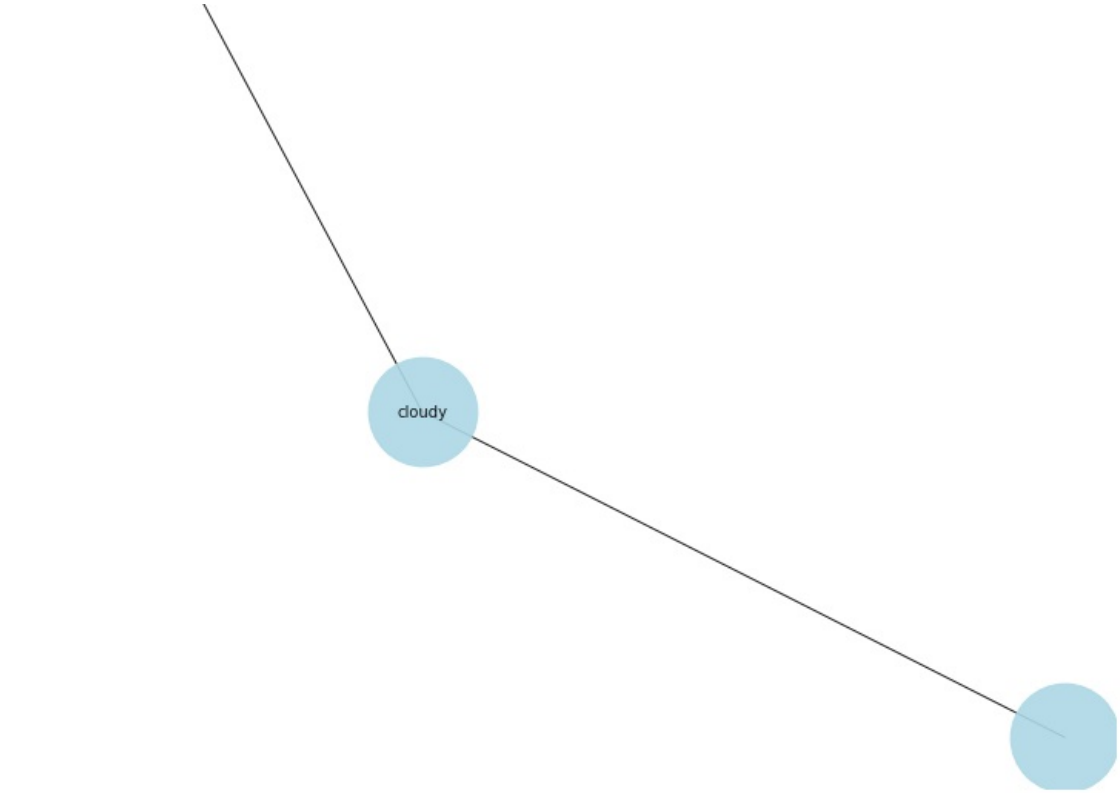


```
Foggy starting in the morning continuing until afternoon.
Foggy -> nsubj
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
Foggy , continue , morning afternoon
```

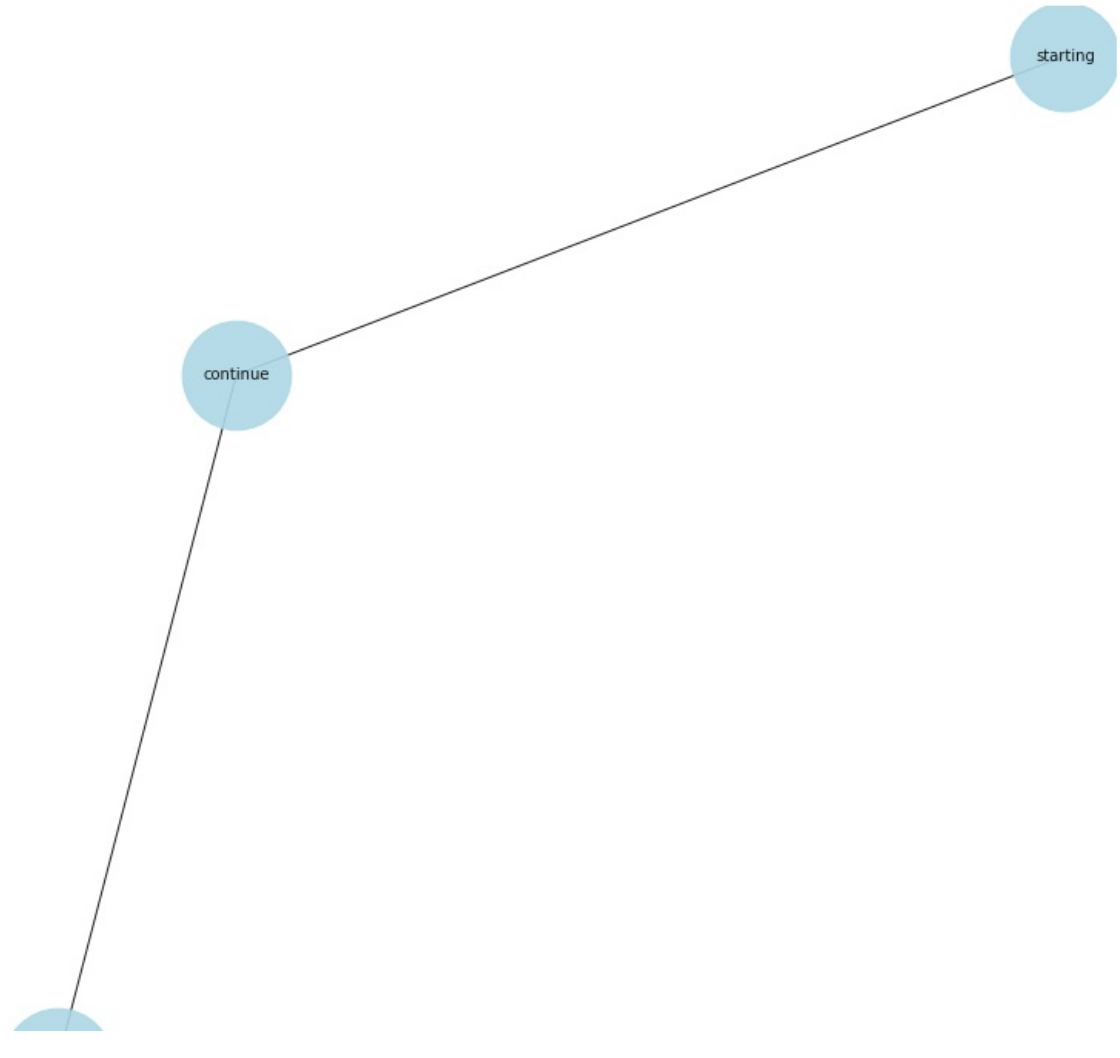


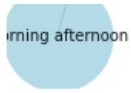

```
Mostly cloudy throughout the day and breezy in the morning.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , cloudy , day morning
```



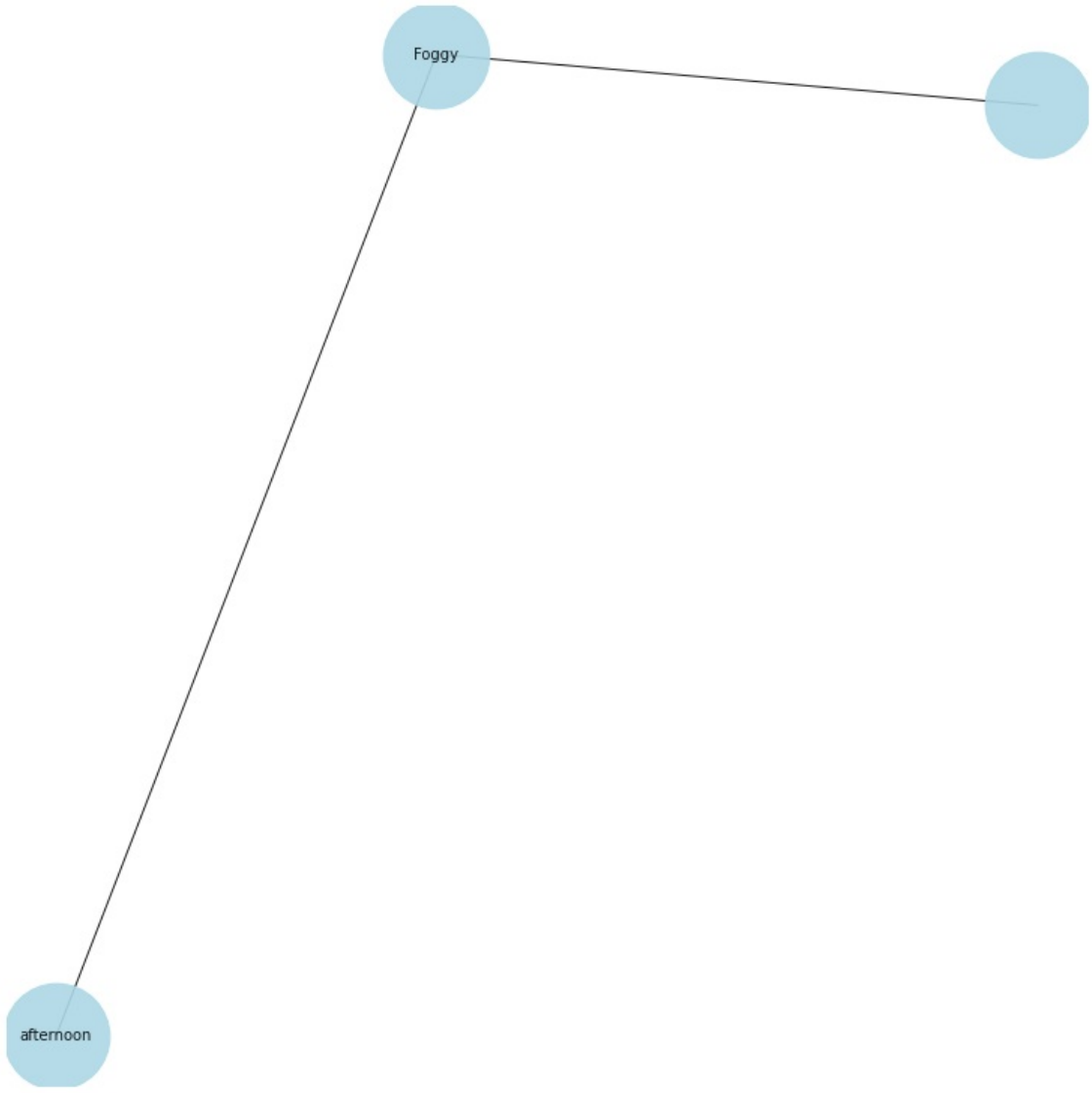



```
Partly cloudy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> advmod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
starting , continue , morning afternoon
```

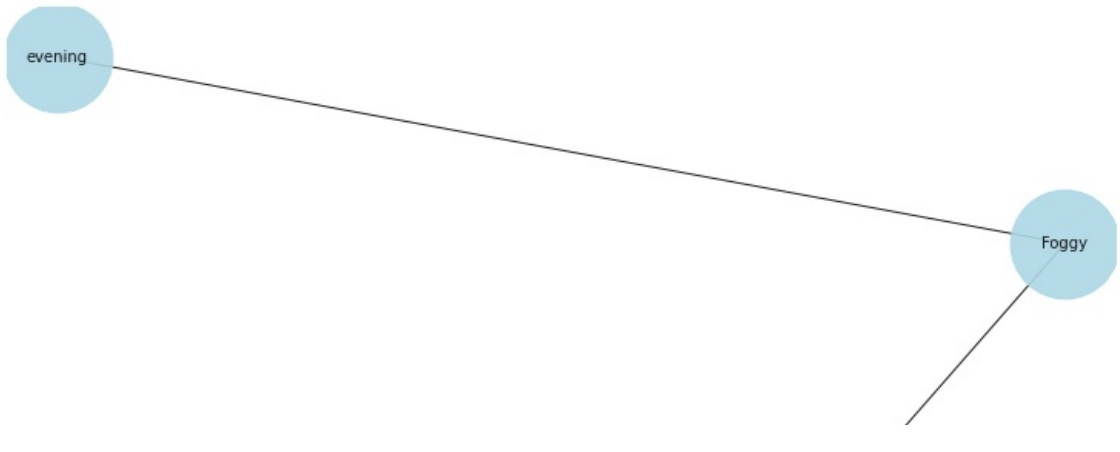


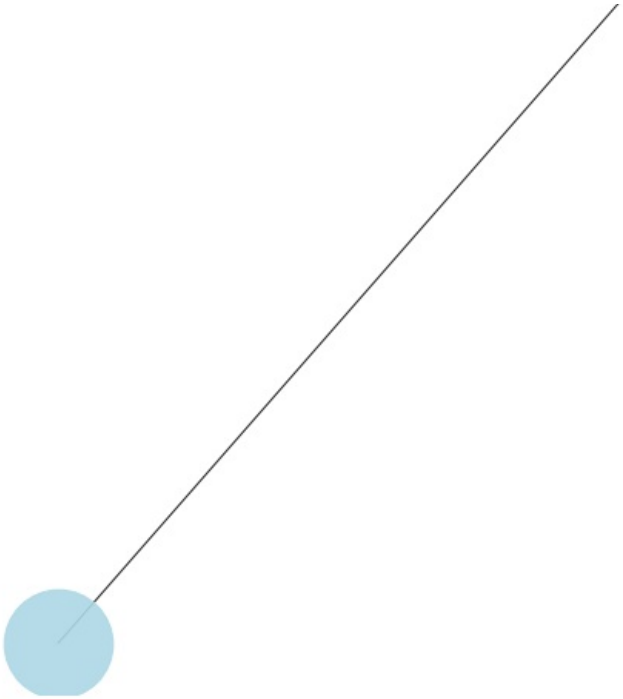



```
Foggy starting in the afternoon.
Foggy -> ROOT
starting -> acl
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , Foggy , afternoon
```

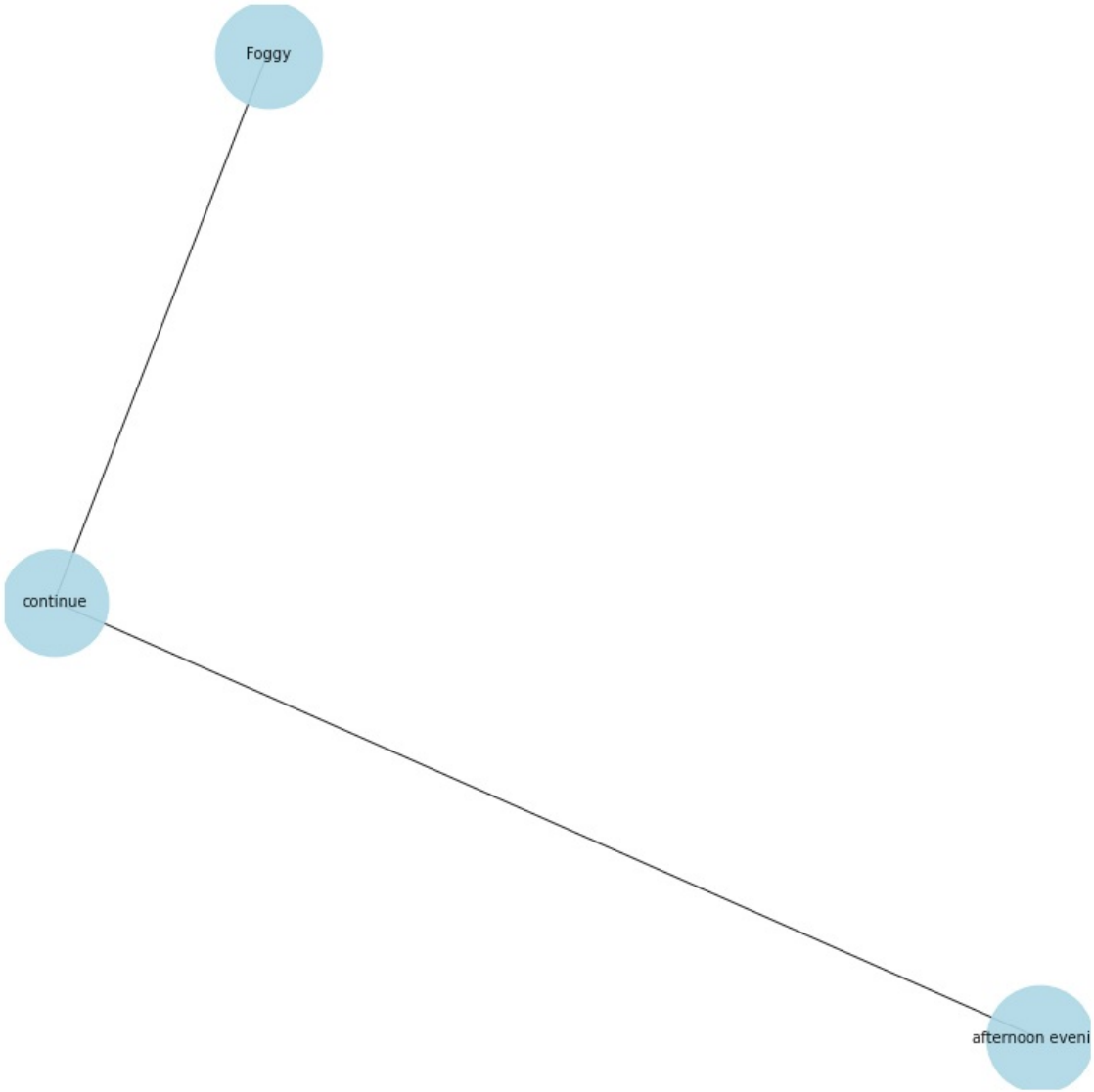



```
Foggy starting overnight continuing until evening.
Foggy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
evening -> pobj
. -> punct
 , Foggy , evening
```

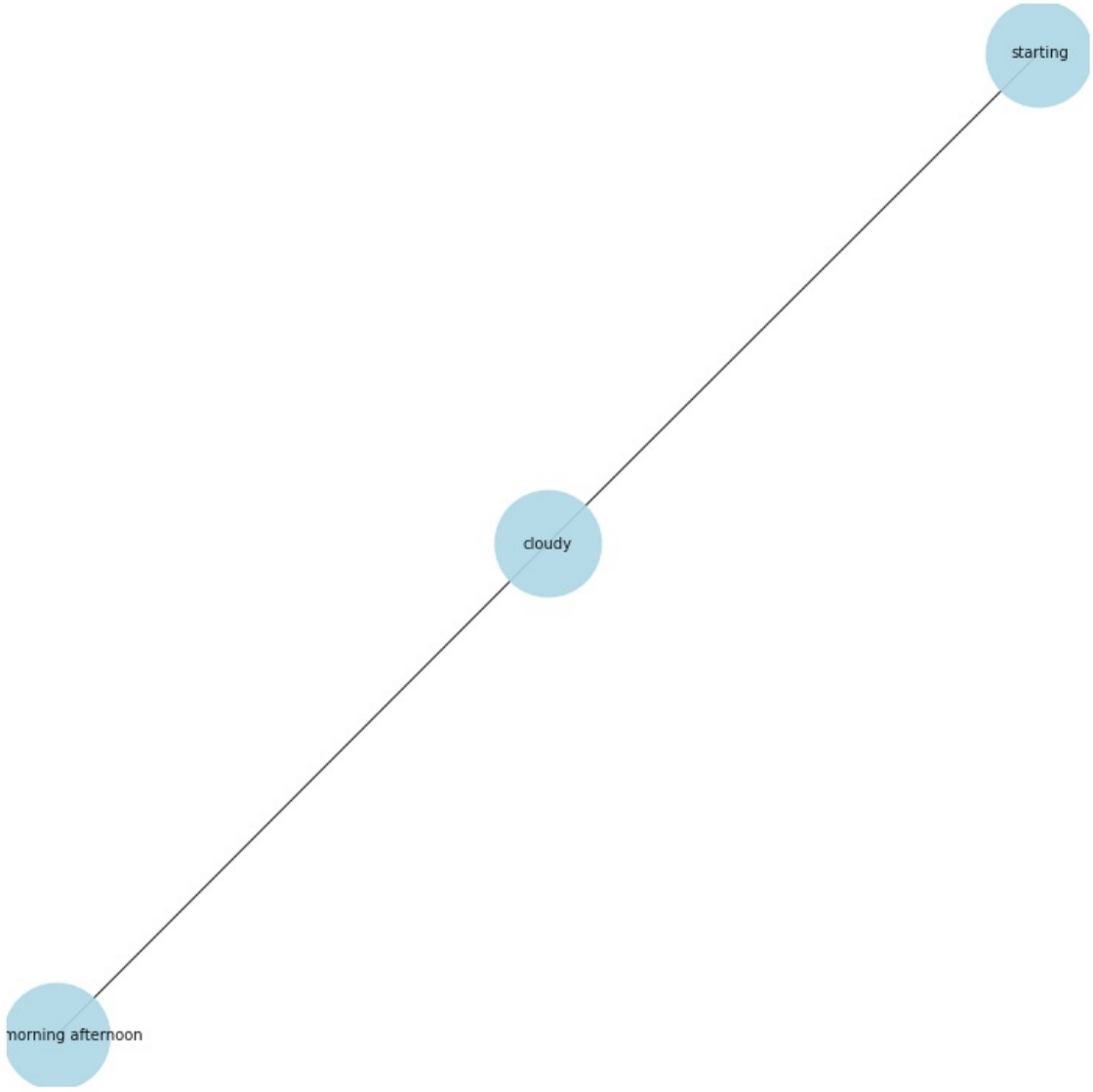




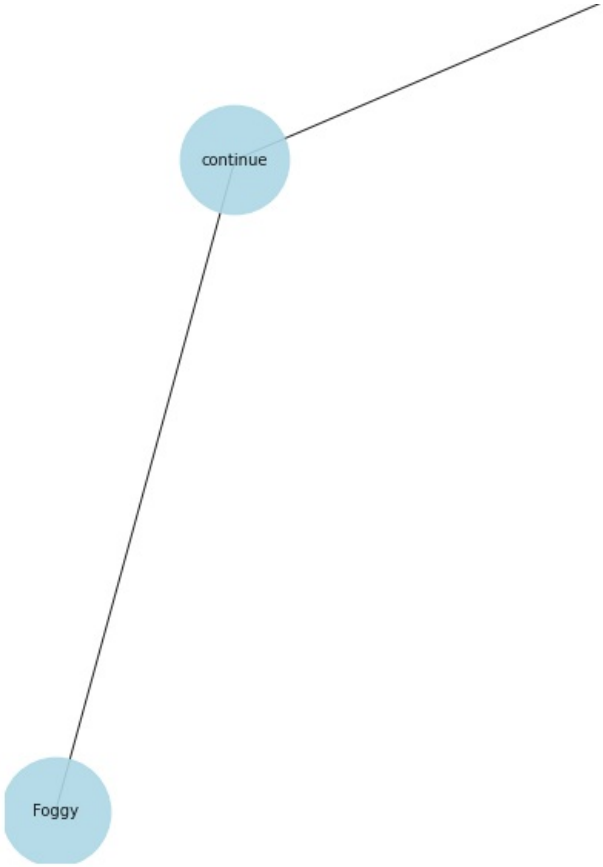
```
Foggy starting in the afternoon continuing until evening.
Foggy -> nsubj
starting -> acl
in -> prep
the -> det
afternoon -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
Foggy , continue , afternoon evening
```



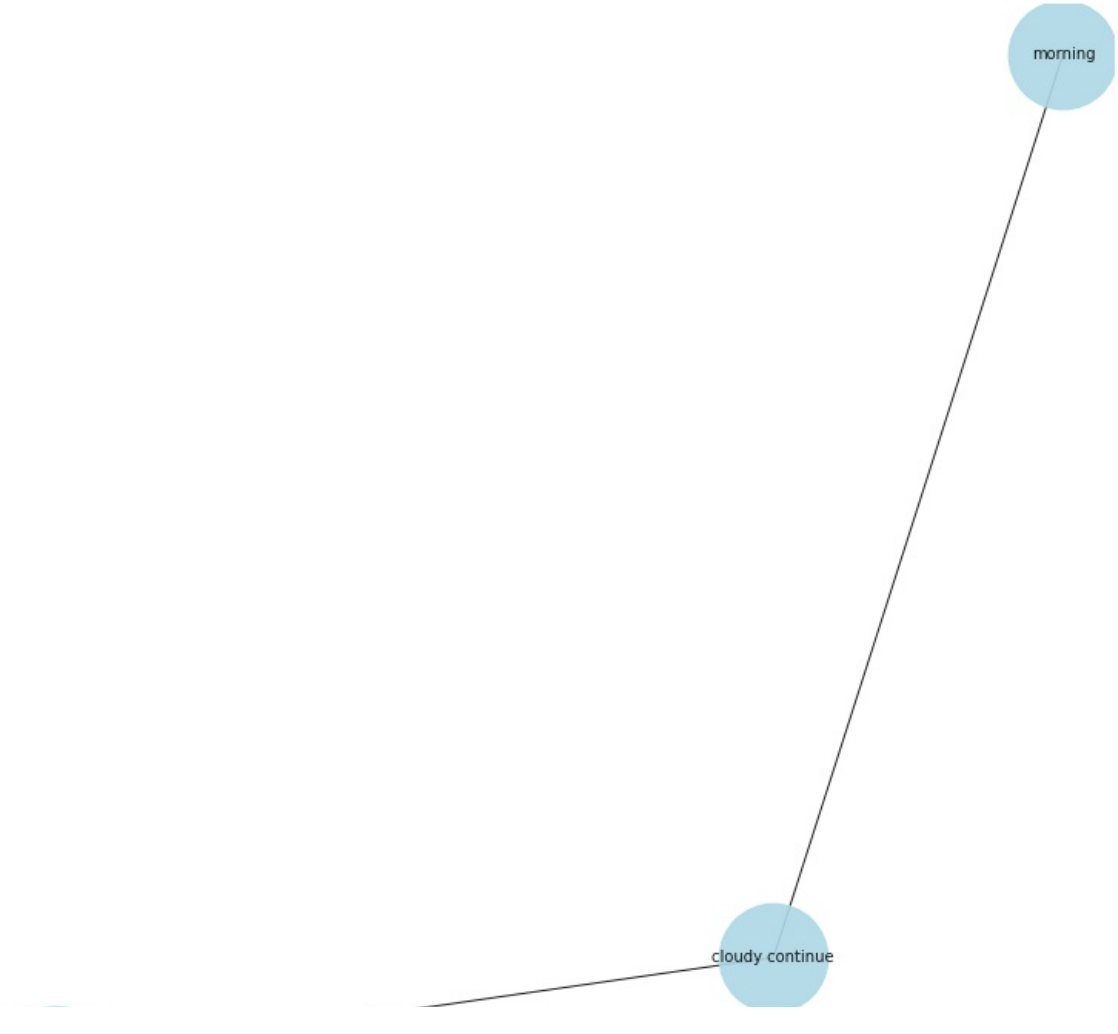
```
Mostly cloudy throughout the day and breezy starting in the morning continuing until afternoon.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy , day morning afternoon
```



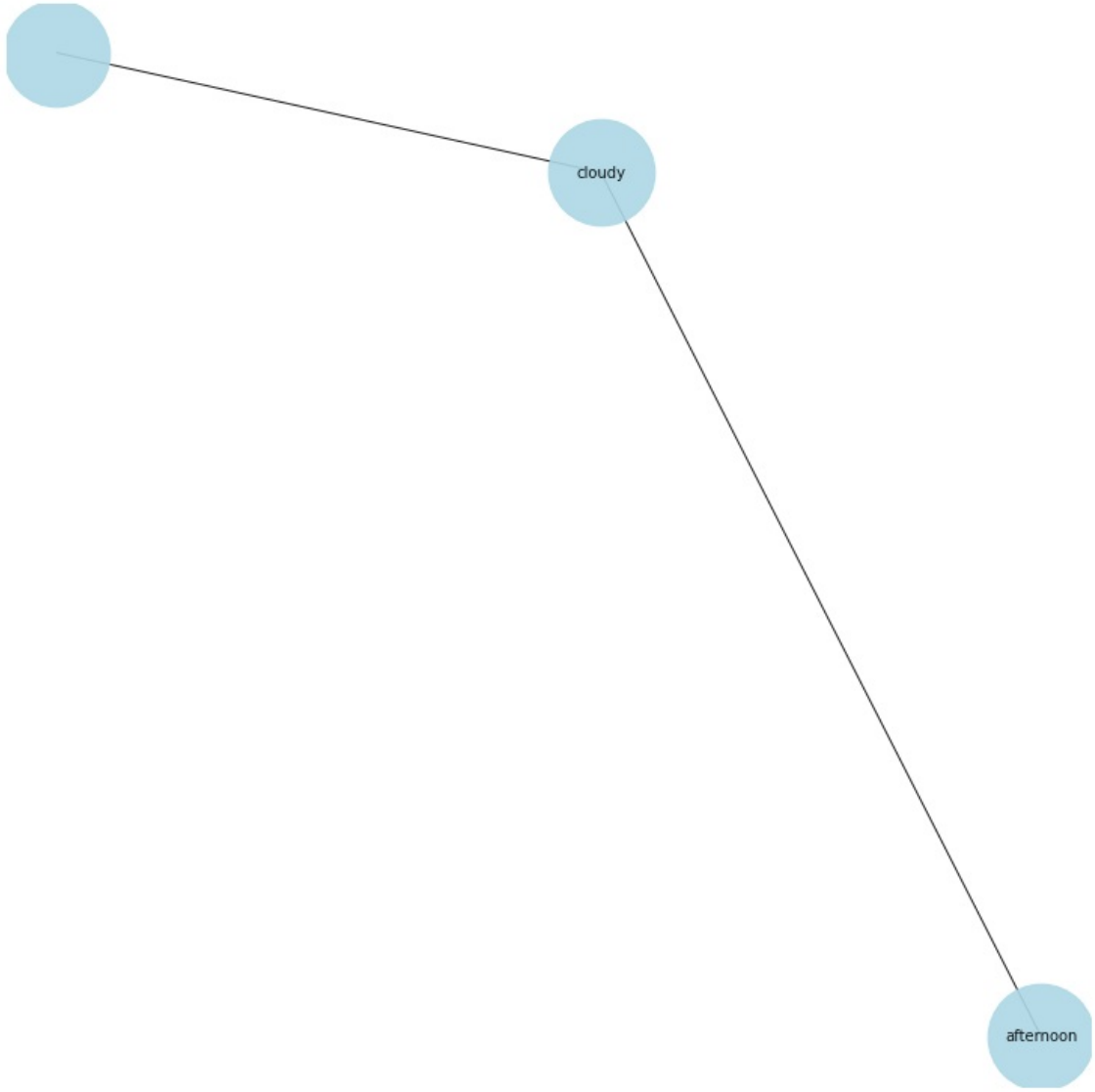

```
Foggy starting in the morning continuing until night.
Foggy -> nsubj
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
Foggy , continue , morning night
```



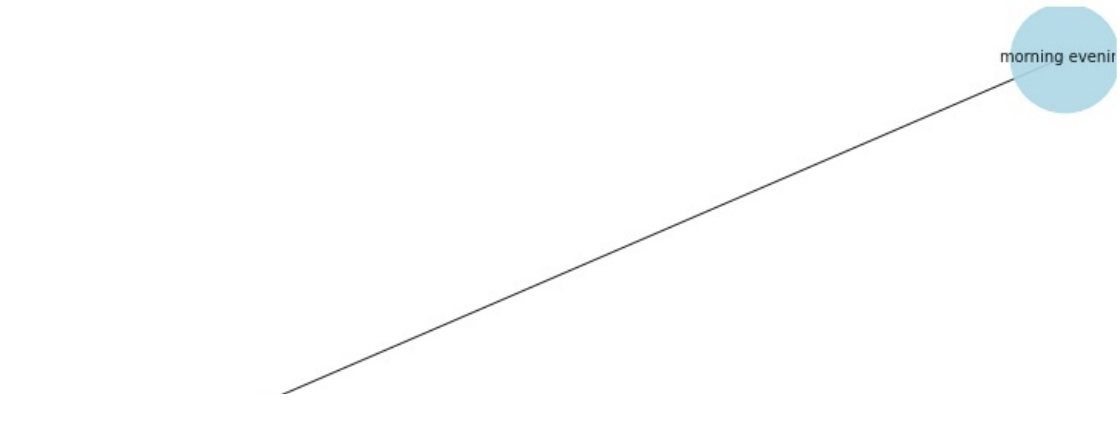

```
Partly cloudy starting overnight continuing until morning.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
morning -> pobj
. -> punct
starting , cloudy continue , morning
```

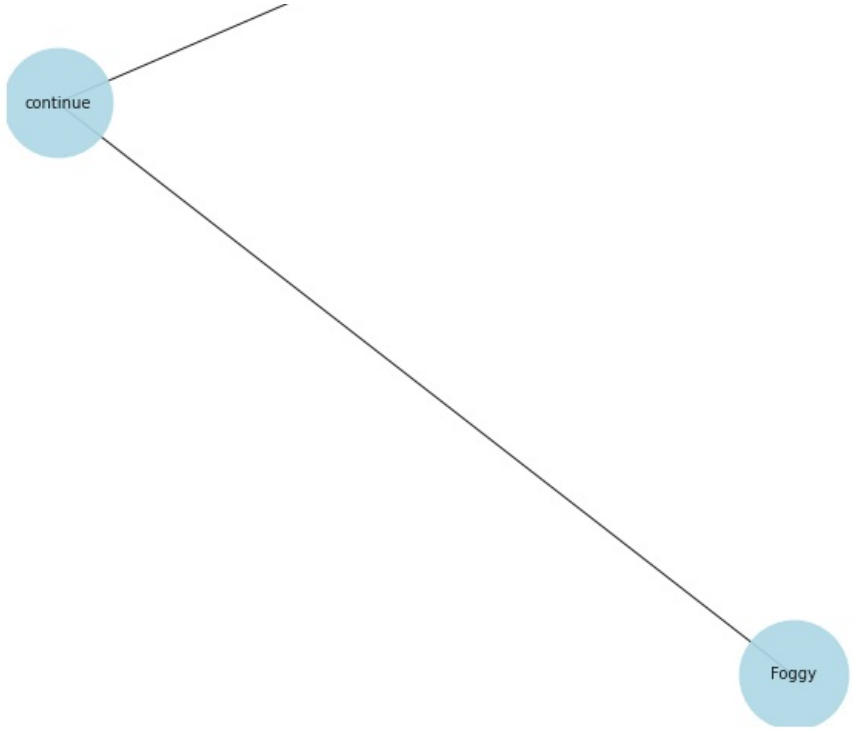



```
Partly cloudy in the afternoon.
Partly -> advmod
cloudy -> ROOT
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , afternoon
```

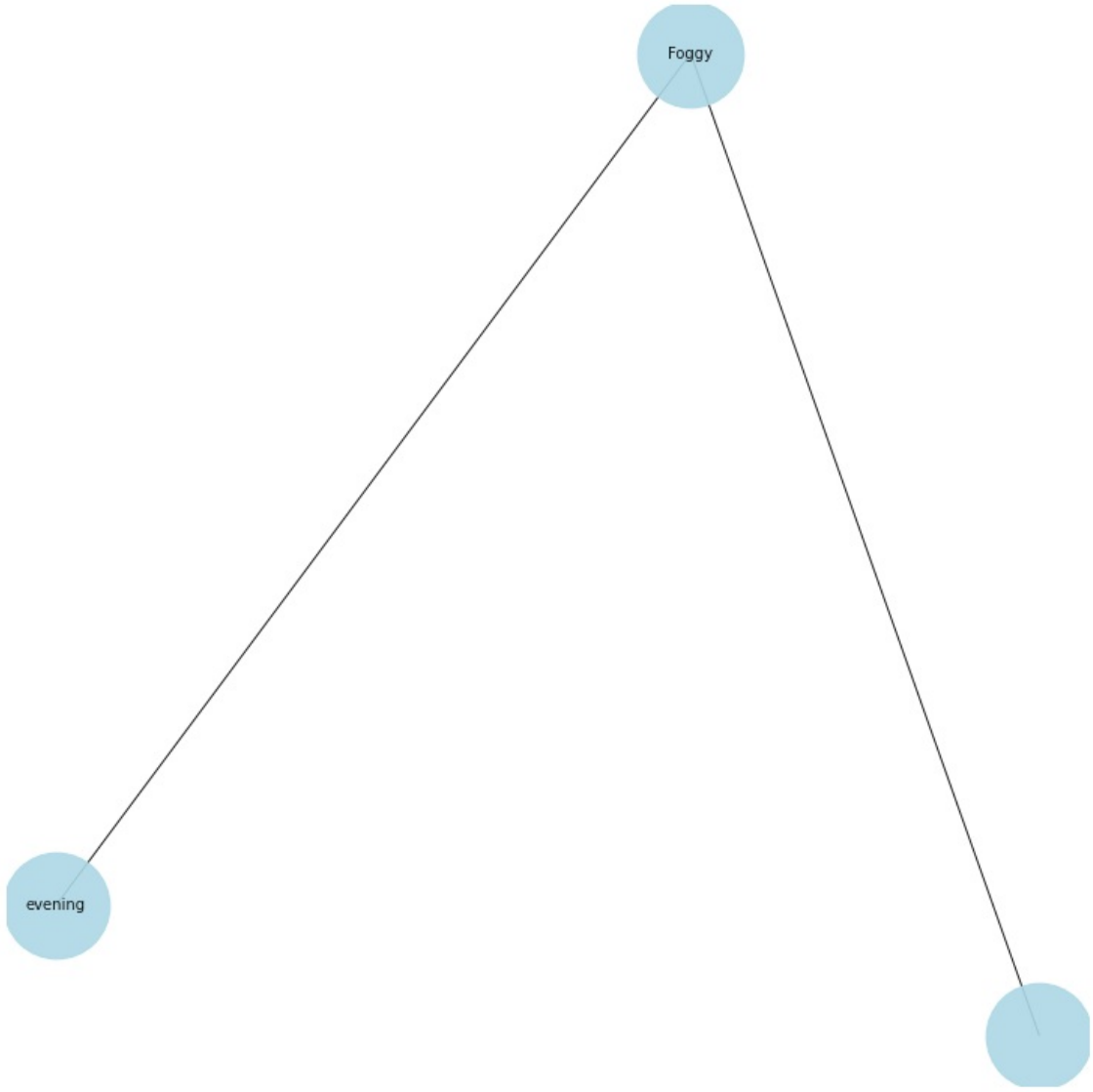



```
Foggy starting in the morning continuing until evening.
Foggy -> nsubj
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
Foggy , continue , morning evening
```



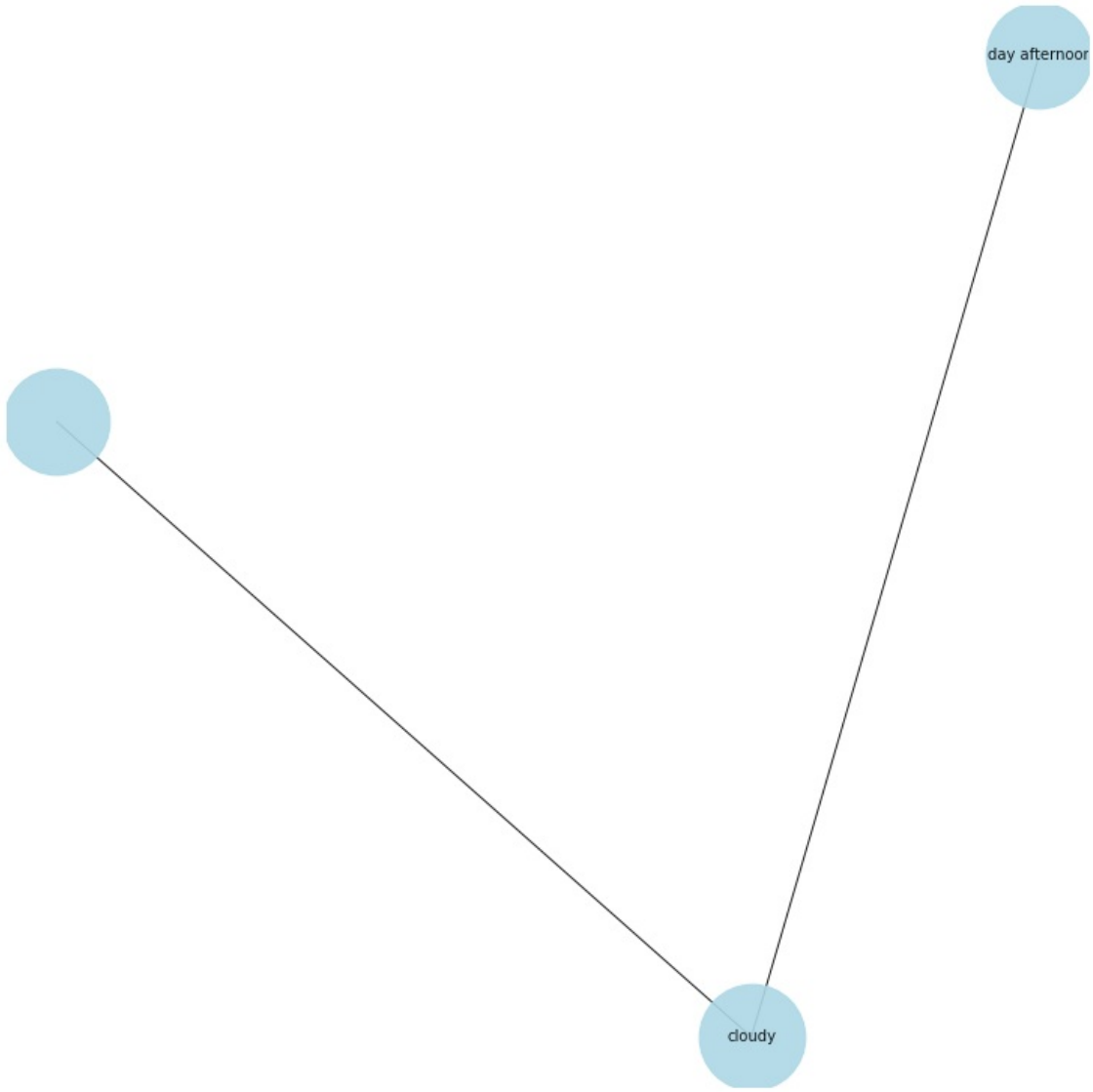



```
Foggy in the evening.
Foggy -> ROOT
in -> prep
the -> det
evening -> pobj
. -> punct
 , Foggy , evening
```

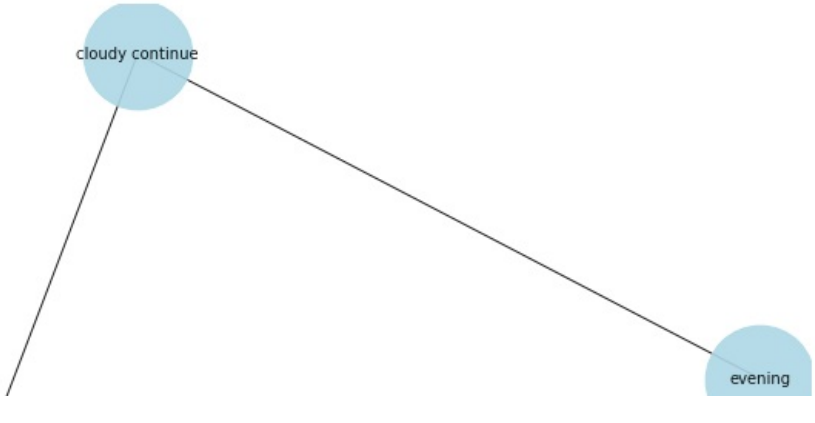



```
Mostly cloudy throughout the day and breezy in the afternoon.
Mostly -> advmod
```

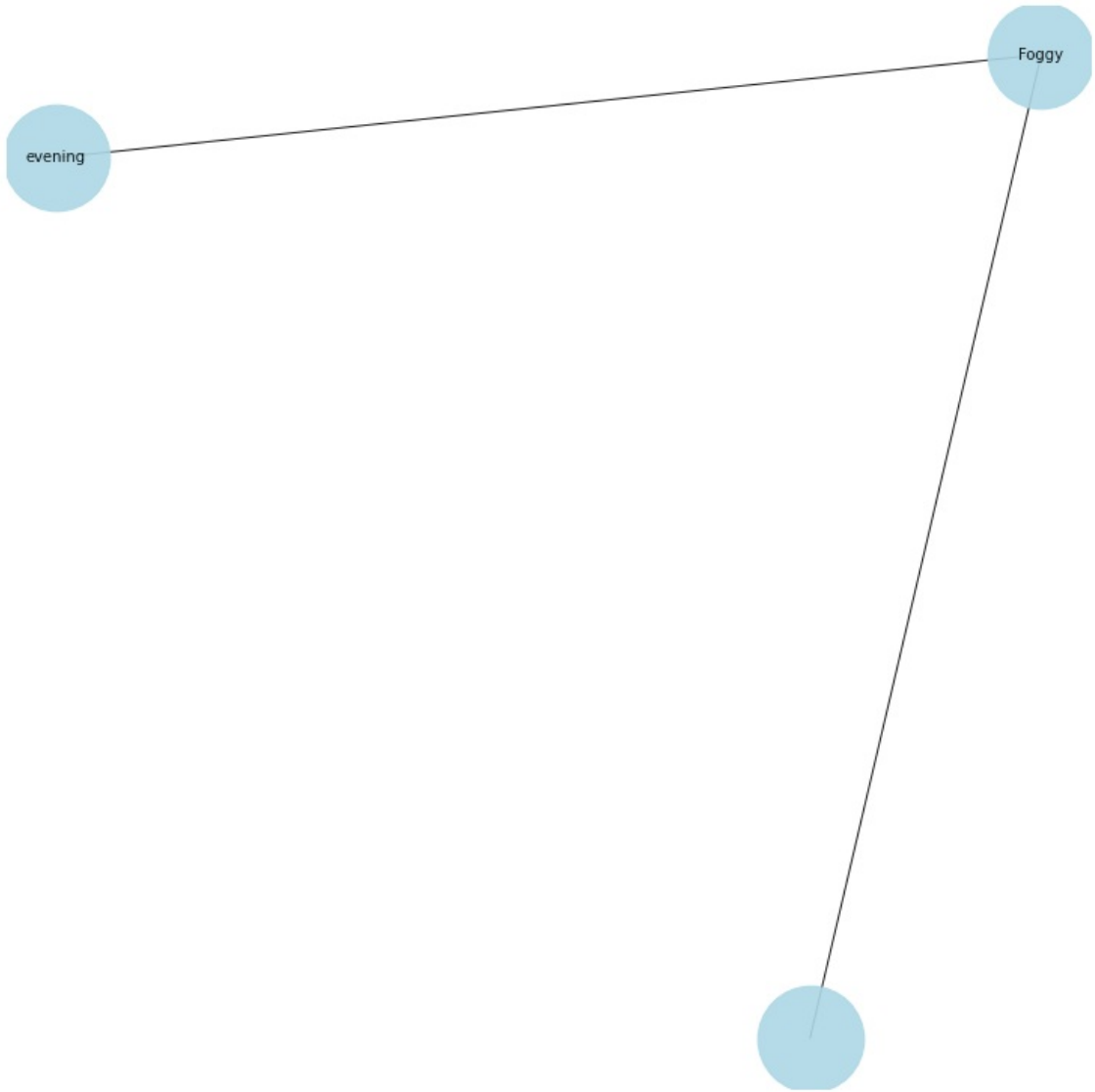
```
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , day afternoon
```




```
Mostly cloudy starting overnight continuing until evening.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
starting , cloudy continue , evening
```

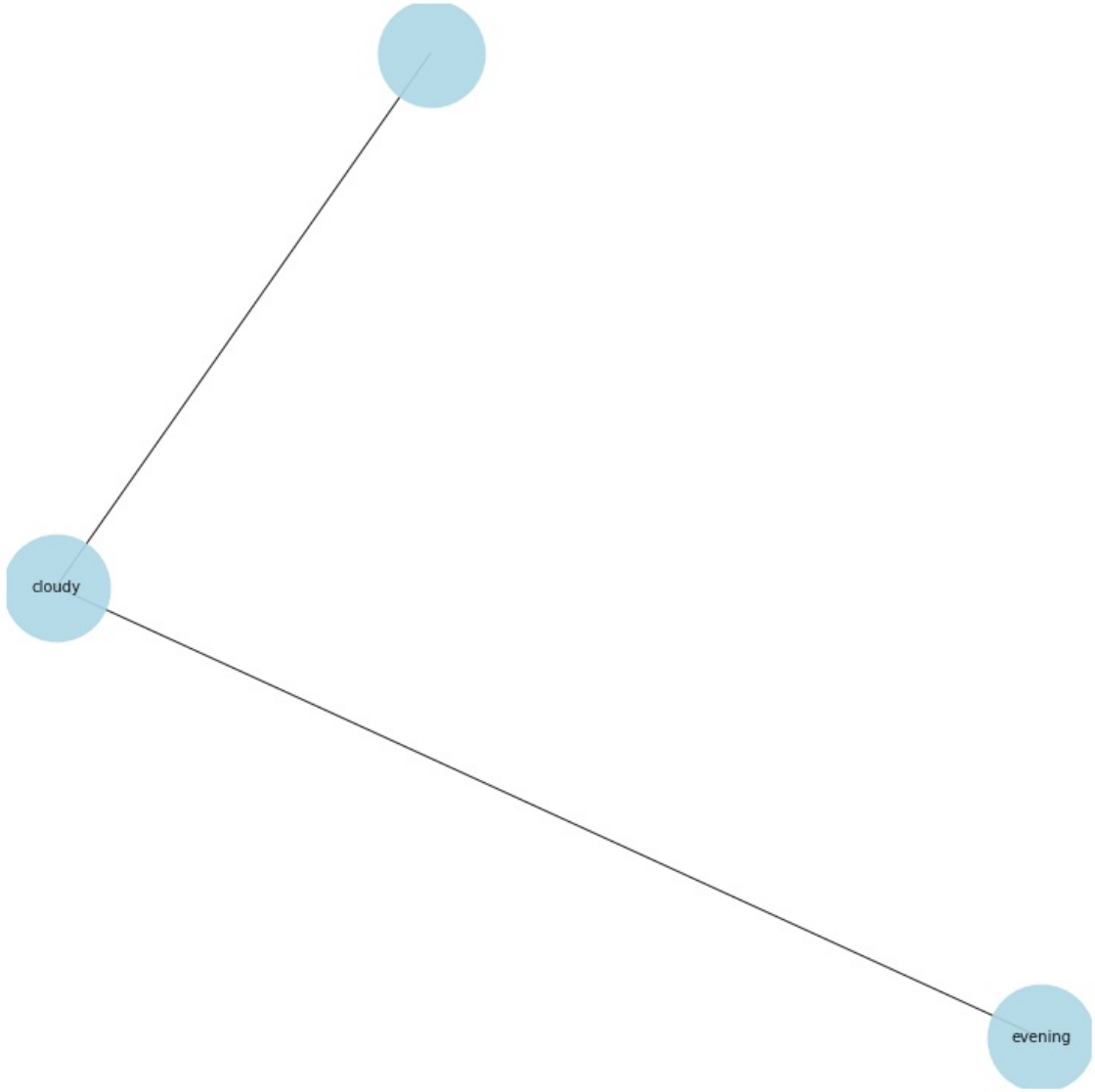



```
Foggy until evening.
Foggy -> ROOT
until -> prep
evening -> pobj
. -> punct
 , Foggy , evening
```

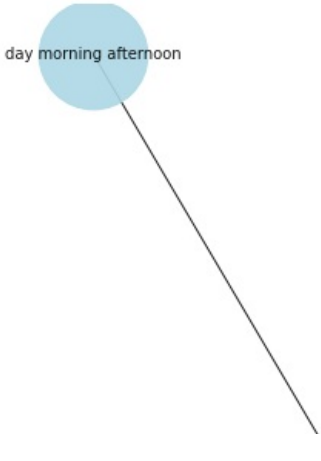


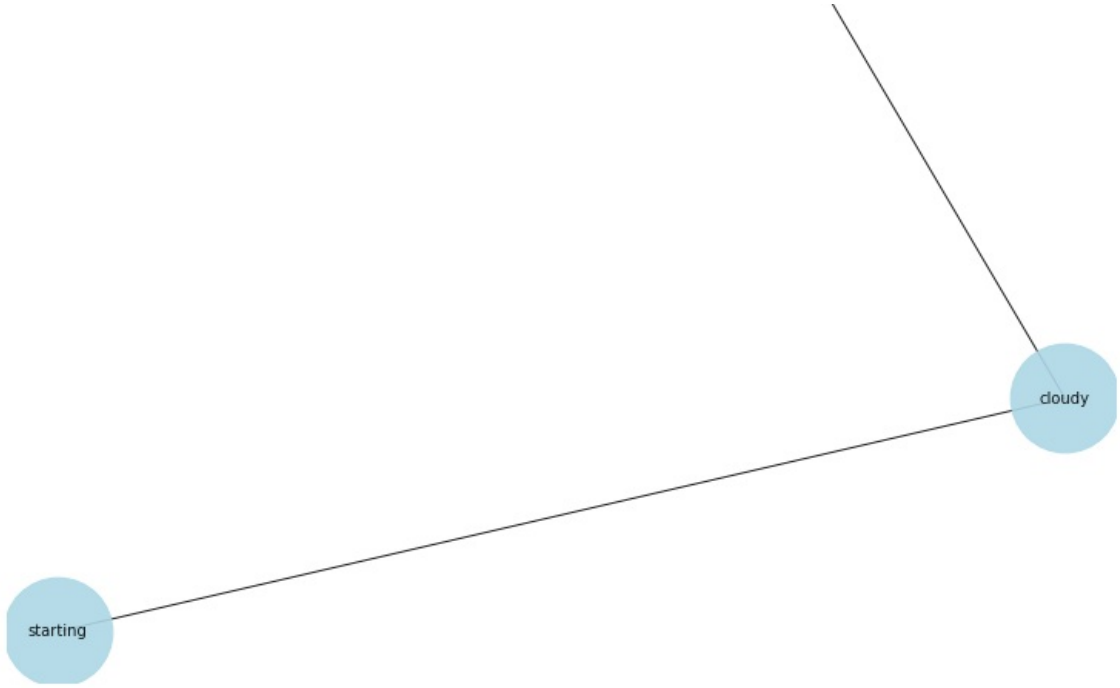

```
Partly cloudy in the evening.
Partly -> advmod
cloudy -> ROOT
```

```
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , evening
```

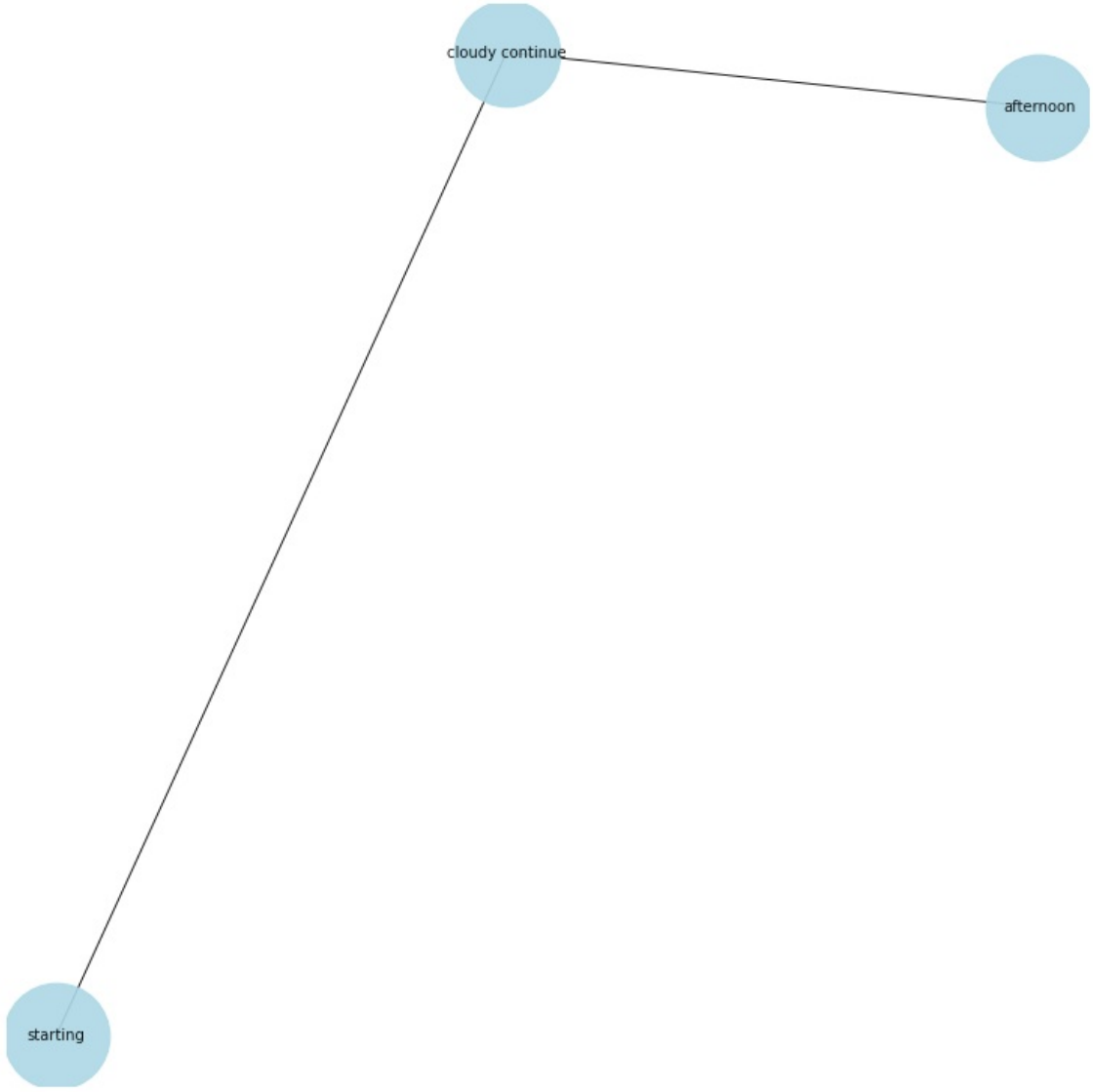



```
Partly cloudy throughout the day and breezy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy , day morning afternoon
```



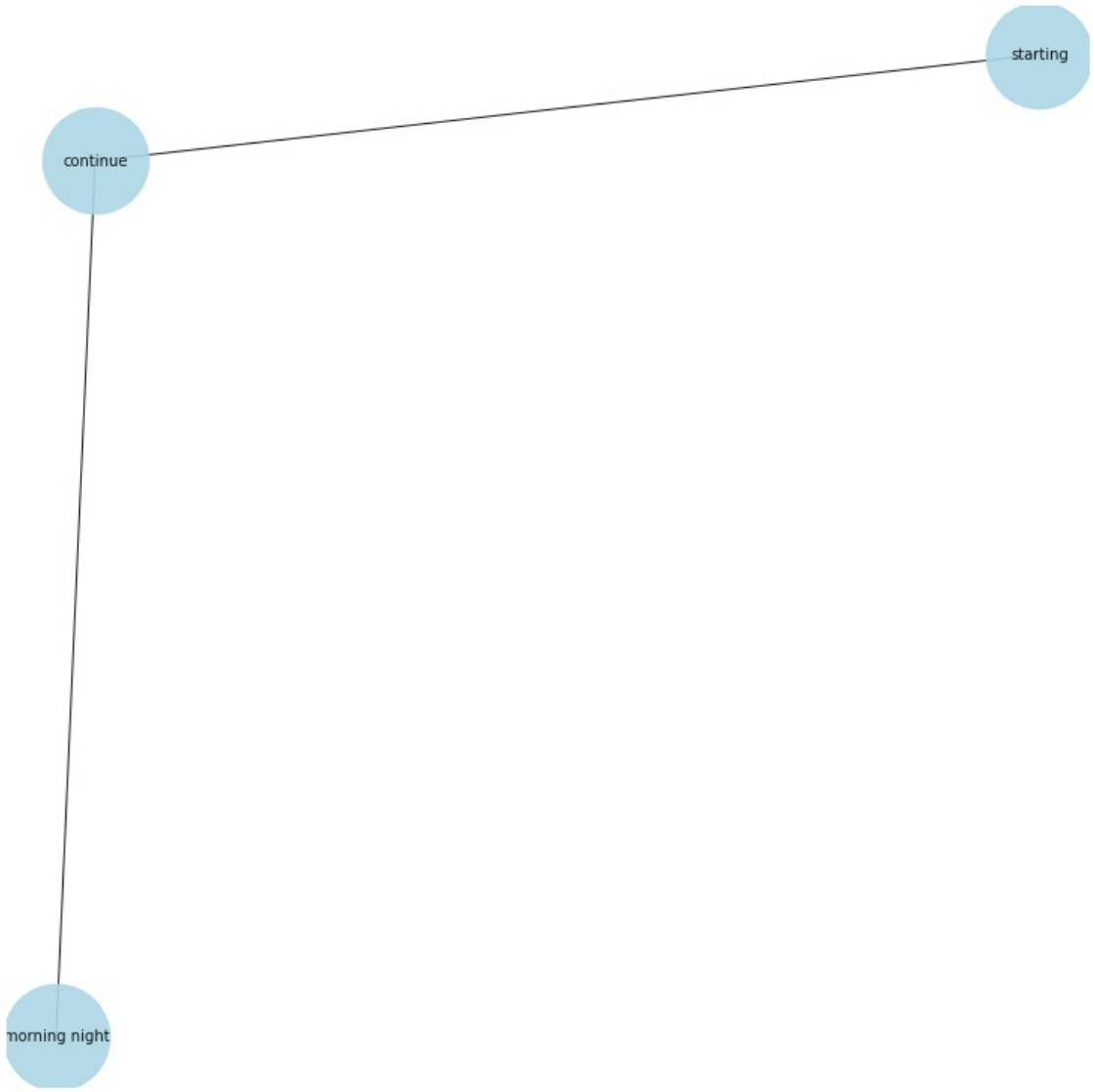



```
Partly cloudy starting overnight continuing until afternoon.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy continue , afternoon
```

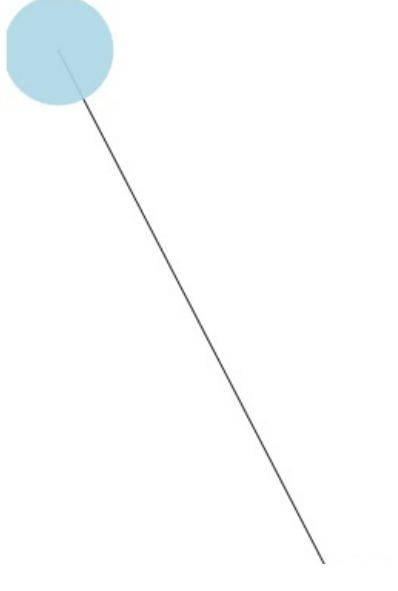


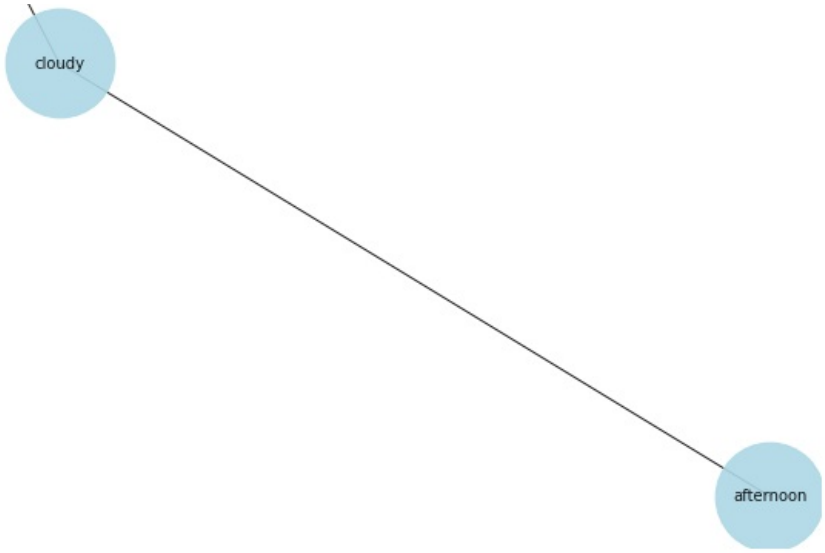

```
Mostly cloudy starting in the morning continuing until night.
Mostly -> advmod
cloudy -> advmod
```

```
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
starting , continue , morning night
```

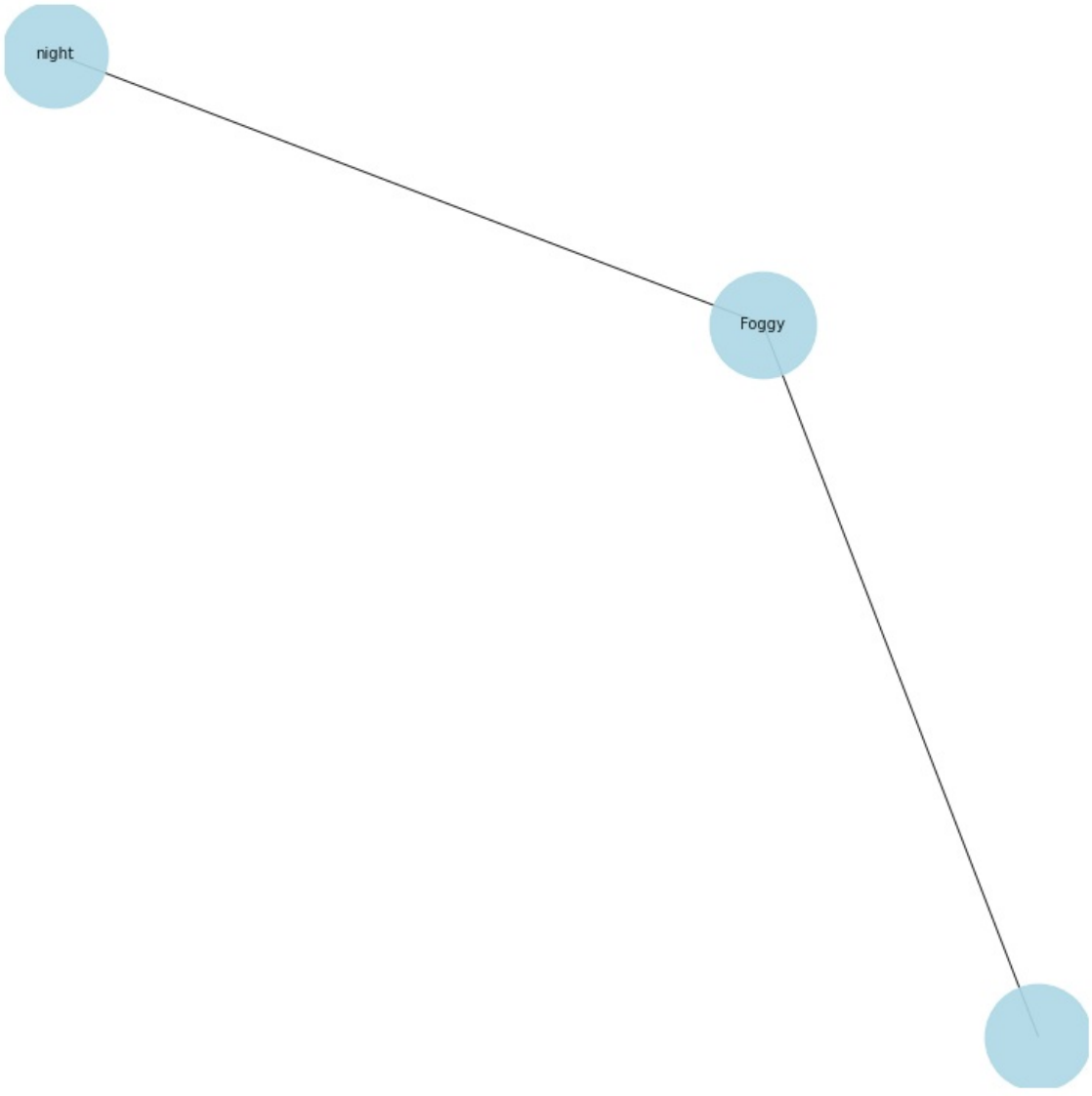



```
Mostly cloudy until afternoon.
Mostly -> advmod
cloudy -> ROOT
until -> prep
afternoon -> pobj
. -> punct
 , cloudy , afternoon
```

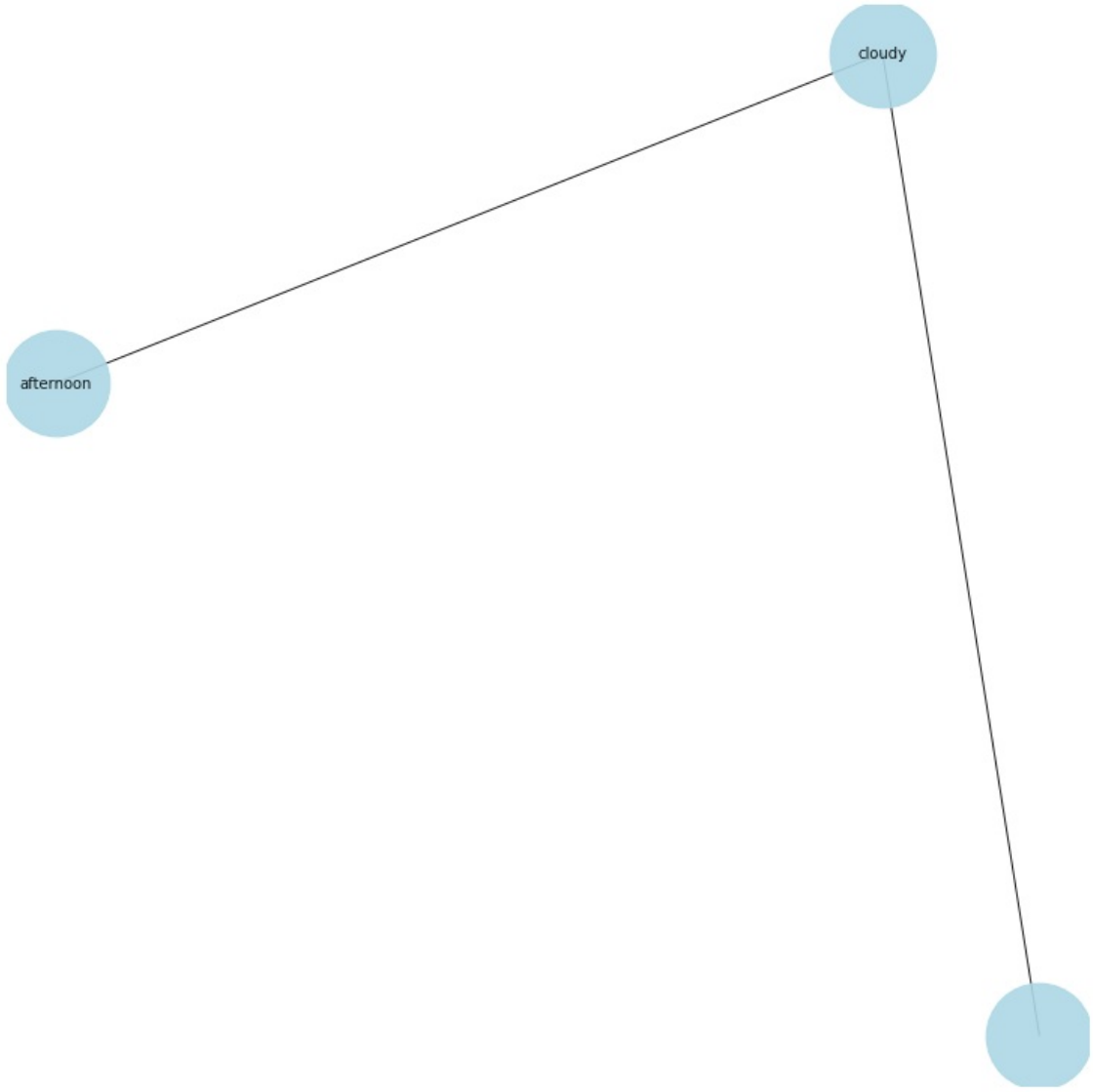




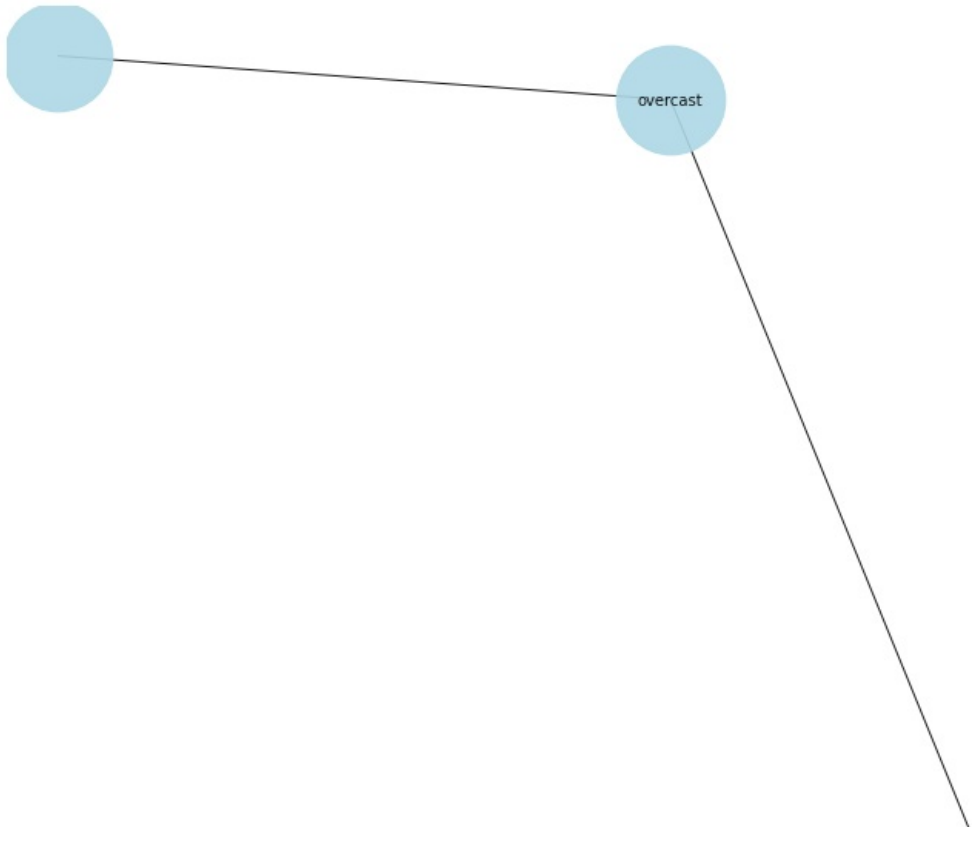

```
Foggy starting overnight continuing until night.
Foggy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
night -> pobj
. -> punct
 , Foggy , night
```



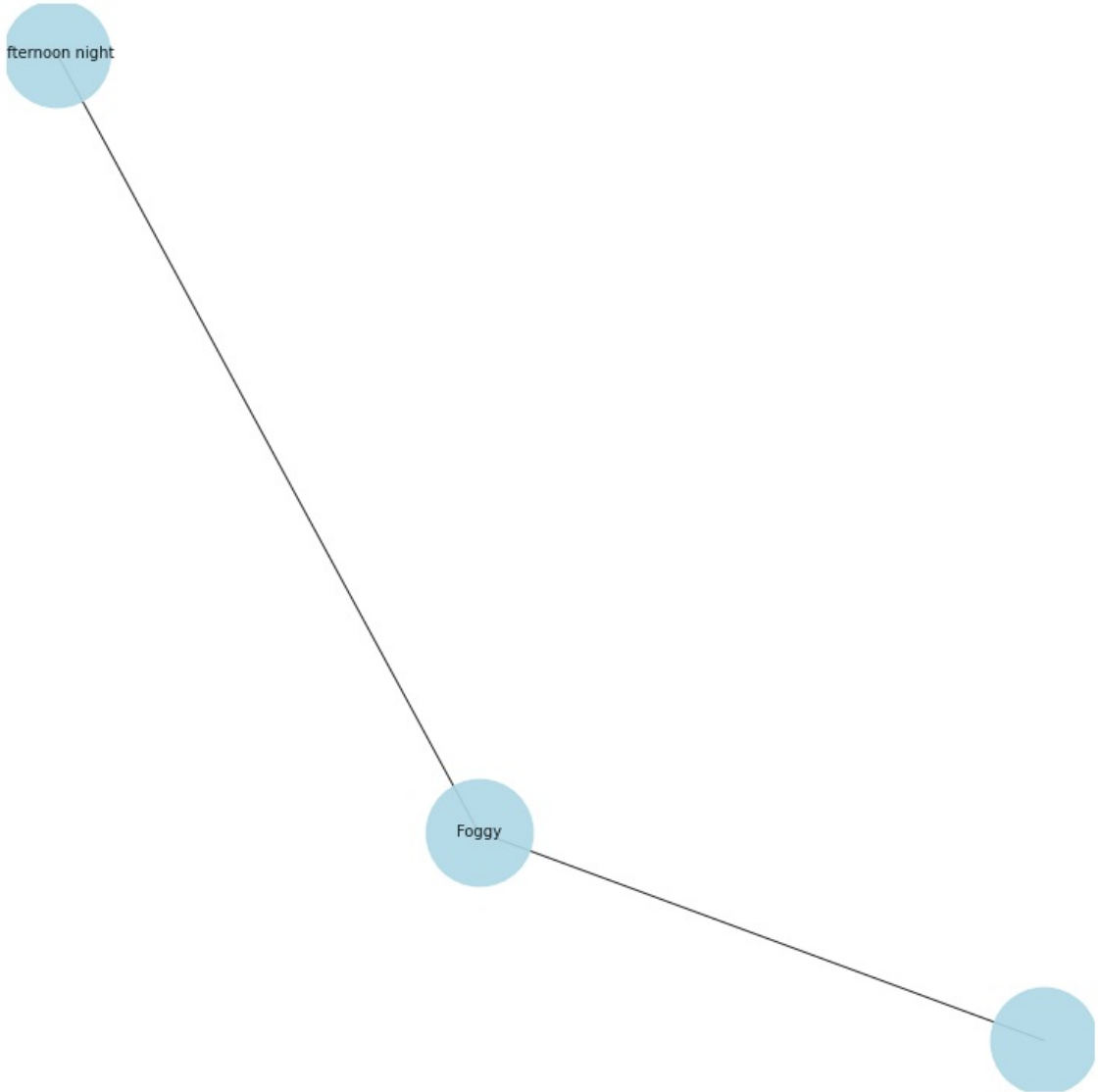

```
Partly cloudy until afternoon.
Partly -> advmod
cloudy -> ROOT
until -> prep
afternoon -> pobj
. -> punct
 , cloudy , afternoon
```



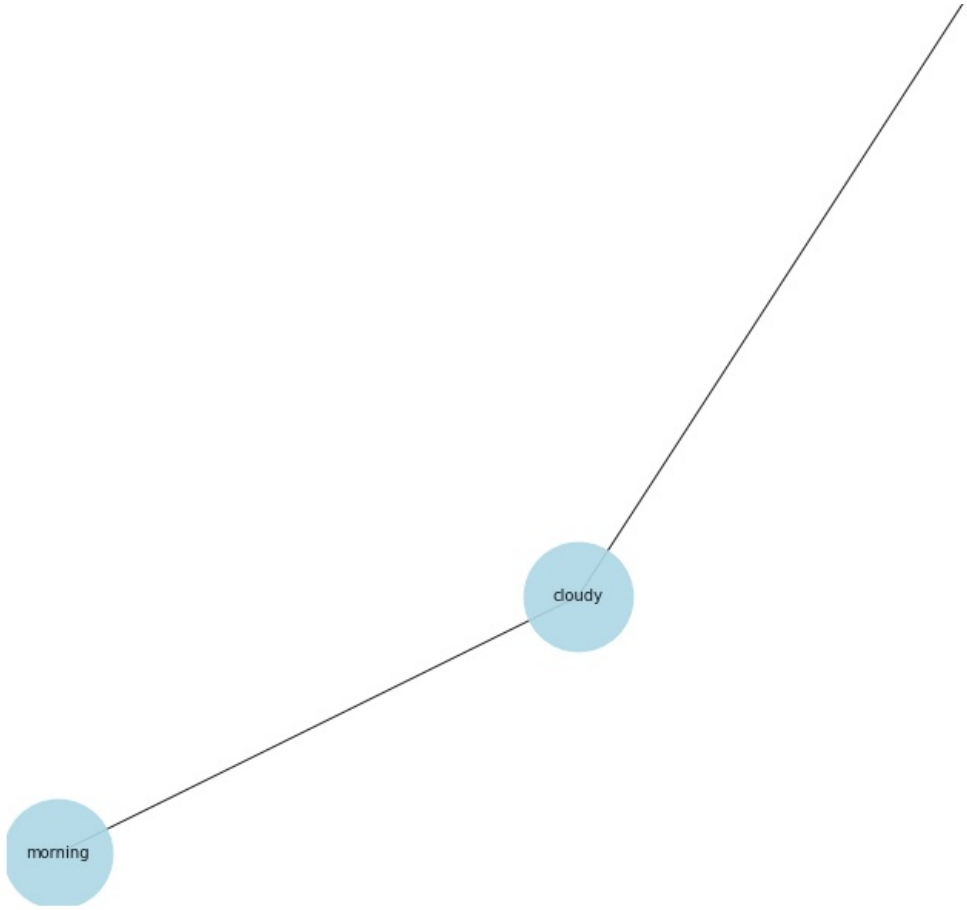

```
Overcast until night.
Overcast -> ROOT
until -> prep
night -> pobj
. -> punct
 , overcast , night
```



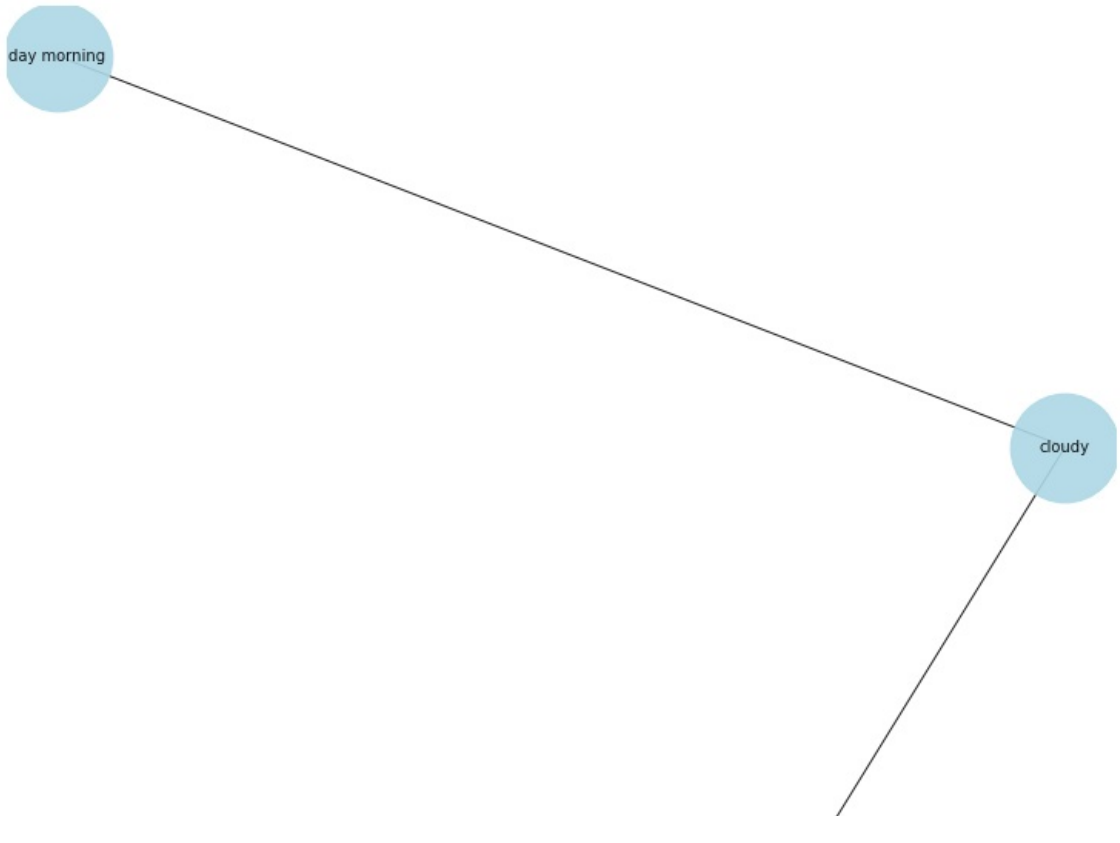

```
Foggy starting in the afternoon continuing until night.
Foggy -> ROOT
starting -> acl
in -> prep
the -> det
afternoon -> pobj
continuing -> acl
until -> prep
night -> pobj
. -> punct
 , Foggy , afternoon night
```

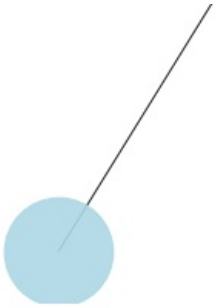



```
Mostly cloudy until morning.
Mostly -> advmod
cloudy -> ROOT
until -> prep
morning -> pobj
. -> punct
 , cloudy , morning
```

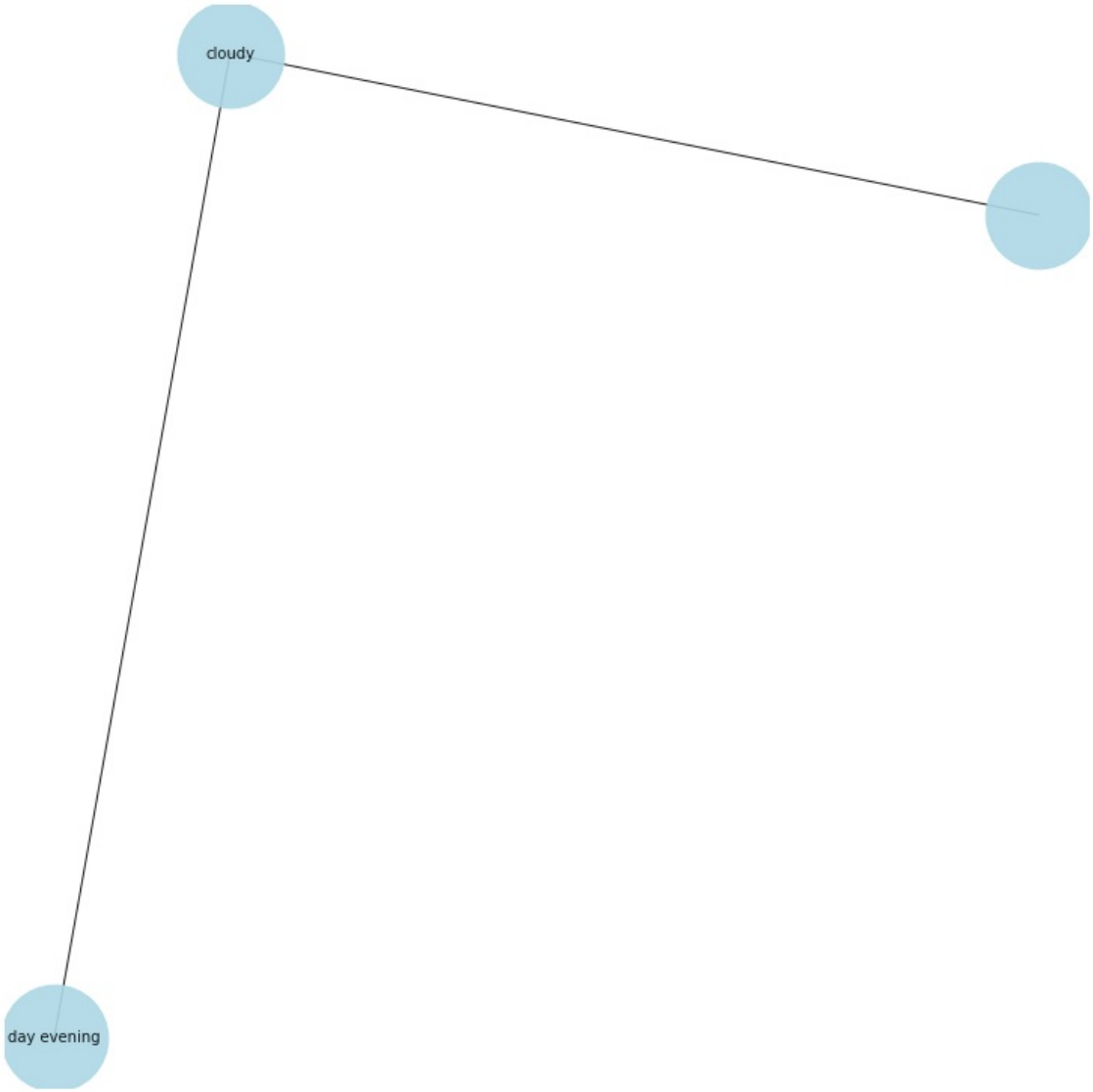



```
Mostly cloudy throughout the day and breezy starting overnight continuing until morning.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
. -> punct
 , cloudy , day morning
```

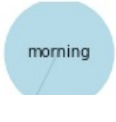


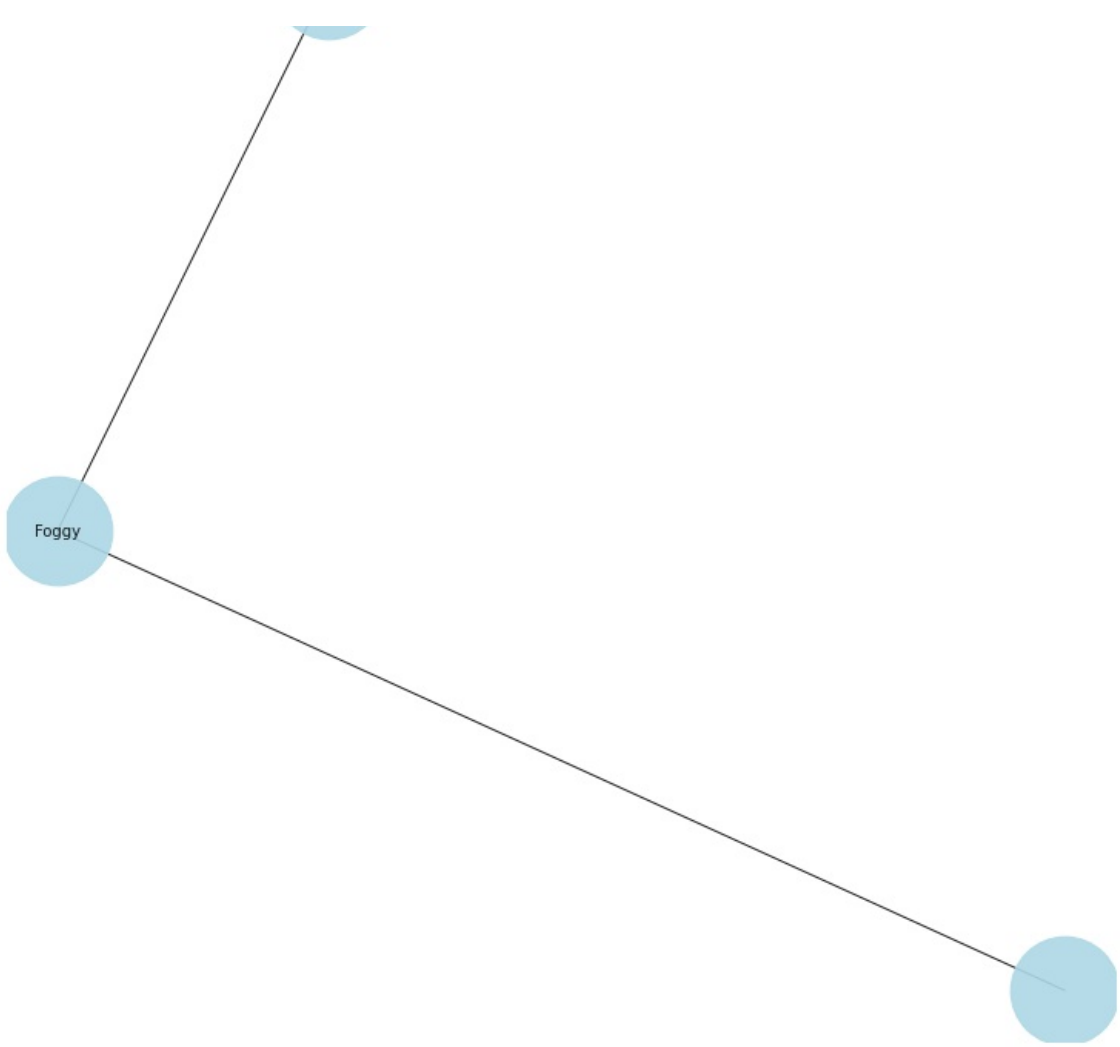


```
Mostly cloudy throughout the day and breezy starting in the evening.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , day evening
```

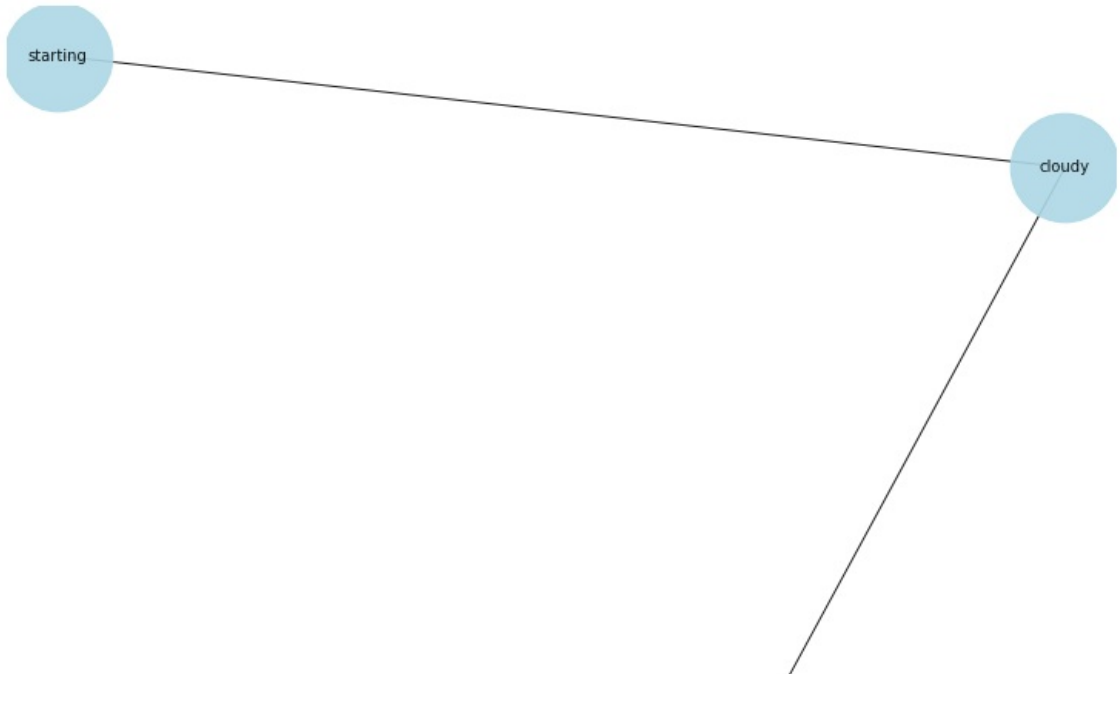


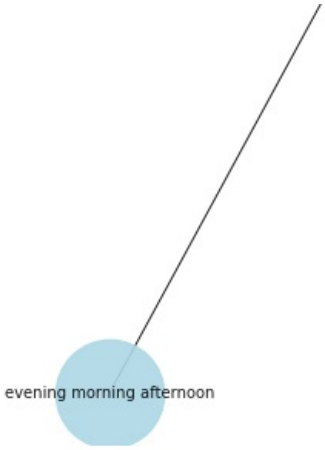

```
Foggy starting in the morning.
Foggy -> ROOT
starting -> acl
in -> prep
the -> det
morning -> pobj
. -> punct
 , Foggy , morning
```



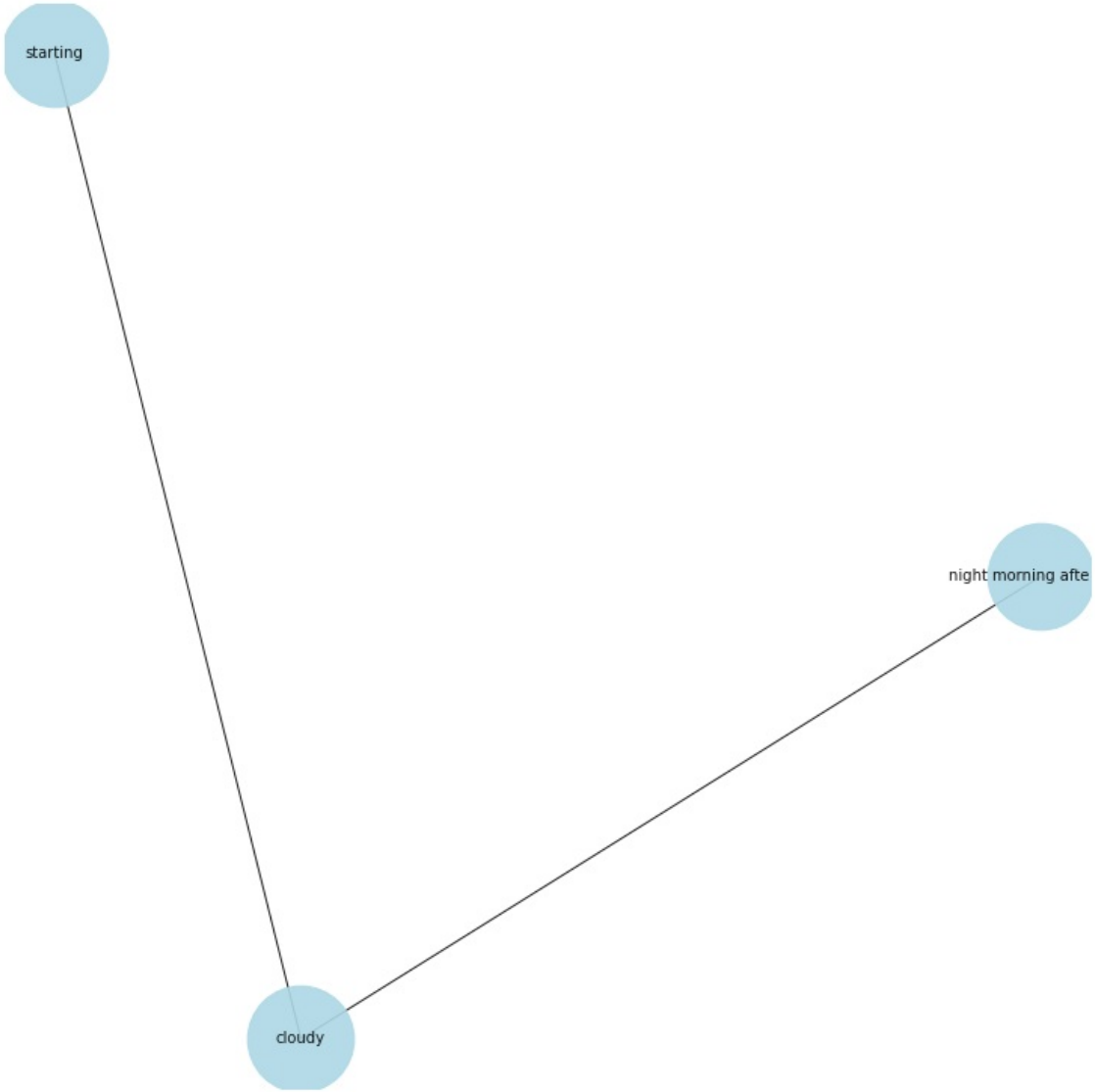



```
Mostly cloudy until evening and breezy starting in the morning continuing until afternoon.
Mostly -> advmod
cloudy -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy , evening morning afternoon
```



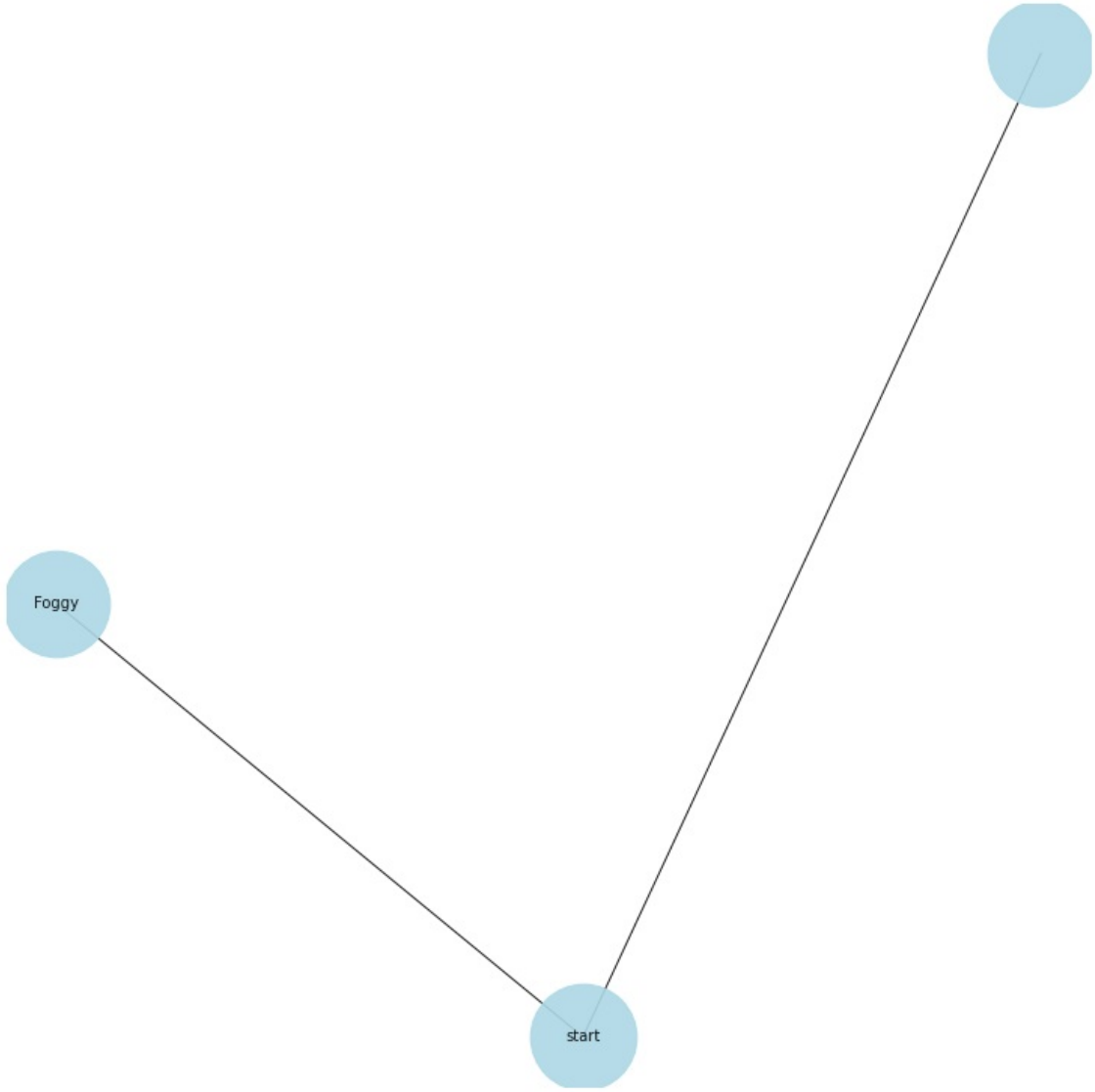



```
Partly cloudy until night and breezy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy , night morning afternoon
```

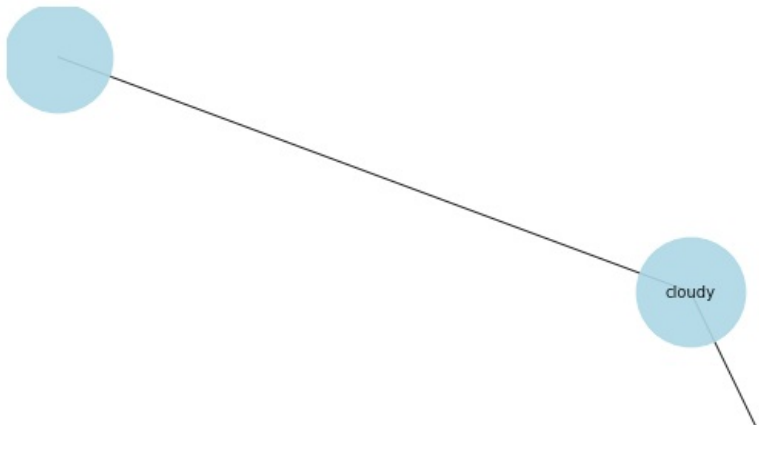



```
Foggy starting overnight.
Foggy -> nsubj
starting -> ROOT
```

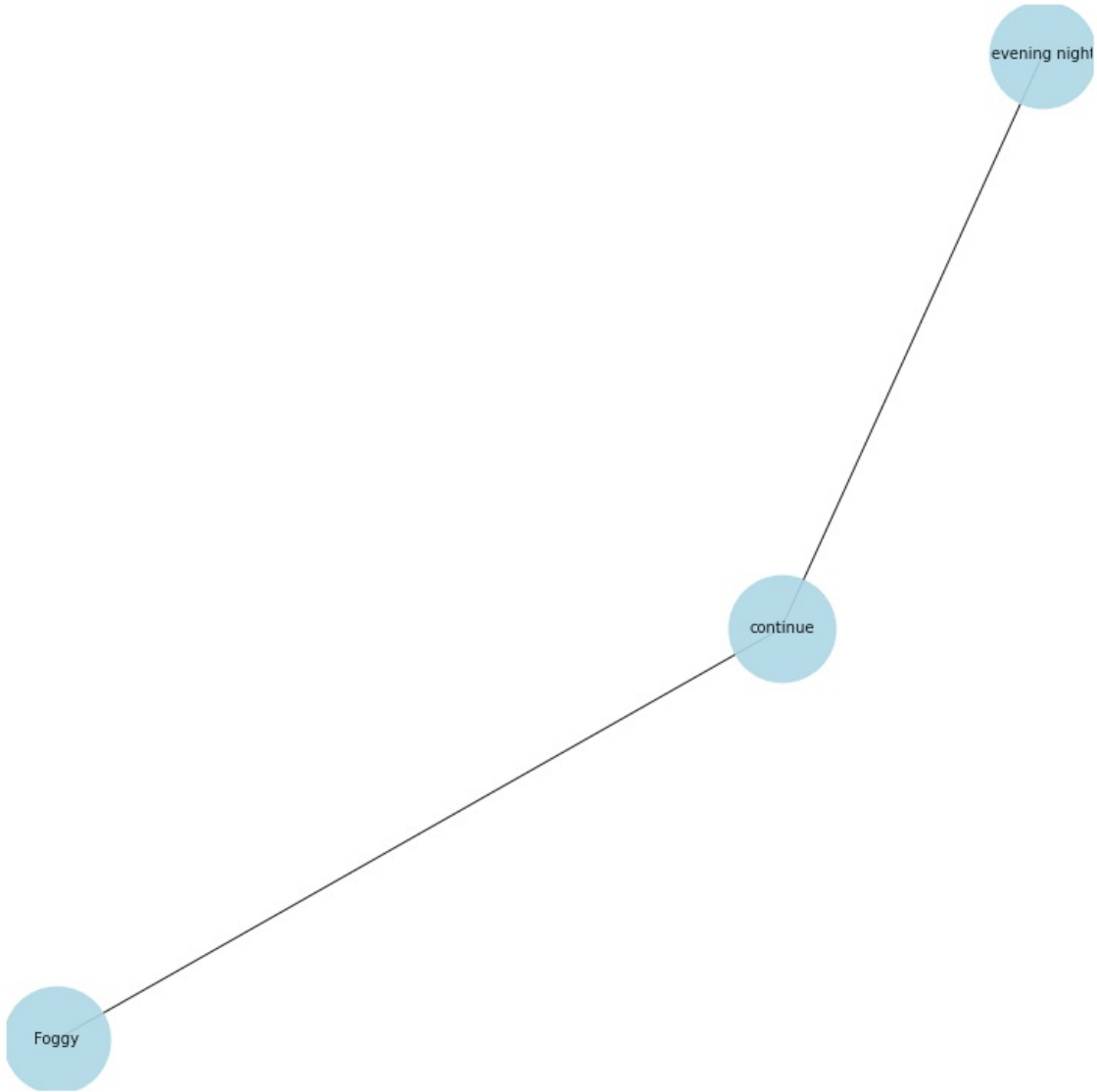
```
overnight -> advmod
. -> punct
Foggy , start ,
```




```
Mostly cloudy throughout the day and breezy starting in the afternoon continuing until evening.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
 , cloudy , day afternoon evening
```

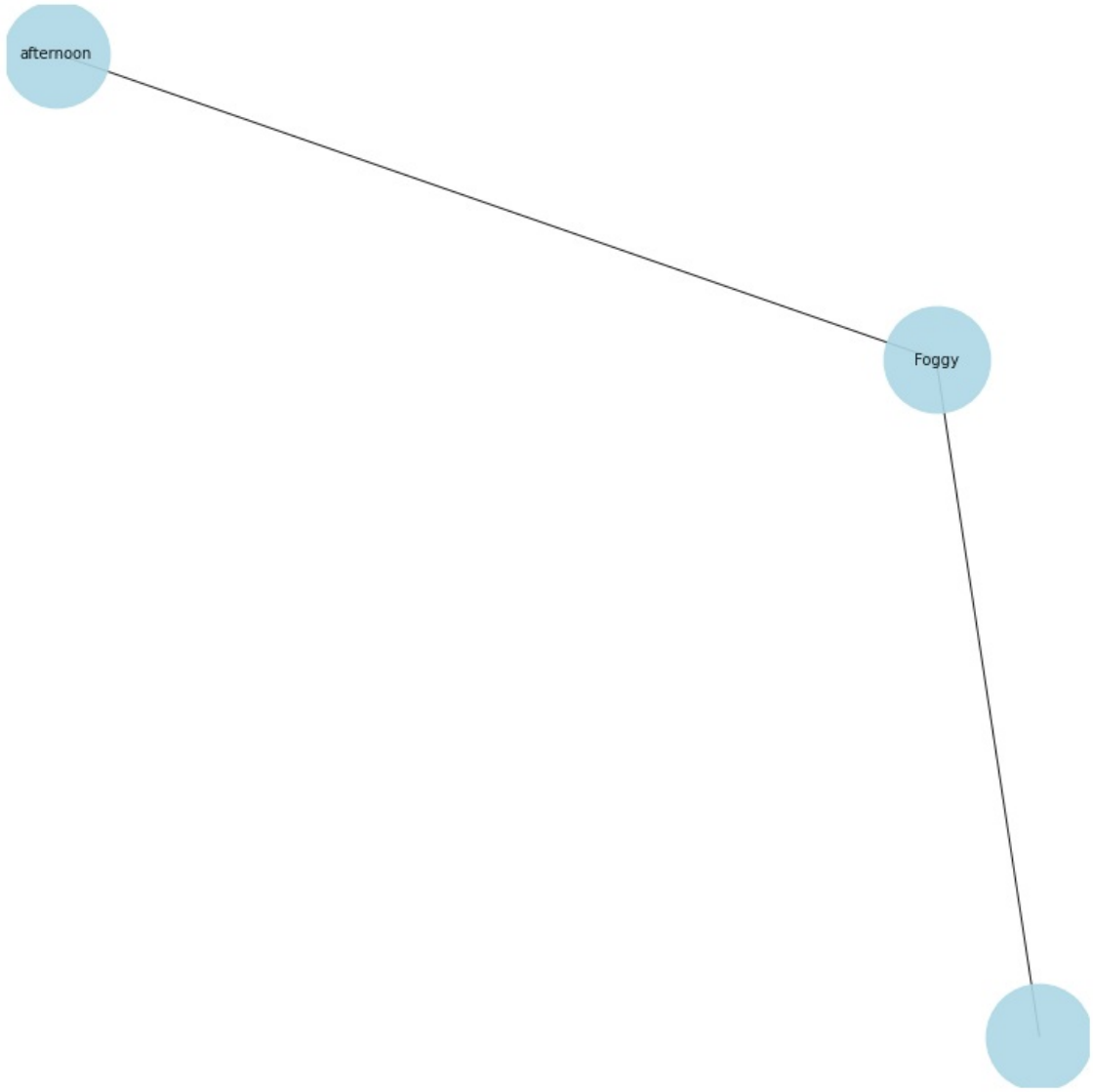



```
Foggy starting in the evening continuing until night.
Foggy -> nsubj
starting -> acl
in -> prep
the -> det
evening -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
Foggy , continue , evening night
```

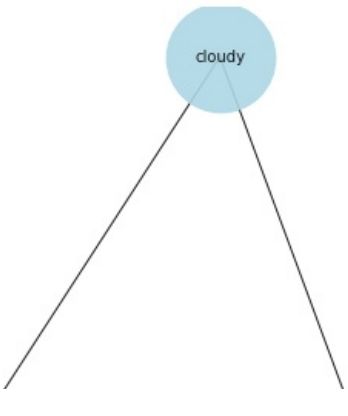


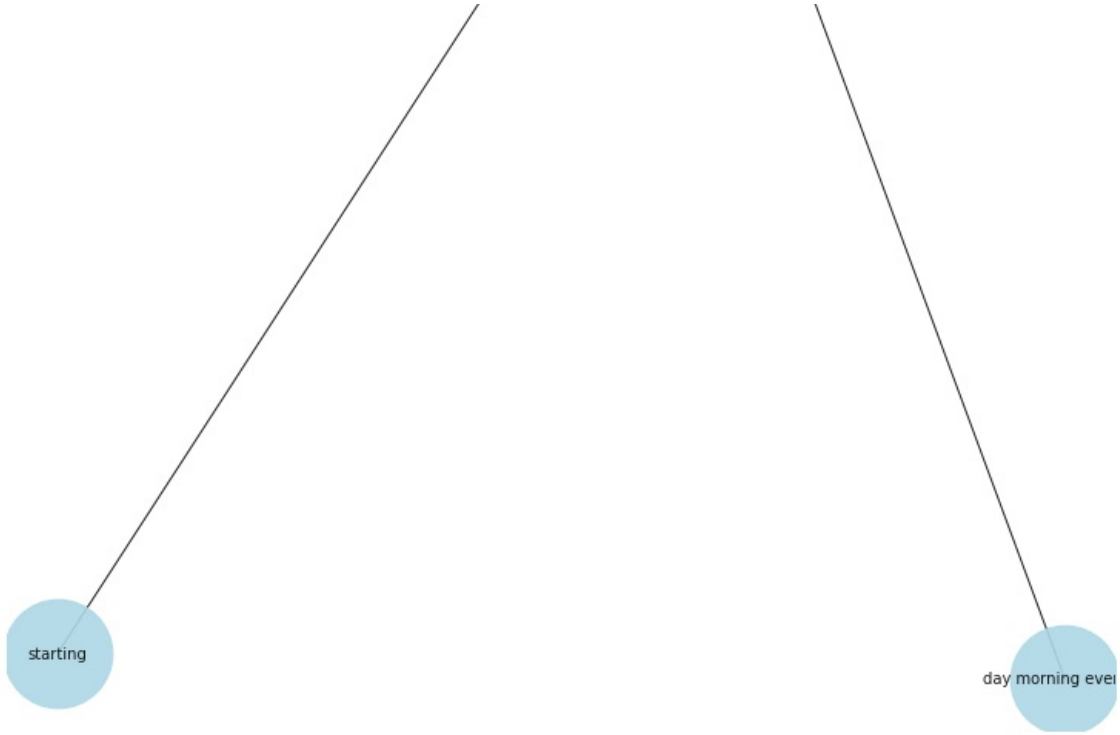

```
Foggy in the afternoon.
```

```
Foggy -> ROOT
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , Foggy , afternoon
```

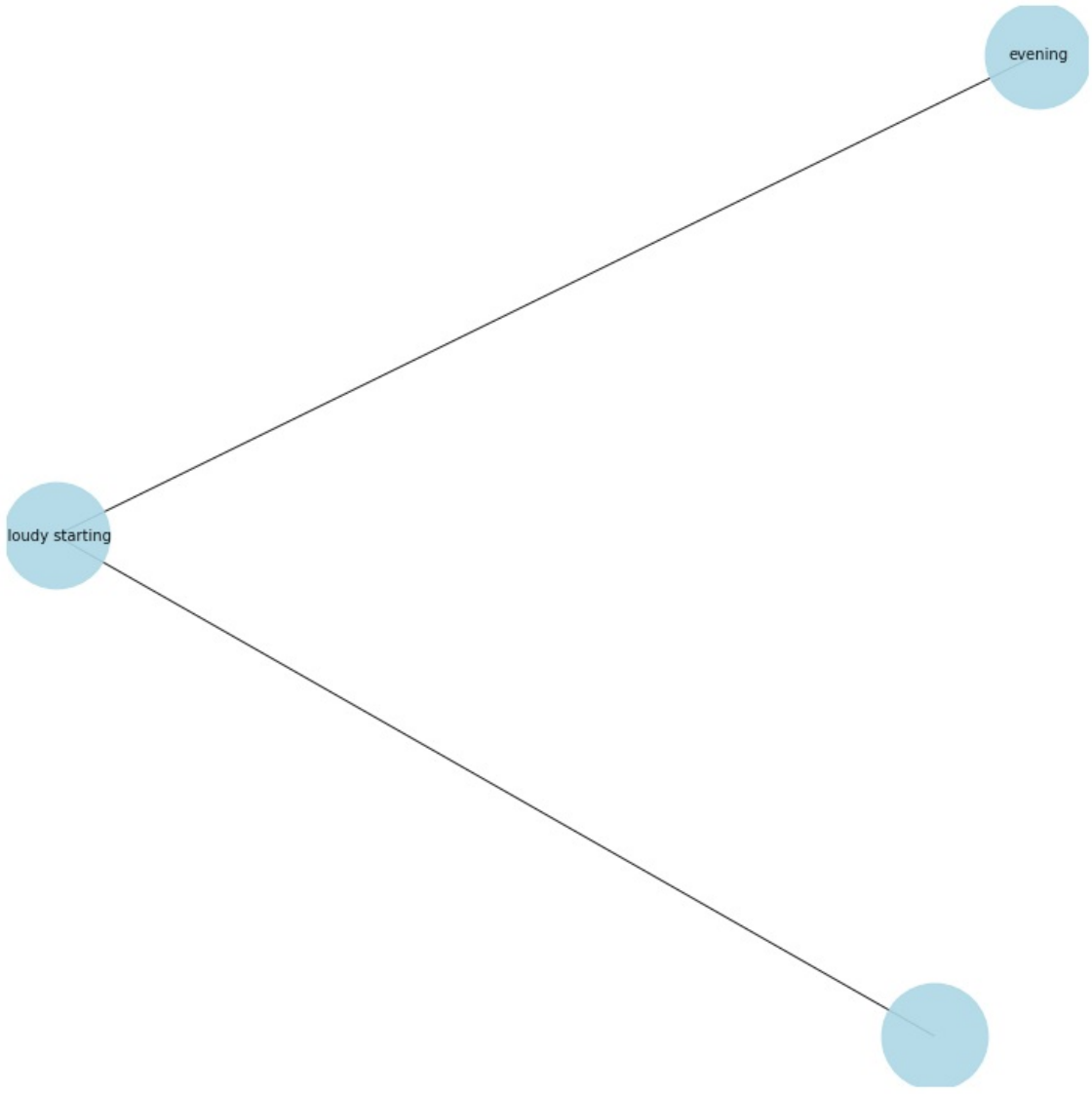



```
Mostly cloudy throughout the day and breezy starting in the morning continuing until evening.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
starting , cloudy , day morning evening
```



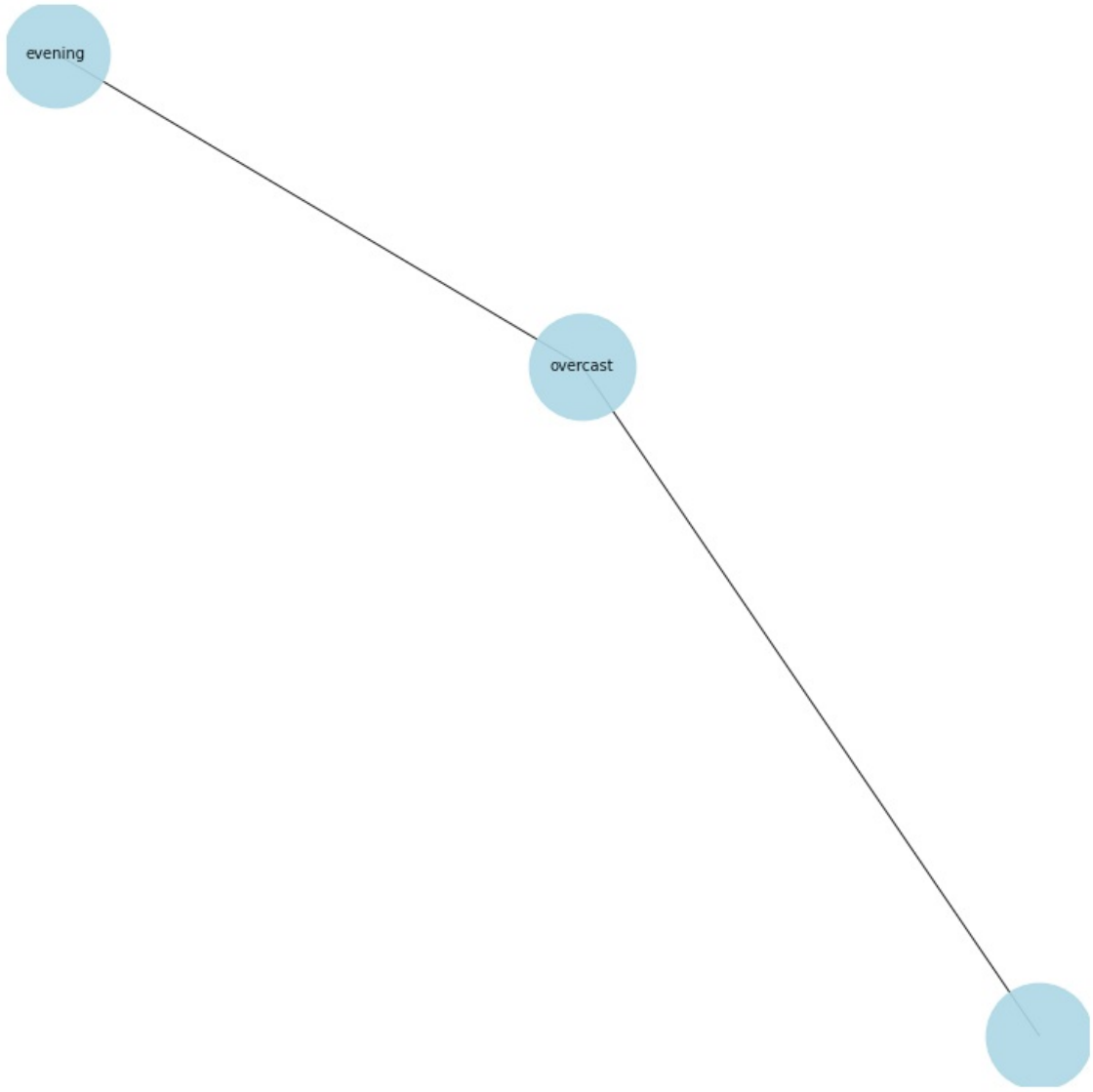



```
Partly cloudy starting in the evening.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy starting , evening
```

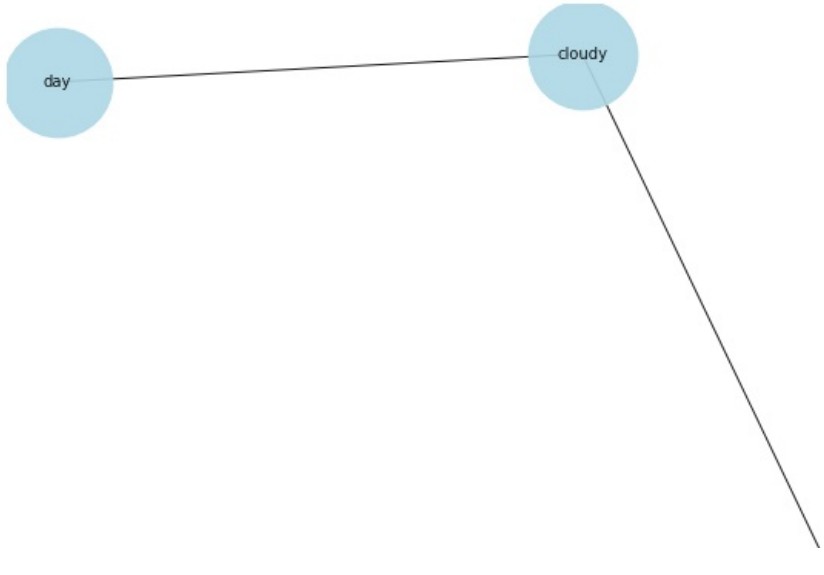


Overcast until evening.

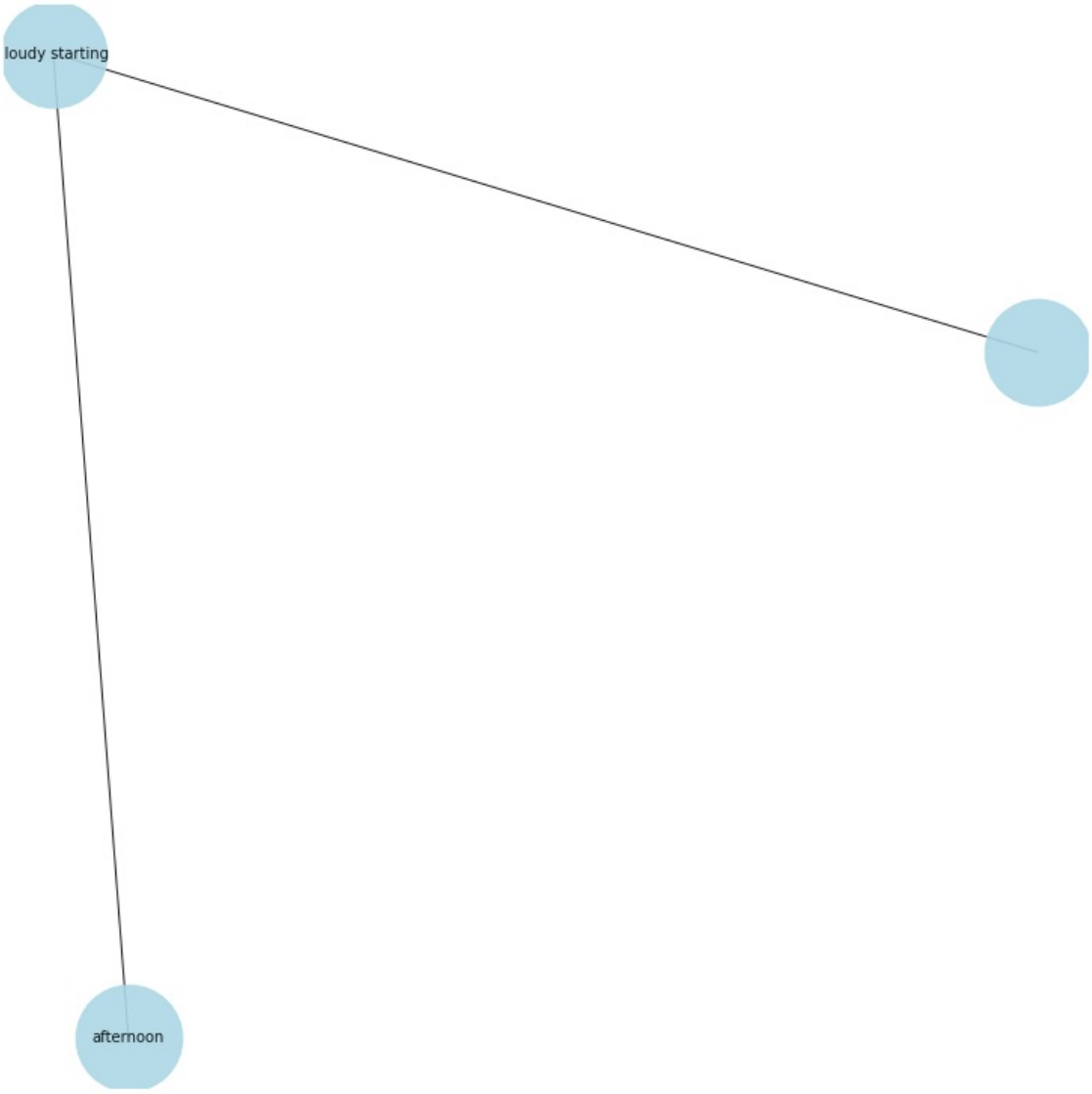
```
Overcast -> ROOT
until -> prep
evening -> pobj
. -> punct
 , overcast , evening
```




```
Mostly cloudy throughout the day and breezy overnight.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
overnight -> advmod
. -> punct
 , cloudy , day
```

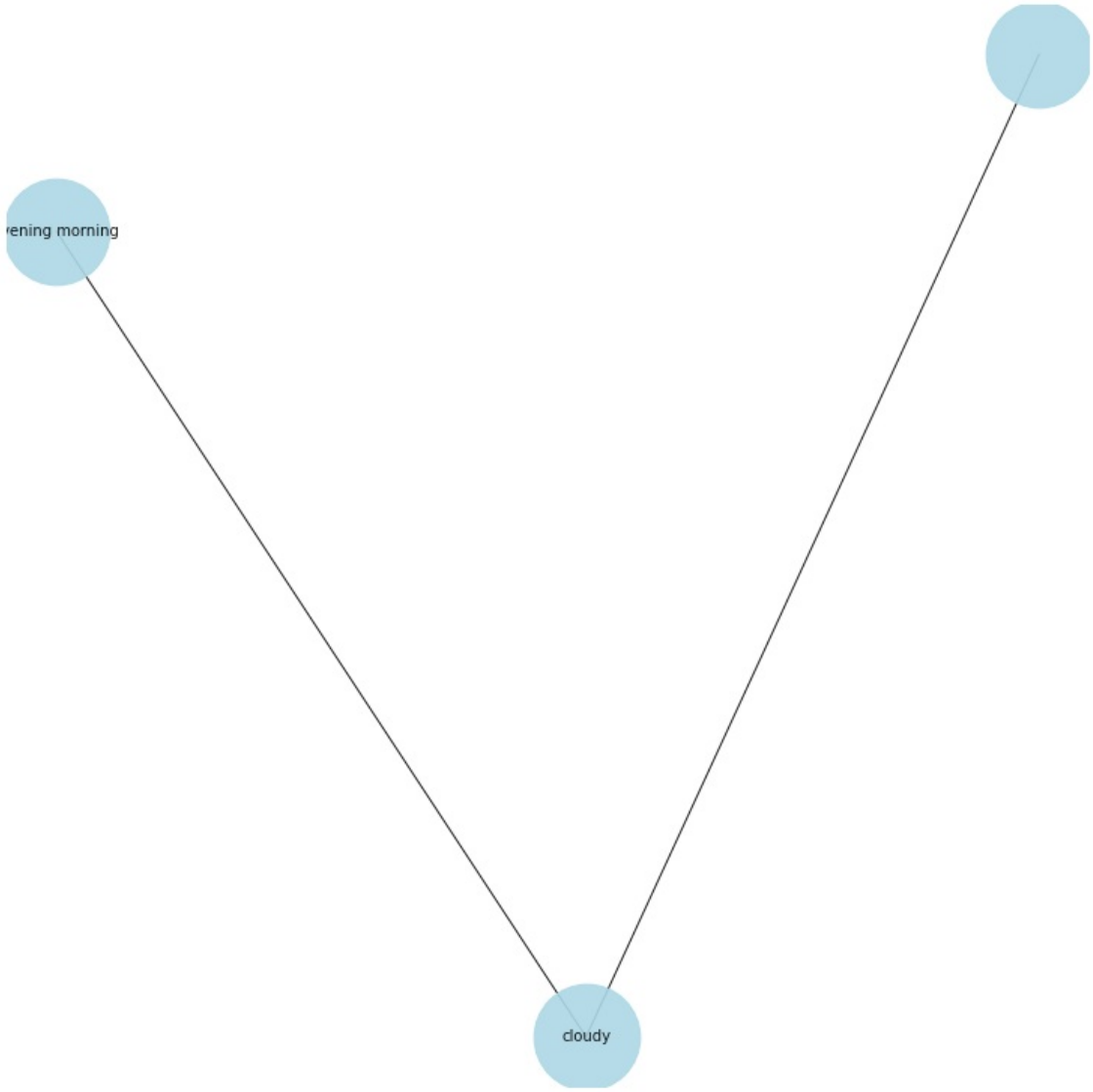


```
Mostly cloudy starting in the afternoon.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy starting , afternoon
```

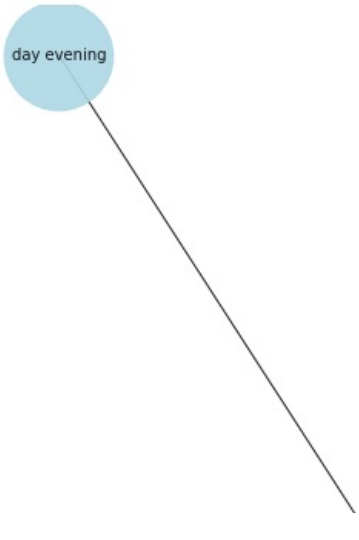


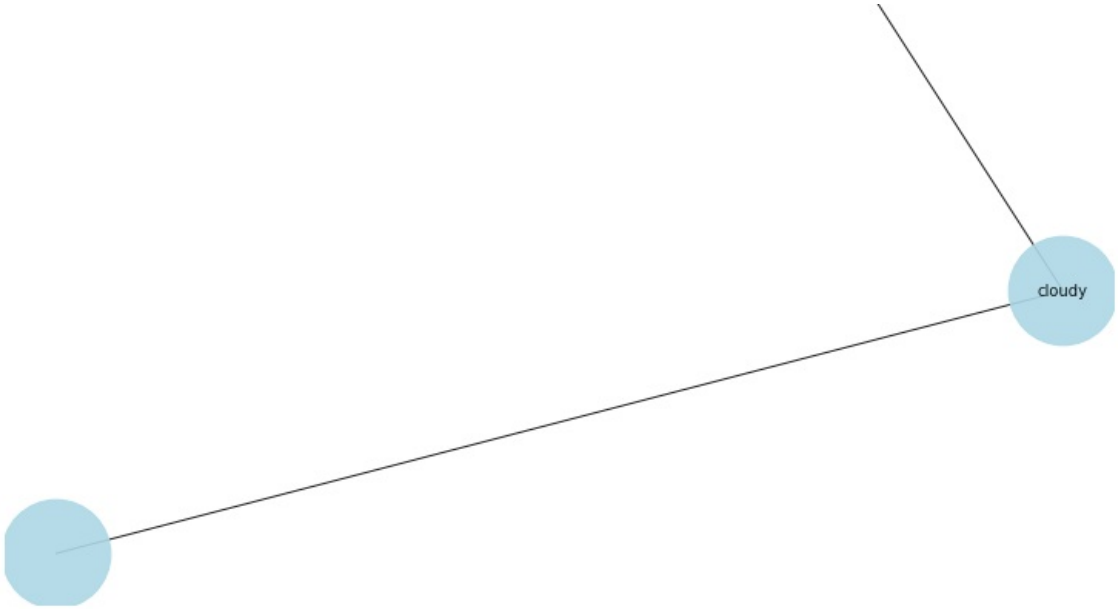

```
Partly cloudy until evening and breezy in the morning.
Partly -> advmod
cloudy -> ROOT
until -> prep
evening -> pobj
and -> cc
```

```
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , cloudy , evening morning
```

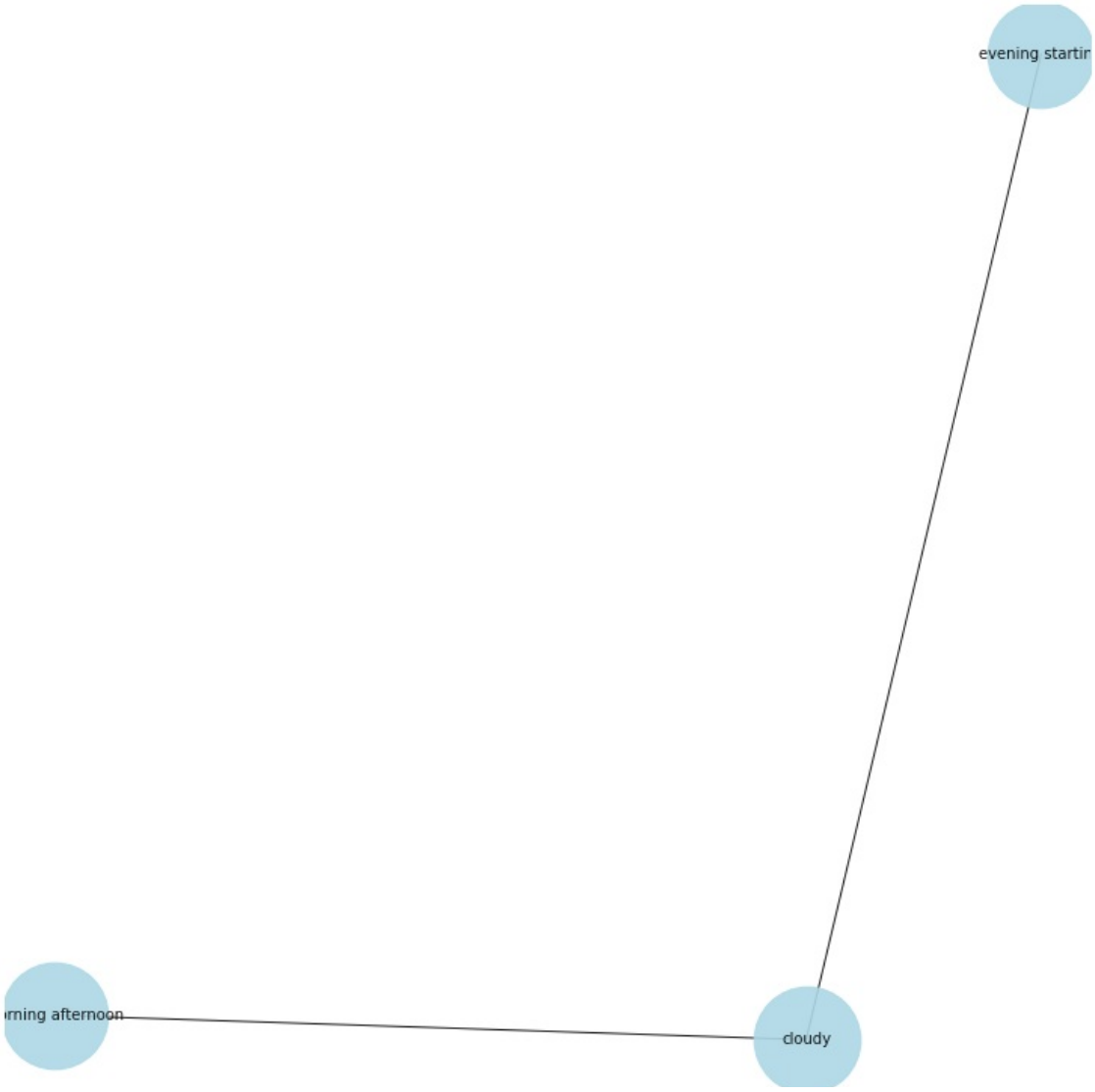



```
Partly cloudy throughout the day and breezy in the evening.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , day evening
```

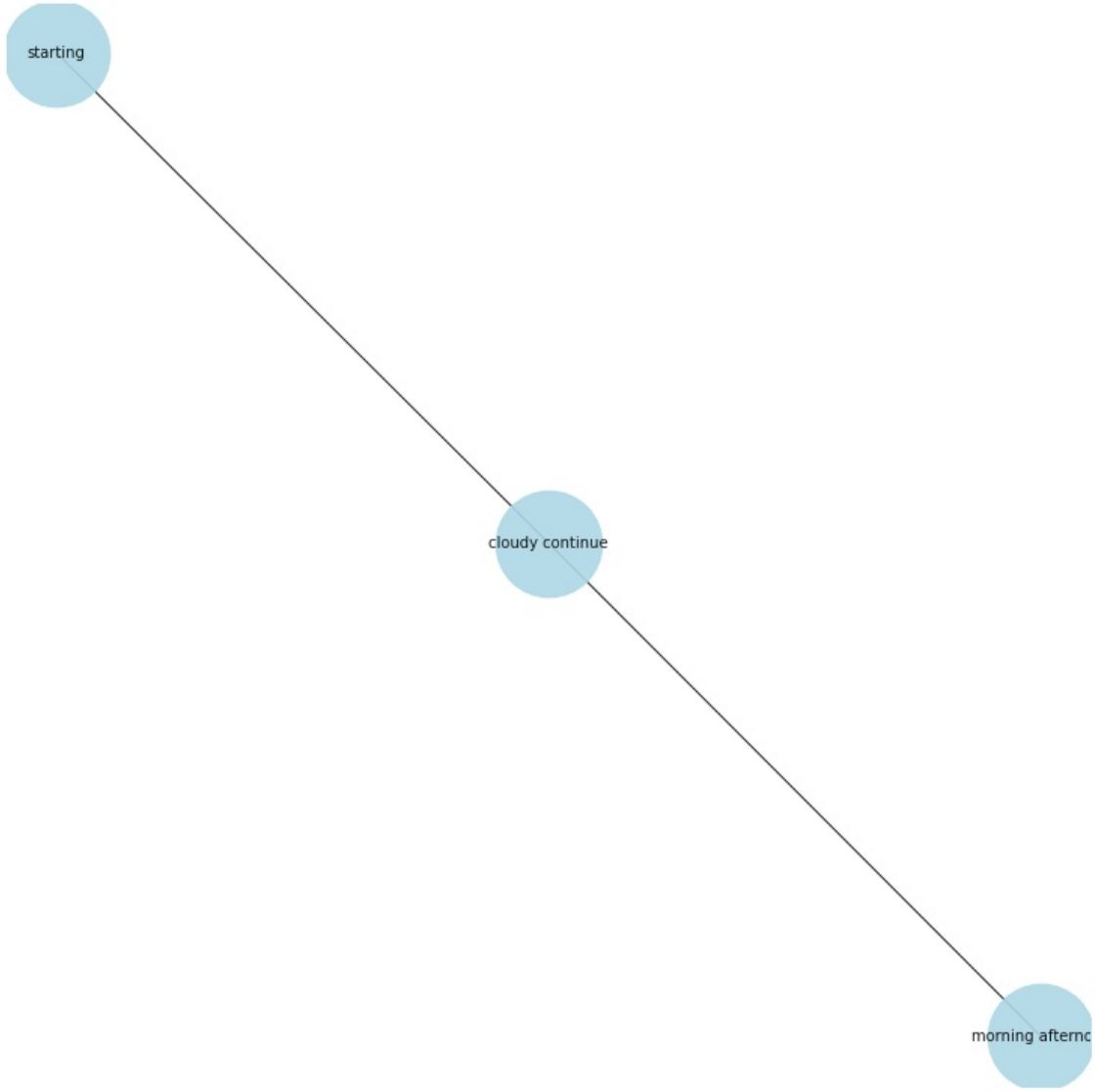




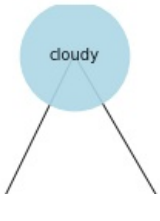

```
Partly cloudy until evening and breezy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> ROOT
until -> prep
evening -> nsubj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> pcomp
until -> prep
afternoon -> pobj
. -> punct
evening starting , cloudy , morning afternoon
```

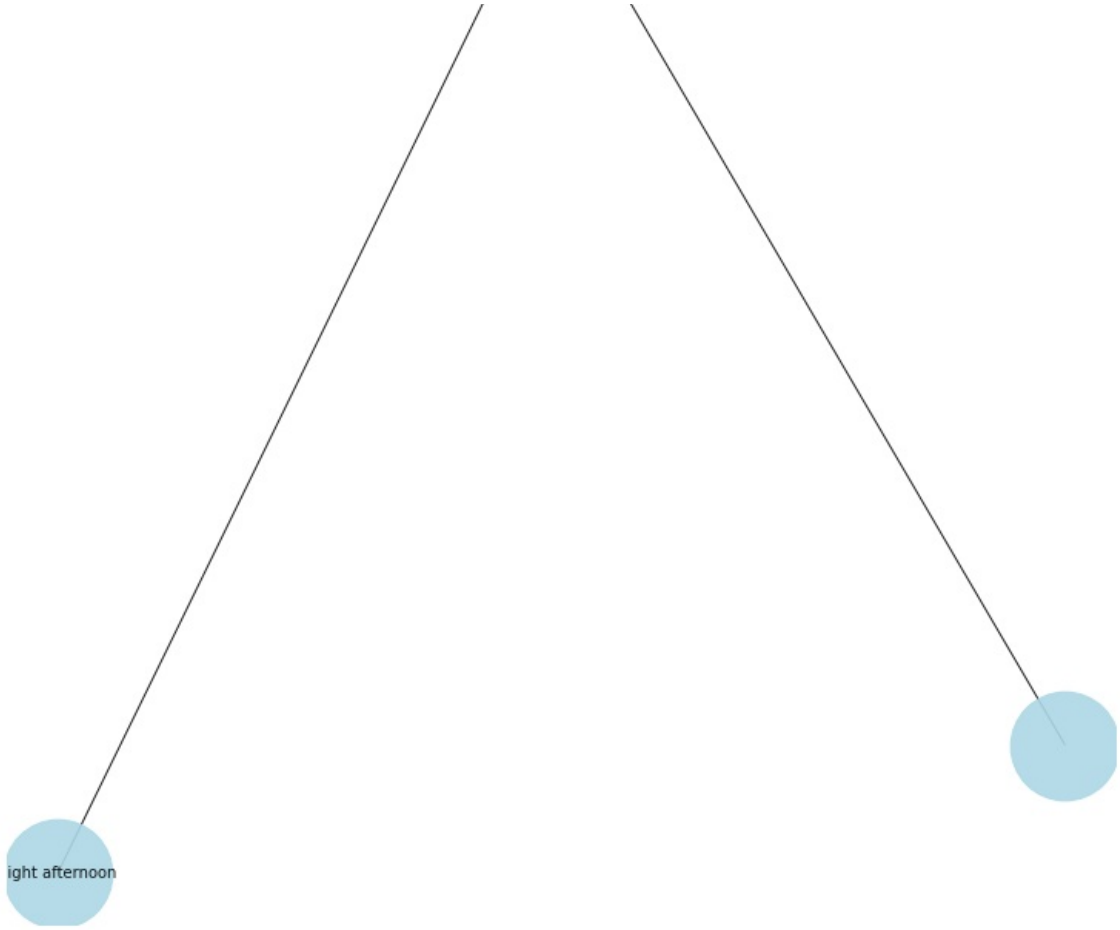


```
Mostly cloudy starting overnight and breezy starting in the morning continuing until afternoon.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy continue , morning afternoon
```

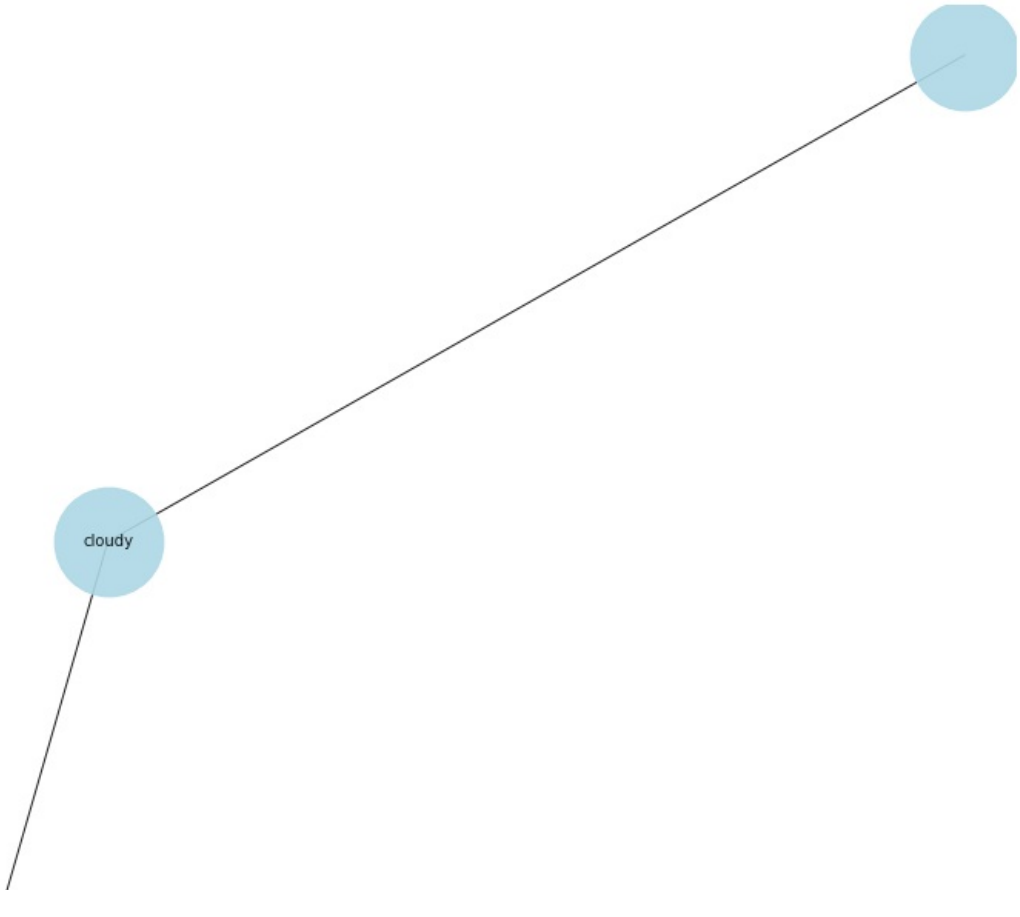



```
Mostly cloudy until night and breezy in the afternoon.
Mostly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , night afternoon
```

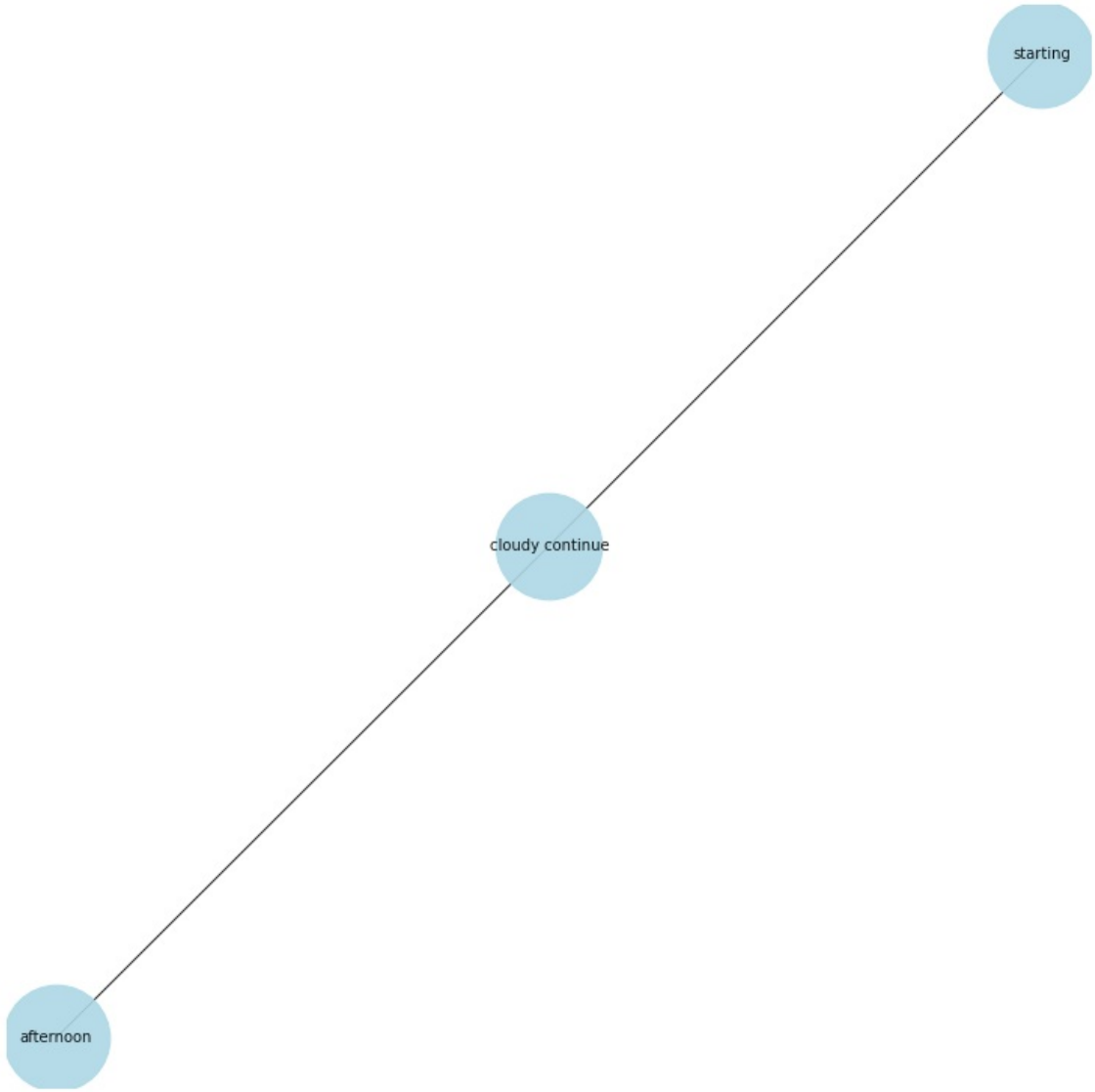




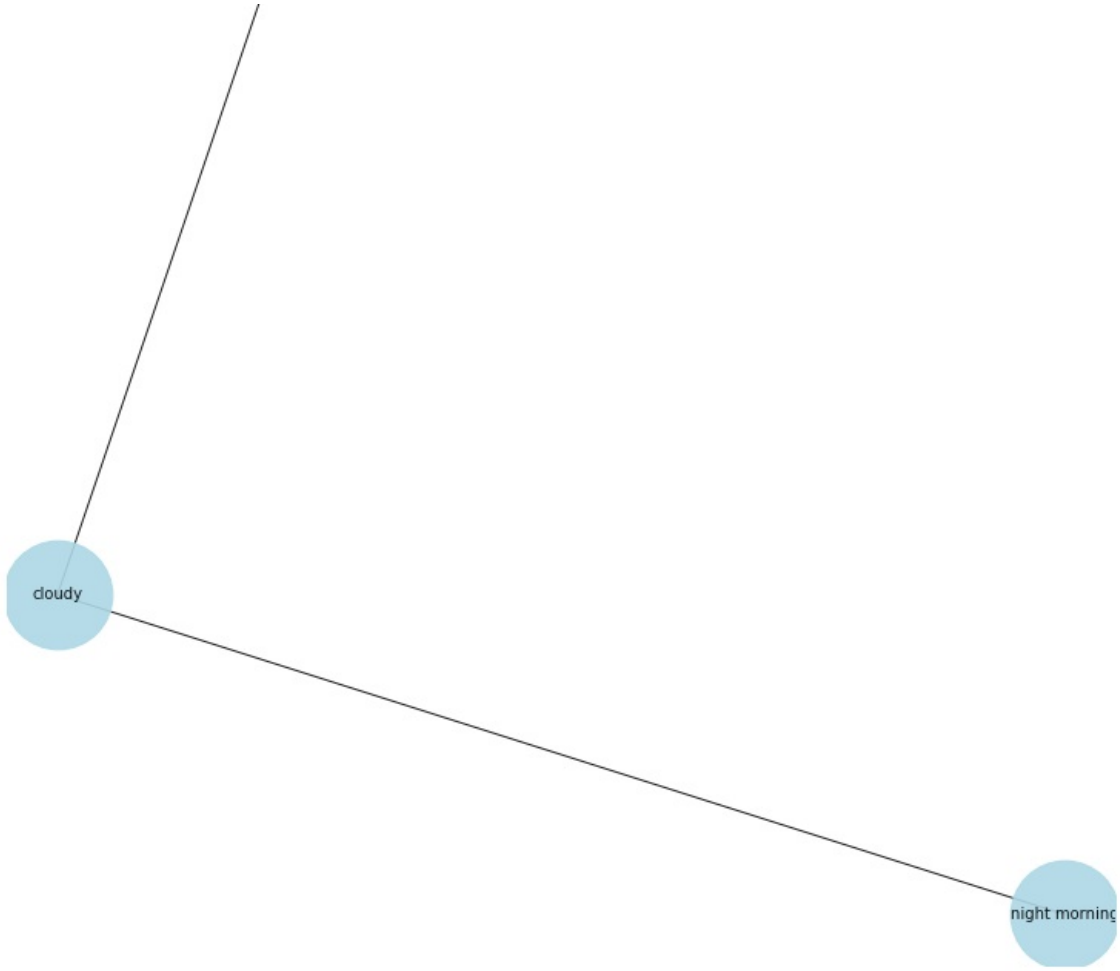

```
Mostly cloudy throughout the day and breezy in the evening.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , day evening
```



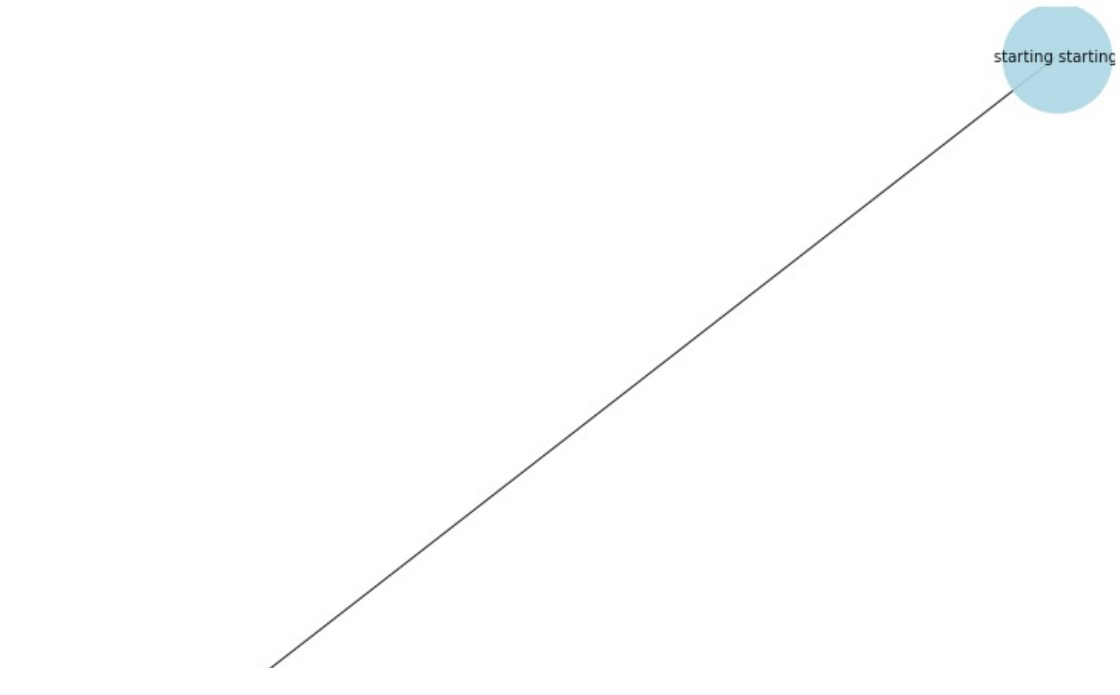

```
Mostly cloudy starting overnight continuing until afternoon.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy continue , afternoon
```

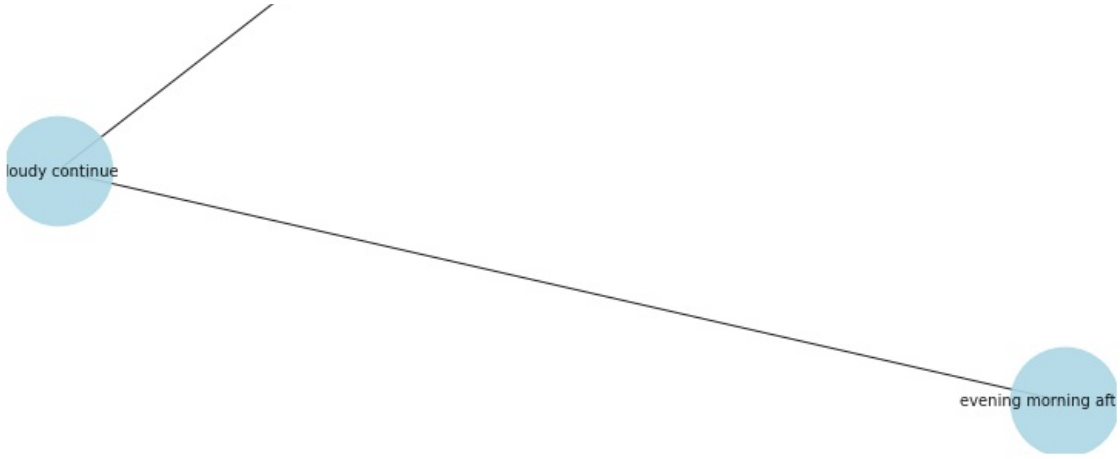



```
Partly cloudy until night and breezy in the morning.
Partly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , cloudy , night morning
```

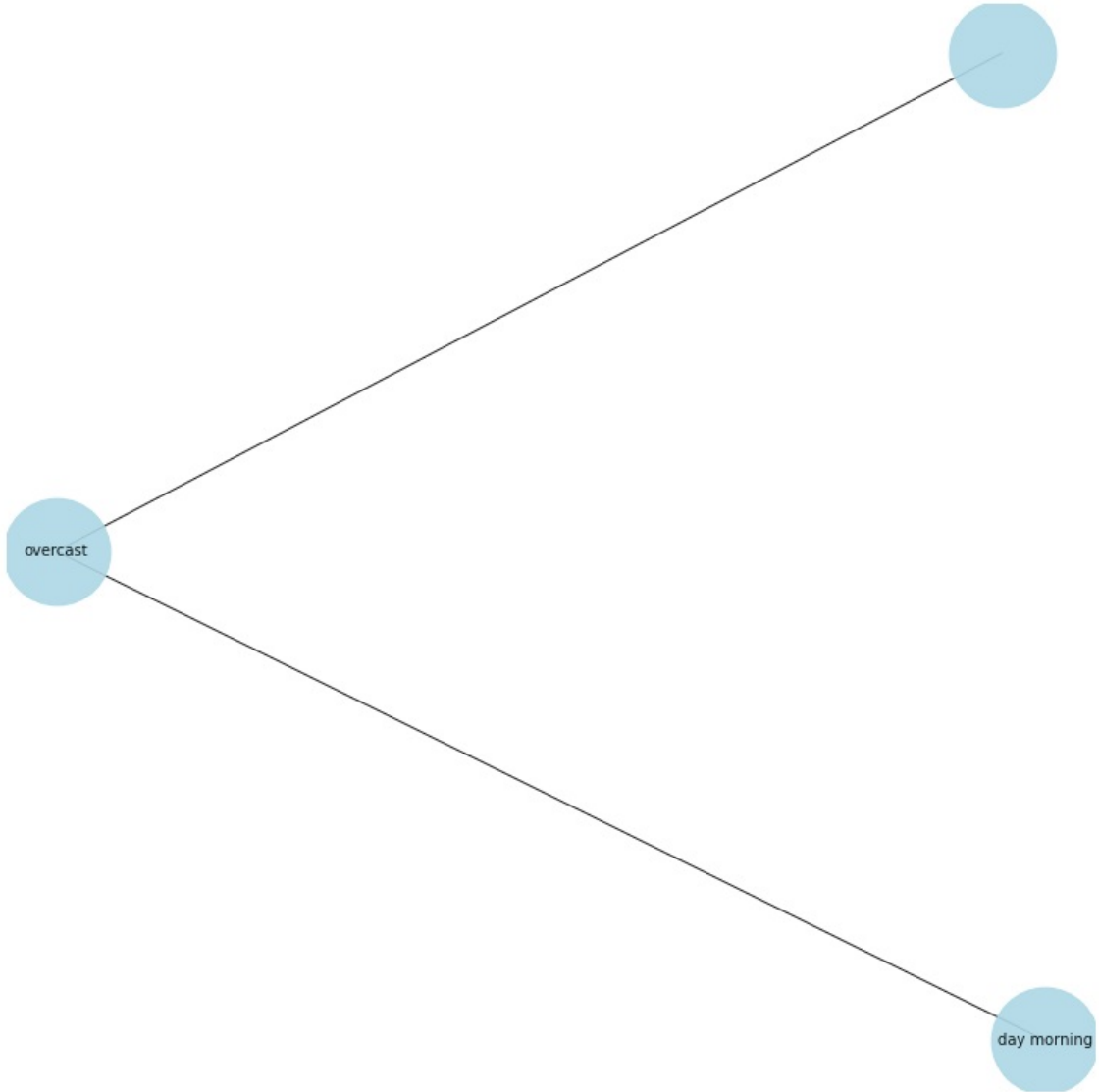



```
Partly cloudy starting overnight continuing until evening and breezy starting in the morning continuing until aft
ernoon.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting starting , cloudy continue , evening morning afternoon
```



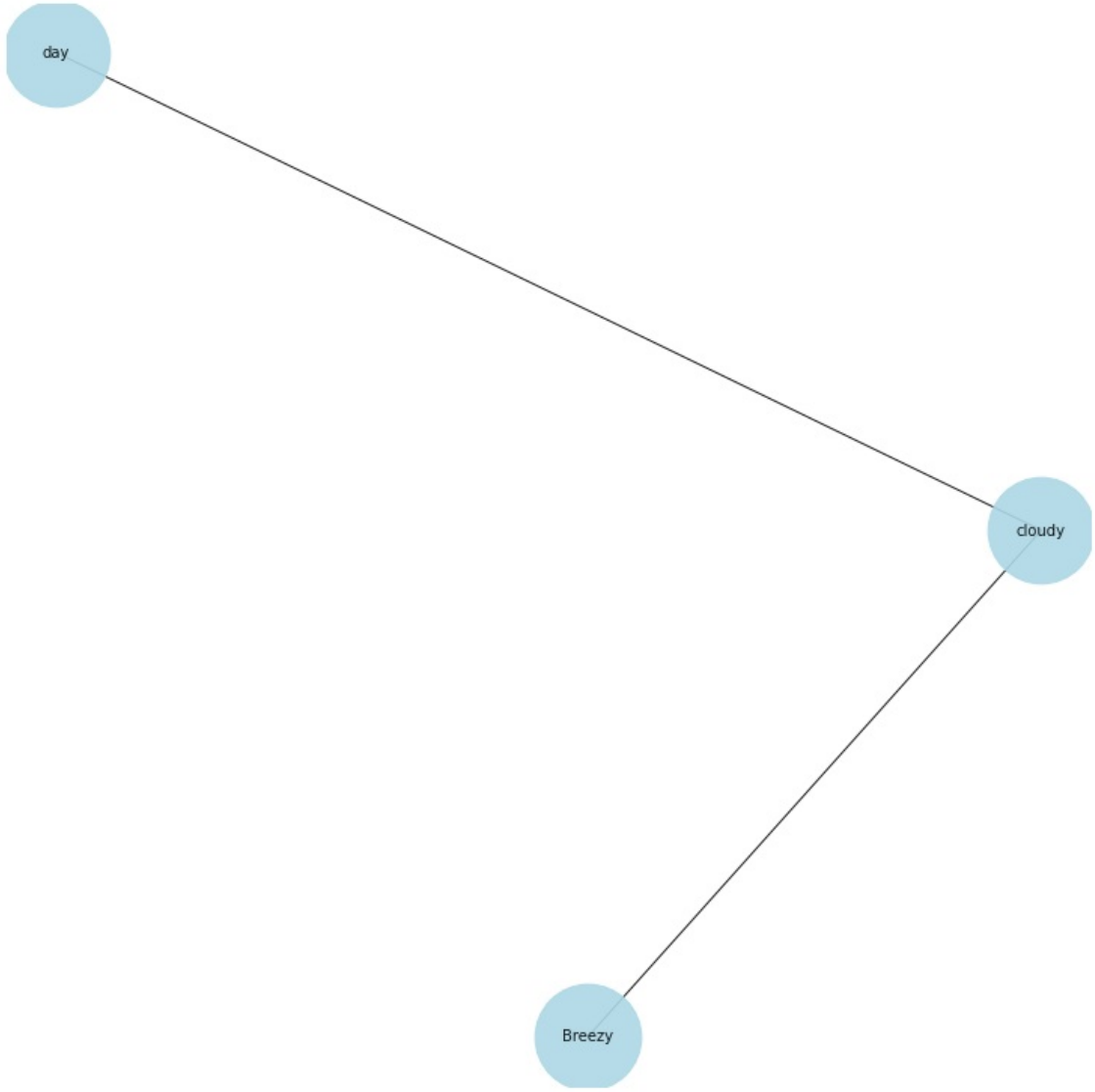



```
Overcast throughout the day and breezy in the morning.
Overcast -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , overcast , day morning
```

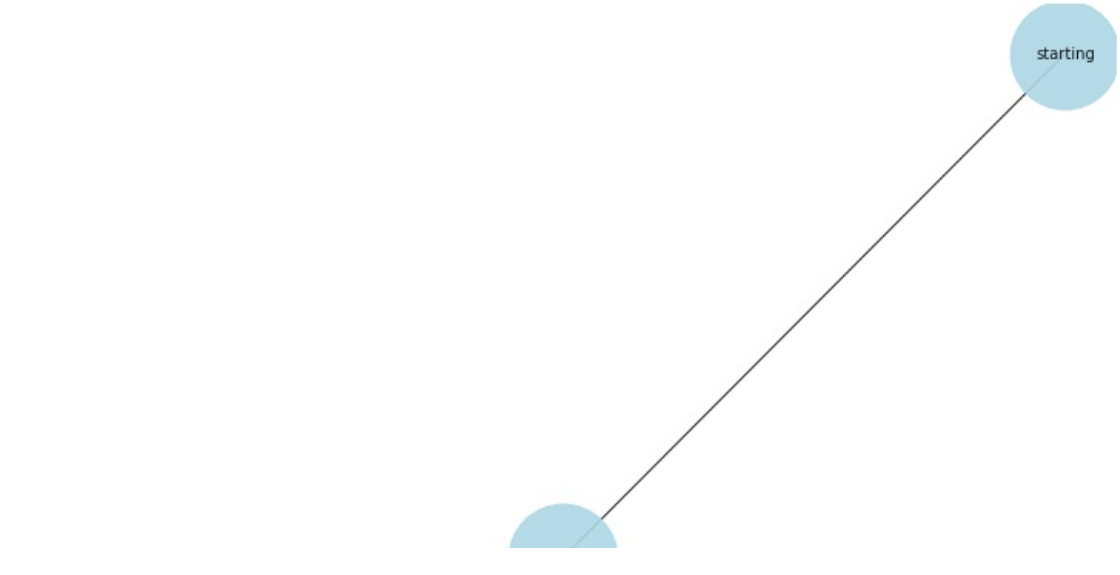


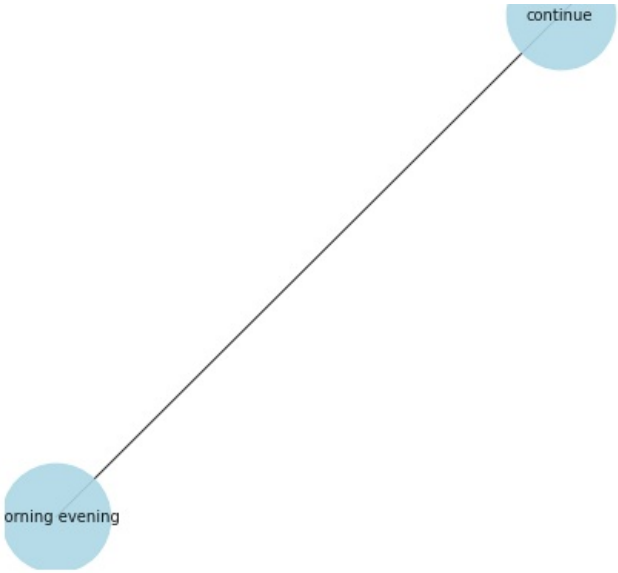

```
Breezy overnight and mostly cloudy throughout the day.
Breezy -> nsubj
overnight -> advmod
and -> cc
mostly -> advmod
cloudy -> ROOT
throughout -> prep
```

```
the -> det
day -> pobj
. -> punct
Breezy , cloudy , day
```

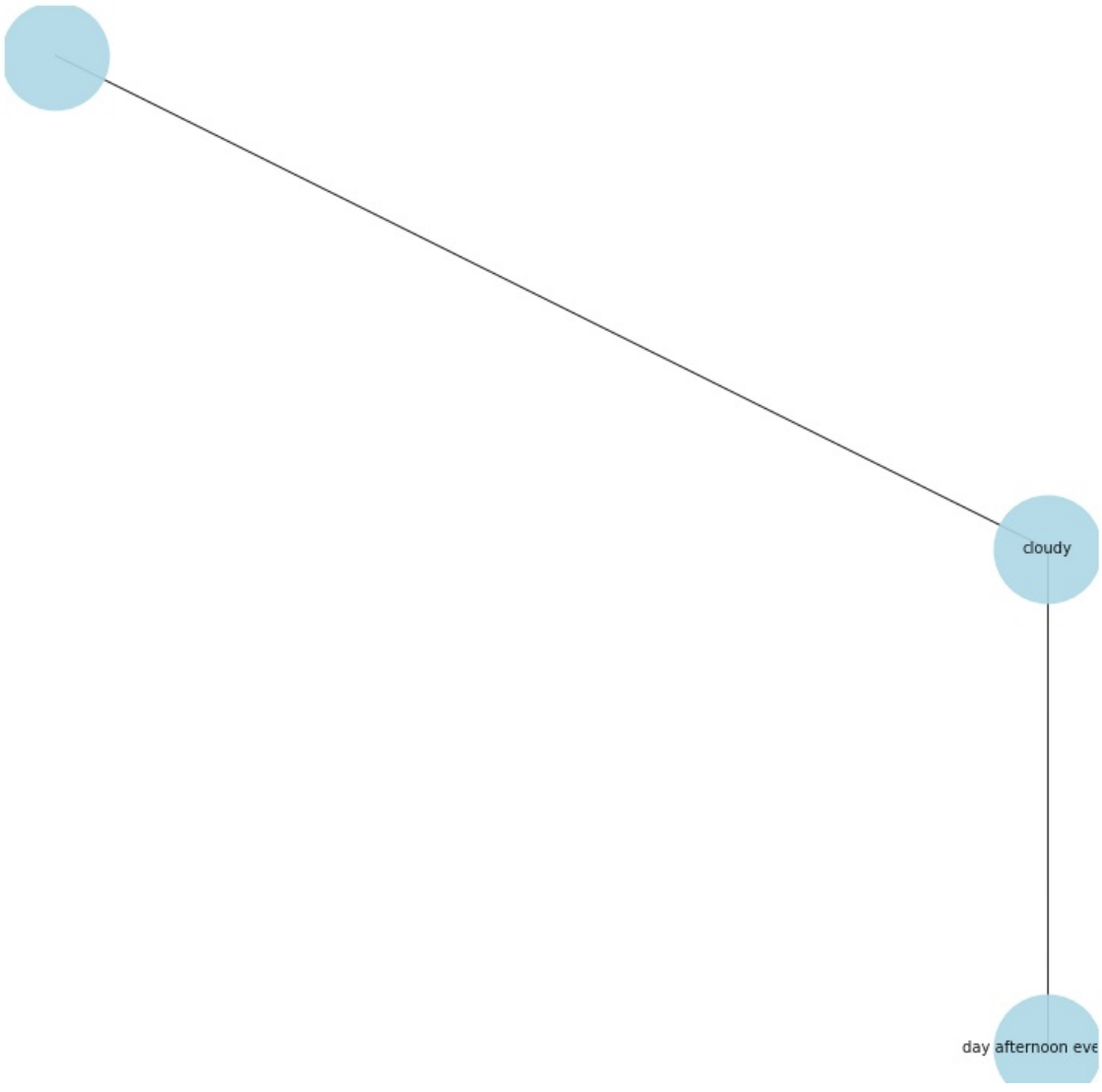



```
Mostly cloudy starting in the morning continuing until evening.
Mostly -> advmod
cloudy -> advmod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
starting , continue , morning evening
```

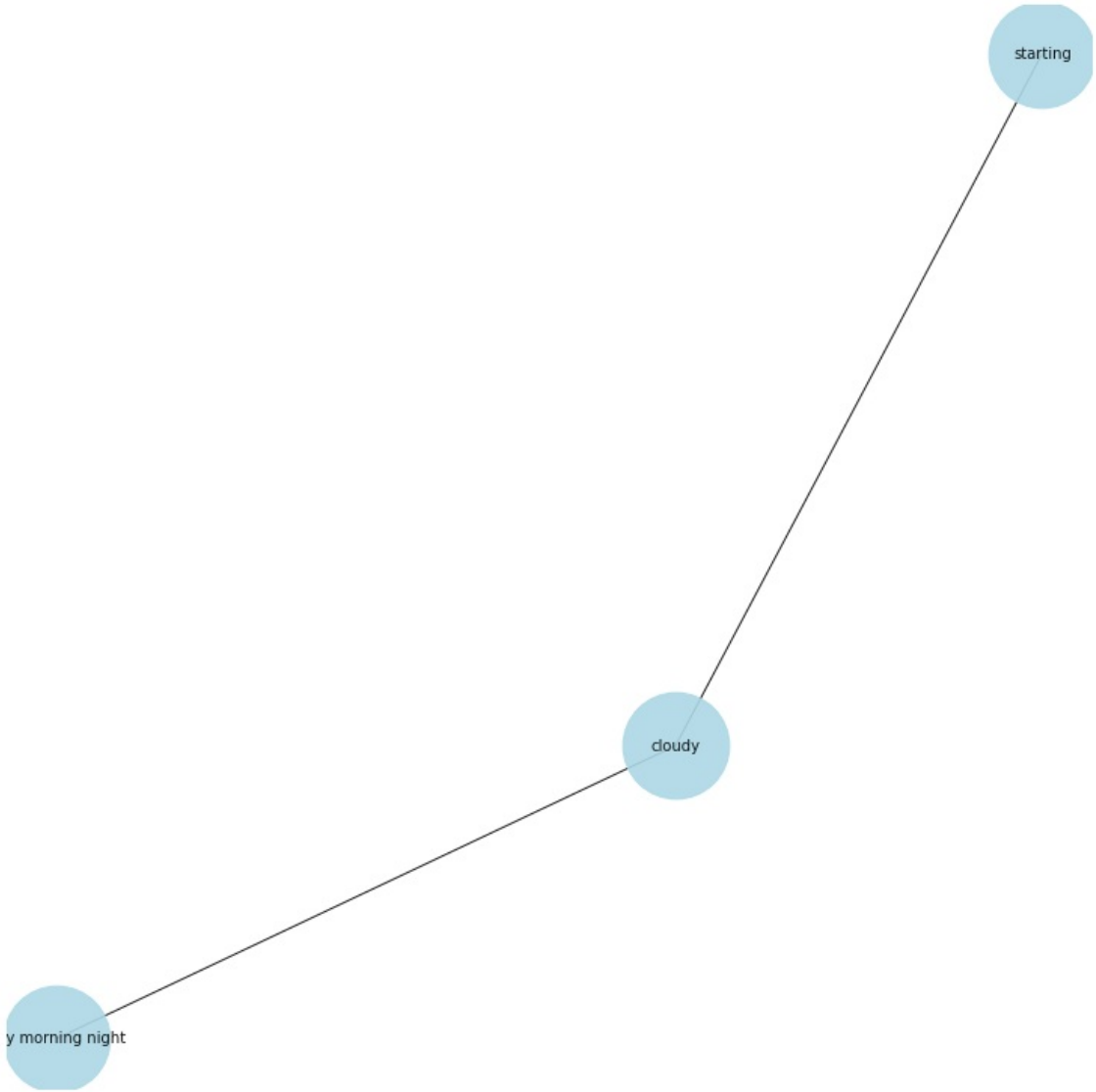




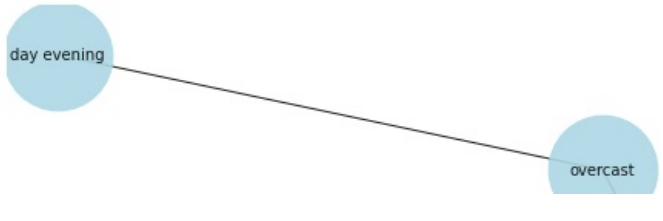

```
Partly cloudy throughout the day and breezy starting in the afternoon continuing until evening.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
 , cloudy , day afternoon evening
```

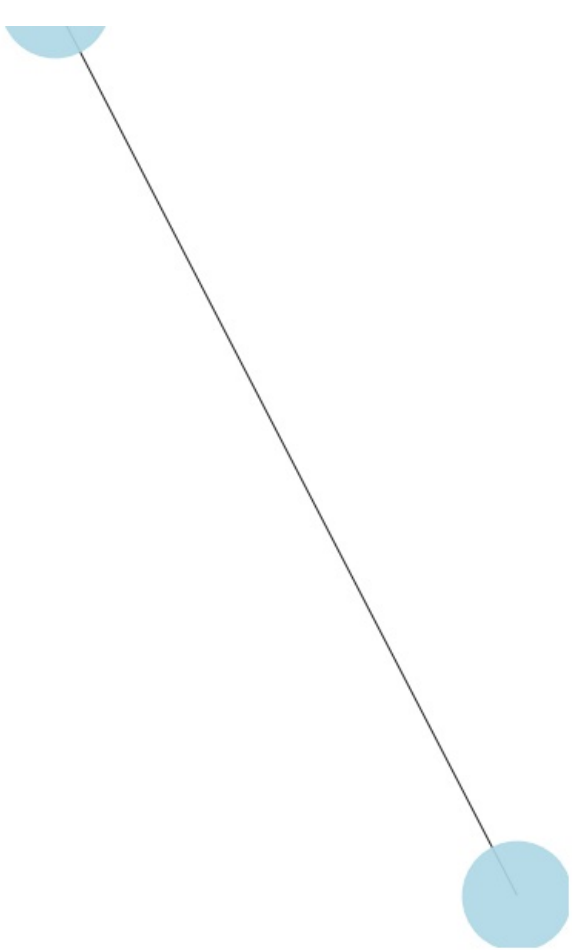


Mostly cloudy throughout the day and breezy starting in the morning continuing until night.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
night -> pobj
. -> punct
starting , cloudy , day morning night

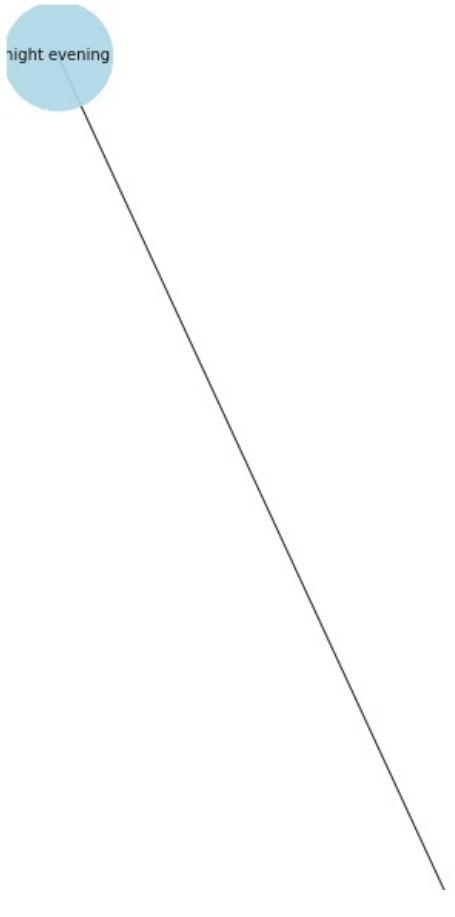


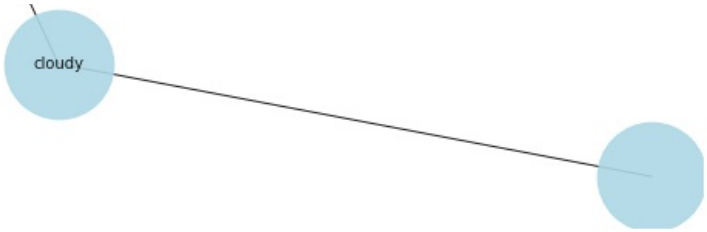
Overcast throughout the day and breezy starting in the evening.
Overcast -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , overcast , day evening



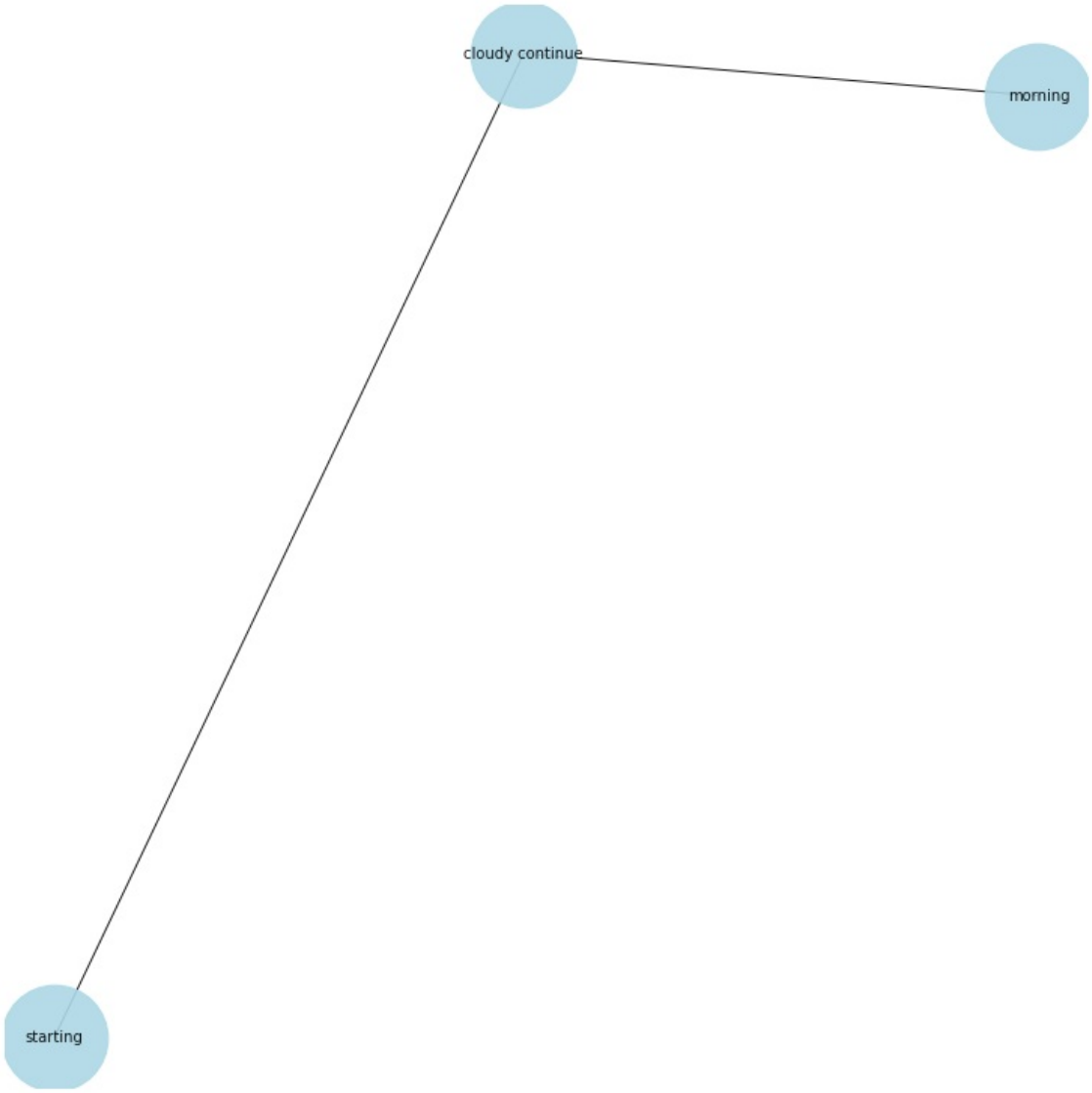


Mostly cloudy until night and breezy starting in the evening.
Mostly -> advmod
cloudy -> ROOT
until -> mark
night -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , night evening

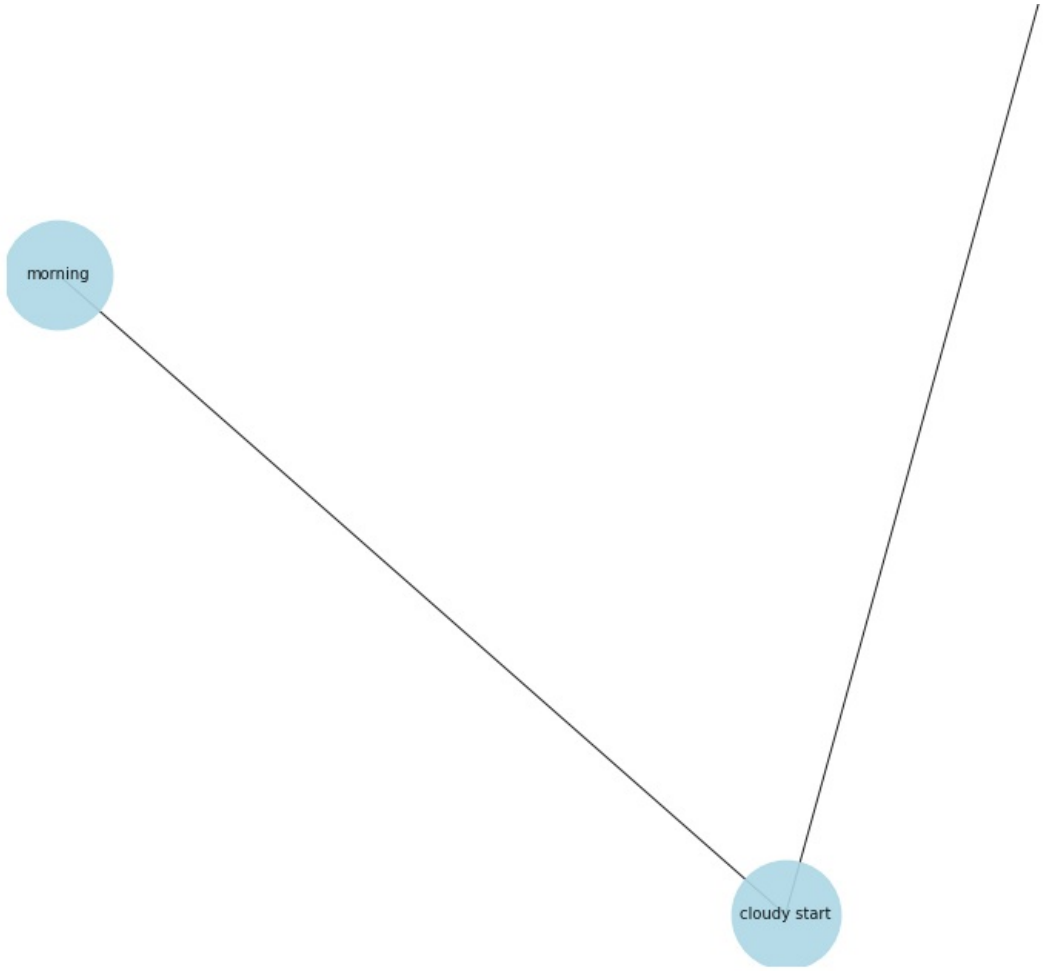




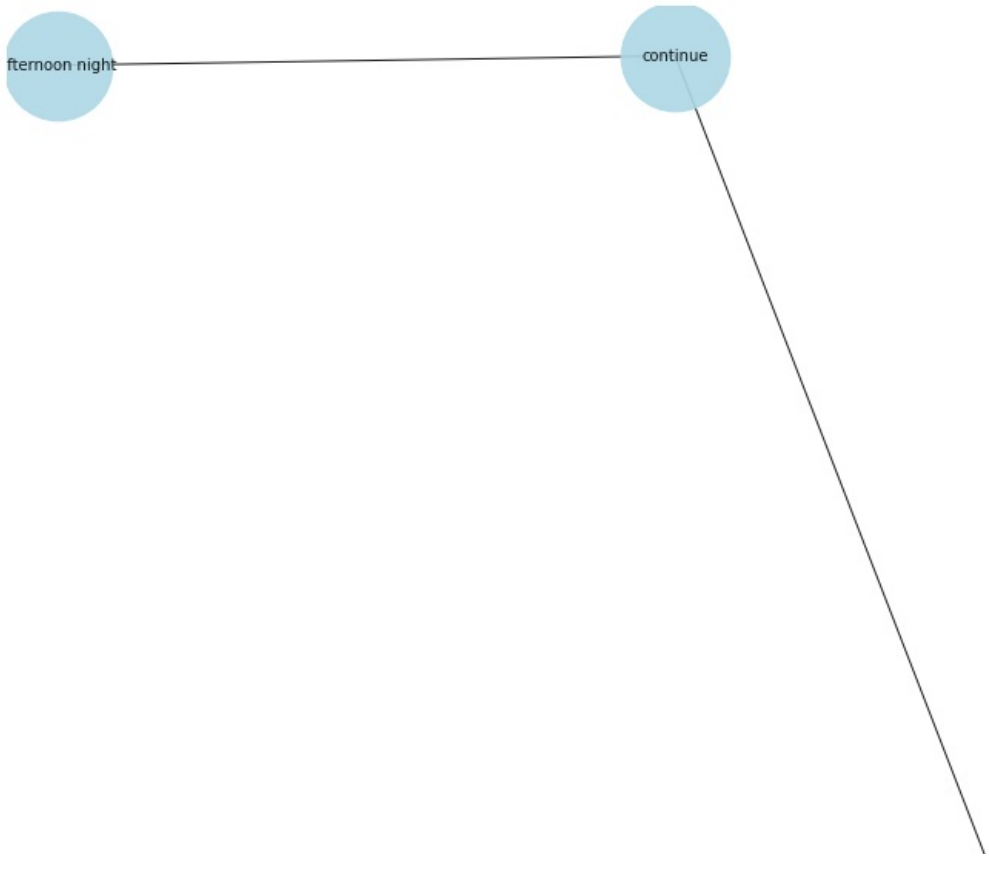

```
Mostly cloudy starting overnight continuing until morning.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
morning -> pobj
. -> punct
starting , cloudy continue , morning
```



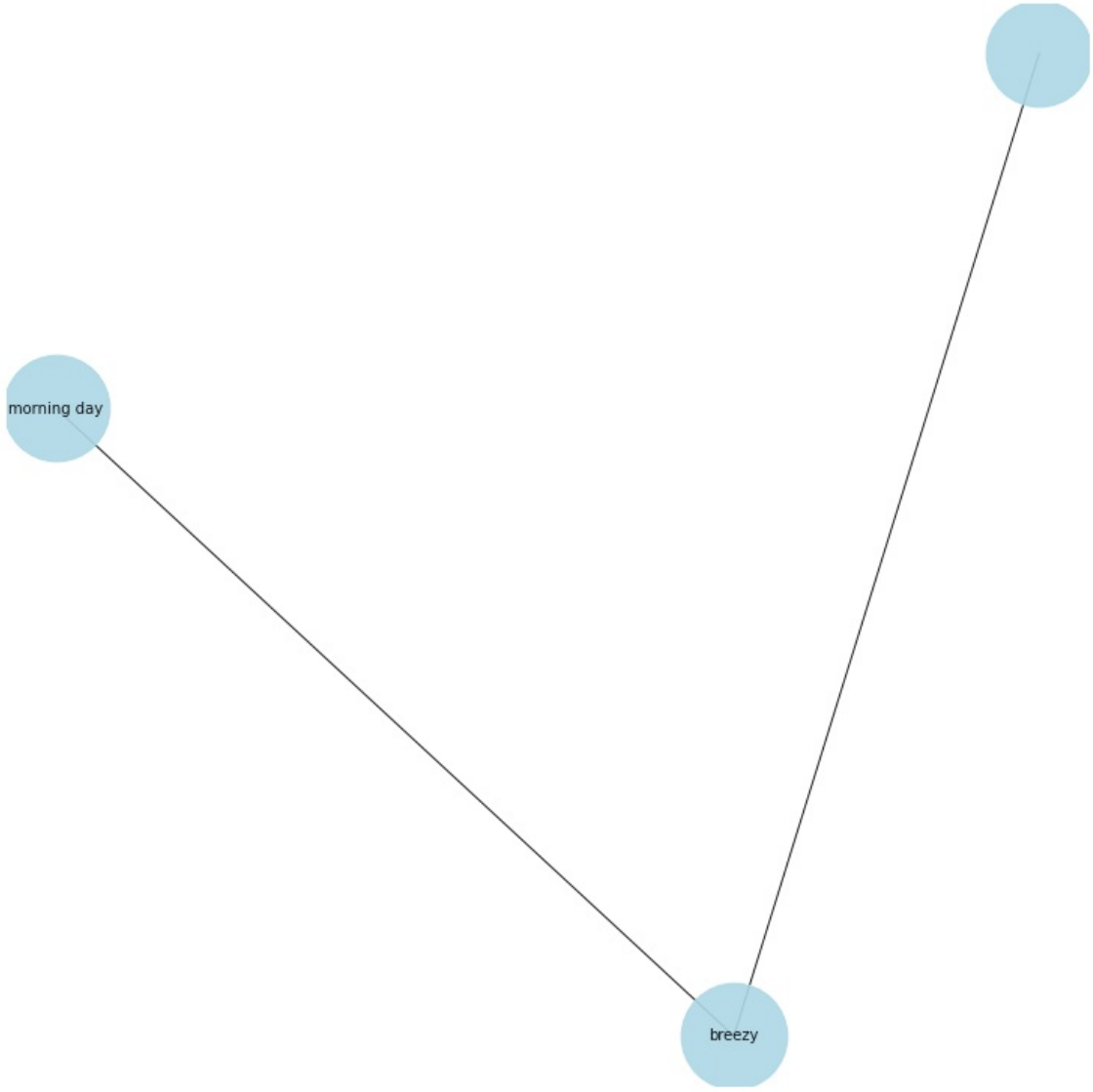

```
Mostly cloudy starting overnight and breezy in the morning.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
overnight -> advmod
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , cloudy start , morning
```



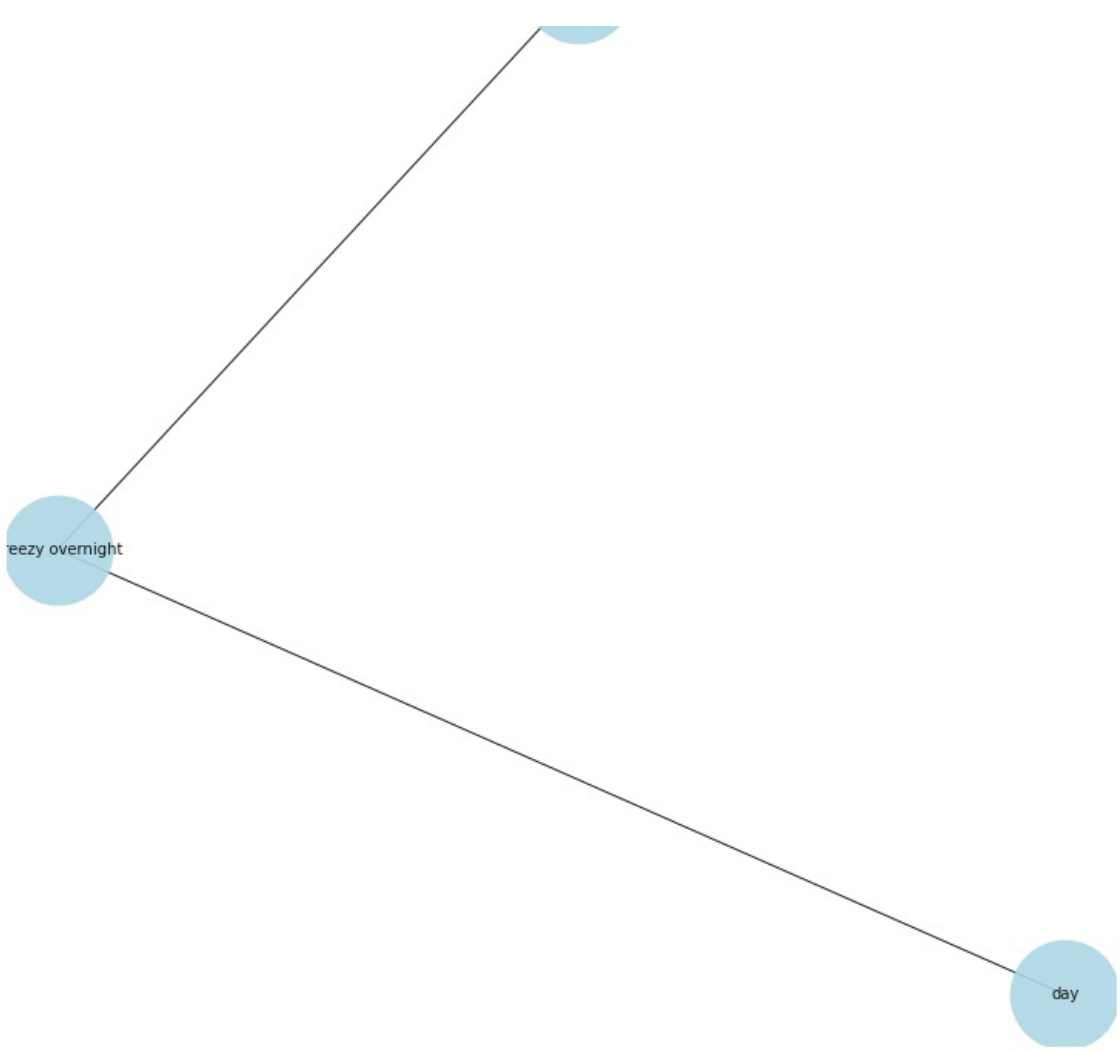

```
Mostly cloudy starting in the afternoon continuing until night.
Mostly -> advmod
cloudy -> nsubj
starting -> acl
in -> prep
the -> det
afternoon -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
cloudy , continue , afternoon night
```



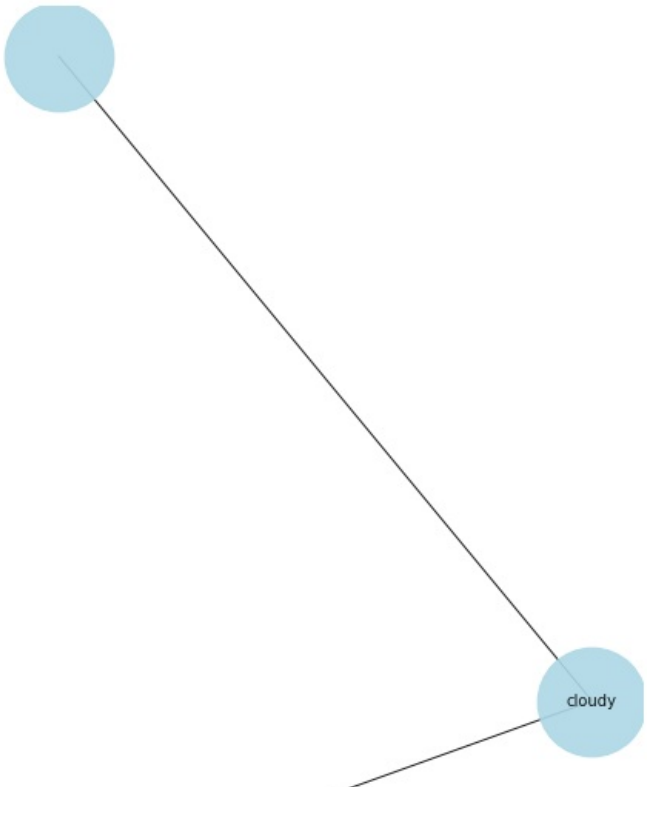

```
Breezy until morning and mostly cloudy throughout the day.
Breezy -> ROOT
until -> prep
morning -> pobj
and -> cc
mostly -> advmod
cloudy -> conj
throughout -> prep
the -> det
day -> pobj
. -> punct
 , breezy , morning day
```

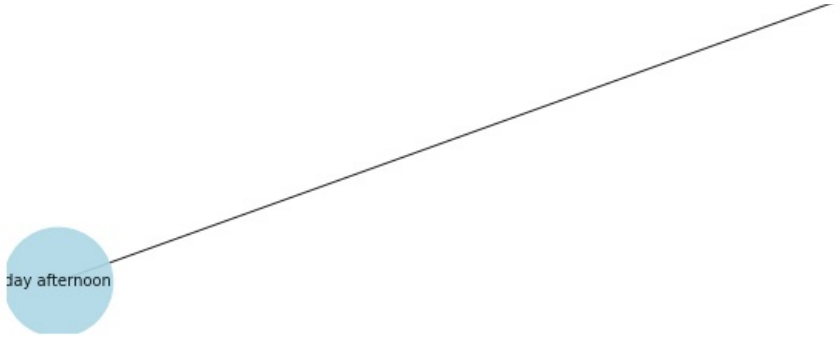



```
Breezy overnight and overcast throughout the day.
Breezy -> amod
overnight -> ROOT
and -> cc
overcast -> conj
throughout -> prep
the -> det
day -> pobj
. -> punct
 , breezy overnight , day
```

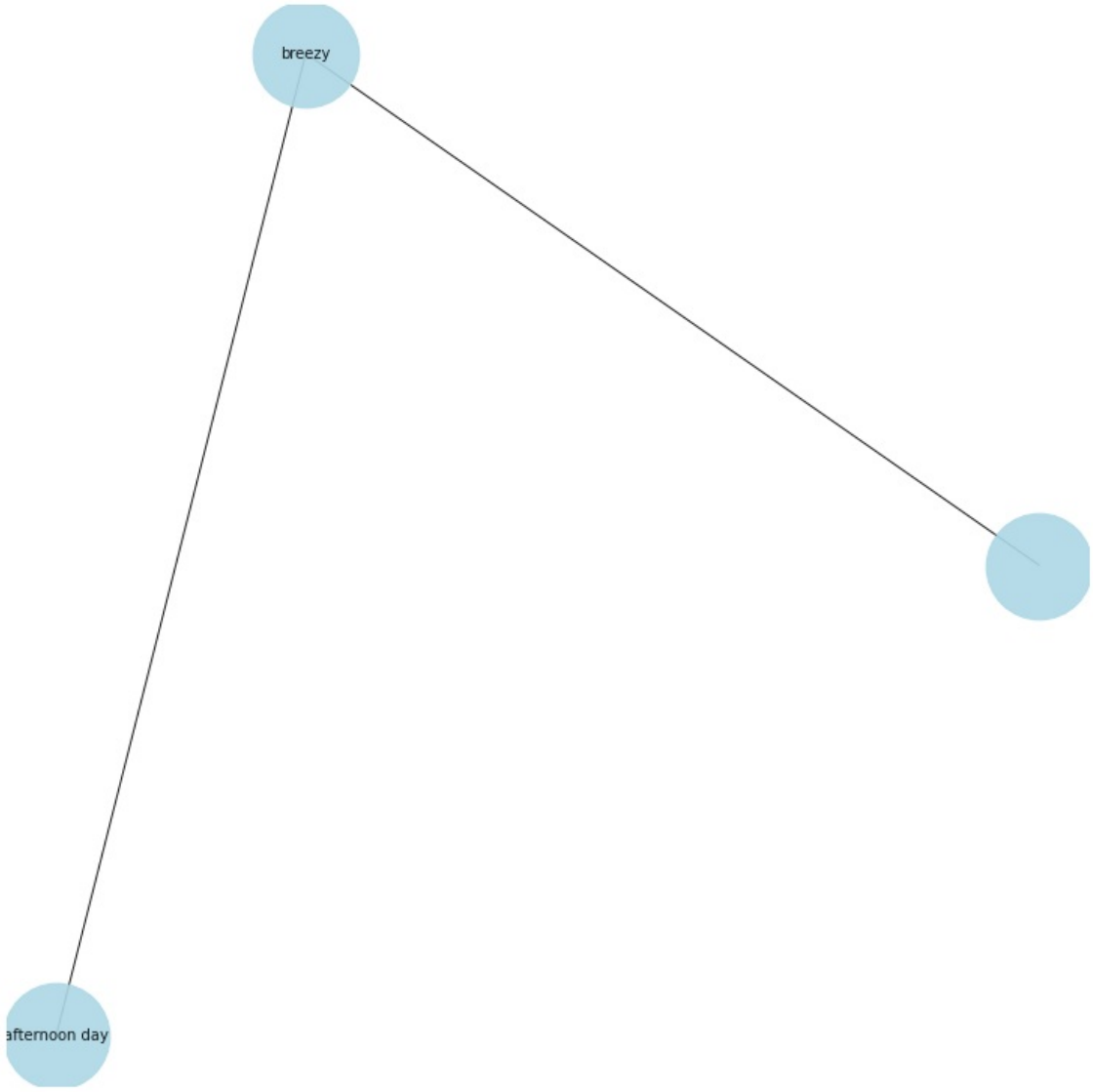



```
Mostly cloudy throughout the day and breezy starting in the afternoon.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , day afternoon
```



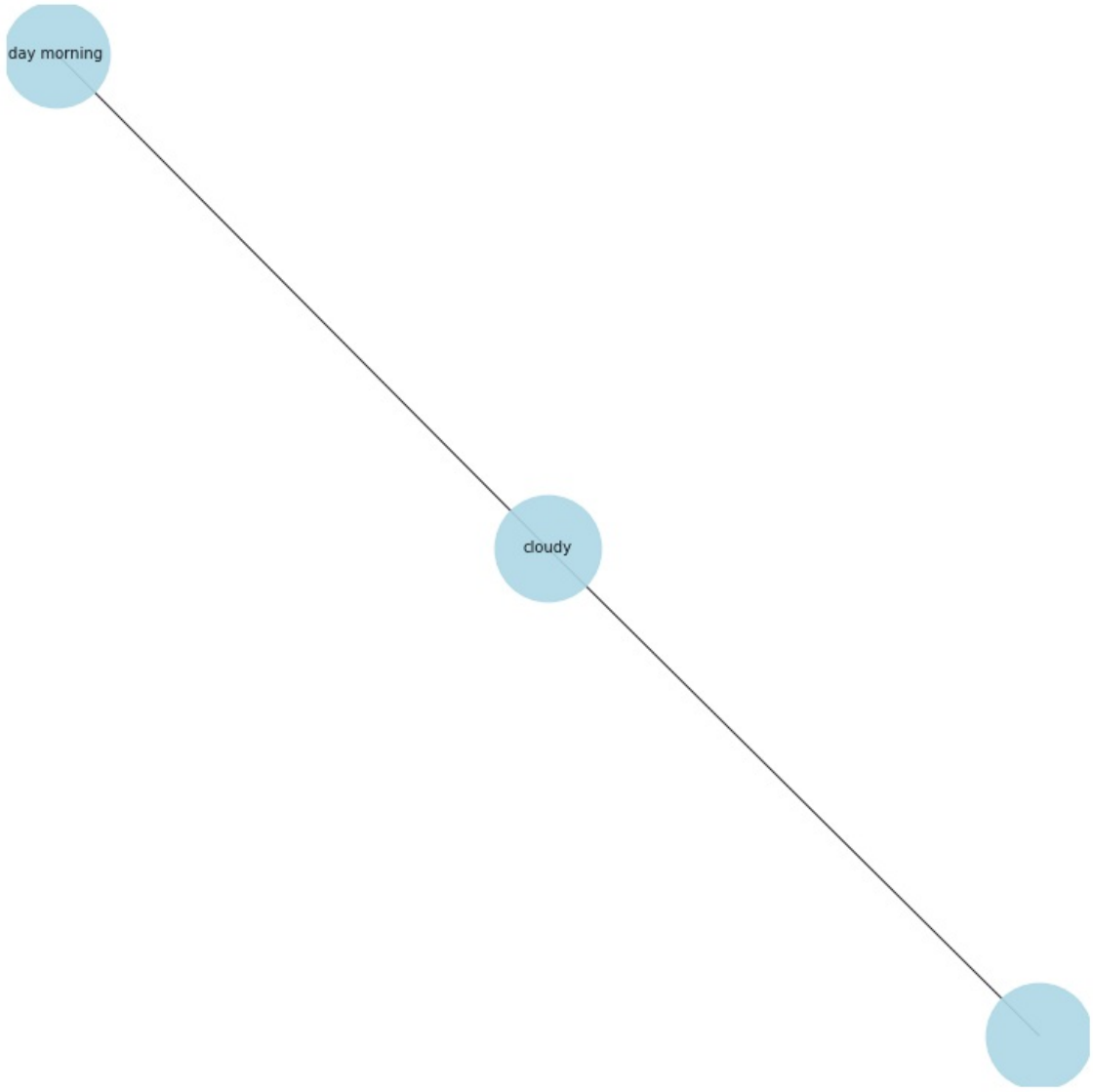



```
Breezy until afternoon and mostly cloudy throughout the day.
Breezy -> ROOT
until -> prep
afternoon -> pobj
and -> cc
mostly -> advmod
cloudy -> conj
throughout -> prep
the -> det
day -> pobj
. -> punct
 , breezy , afternoon day
```

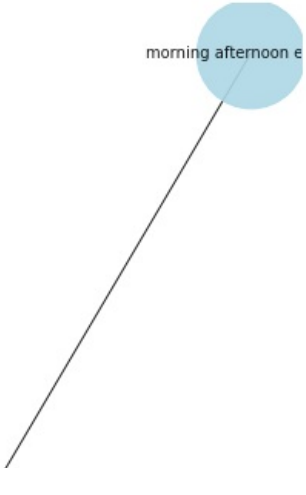


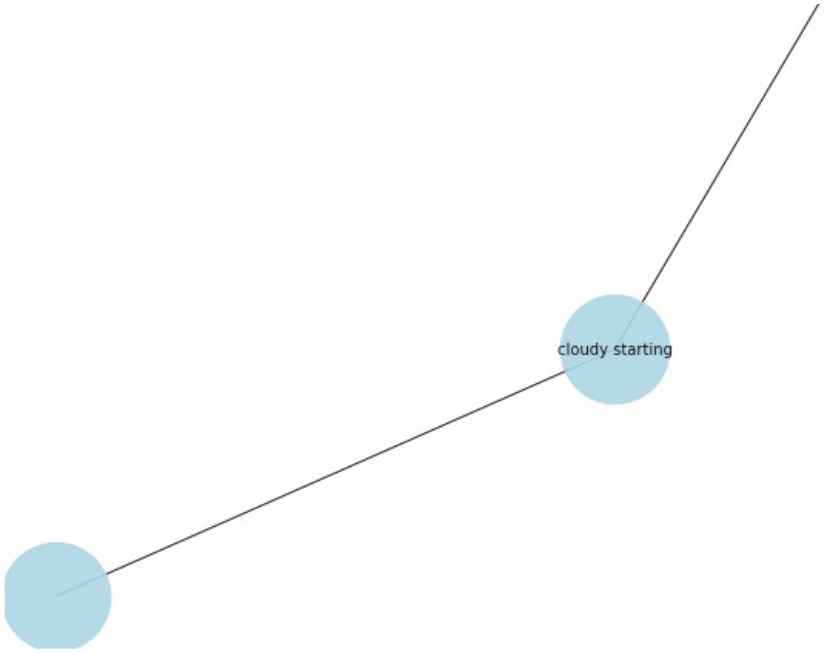

```
Partly cloudy throughout the day and breezy in the morning.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
```

, cloudy , day morning

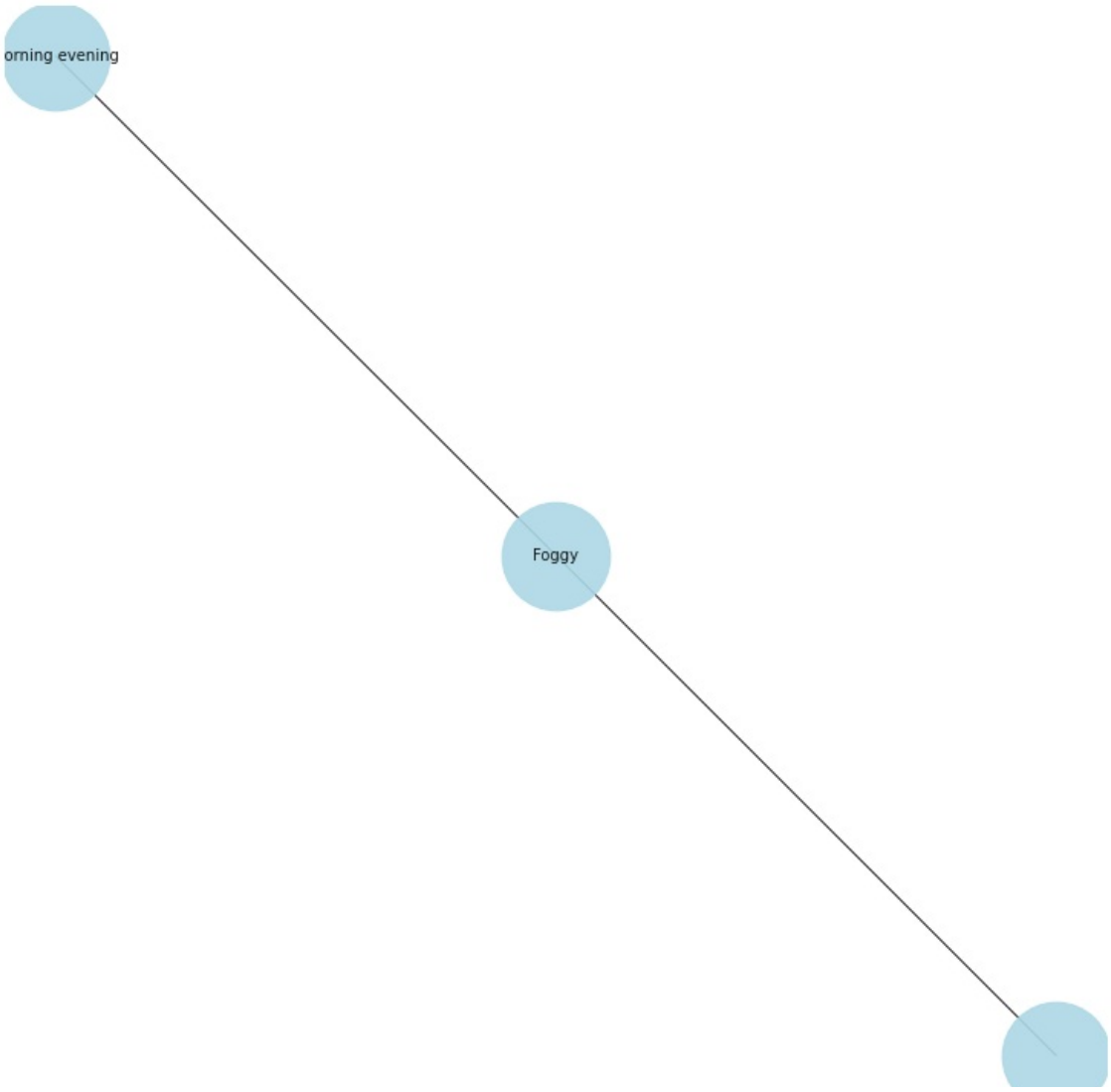


Mostly cloudy starting in the morning and breezy starting in the afternoon continuing until evening.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
starting -> xcomp
in -> prep
the -> det
afternoon -> pobj
continuing -> xcomp
until -> prep
evening -> pobj
. -> punct
, cloudy starting , morning afternoon evening

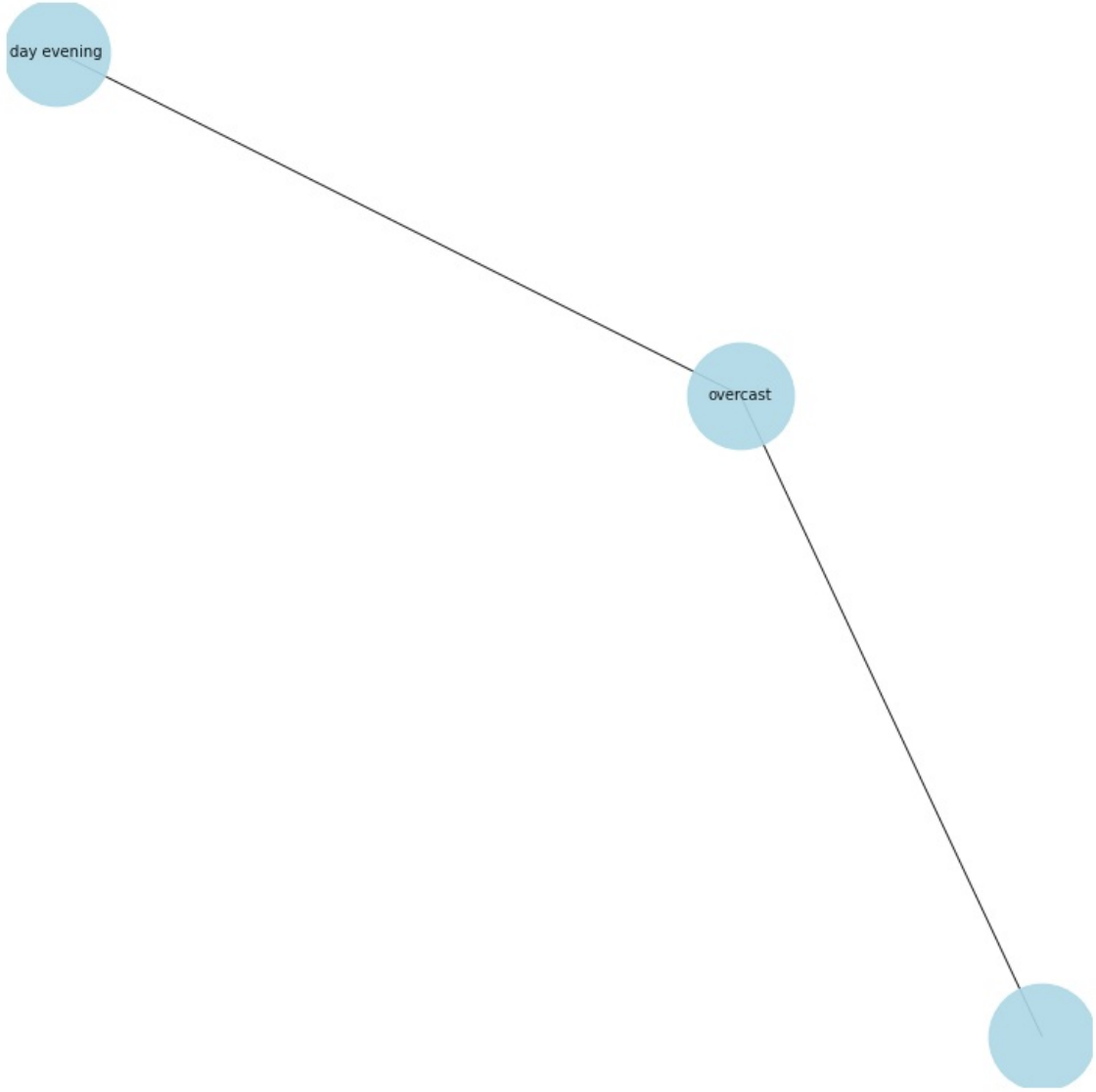




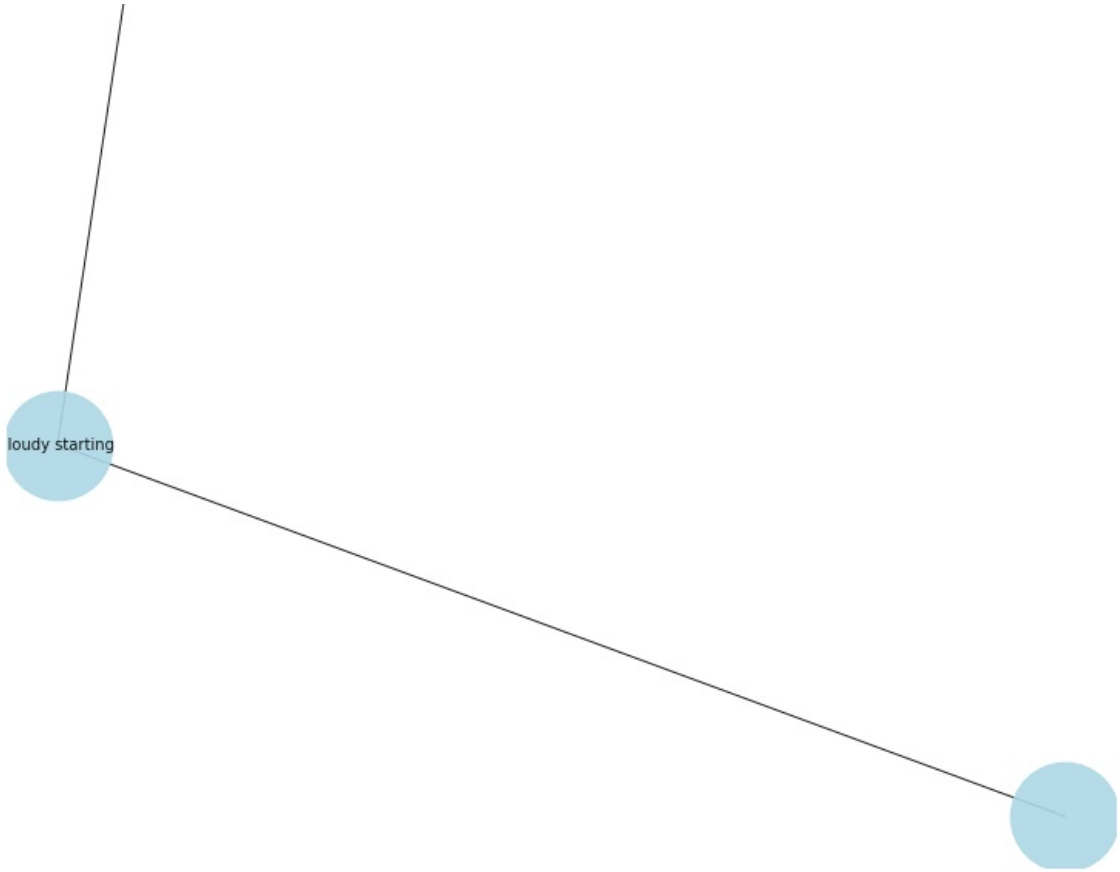

```
Foggy starting overnight continuing until morning and breezy starting in the evening.
Foggy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , Foggy , morning evening
```



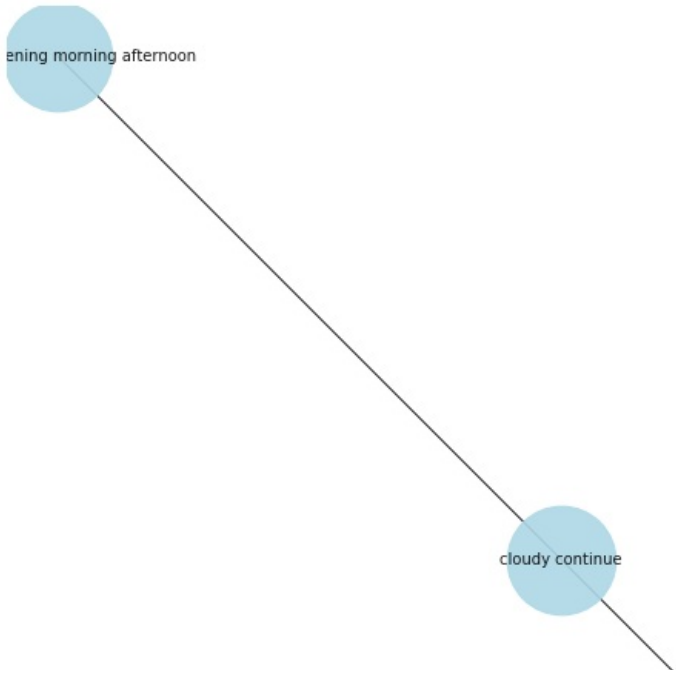
Overcast throughout the day and breezy in the evening.
Overcast -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , overcast , day evening

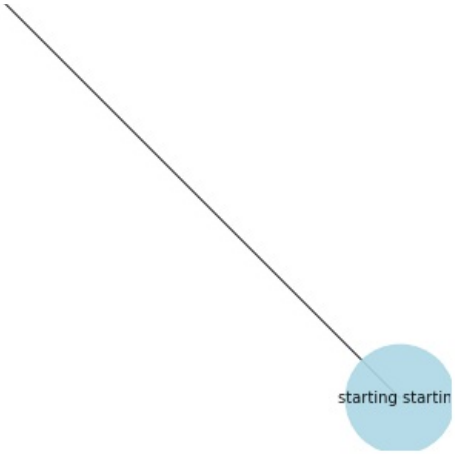


Mostly cloudy starting in the morning and breezy in the afternoon.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy starting , morning afternoon

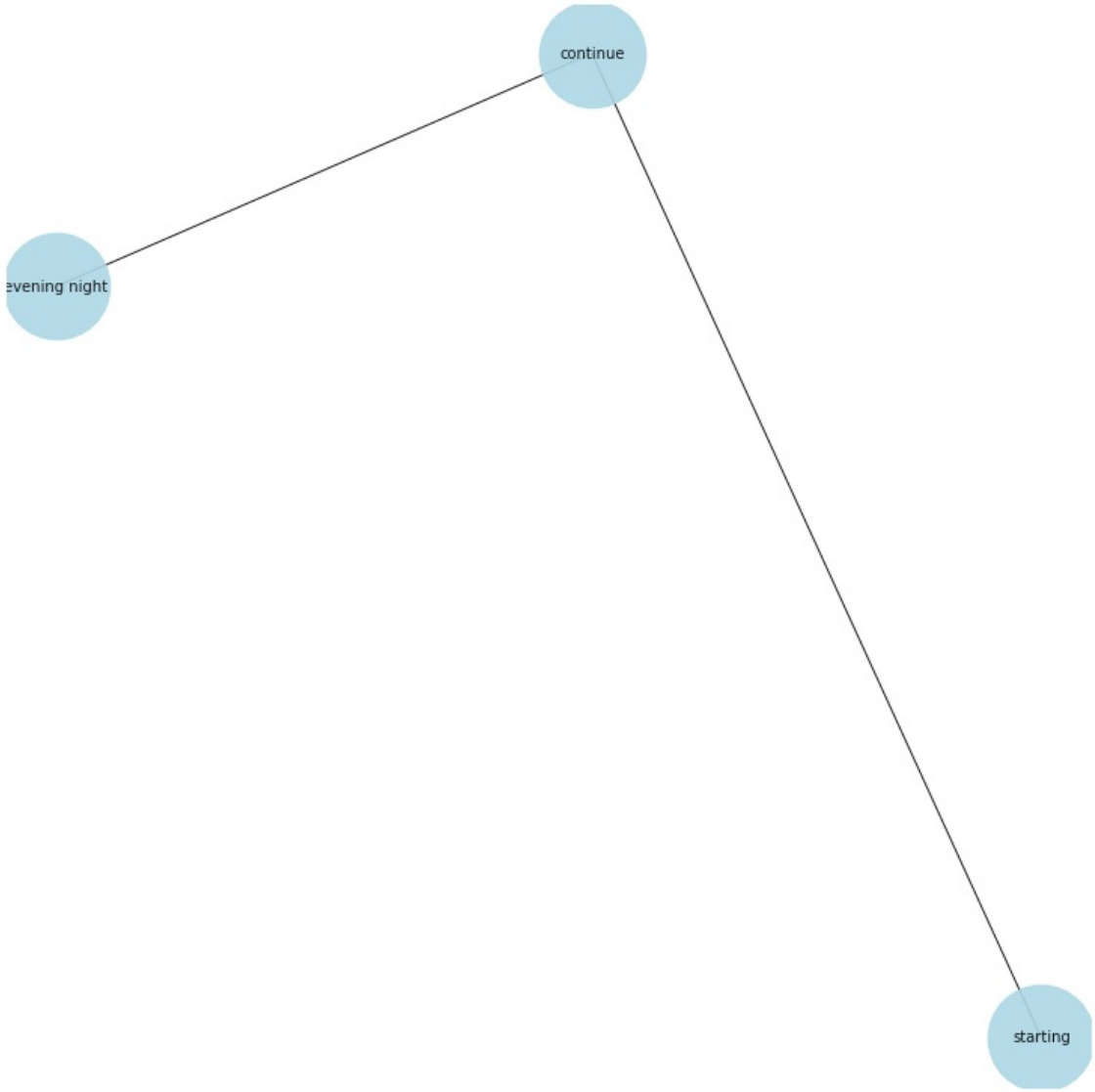



```
Partly cloudy starting in the morning continuing until evening and breezy starting in the morning continuing unti
l afternoon.
Partly -> advmod
cloudy -> amod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting starting , cloudy continue , morning evening morning afternoon
```



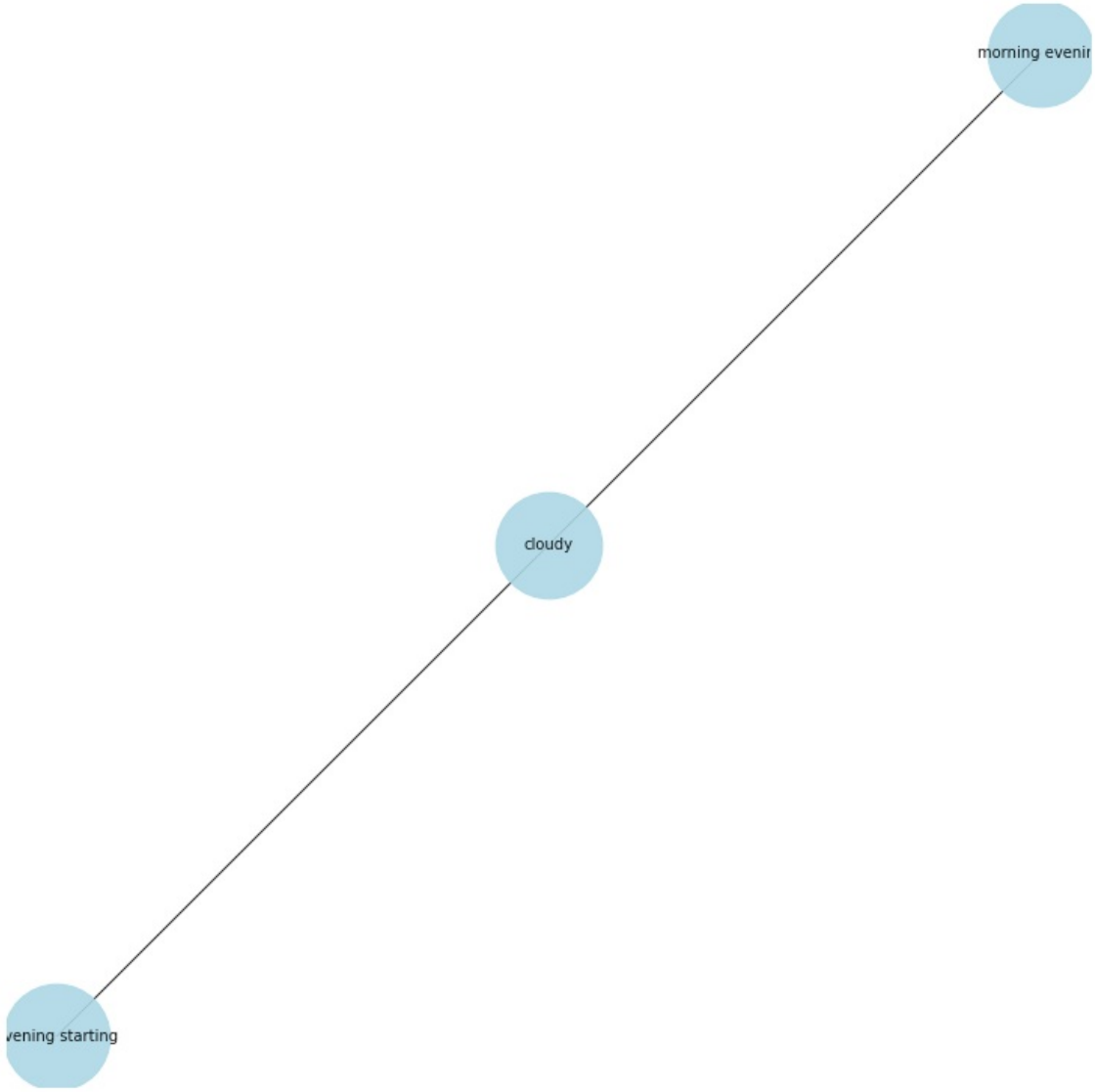



```
Partly cloudy starting in the evening continuing until night.
Partly -> advmod
cloudy -> advmod
starting -> nsubj
in -> prep
the -> det
evening -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
starting , continue , evening night
```

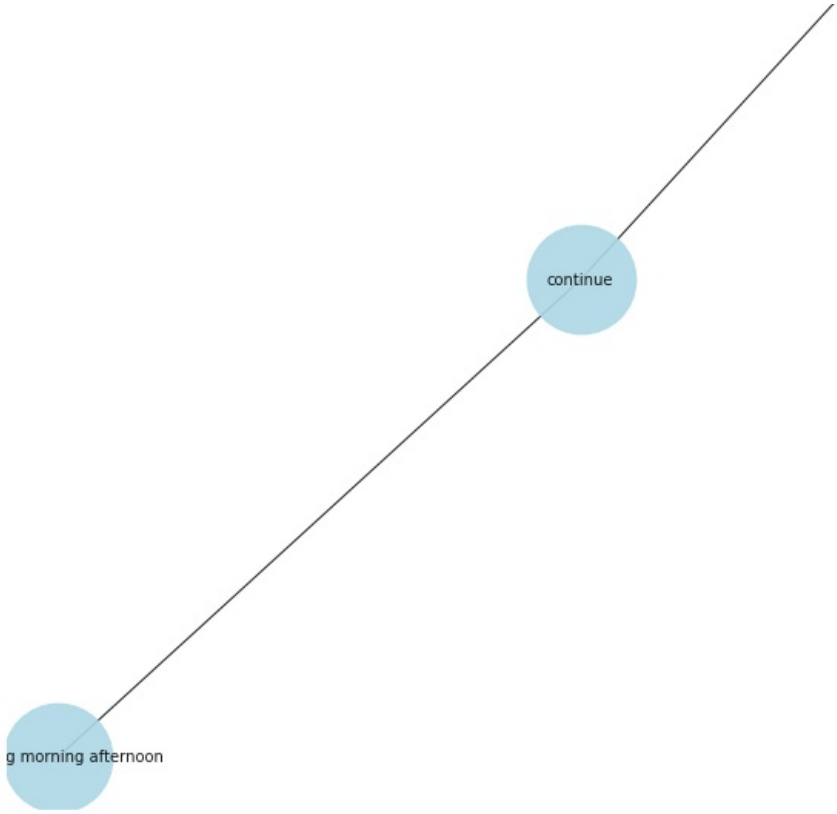



```
Partly cloudy until evening and breezy starting in the morning continuing until evening.
Partly -> advmod
cloudy -> ROOT
until -> prep
evening -> nsubj
and -> cc
breezy -> conj
starting -> nsubj
```

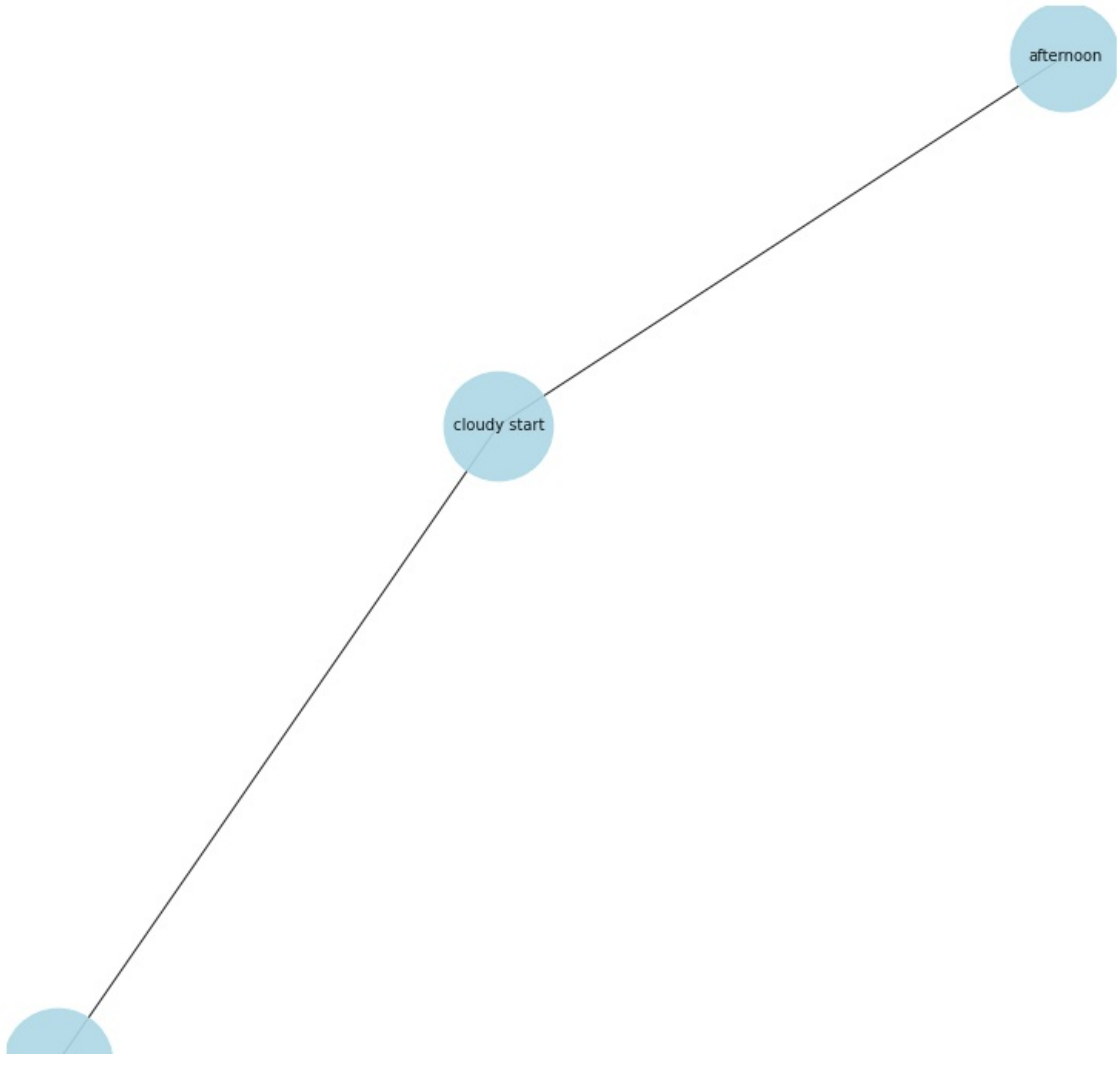
```
in -> prep
the -> det
morning -> pobj
continuing -> pcomp
until -> prep
evening -> pobj
. -> punct
evening starting , cloudy , morning evening
```



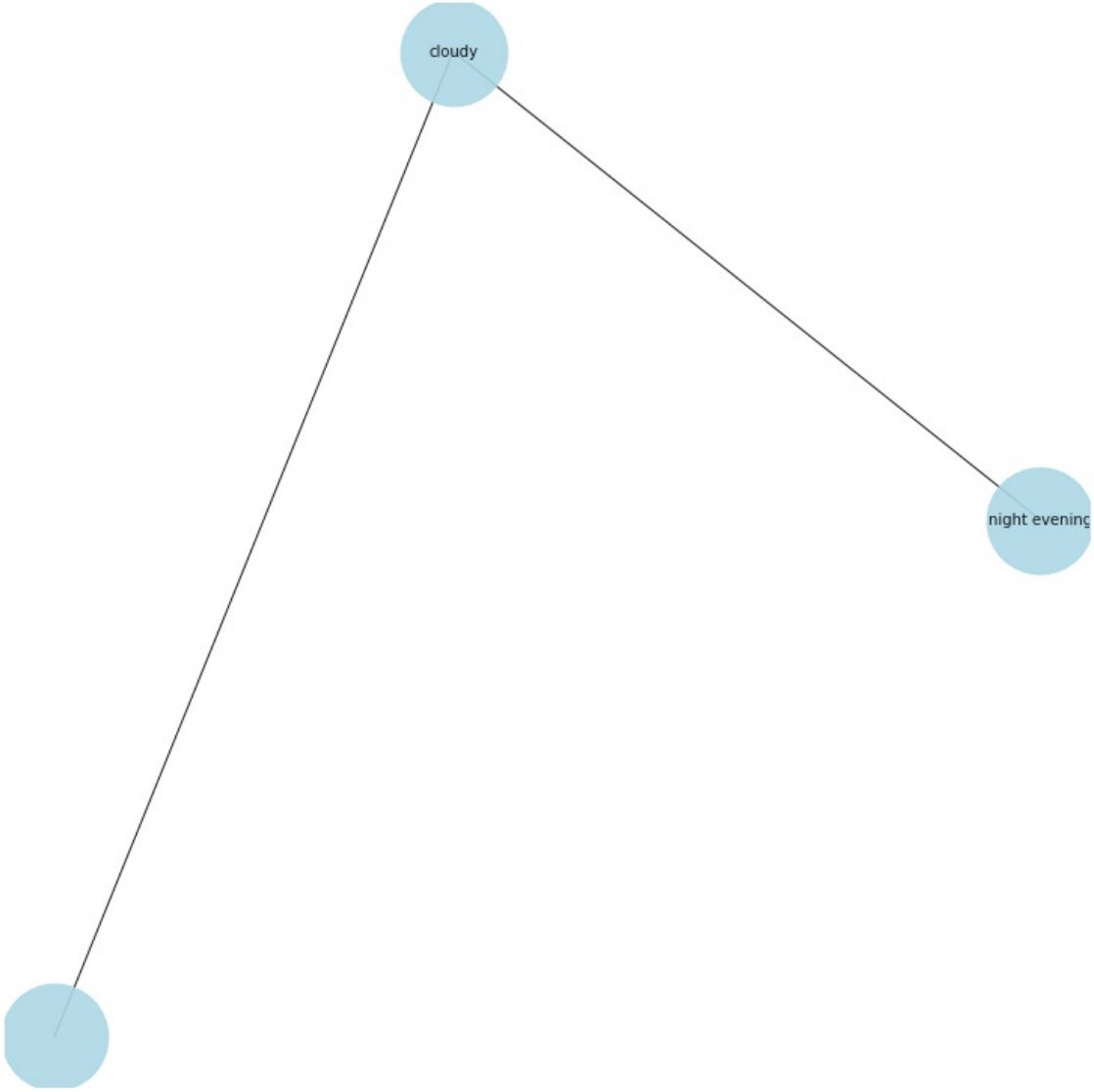

```
Foggy starting overnight continuing until morning and breezy starting in the morning continuing until afternoon.
Foggy -> nsubj
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
Foggy , continue , morning morning afternoon
```



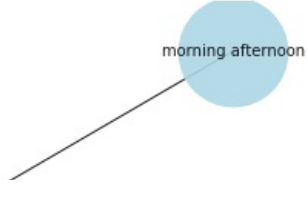

```
Mostly cloudy starting overnight and breezy in the afternoon.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
overnight -> advmod
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy start , afternoon
```

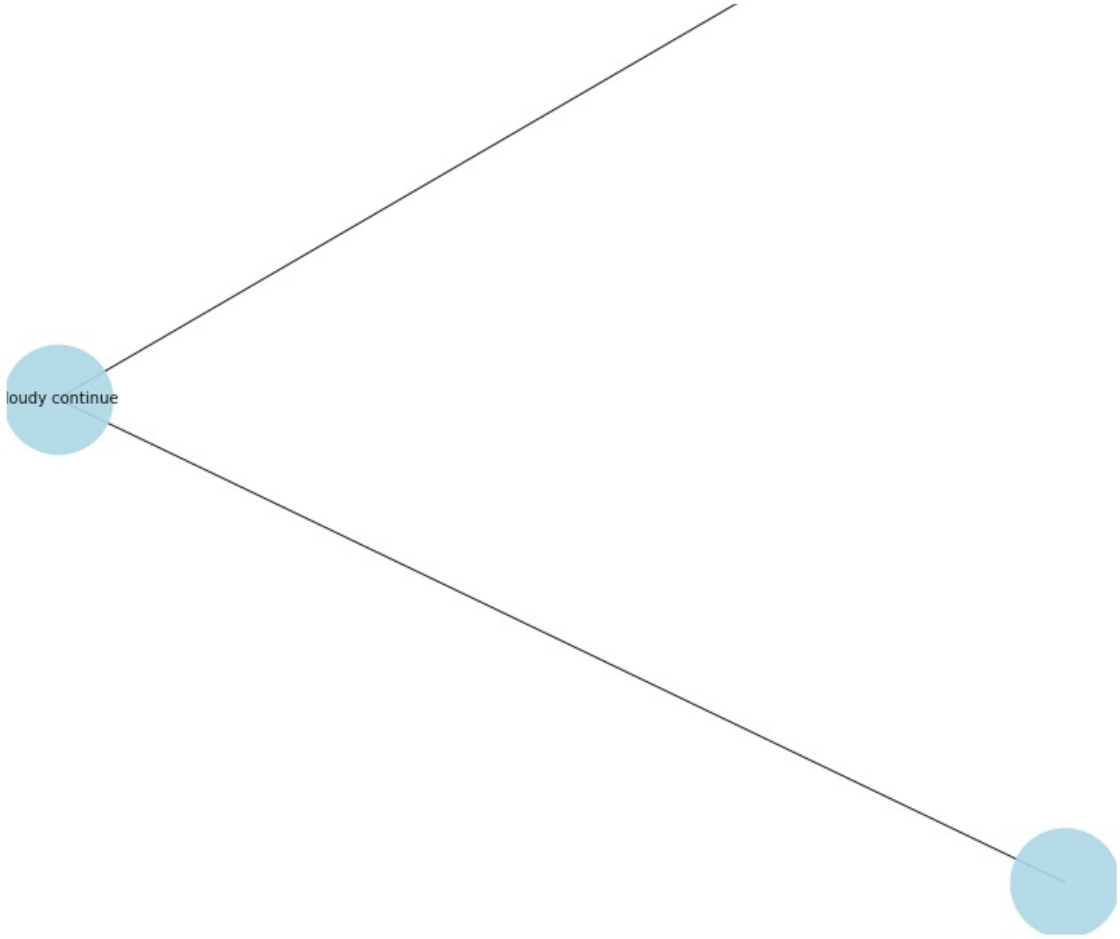


Mostly cloudy until night and breezy in the evening.
Mostly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , night evening

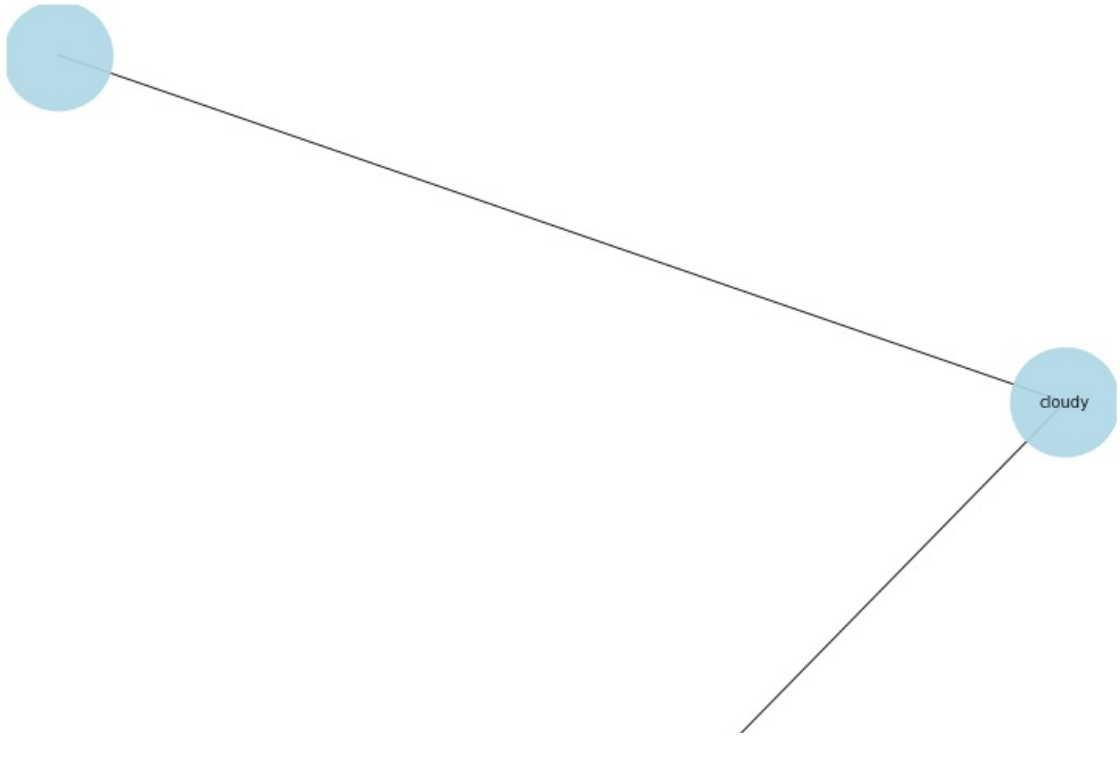


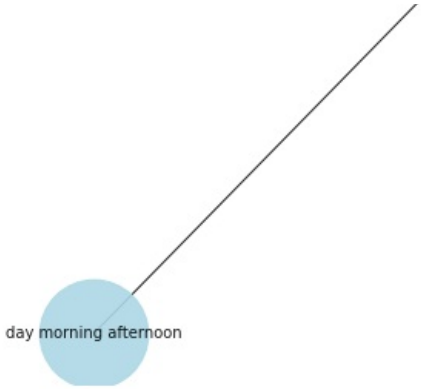
Partly cloudy overnight and breezy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> amod
overnight -> advmod
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
 , cloudy continue , morning afternoon



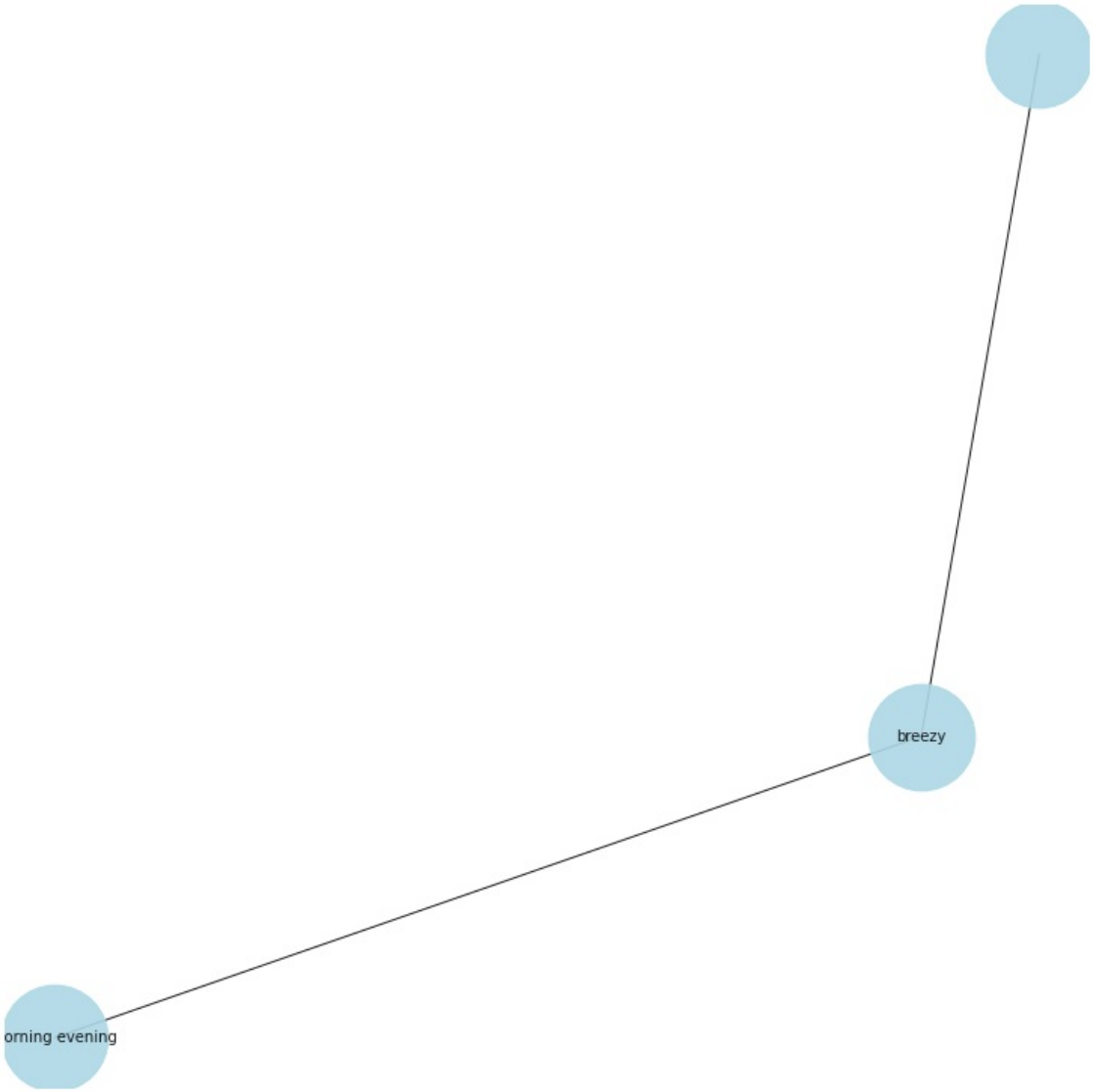



```
Partly cloudy throughout the day and windy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
windy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
 , cloudy , day morning afternoon
```



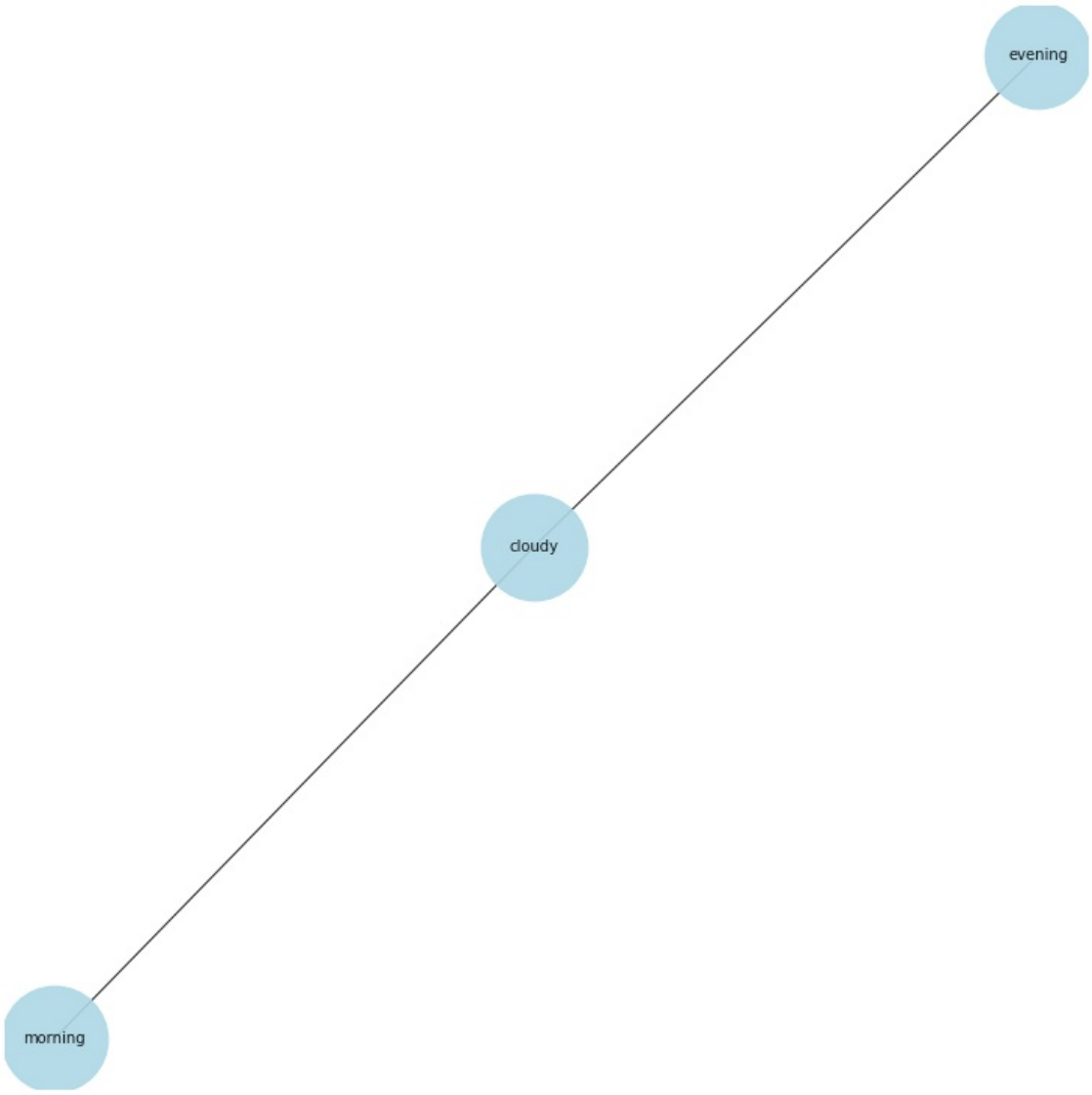



```
Breezy starting overnight continuing until morning and foggy in the evening.
Breezy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
foggy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , breezy , morning evening
```

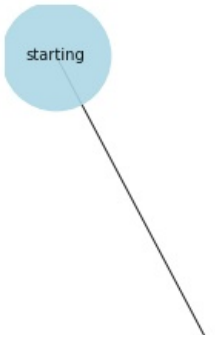


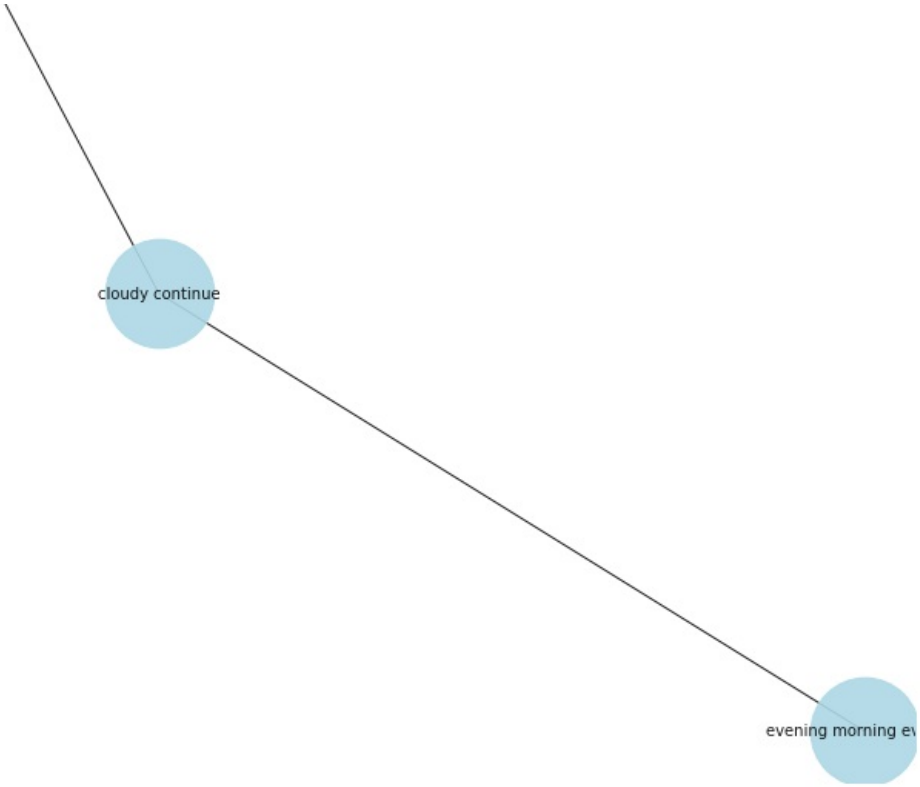

```
Mostly cloudy until evening and breezy starting overnight continuing until morning.
Mostly -> advmod
cloudy -> ROOT
until -> prep
evening -> nsubj
and -> cc
breezy -> conj
starting -> pcomp
overnight -> advmod
```

```
continuing -> xcomp
until -> prep
morning -> pobj
. -> punct
evening , cloudy , morning
```

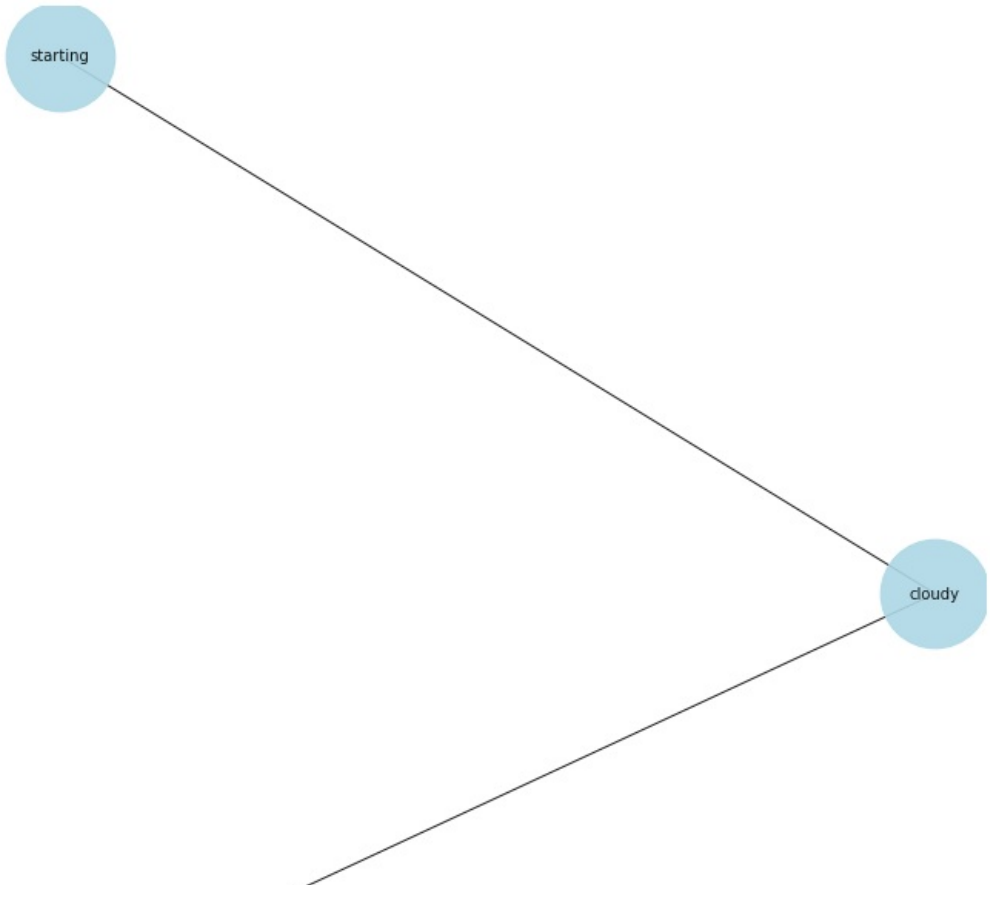


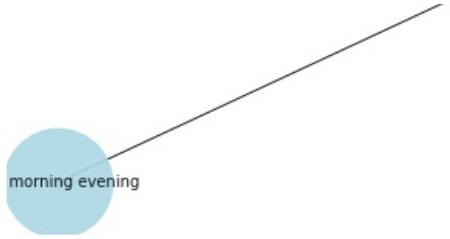

```
Partly cloudy starting overnight continuing until evening and breezy starting in the morning continuing until eve
ning.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
starting , cloudy continue , evening morning evening
```



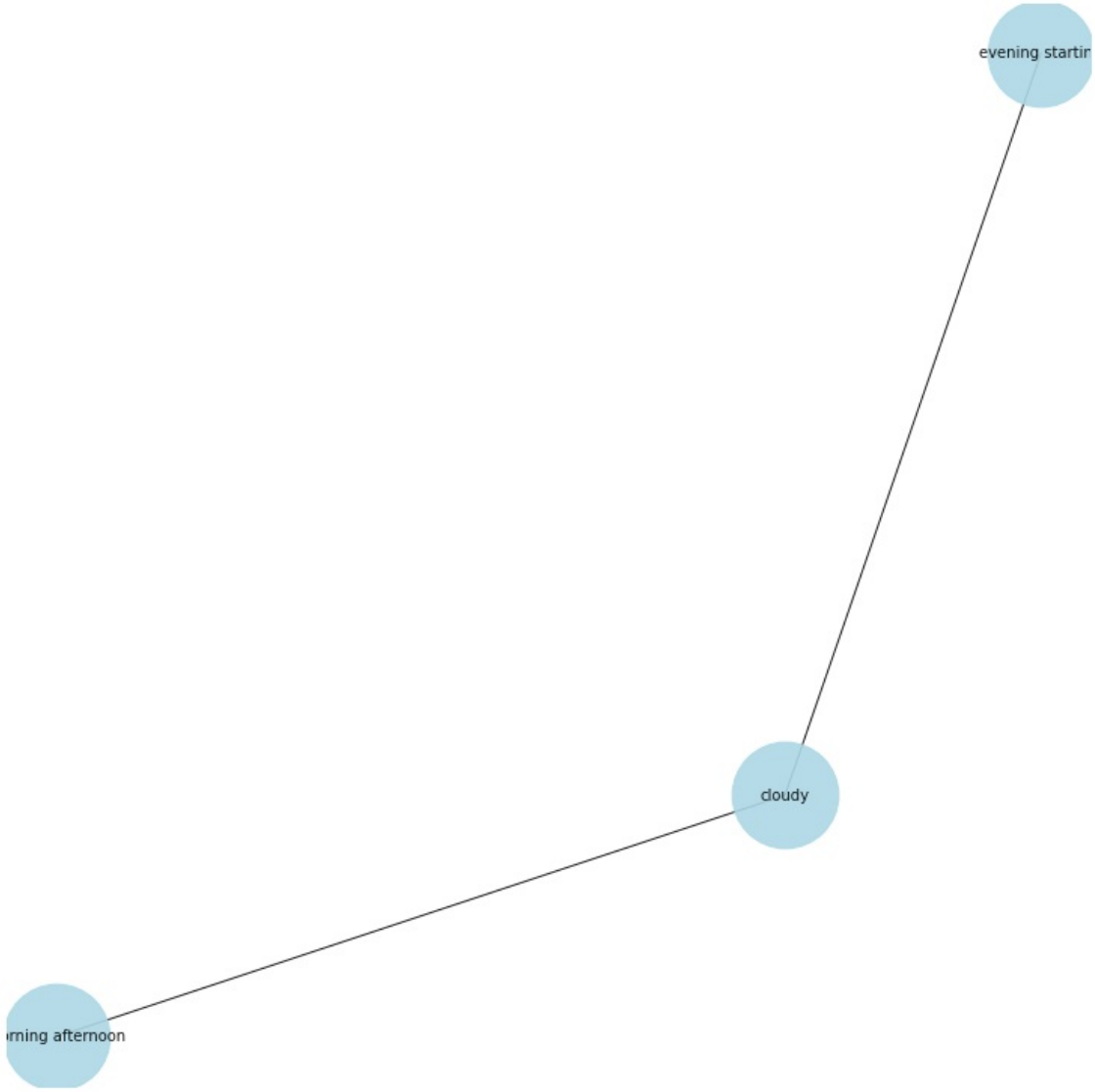



```
Mostly cloudy throughout the day and windy starting in the morning continuing until evening.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
windy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
starting , cloudy , day morning evening
```

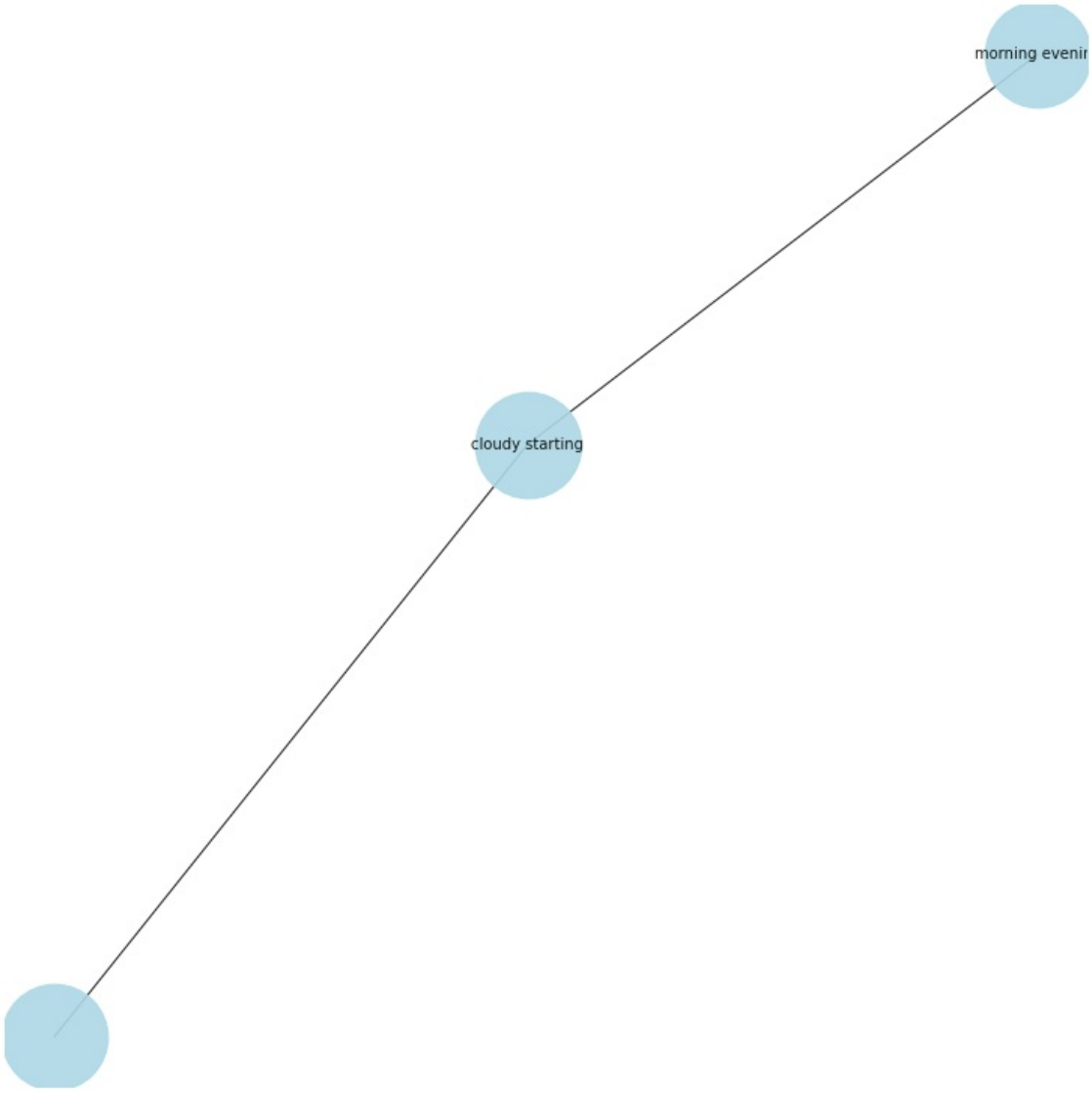




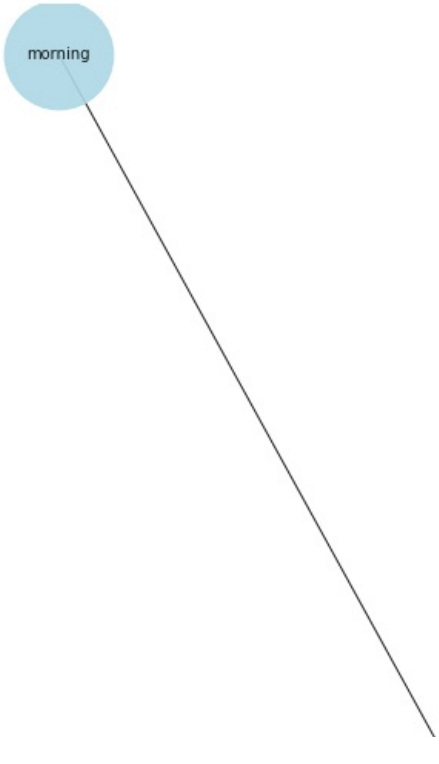
Mostly cloudy until evening and windy starting in the morning continuing until afternoon.
Mostly -> advmod
cloudy -> ROOT
until -> prep
evening -> nsubj
and -> cc
windy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> pcomp
until -> prep
afternoon -> pobj
. -> punct
evening starting , cloudy , morning afternoon

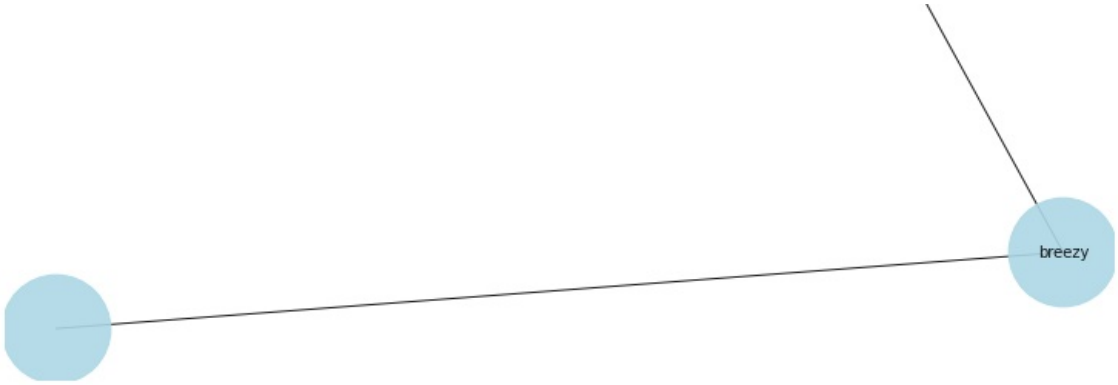


Mostly cloudy starting in the morning and breezy in the evening.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy starting , morning evening

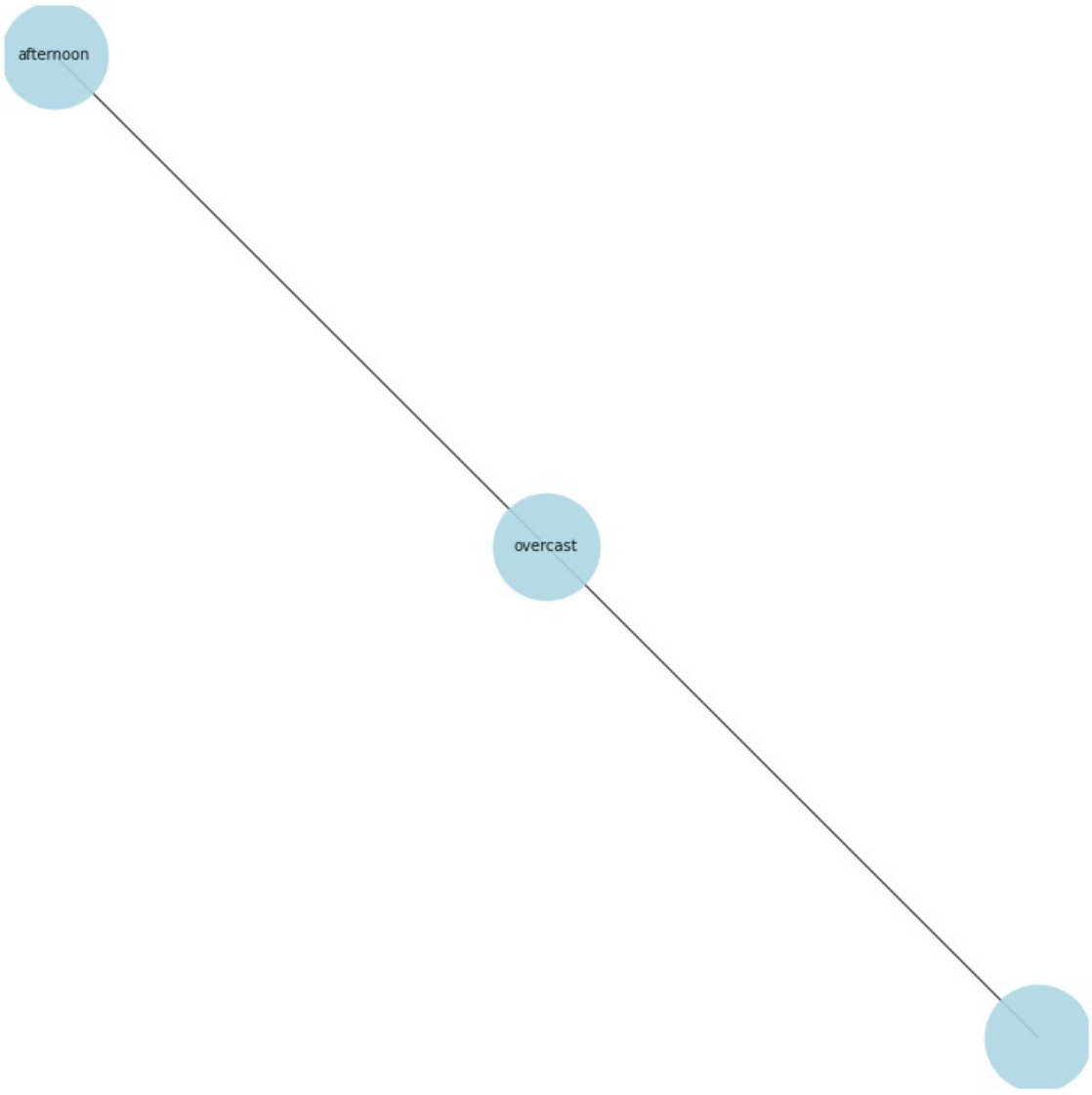



```
Breezy starting overnight continuing until morning.
Breezy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
. -> punct
 , breezy , morning
```



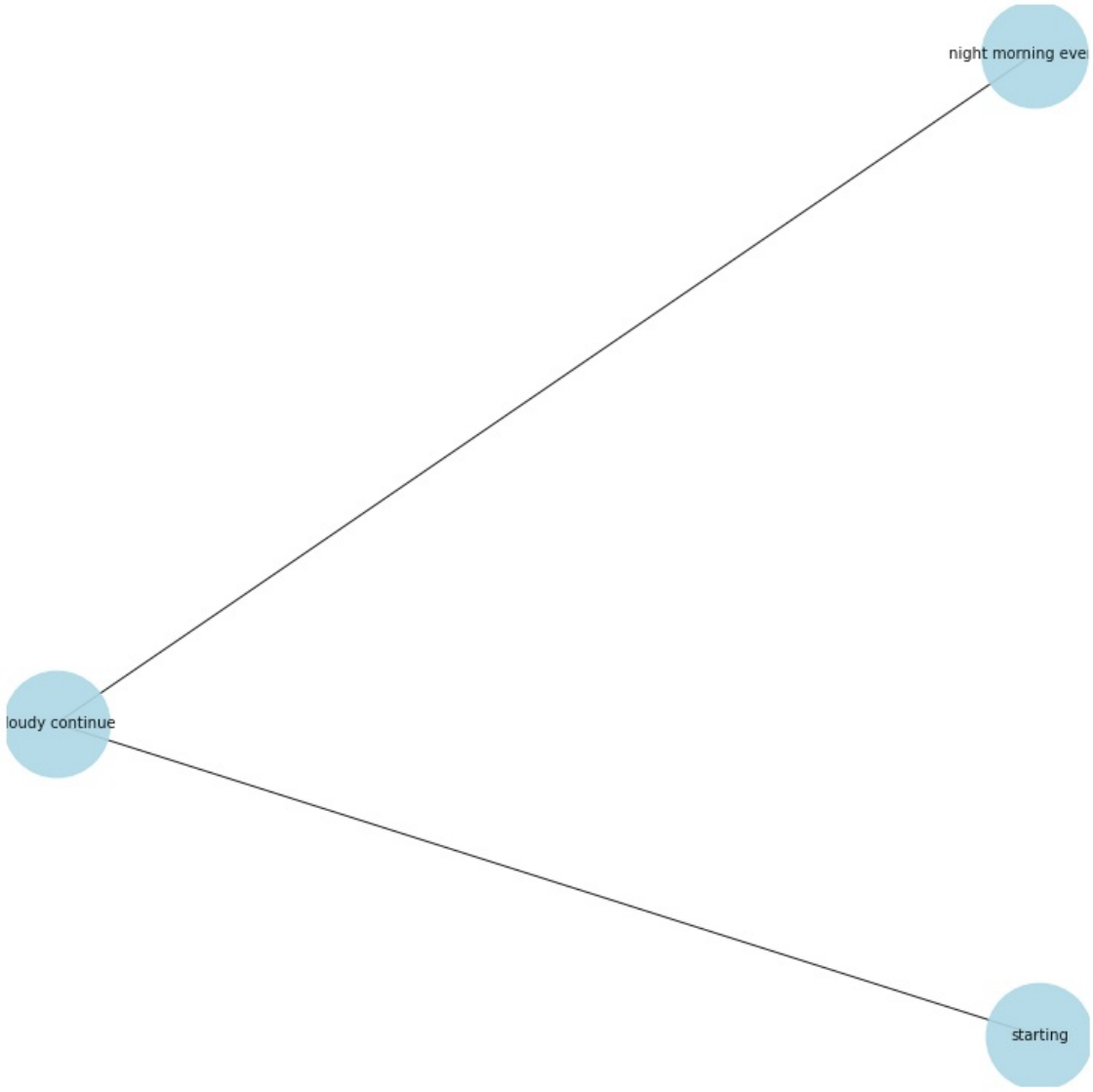



```
Overcast starting in the afternoon.
Overcast -> ROOT
starting -> acl
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , overcast , afternoon
```

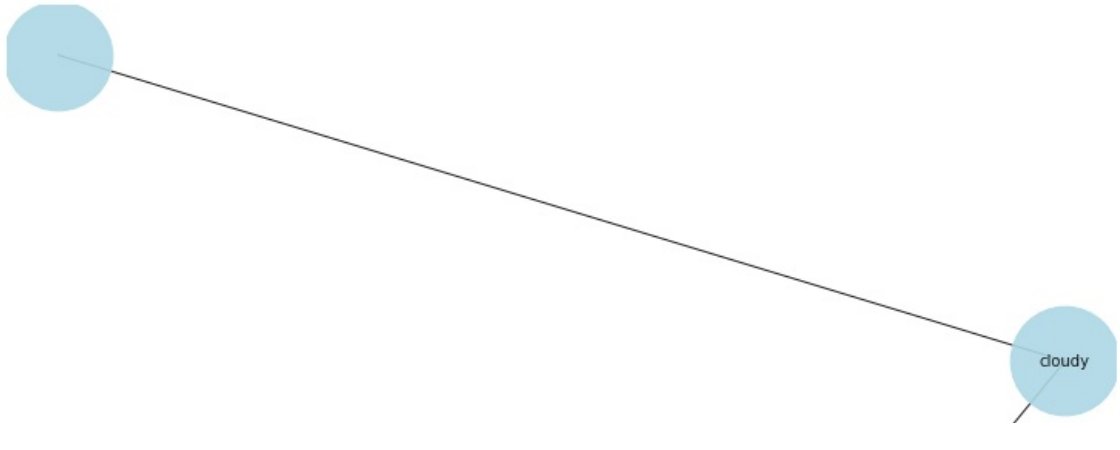


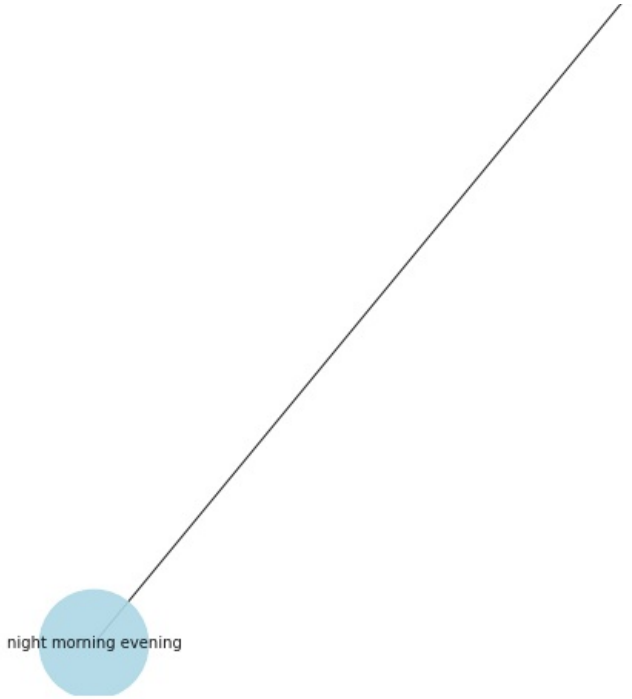

```
Mostly cloudy starting overnight continuing until night and breezy starting in the morning continuing until eveni
ng.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> advcl
```

```
until -> prep
evening -> pobj
. -> punct
starting , cloudy continue , night morning evening
```

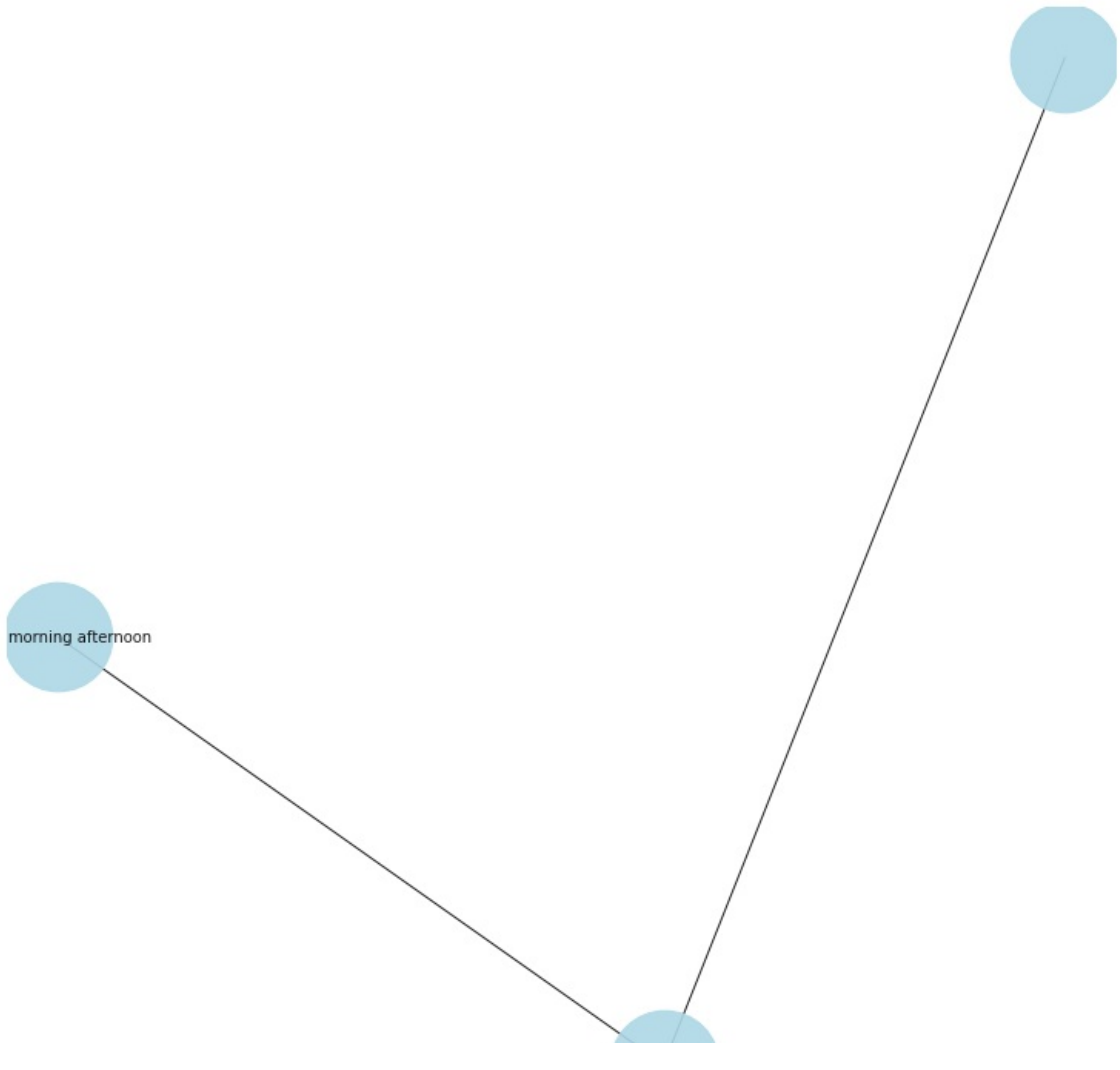



```
Mostly cloudy until night and breezy starting in the morning continuing until evening.
Mostly -> advmod
cloudy -> ROOT
until -> mark
night -> pobj
and -> cc
breezy -> conj
starting -> pcomp
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
 , cloudy , night morning evening
```

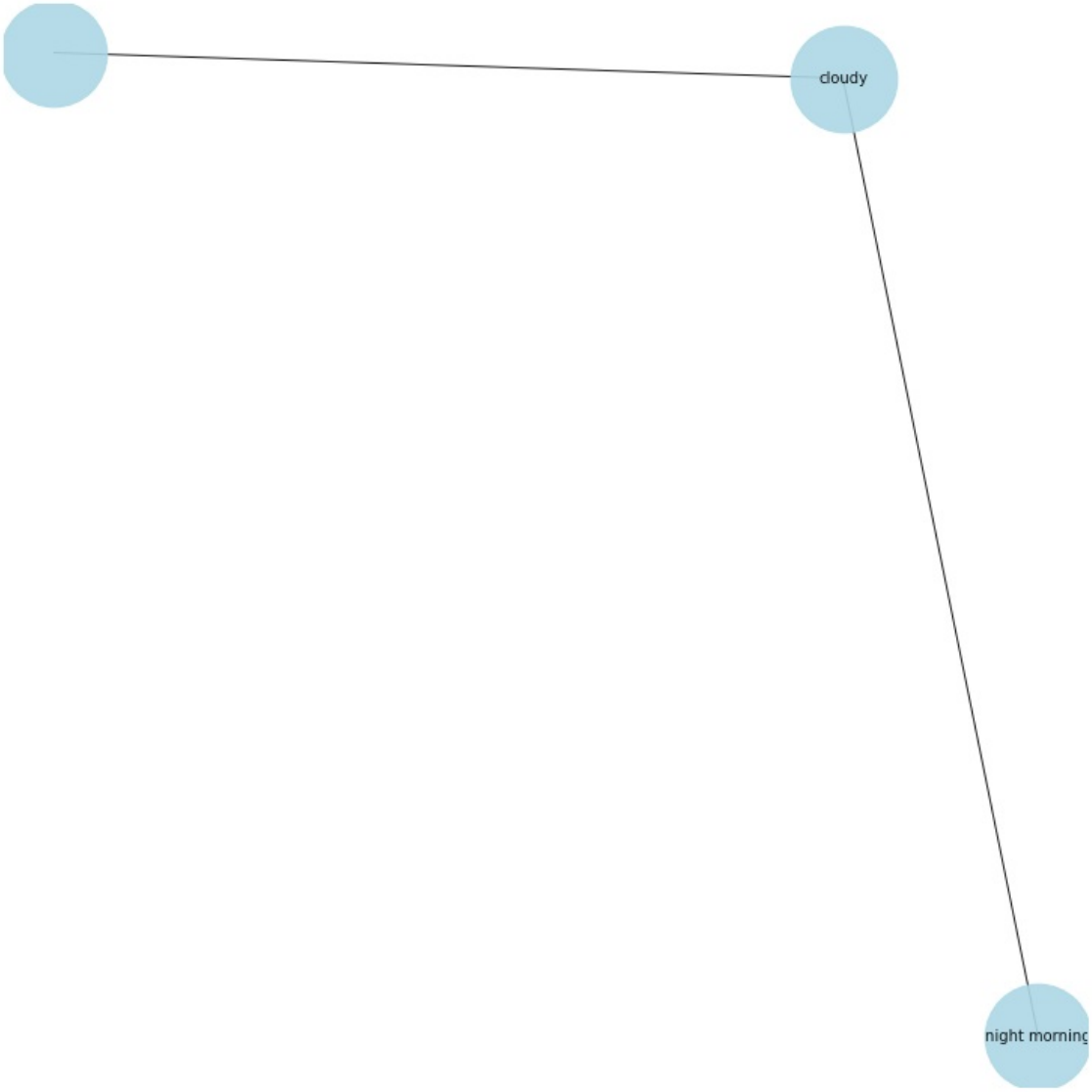




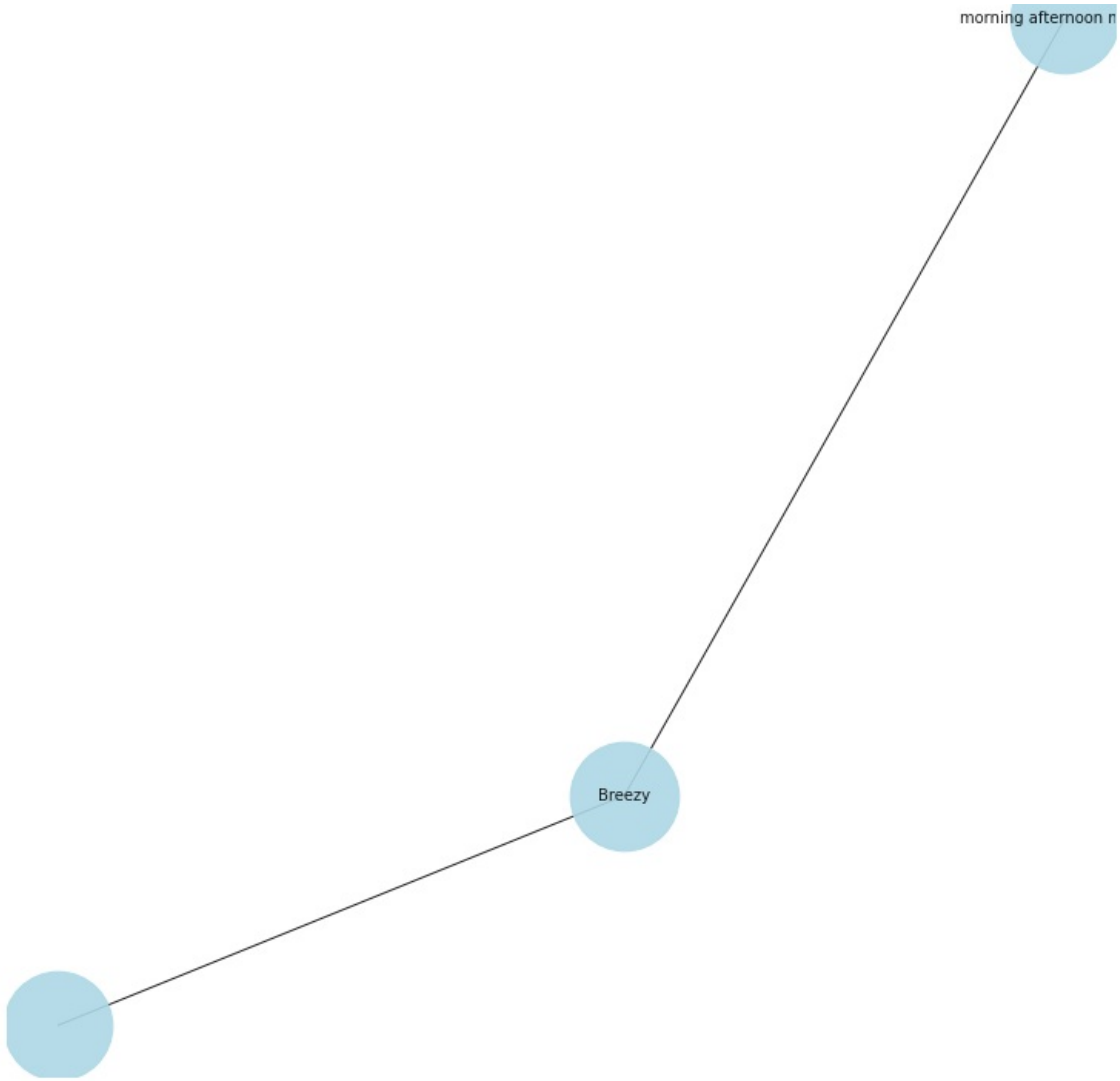

```
Mostly cloudy until night and windy starting in the morning continuing until afternoon.
Mostly -> advmod
cloudy -> advmod
until -> mark
night -> pobj
and -> cc
windy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
 , continue , night morning afternoon
```



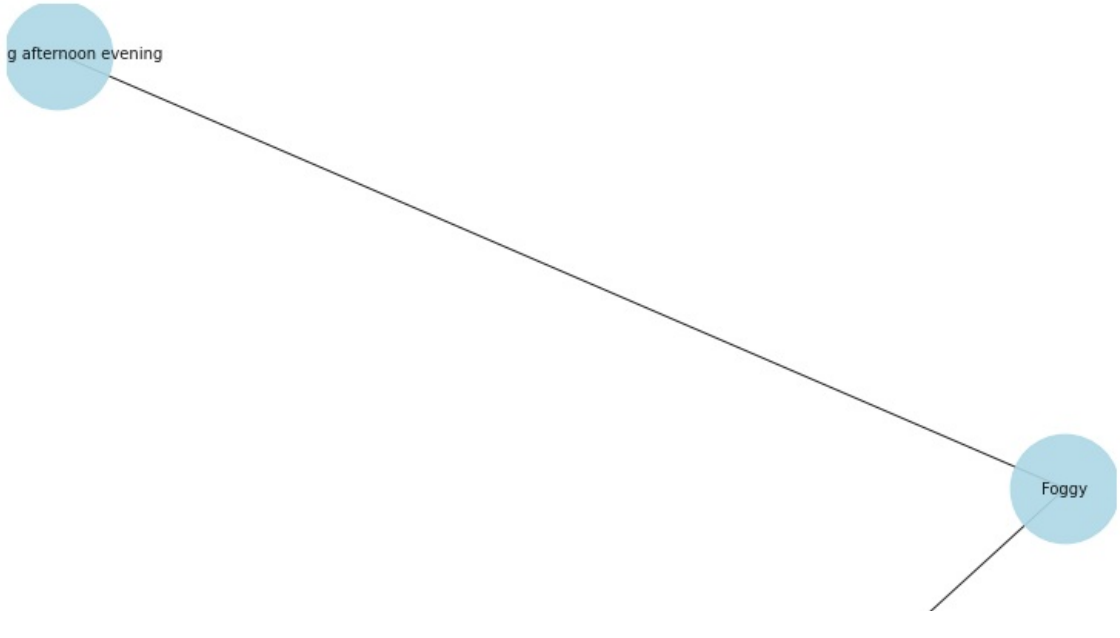

```
Mostly cloudy until night and breezy starting overnight continuing until morning.
Mostly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
starting -> aux
overnight -> advmod
continuing -> advcl
until -> prep
morning -> pobj
. -> punct
 , cloudy , night morning
```

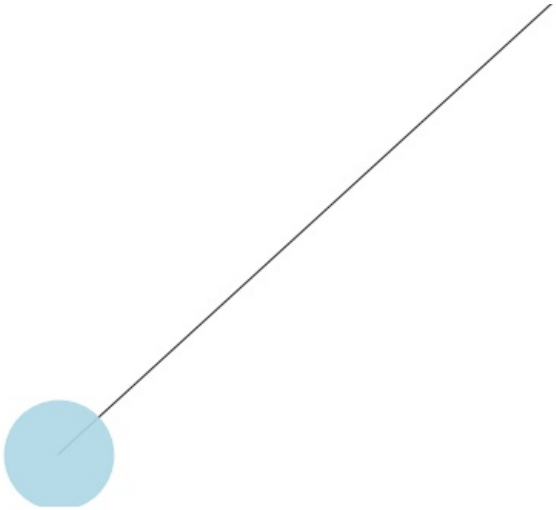



```
Breezy starting in the morning continuing until afternoon and mostly cloudy starting in the morning.
Breezy -> ROOT
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> acl
until -> prep
afternoon -> pobj
and -> cc
mostly -> advmod
cloudy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
. -> punct
 , Breezy , morning afternoon morning
```

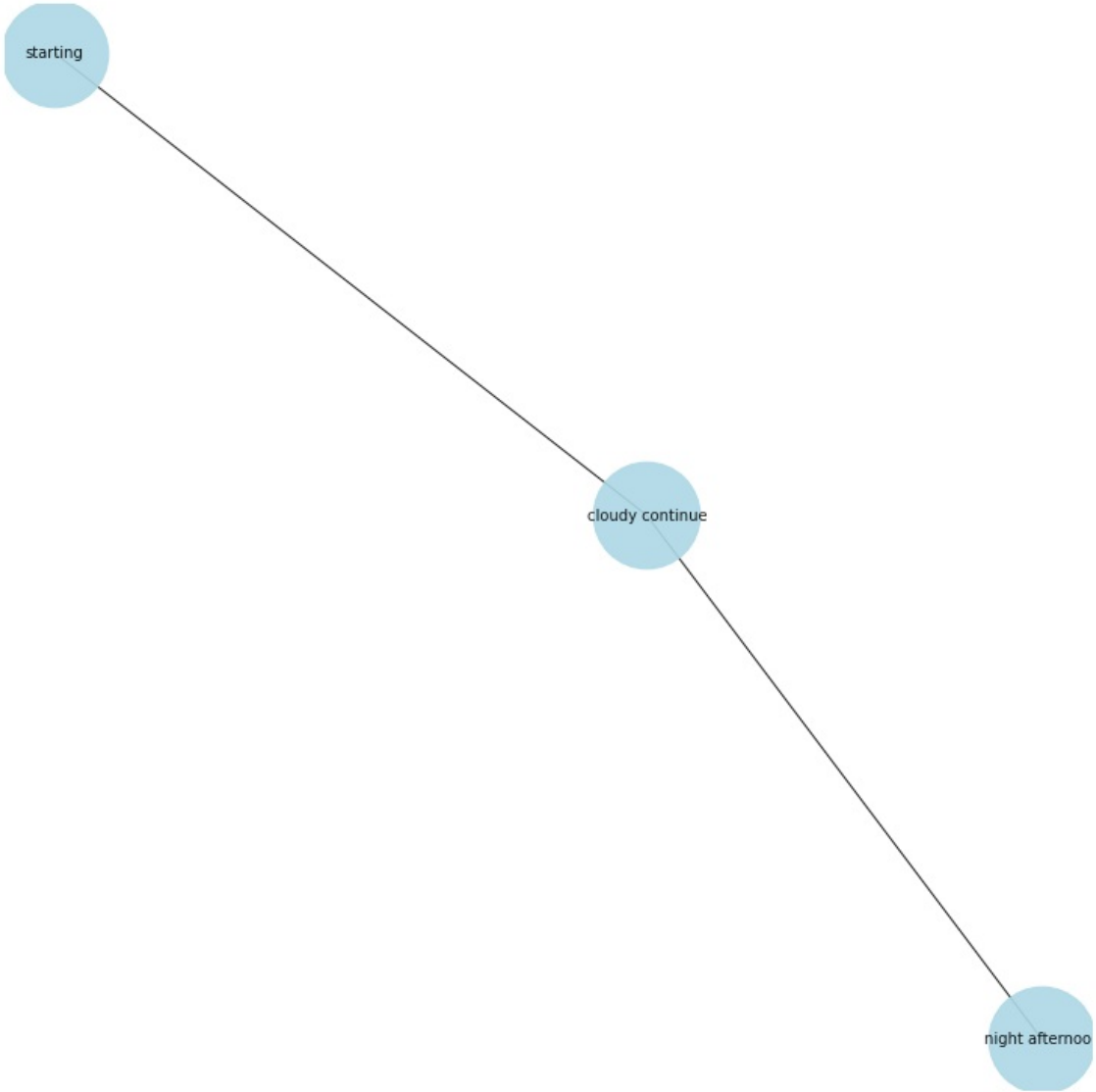


Foggy starting in the morning continuing until afternoon and breezy starting in the evening.
Foggy -> ROOT
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> acl
until -> prep
afternoon -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , Foggy , morning afternoon evening



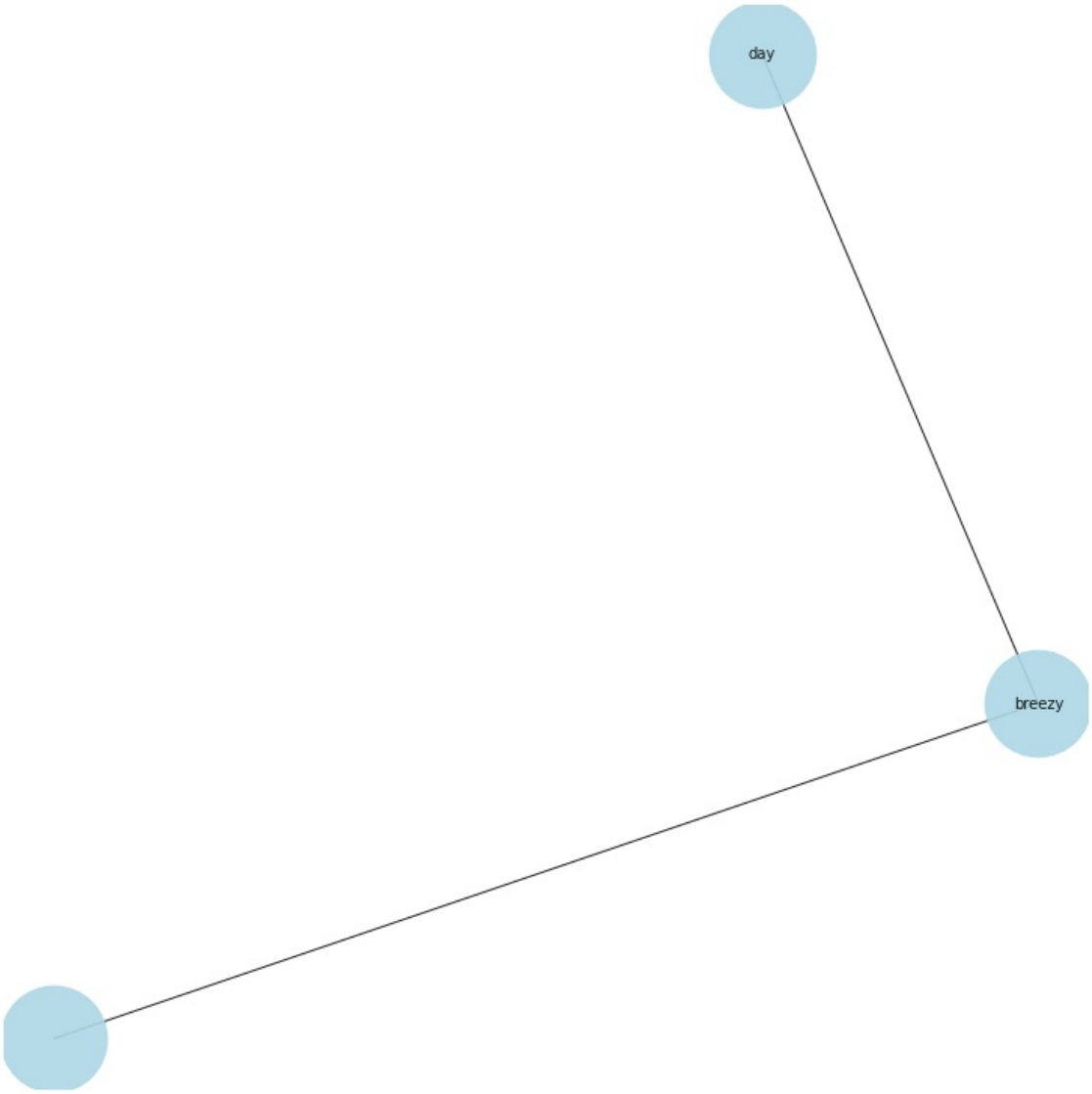


```
Partly cloudy starting overnight continuing until night and breezy in the afternoon.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
starting , cloudy continue , night afternoon
```

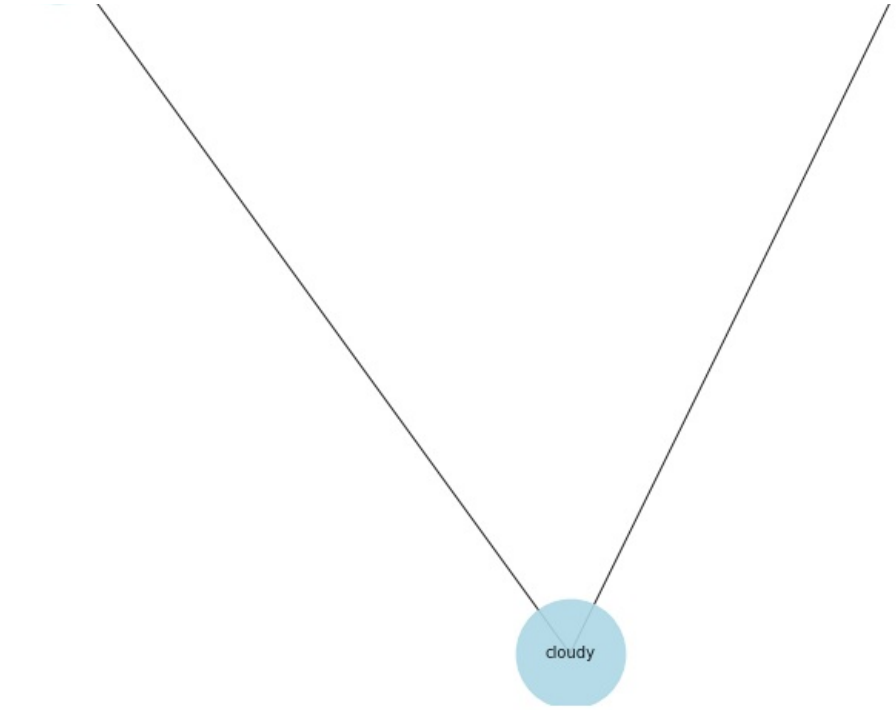



```
Breezy overnight and partly cloudy throughout the day.
Breezy -> ROOT
overnight -> advmod
and -> cc
partly -> advmod
```

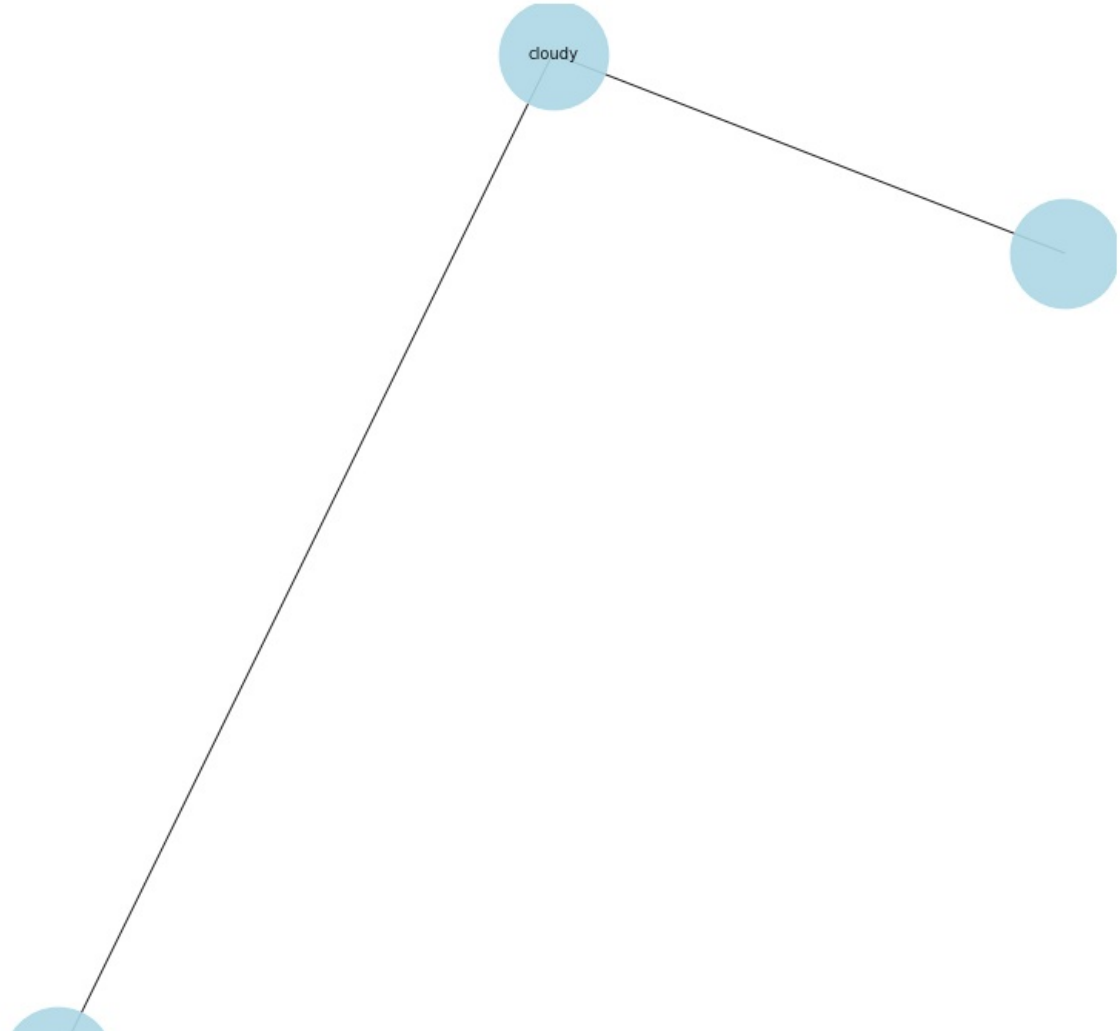
```
cloudy -> conj
throughout -> prep
the -> det
day -> pobj
. -> punct
 , breezy , day
```

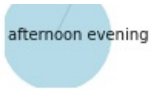



```
Mostly cloudy until night and breezy starting in the afternoon continuing until night.
Mostly -> advmod
cloudy -> ROOT
until -> mark
night -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> advcl
until -> prep
night -> pobj
. -> punct
 , cloudy , night afternoon night
```

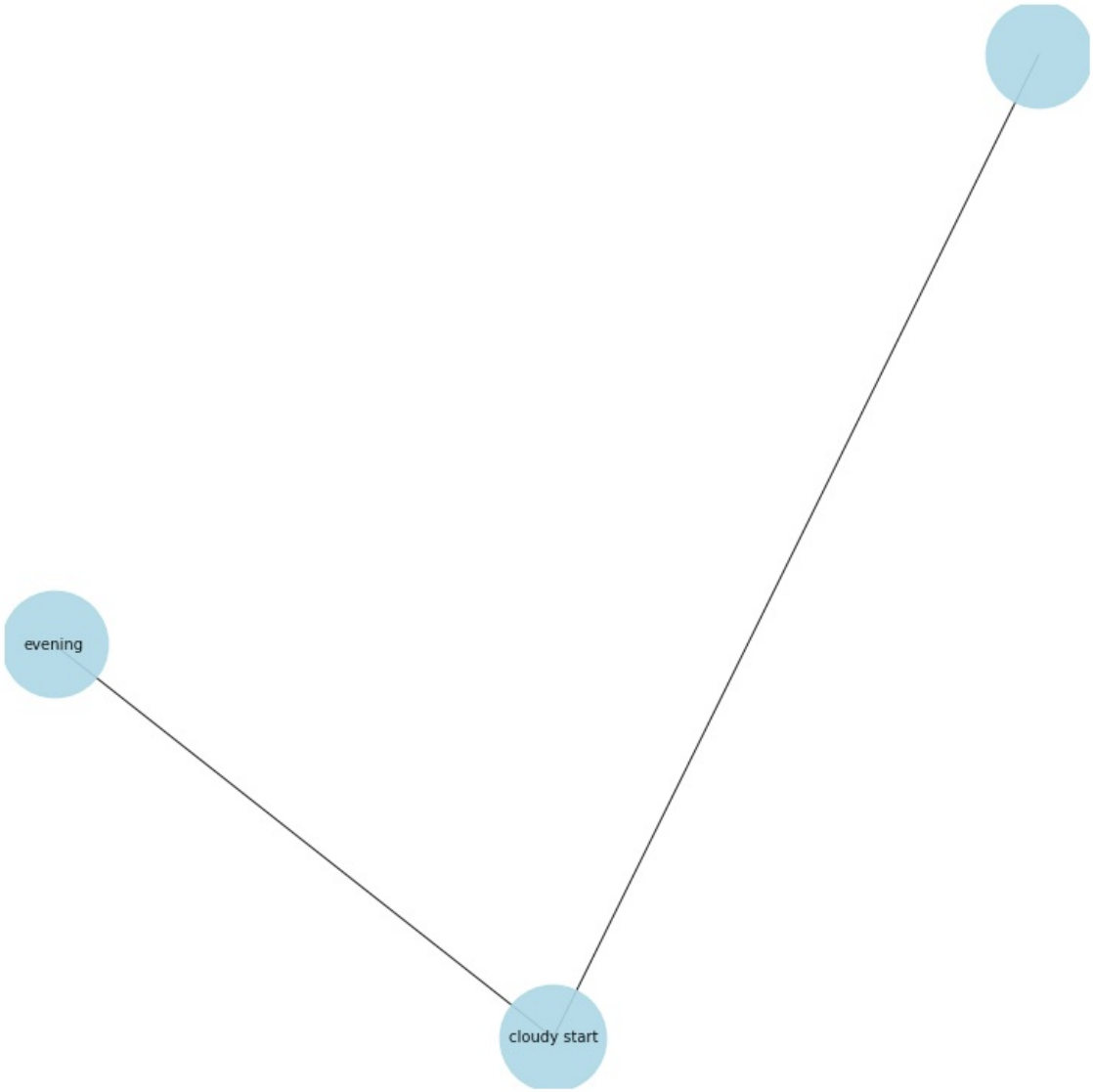


Mostly cloudy until night and breezy starting in the afternoon continuing until evening.
Mostly -> advmod
cloudy -> ROOT
until -> mark
night -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
 , cloudy , night afternoon evening

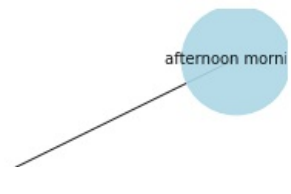


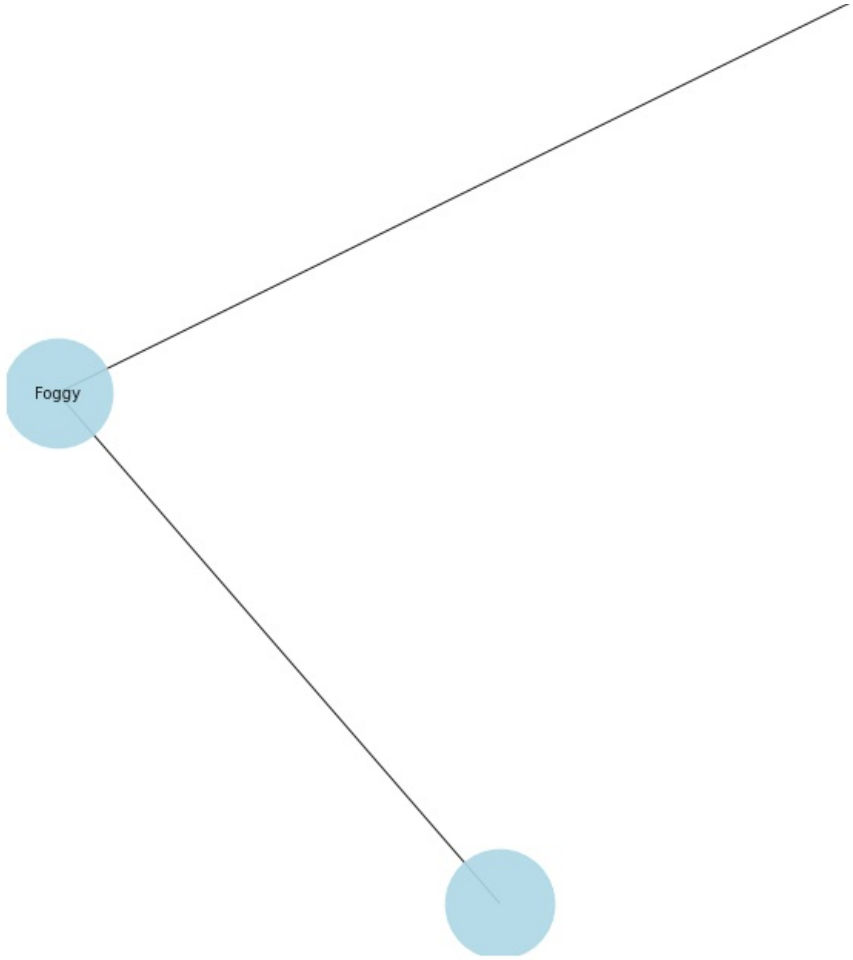



```
Partly cloudy starting overnight and breezy starting in the evening.
Partly -> advmod
cloudy -> amod
starting -> ROOT
overnight -> advmod
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy start , evening
```

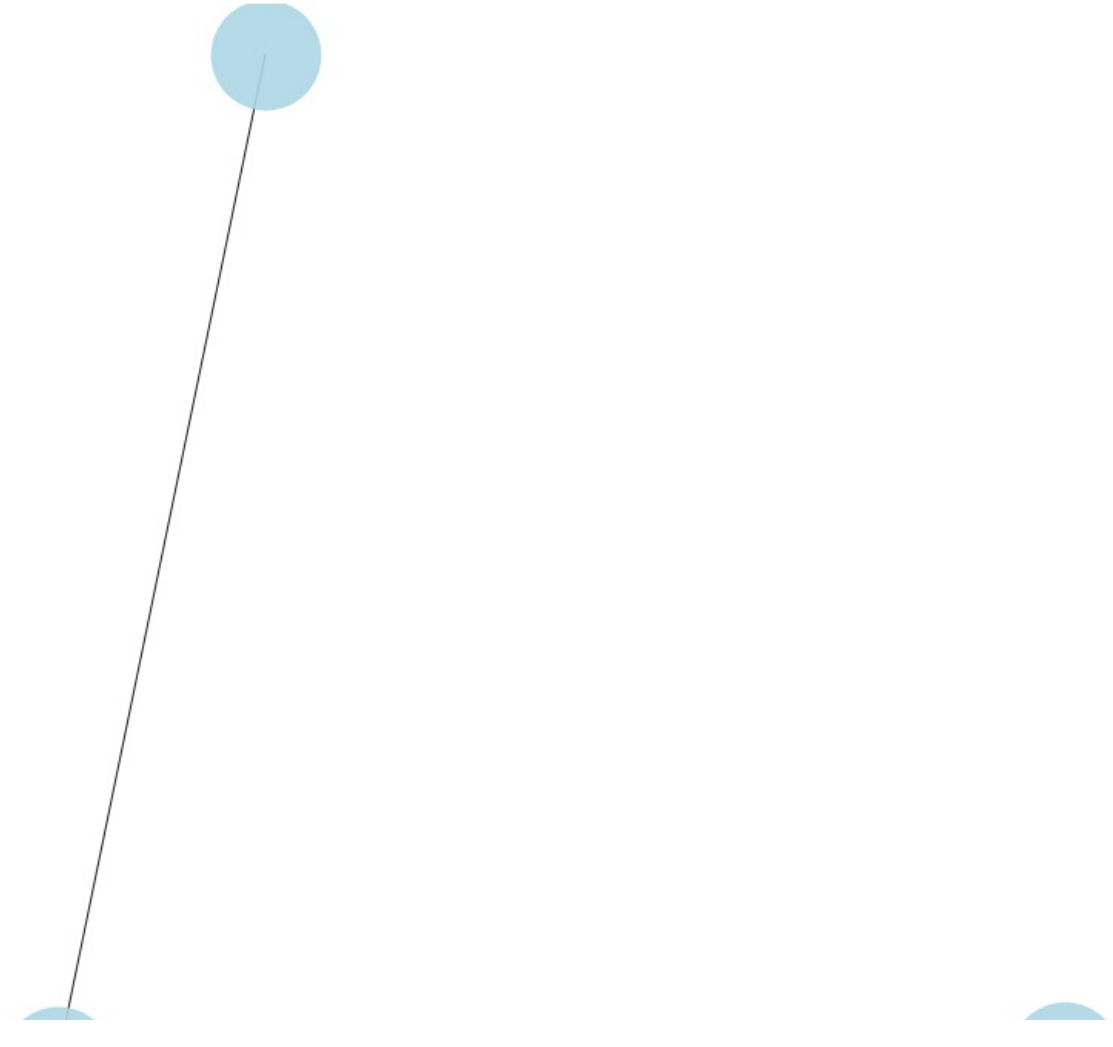



```
Foggy starting overnight continuing until afternoon and breezy in the morning.
Foggy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
afternoon -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , Foggy , afternoon morning
```

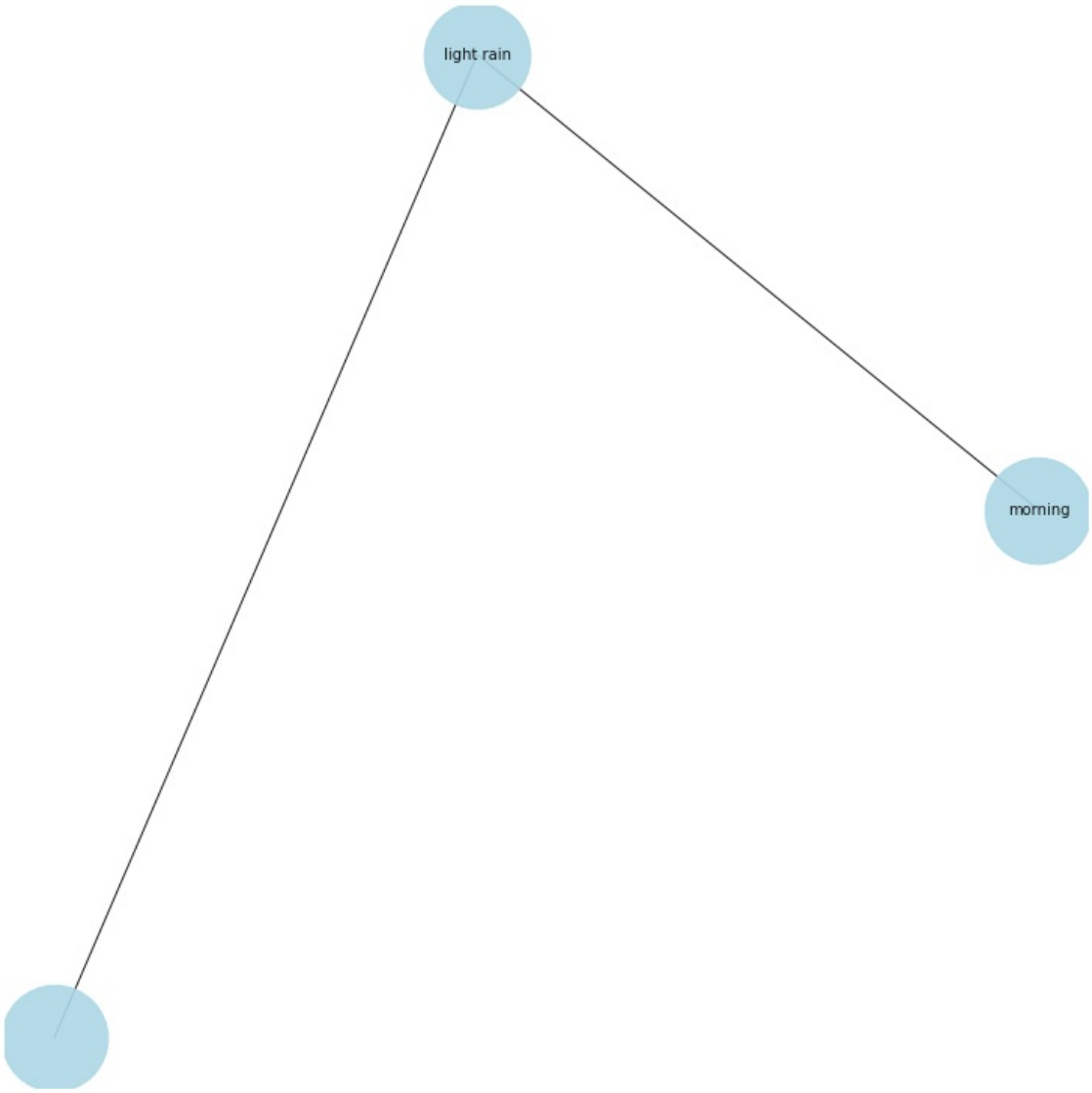




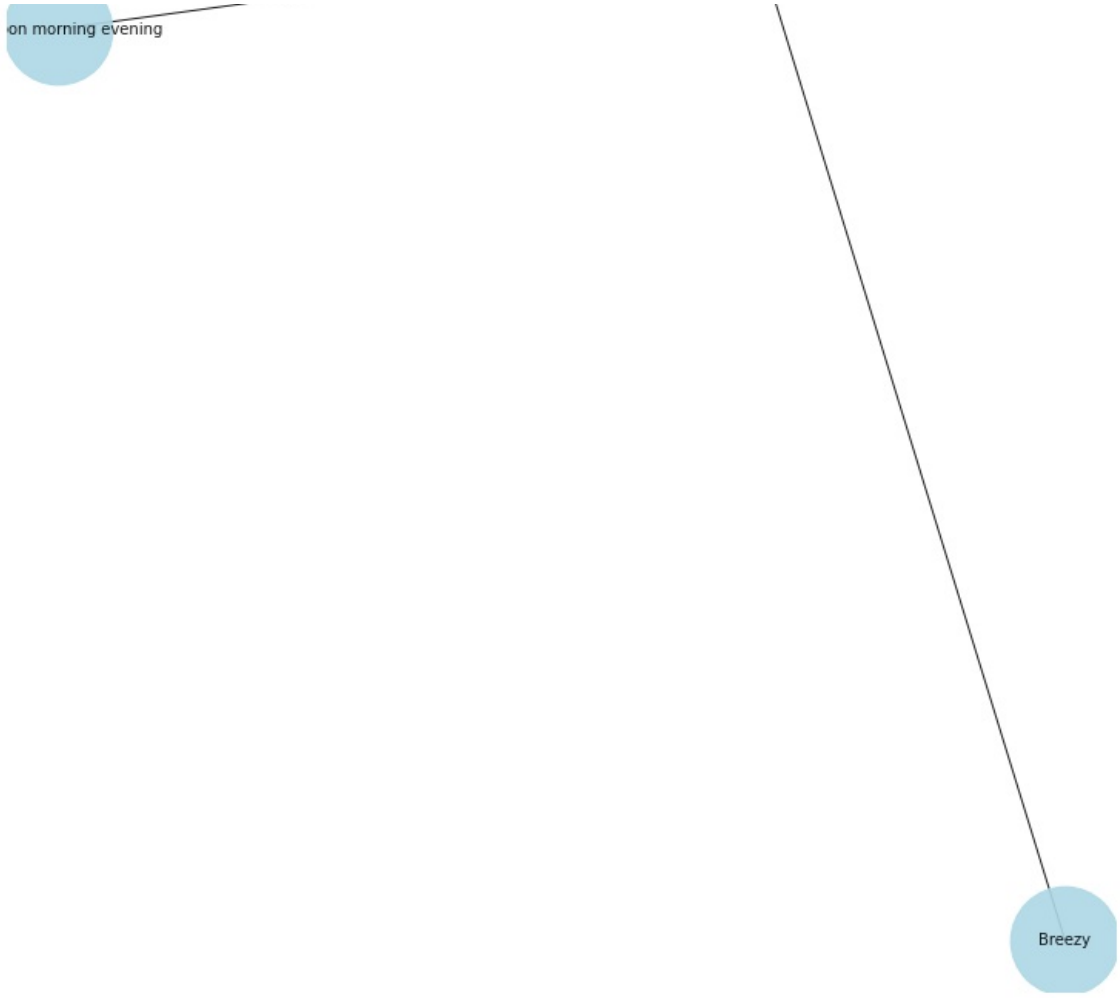

```
Overcast until morning.
Overcast -> ROOT
until -> prep
morning -> pobj
. -> punct
 , overcast , morning
```



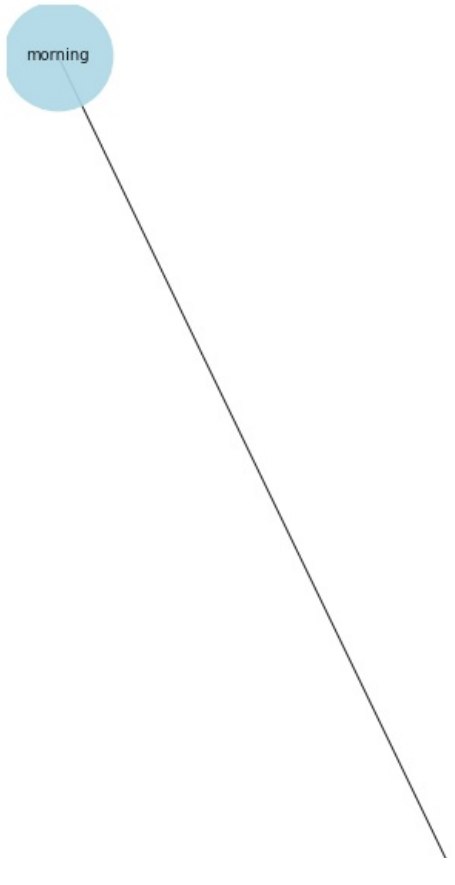

```
Light rain in the morning and afternoon.
Light -> amod
rain -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
afternoon -> conj
. -> punct
 , light rain , morning
```



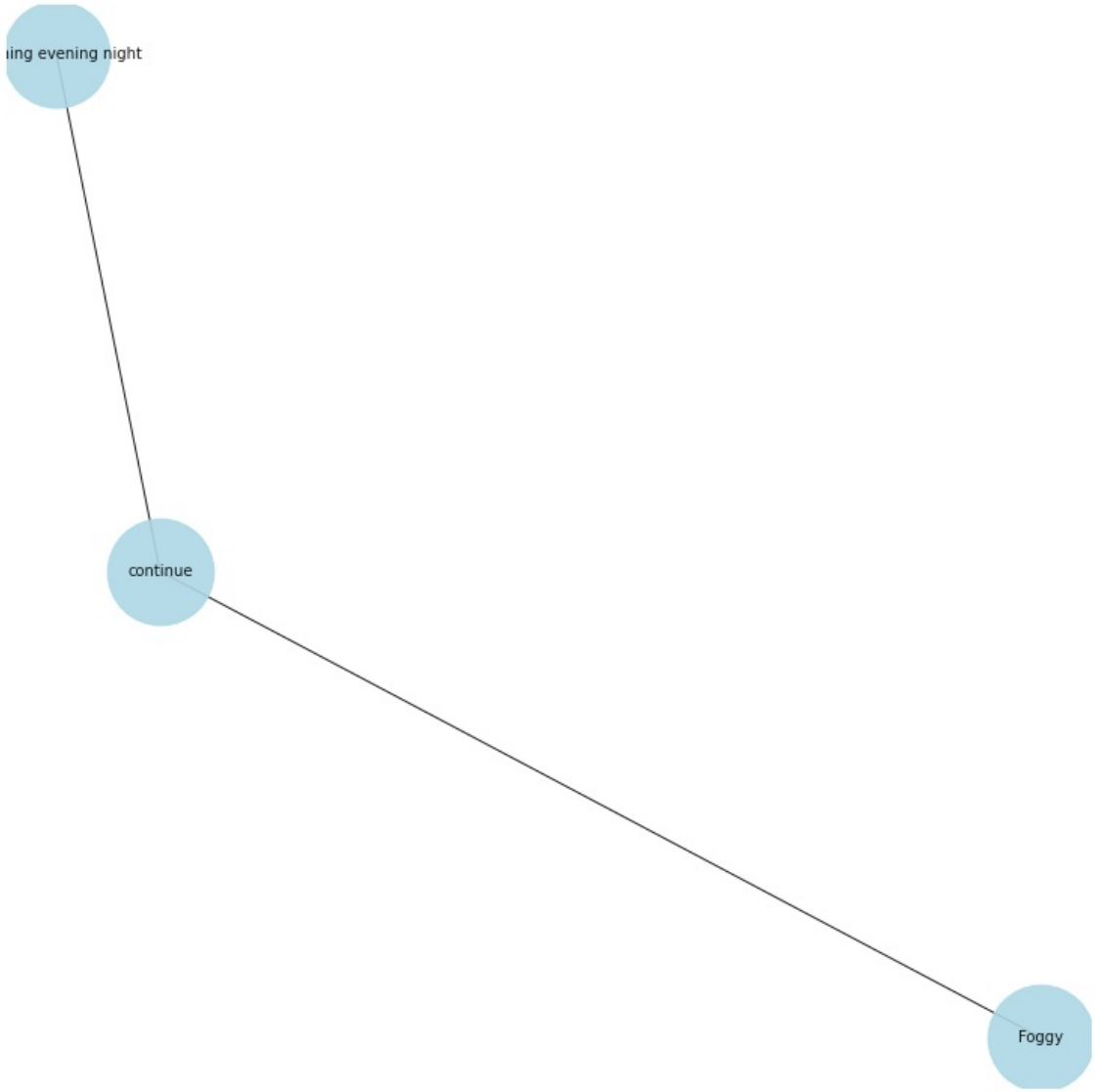

```
Breezy starting overnight continuing until afternoon and foggy starting in the morning continuing until evening.
Breezy -> nsubj
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
afternoon -> pobj
and -> cc
foggy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
Breezy , continue , afternoon morning evening
```




```
Foggy overnight and breezy in the morning.
Foggy -> ROOT
overnight -> advmod
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , Foggy , morning
```

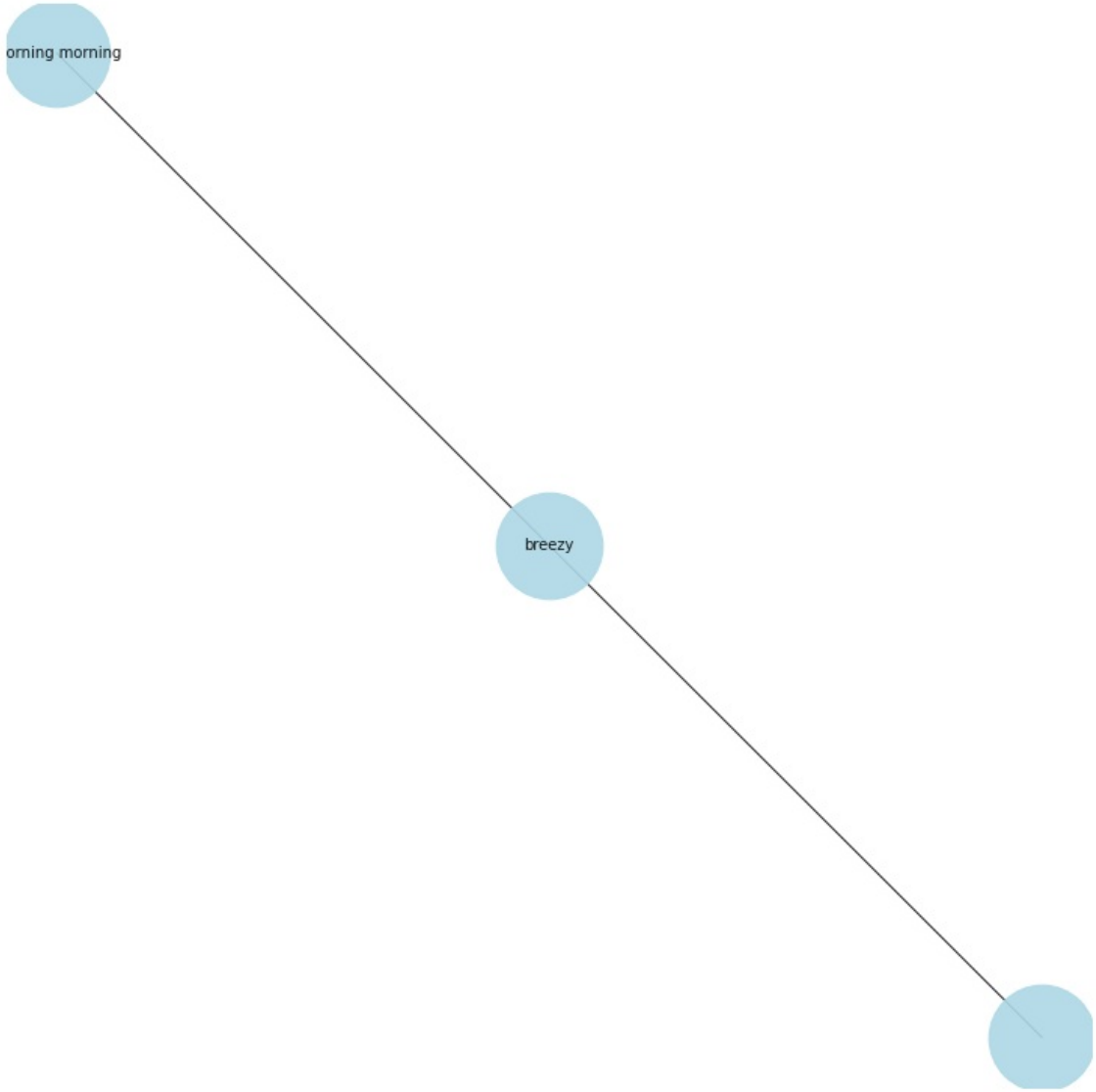


Foggy starting overnight continuing until morning and breezy starting in the evening continuing until night.
Foggy -> nsubj
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
Foggy , continue , morning evening night

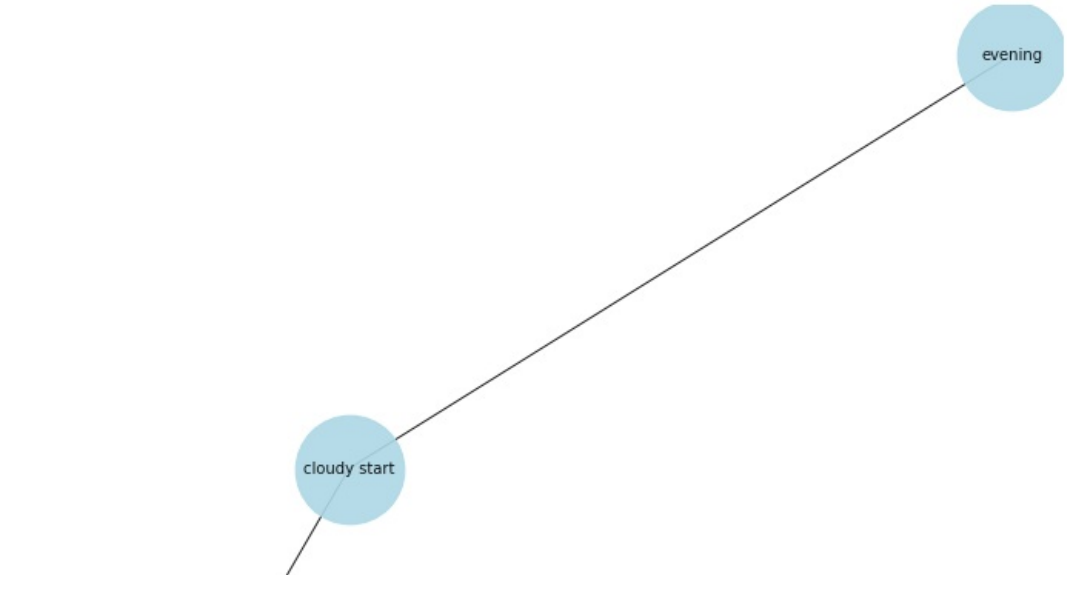


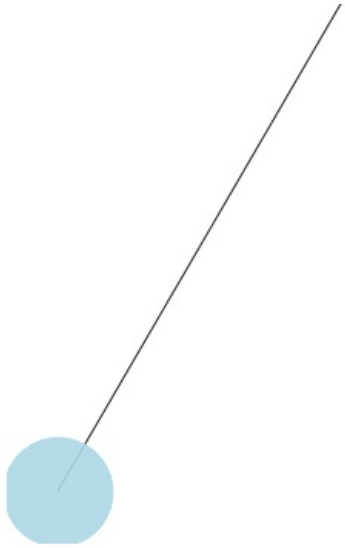
Breezy in the morning and mostly cloudy starting in the morning.
Breezy -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
mostly -> advmod
cloudy -> conj
starting -> acl
in -> prep
the -> det
morning -> pobj

```
. -> punct
 , breezy , morning morning
```

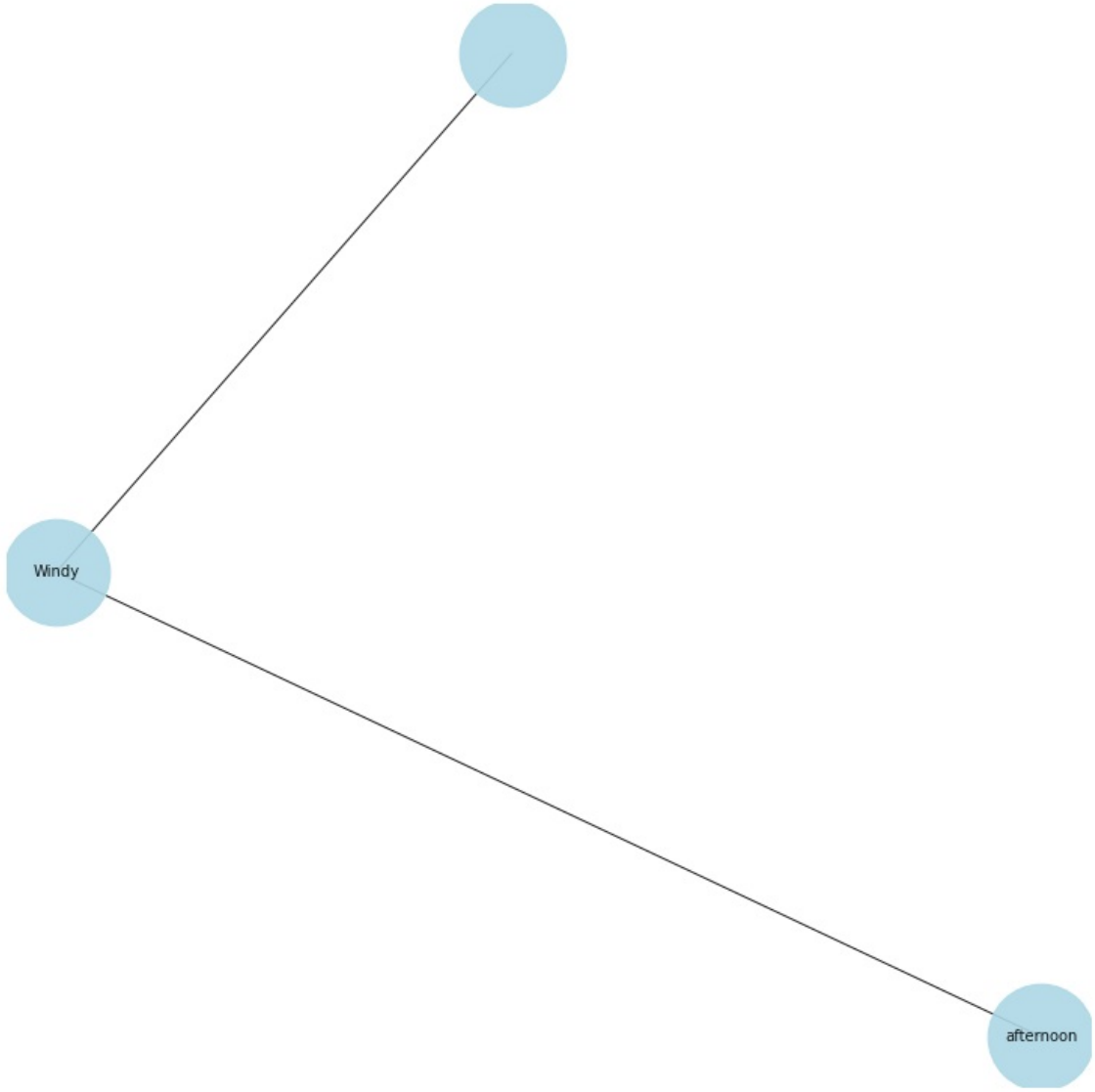



```
Mostly cloudy starting overnight and breezy starting in the evening.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
overnight -> advmod
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy start , evening
```



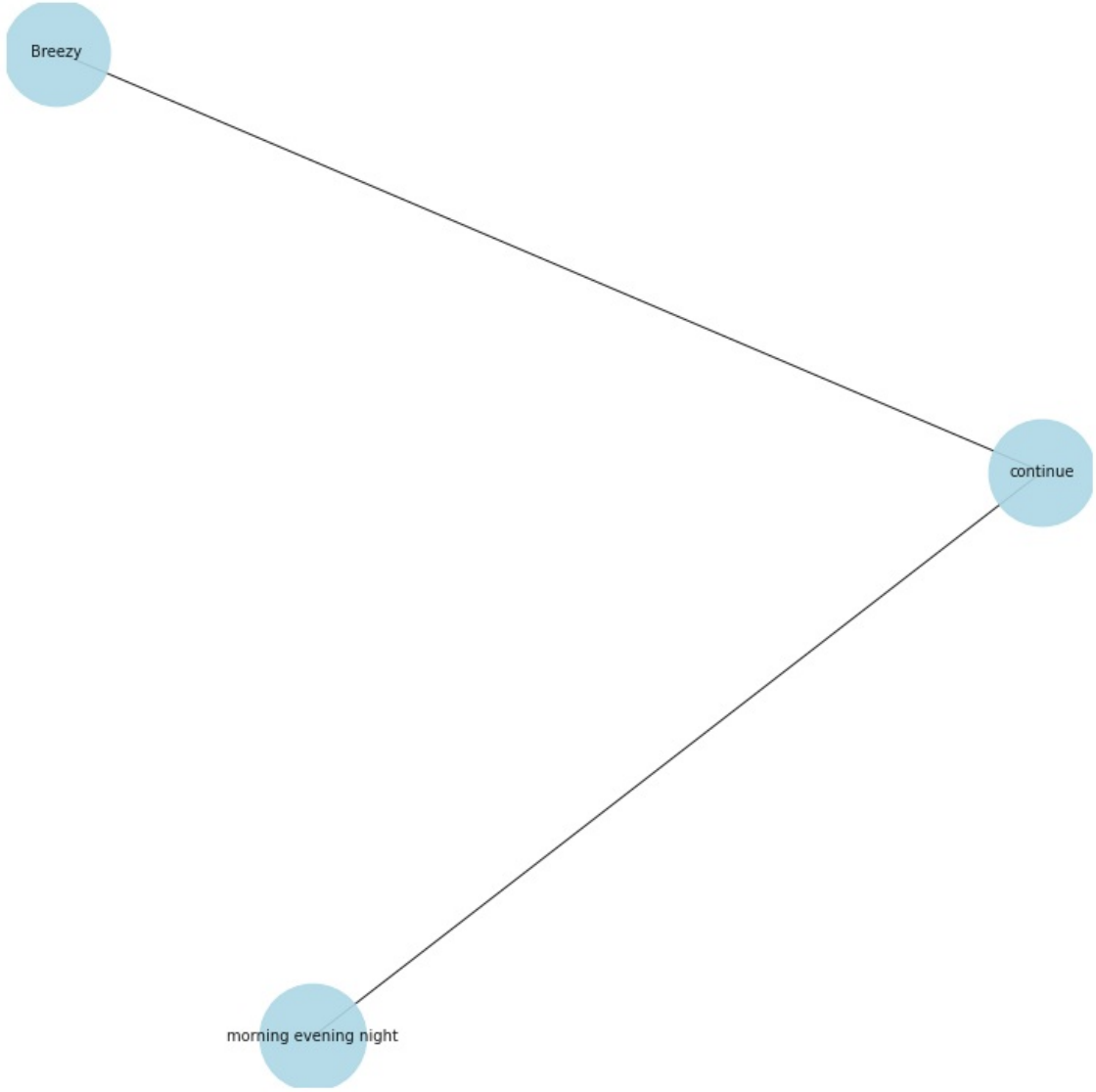


```
Windy in the afternoon.
Windy -> ROOT
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , Windy , afternoon
```

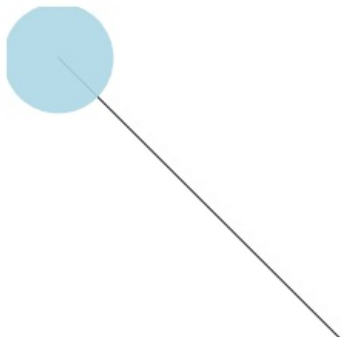


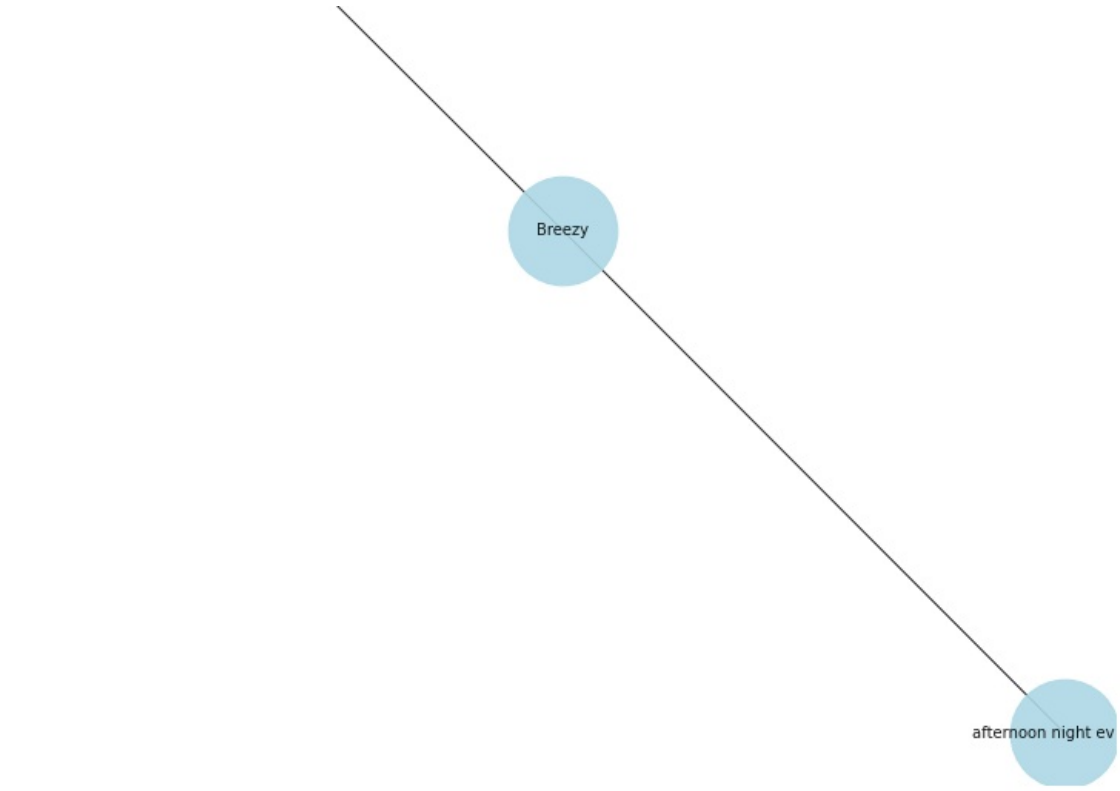

```
Breezy in the morning and partly cloudy starting in the evening continuing until night.
Breezy -> nsubj
in -> prep
the -> det
morning -> pobj
and -> cc
partly -> advmod
cloudy -> conj
starting -> acl
in -> prep
```

```
the -> det
evening -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
Breezy , continue , morning evening night
```

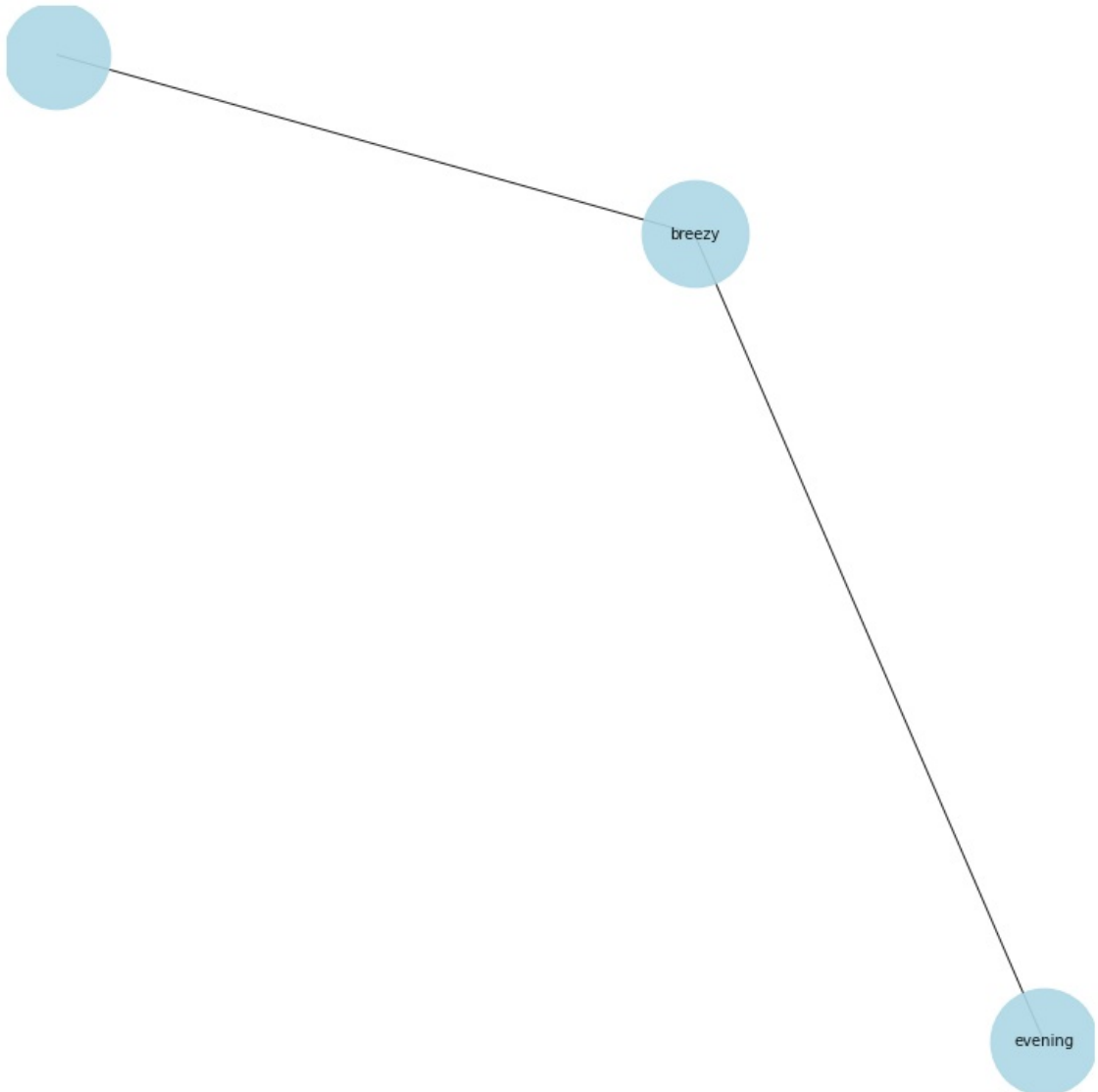



```
Breezy starting in the afternoon continuing until night and mostly cloudy starting in the evening.
Breezy -> ROOT
starting -> acl
in -> prep
the -> det
afternoon -> pobj
continuing -> acl
until -> prep
night -> pobj
and -> cc
mostly -> advmod
cloudy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
 , Breezy , afternoon night evening
```

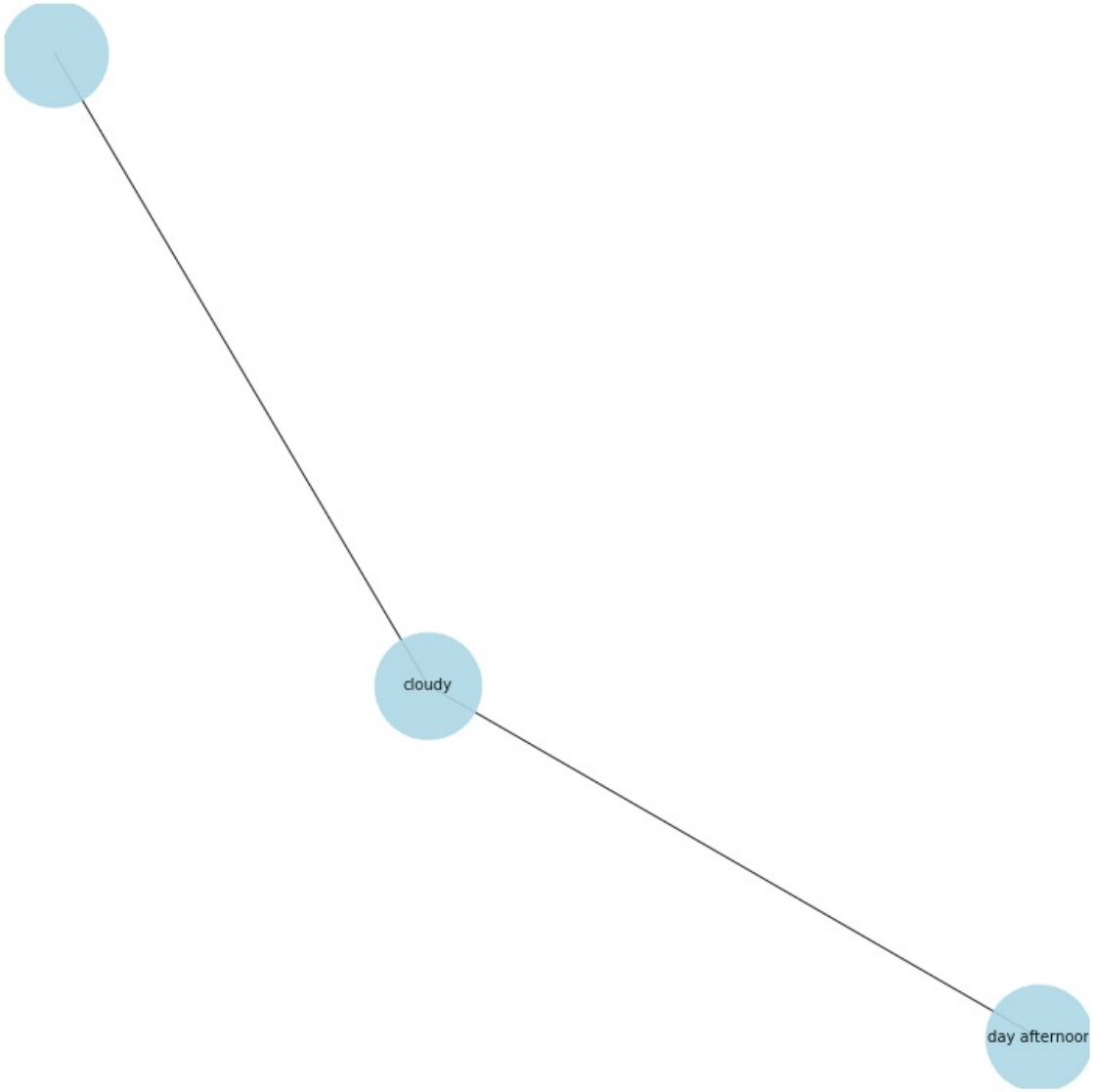




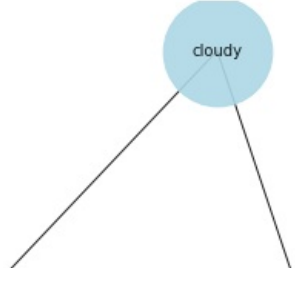

```
Breezy and foggy starting in the evening.
Breezy -> ROOT
and -> cc
foggy -> conj
starting -> acl
in -> prep
the -> det
evening -> pobj
. -> punct
 , breezy , evening
```

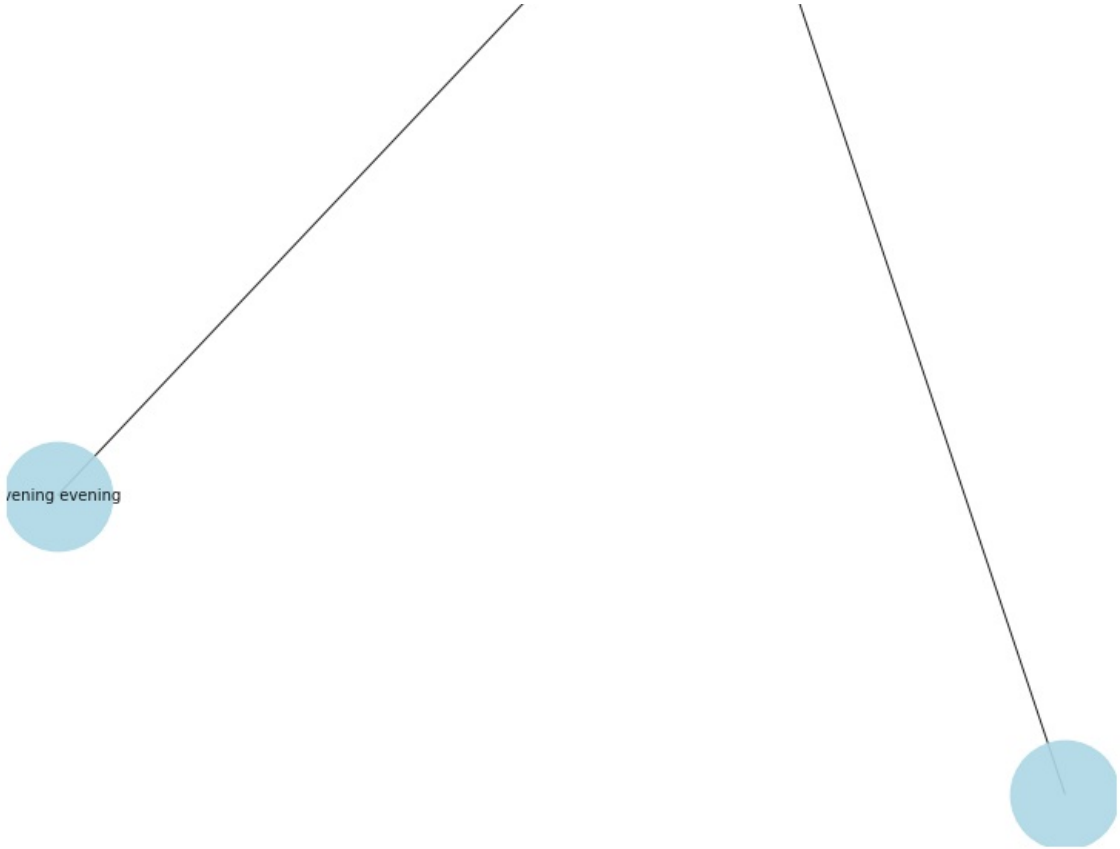


```
Partly cloudy throughout the day and breezy in the afternoon.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , day afternoon
```

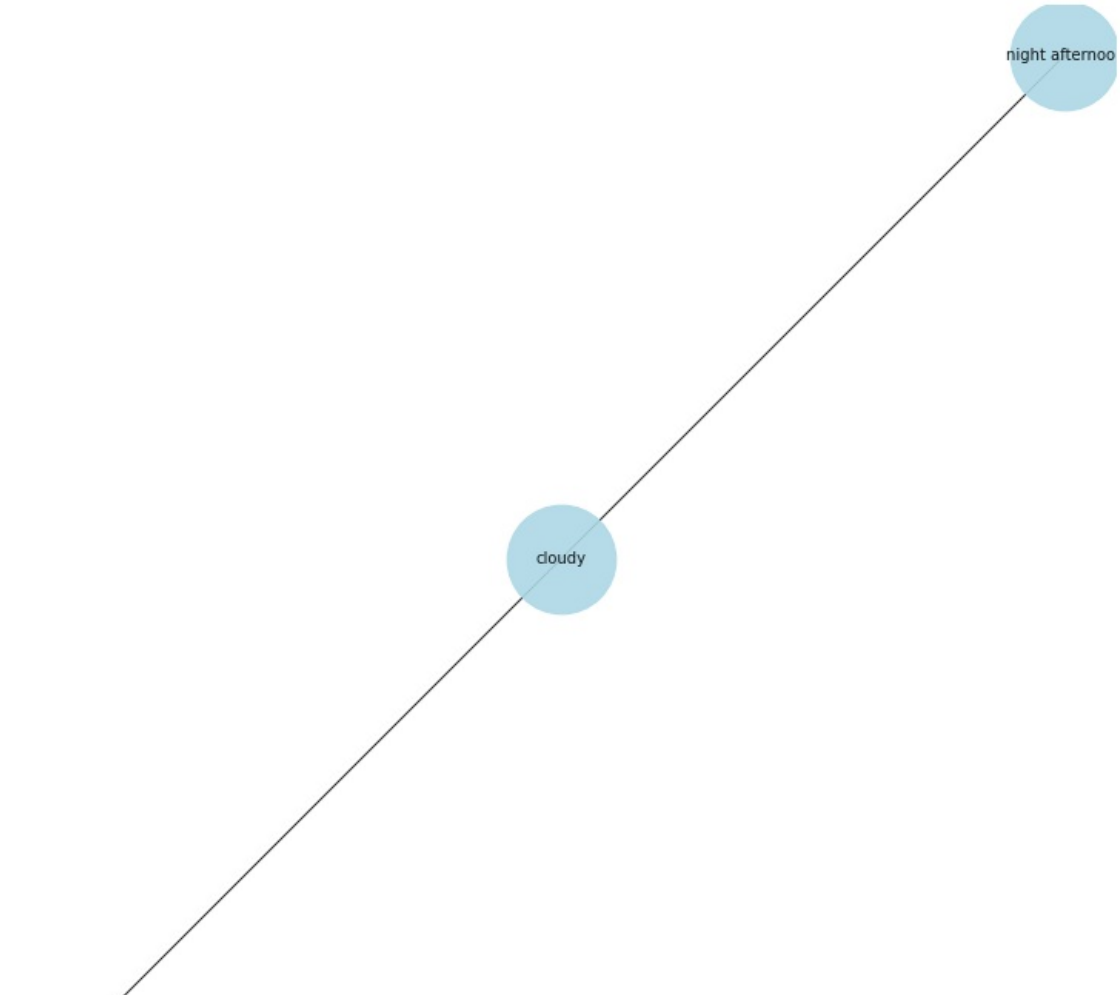



```
Mostly cloudy until evening and breezy in the evening.
Mostly -> advmod
cloudy -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , evening evening
```

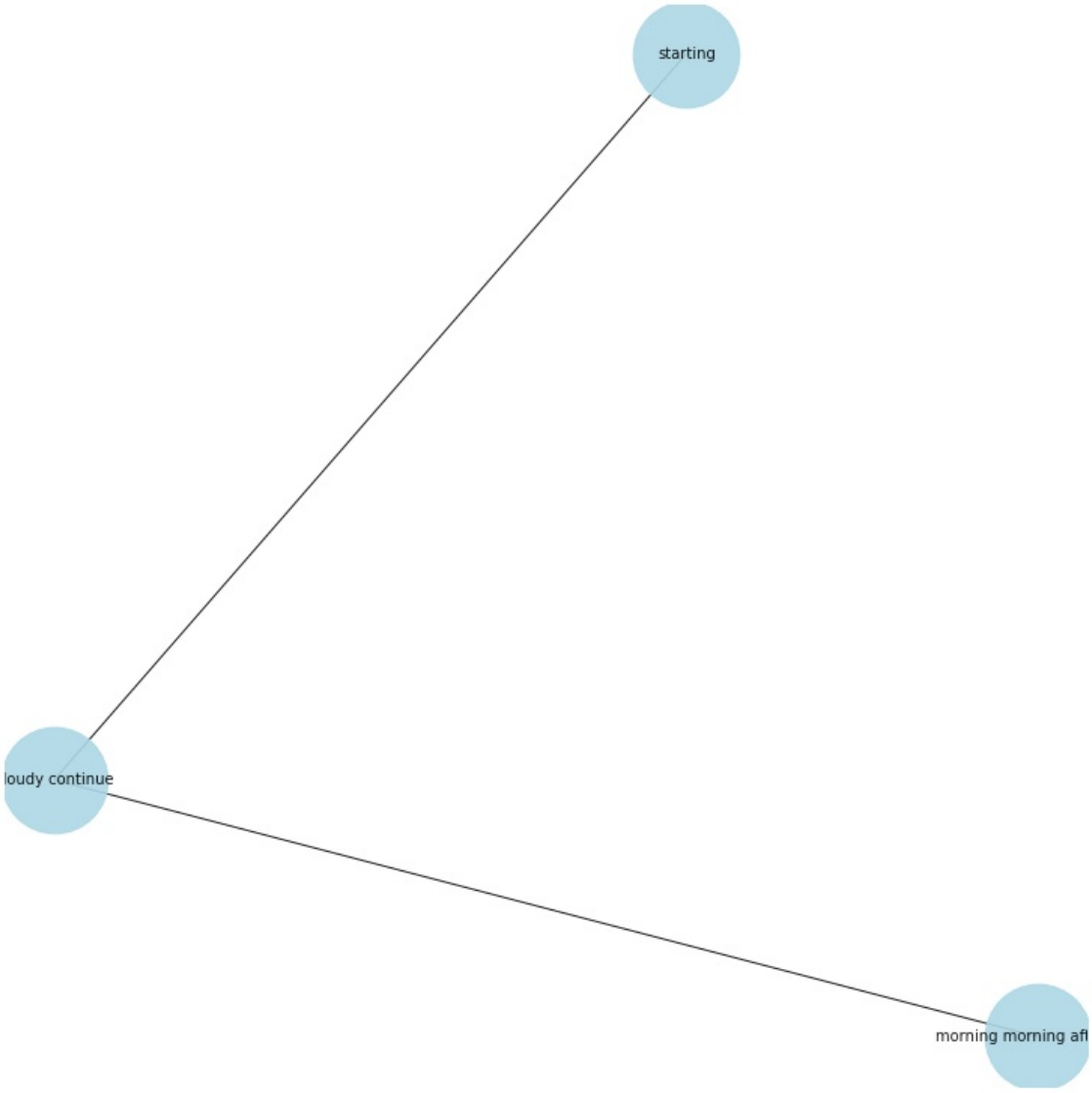




Partly cloudy until night and breezy in the afternoon.
Partly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , night afternoon

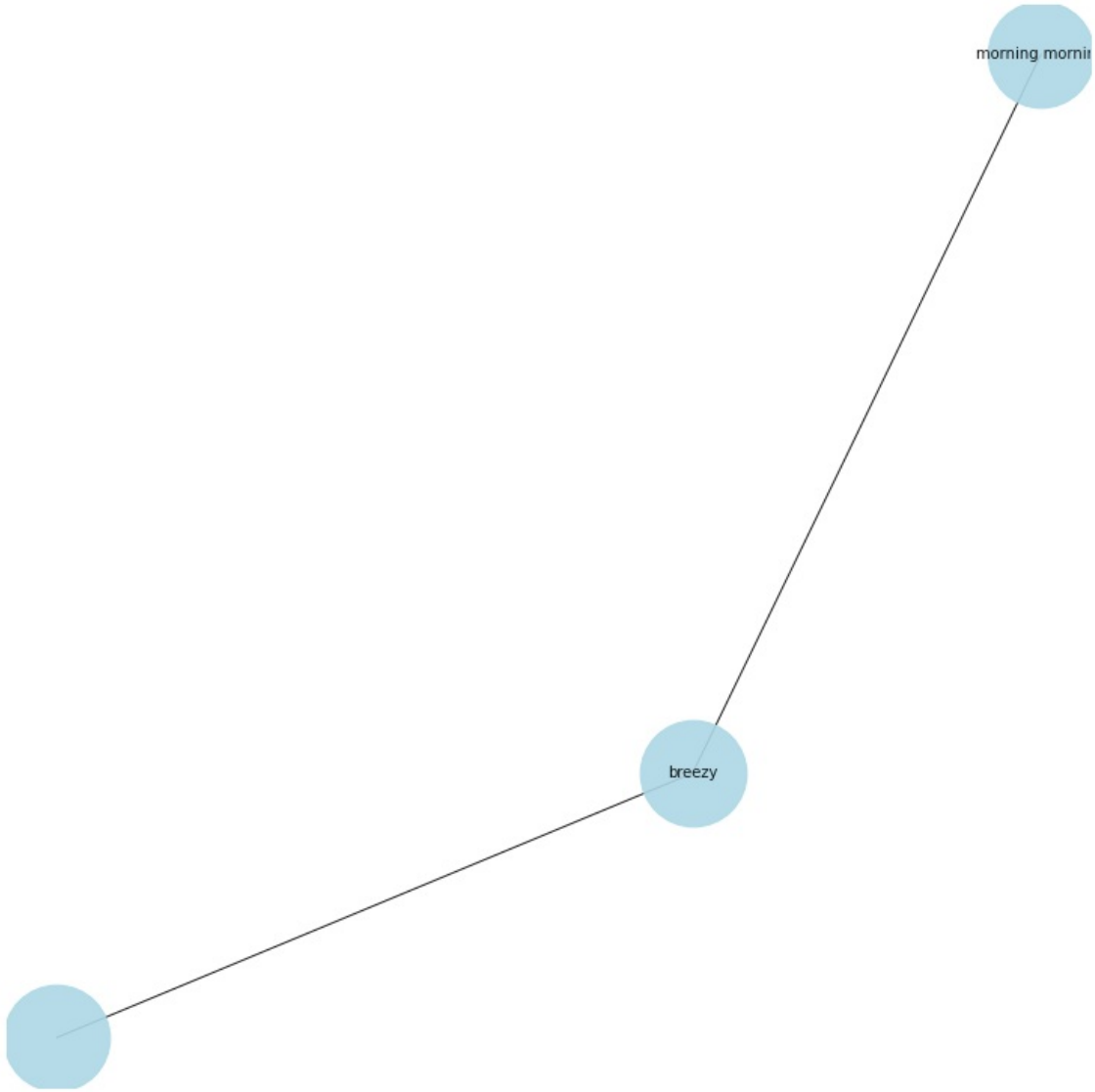


```
Partly cloudy starting overnight continuing until morning and breezy starting in the morning continuing until aft
ernoon.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
morning -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> xcomp
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy continue , morning morning afternoon
```

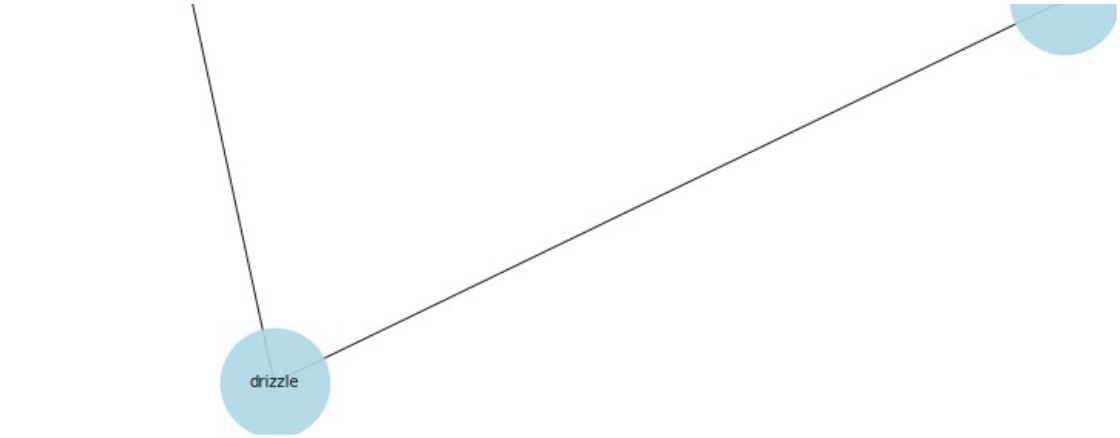



```
Breezy starting overnight continuing until morning and partly cloudy starting in the morning.
Breezy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
partly -> advmod
cloudy -> conj
starting -> advcl
in -> prep
```

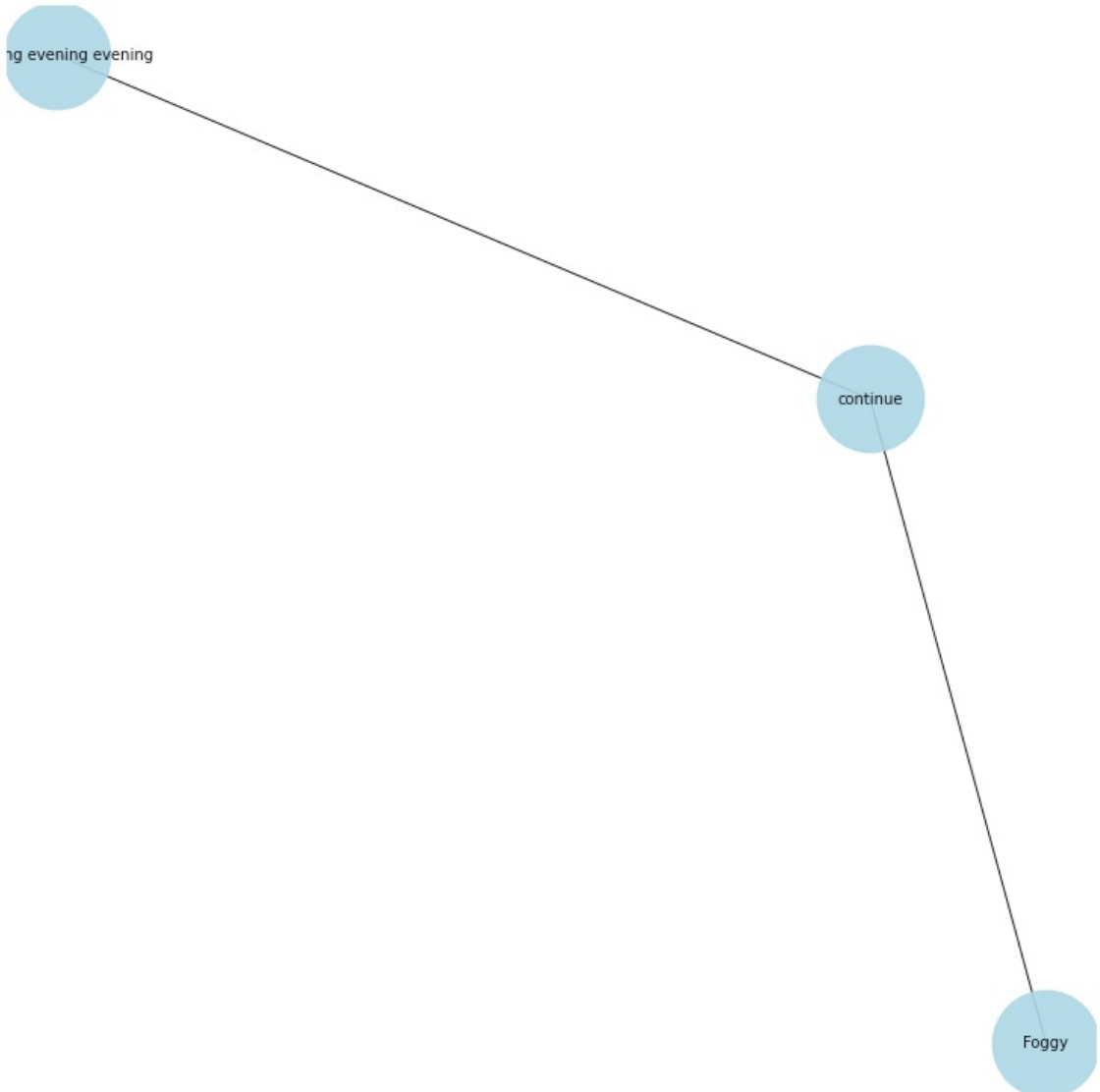
```
the -> det
morning -> pobj
. -> punct
 , breezy , morning morning
```




```
Drizzle until morning.
Drizzle -> ROOT
until -> prep
morning -> pobj
. -> punct
 , drizzle , morning
```

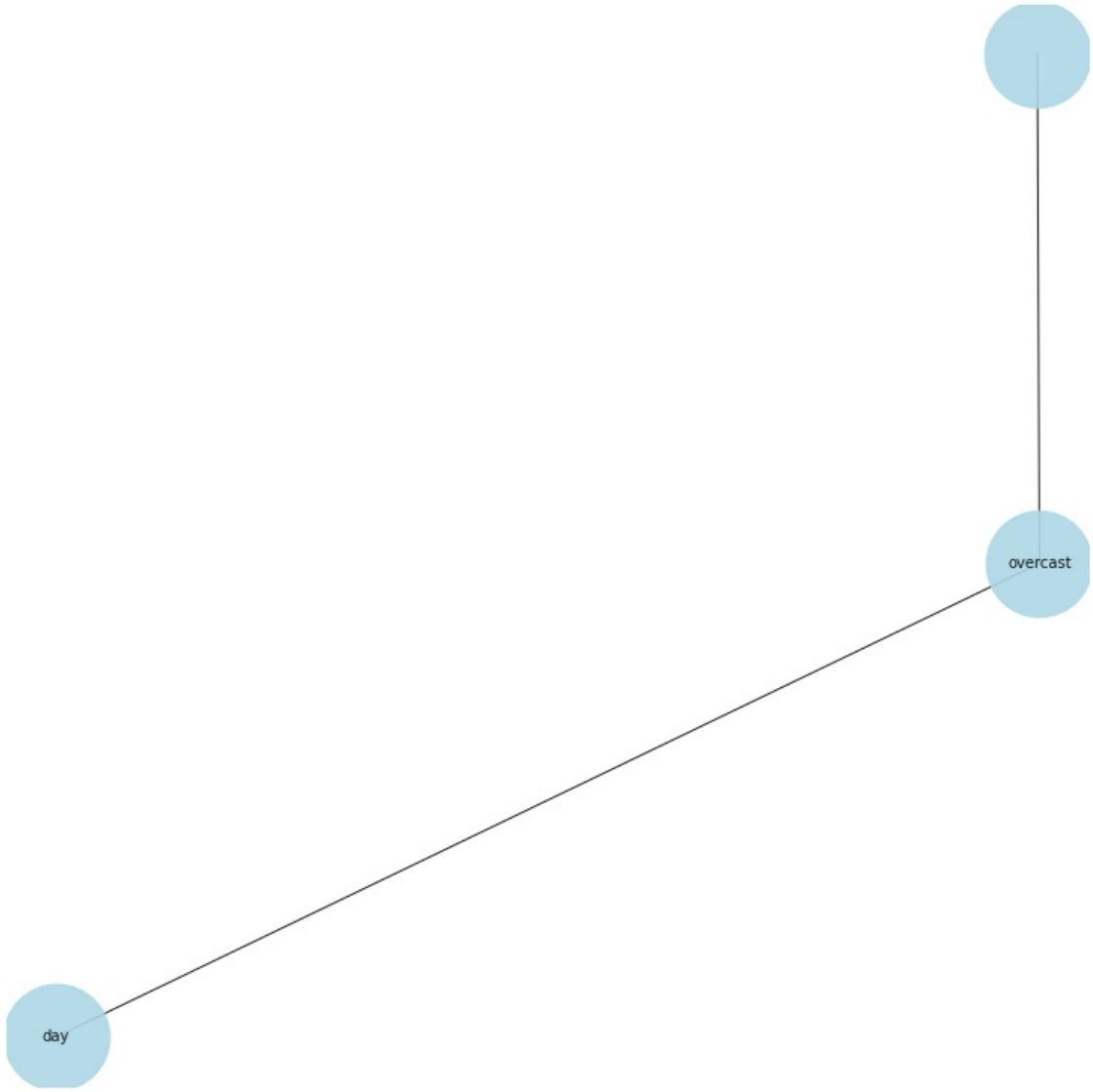



```
Foggy starting in the morning continuing until evening and breezy in the evening.
Foggy -> nsubj
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
Foggy , continue , morning evening evening
```

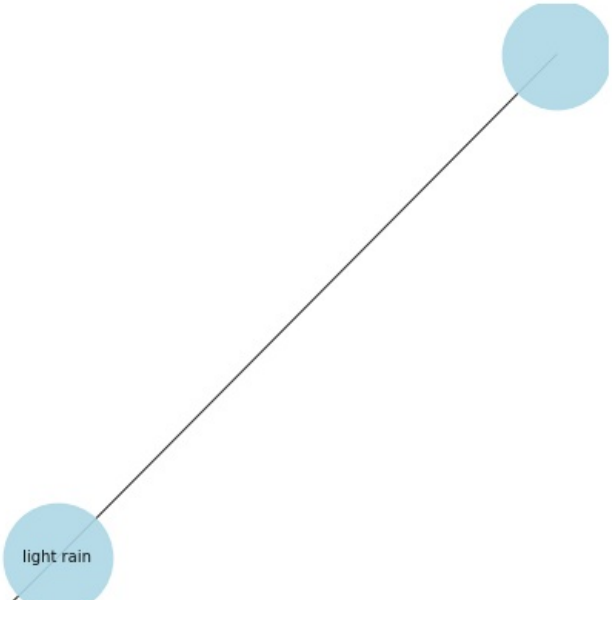


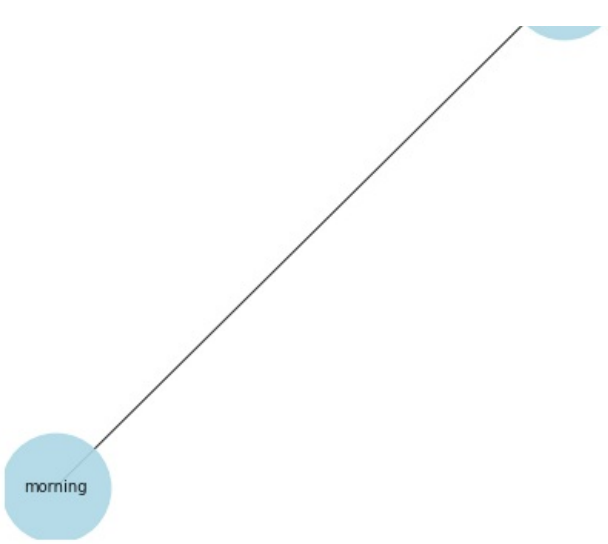

```
Overcast throughout the day and breezy overnight.
Overcast -> ROOT
throughout -> prep
the -> det
```

```
day -> pobj
and -> cc
breezy -> conj
overnight -> advmod
. -> punct
 , overcast , day
```

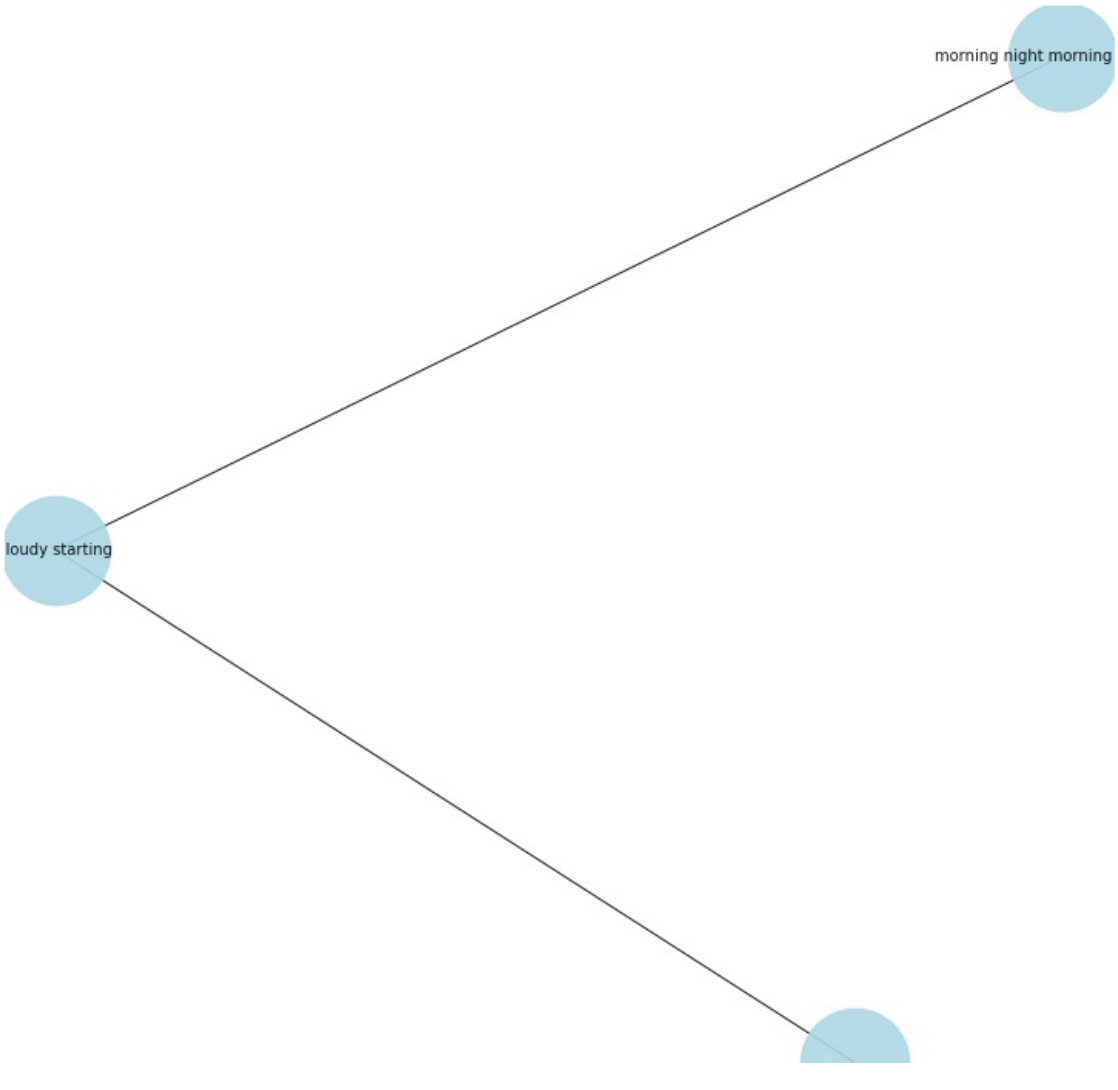



```
Light rain in the morning.
Light -> amod
rain -> ROOT
in -> prep
the -> det
morning -> pobj
. -> punct
 , light rain , morning
```

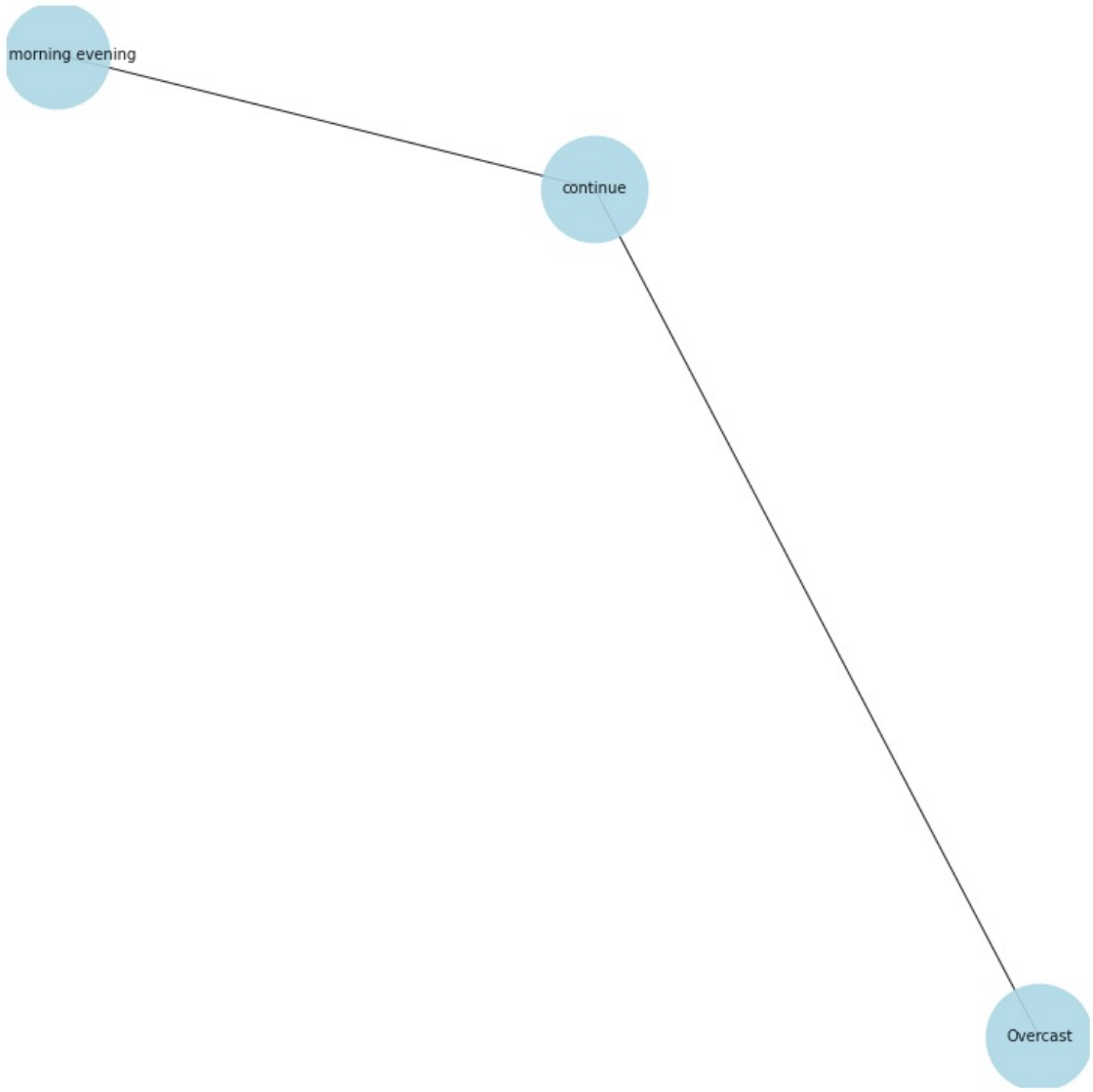




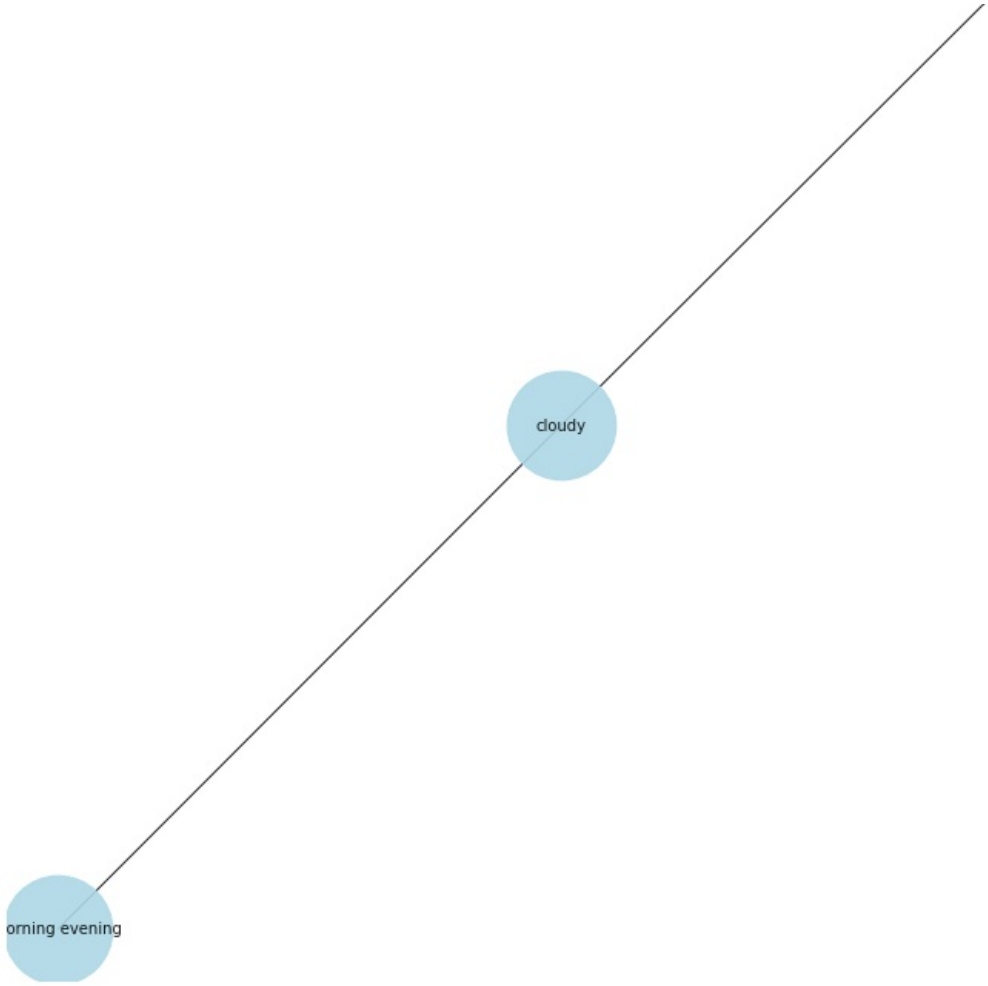
Partly cloudy starting in the morning continuing until night and breezy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
continuing -> xcomp
until -> prep
night -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> xcomp
until -> prep
afternoon -> pobj
. -> punct
 , cloudy starting , morning night morning afternoon



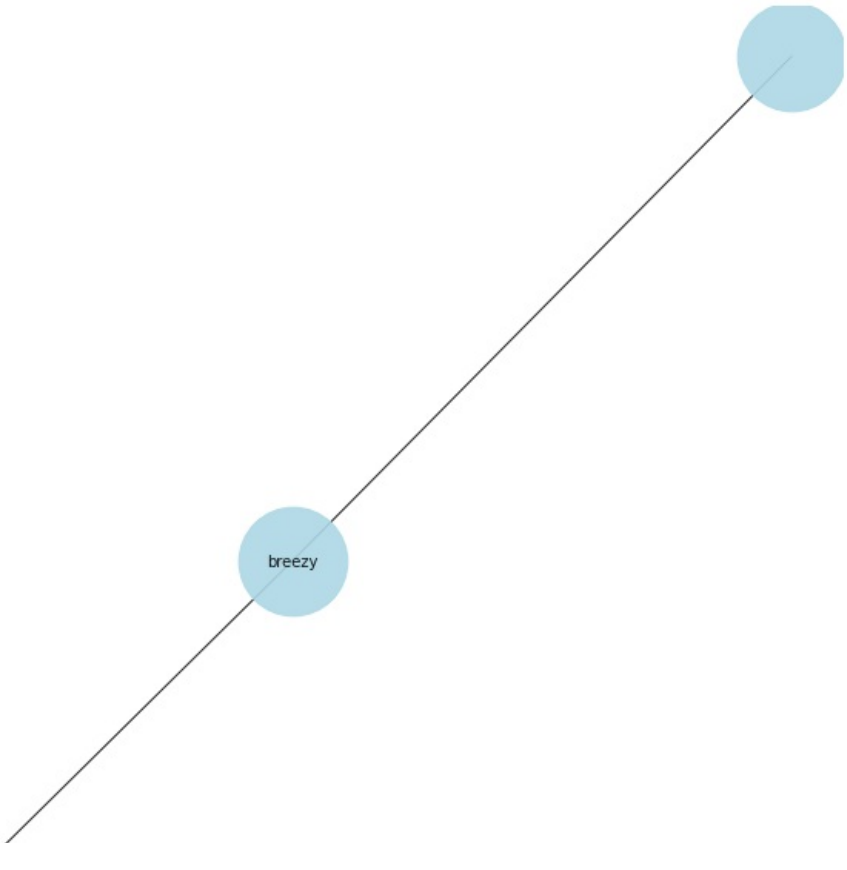
Overcast throughout the day and breezy starting in the morning continuing until evening.
Overcast -> nsubj
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
Overcast , continue , day morning evening

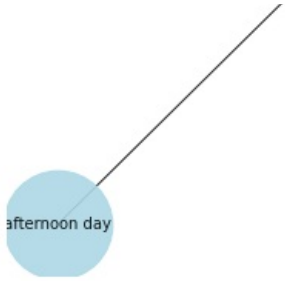


Partly cloudy starting in the morning and breezy in the evening.
Partly -> advmod
cloudy -> ROOT
starting -> acl
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy , morning evening

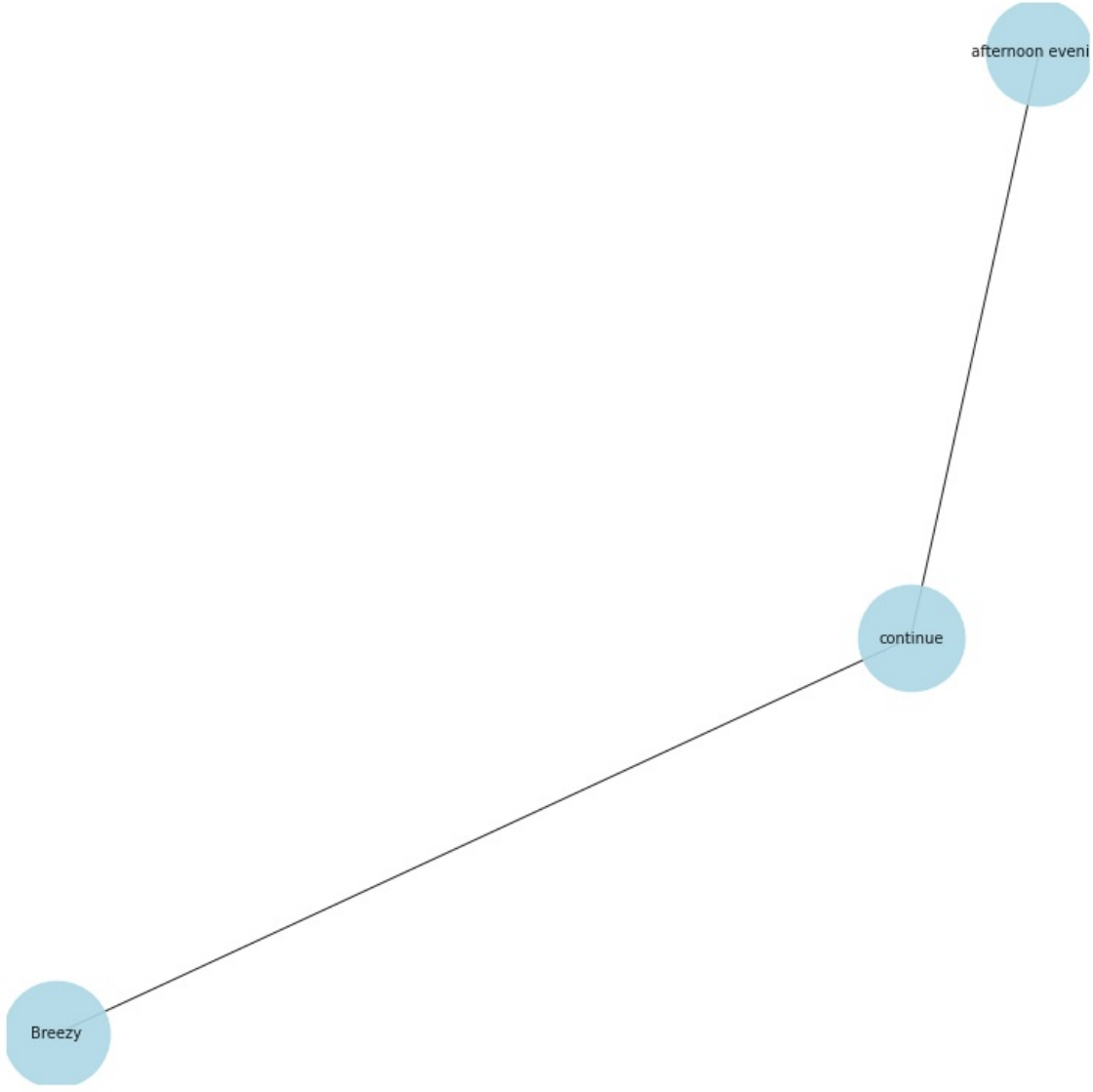



```
Breezy until afternoon and overcast throughout the day.
Breezy -> ROOT
until -> prep
afternoon -> pobj
and -> cc
overcast -> conj
throughout -> prep
the -> det
day -> pobj
. -> punct
 , breezy , afternoon day
```

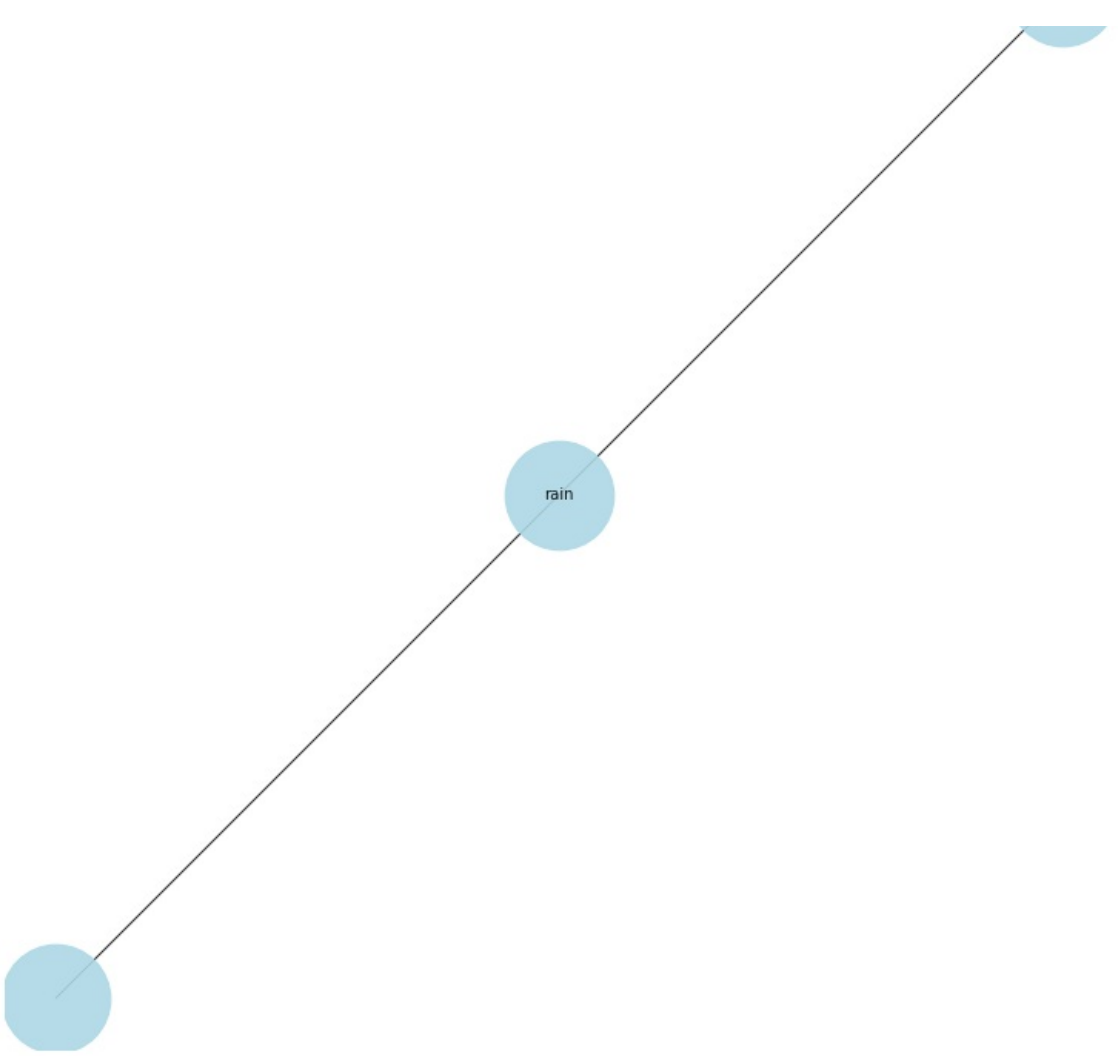




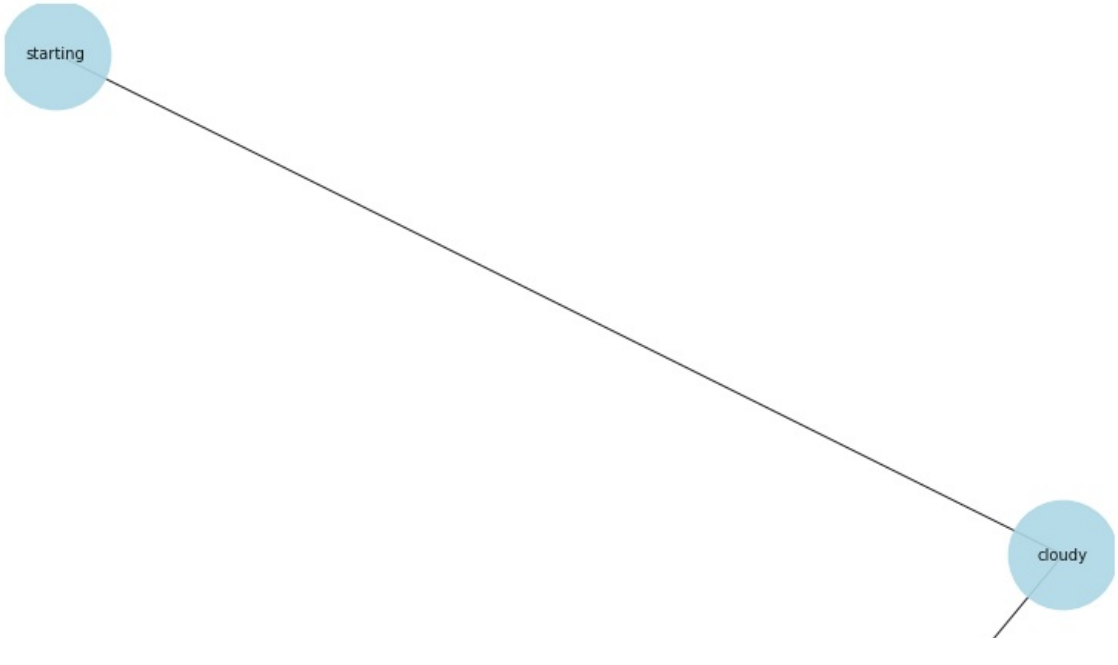
Breezy starting overnight continuing until afternoon and mostly cloudy starting overnight continuing until evenin
g.
Breezy -> nsubj
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
afternoon -> pobj
and -> cc
mostly -> advmod
cloudy -> conj
starting -> dep
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
Breezy , continue , afternoon evening

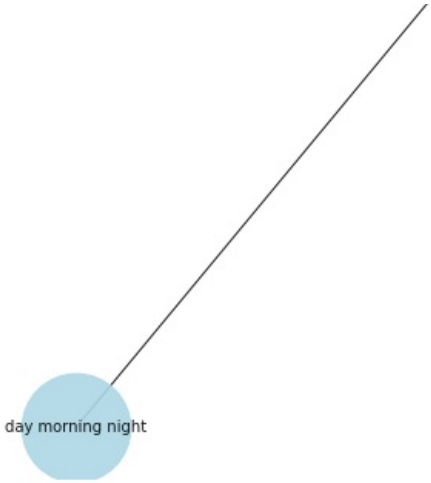


Rain throughout the day.
Rain -> ROOT
throughout -> prep
the -> det
day -> pobj
. -> punct
 , rain , day

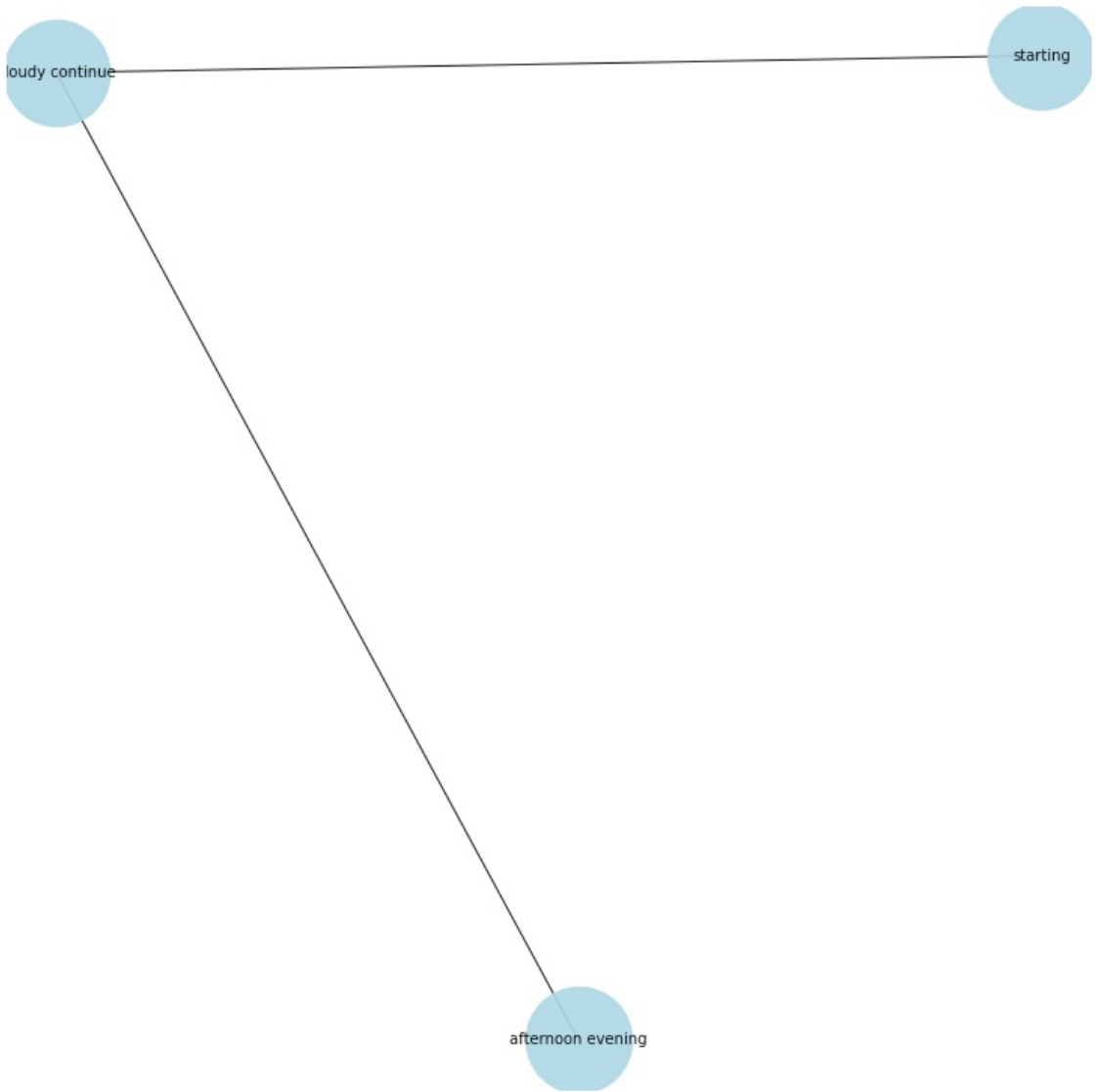


Partly cloudy throughout the day and breezy starting in the morning continuing until night.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
night -> pobj
. -> punct
starting , cloudy , day morning night



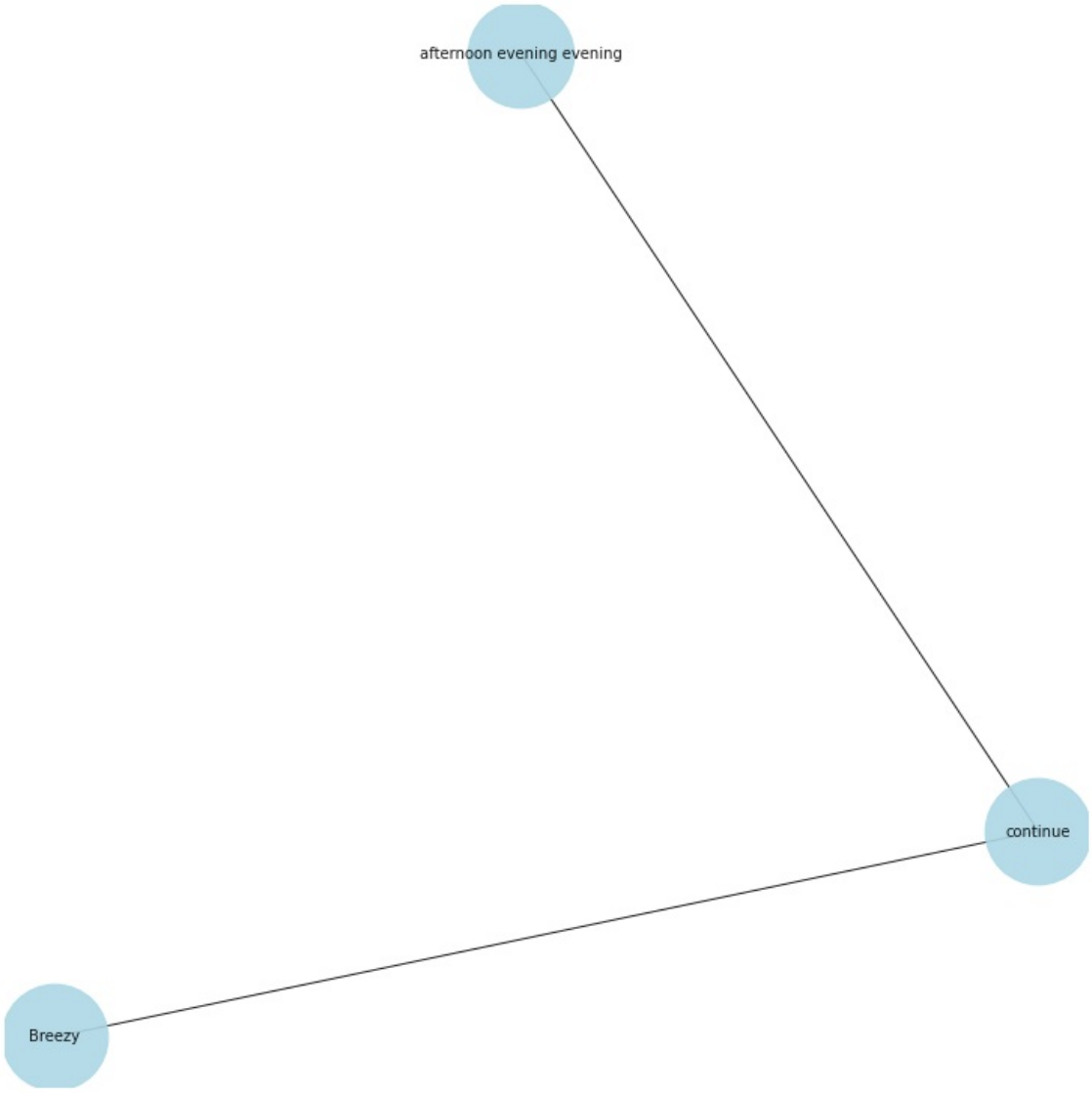



```
Mostly cloudy starting overnight and breezy starting in the afternoon continuing until evening.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
starting , cloudy continue , afternoon evening
```

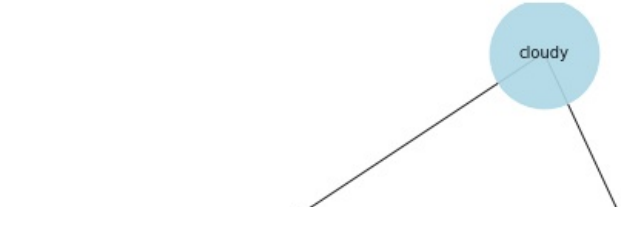


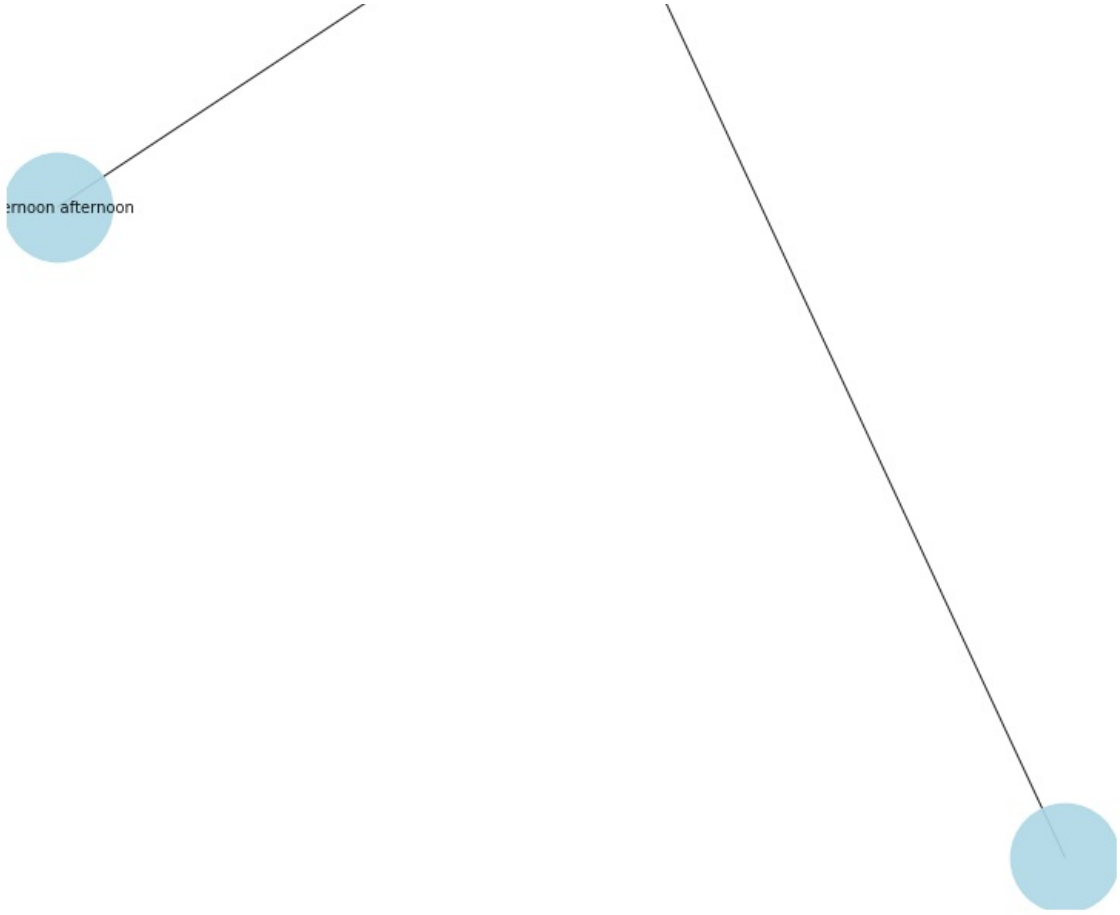

```
Breezy starting in the afternoon continuing until evening and foggy starting in the evening.
Breezy -> nsubj
```

```
starting -> acl
in -> prep
the -> det
afternoon -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
foggy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
. -> punct
Breezy , continue , afternoon evening evening
```

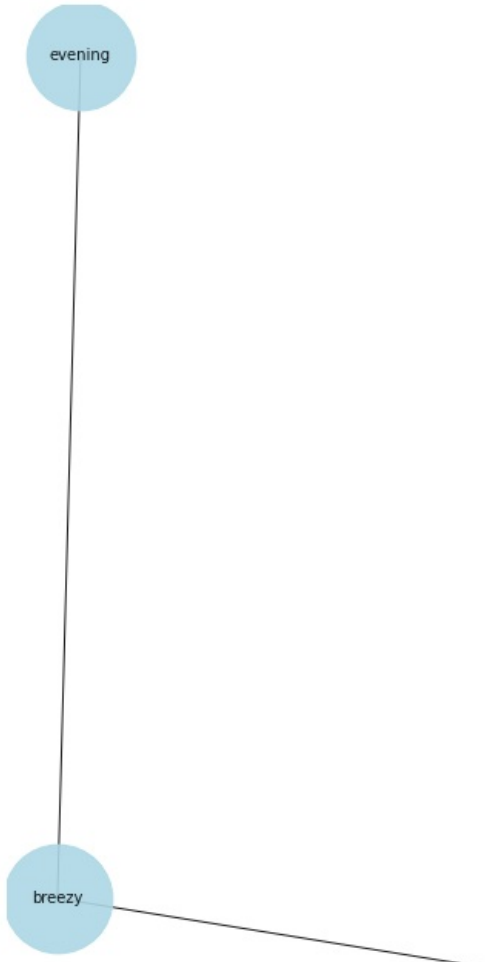



```
Partly cloudy starting in the afternoon and breezy in the afternoon.
Partly -> advmod
cloudy -> ROOT
starting -> acl
in -> prep
the -> det
afternoon -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , afternoon afternoon
```

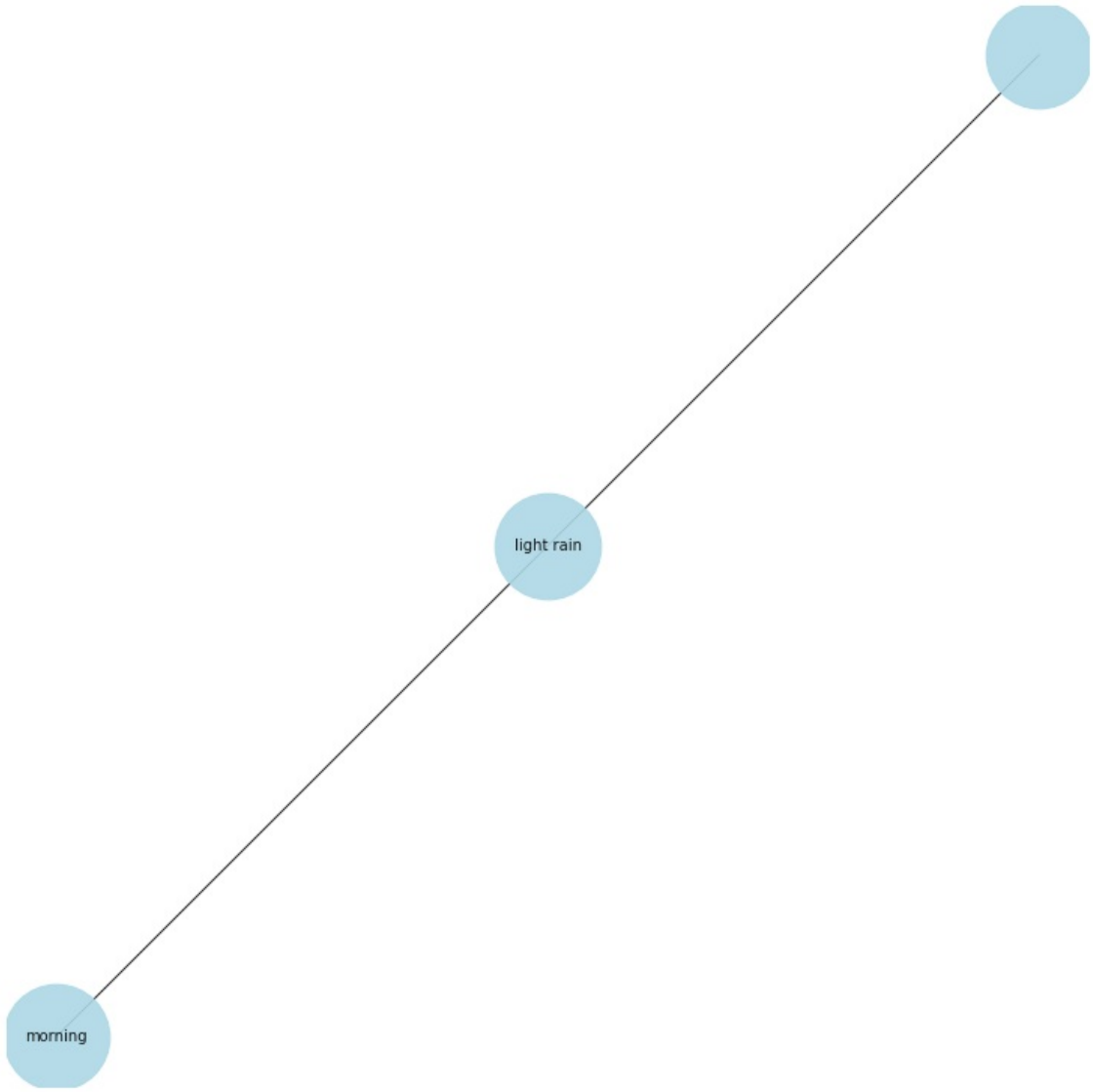




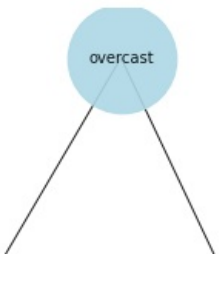

```
Breezy overnight and partly cloudy until evening.
Breezy -> ROOT
overnight -> advmod
and -> cc
partly -> advmod
cloudy -> conj
until -> prep
evening -> pobj
. -> punct
 , breezy , evening
```

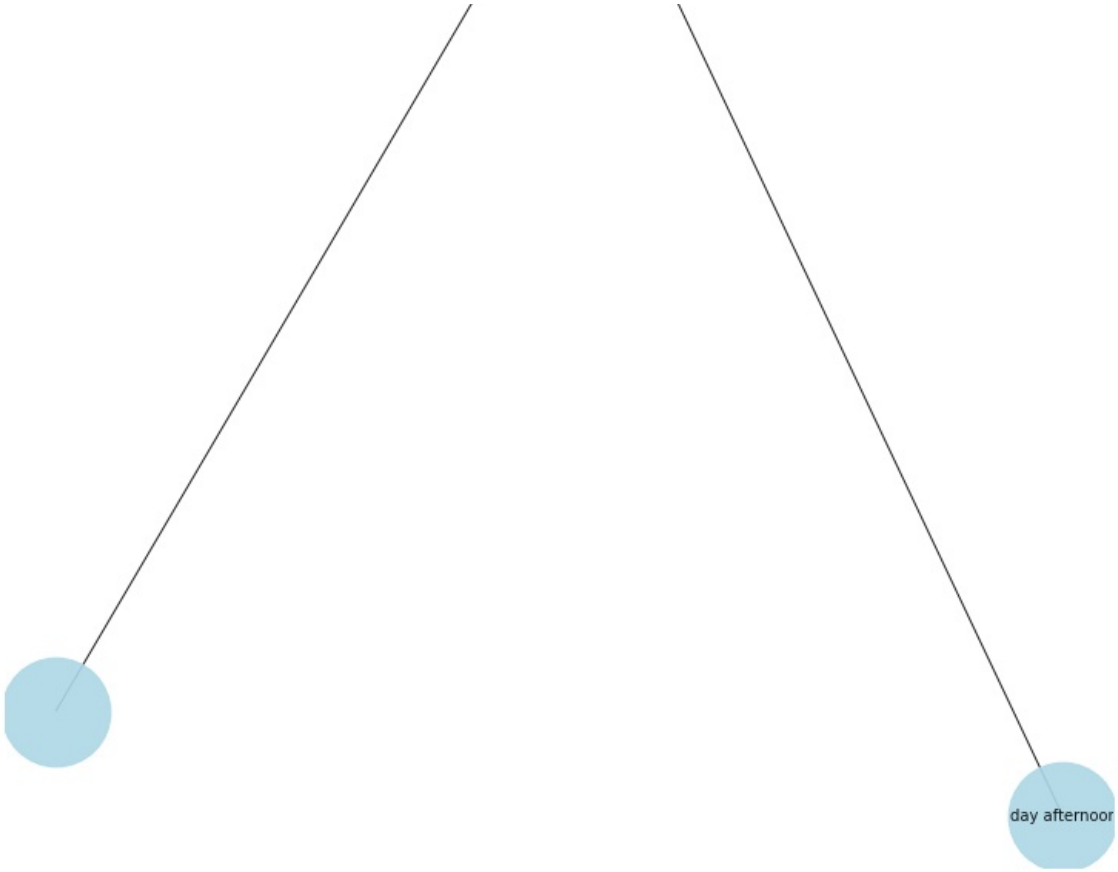


```
Light rain until morning.
Light -> amod
rain -> ROOT
until -> prep
morning -> pobj
. -> punct
 , light rain , morning
```

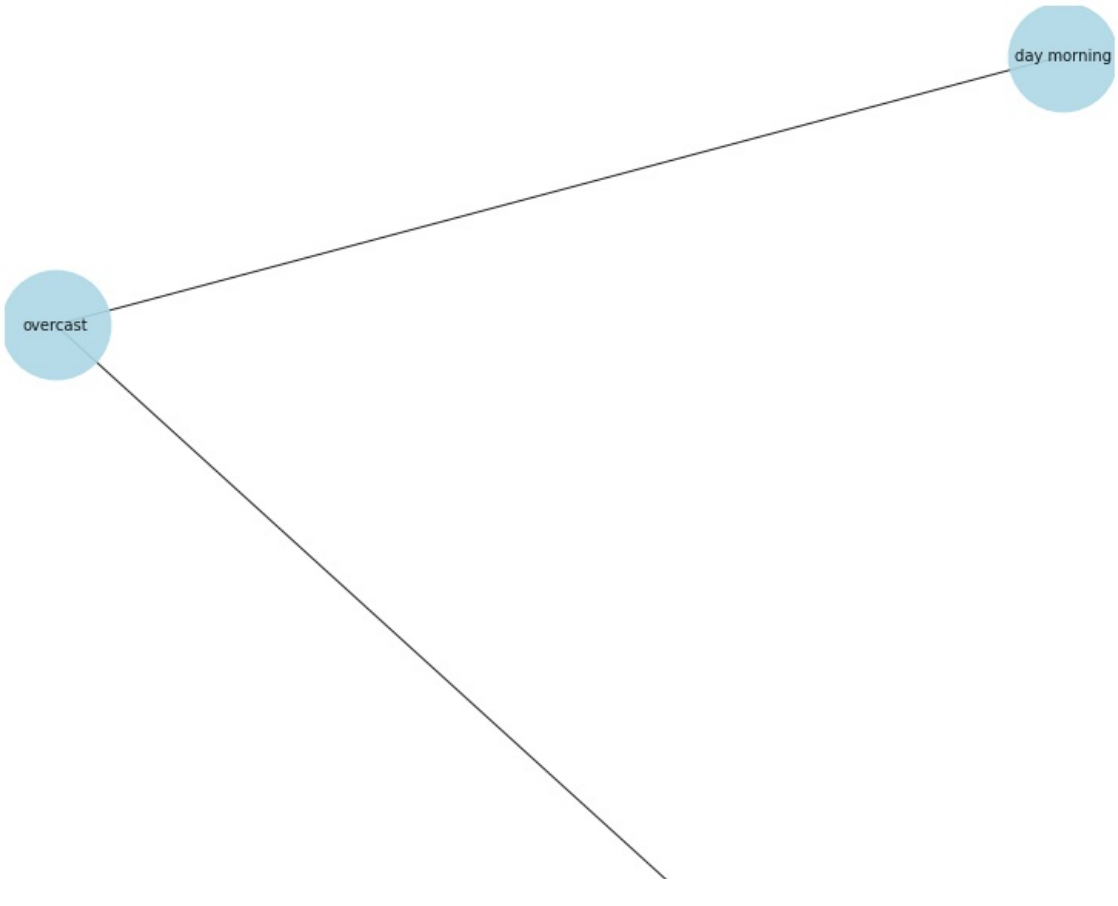


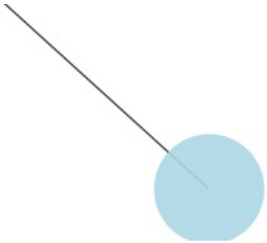

```
Overcast throughout the day and breezy starting overnight continuing until afternoon.
Overcast -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
overnight -> advmod
continuing -> xcomp
until -> prep
afternoon -> pobj
. -> punct
 , overcast , day afternoon
```



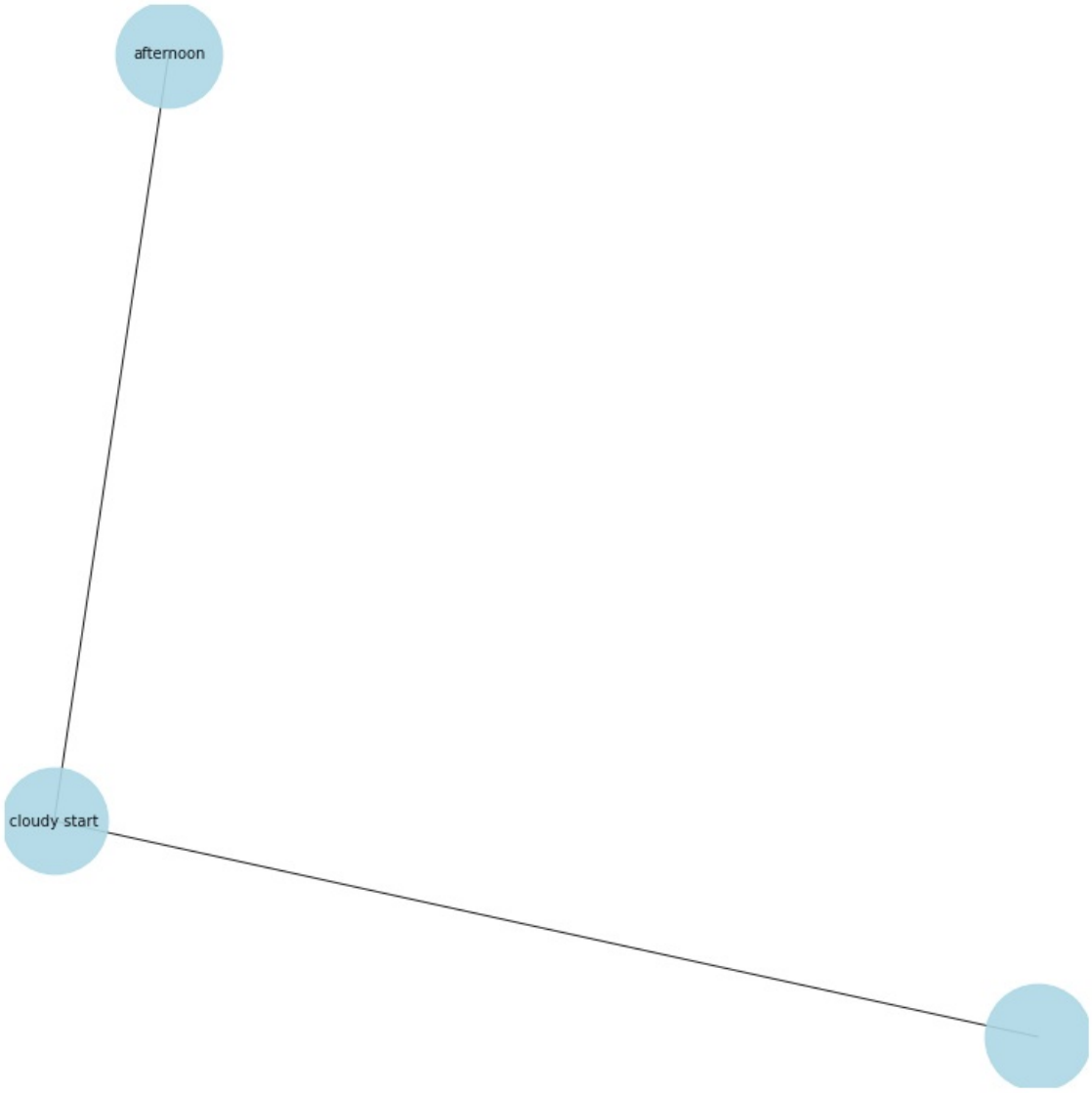



```
Overcast throughout the day and breezy starting overnight continuing until morning.
Overcast -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
. -> punct
 , overcast , day morning
```

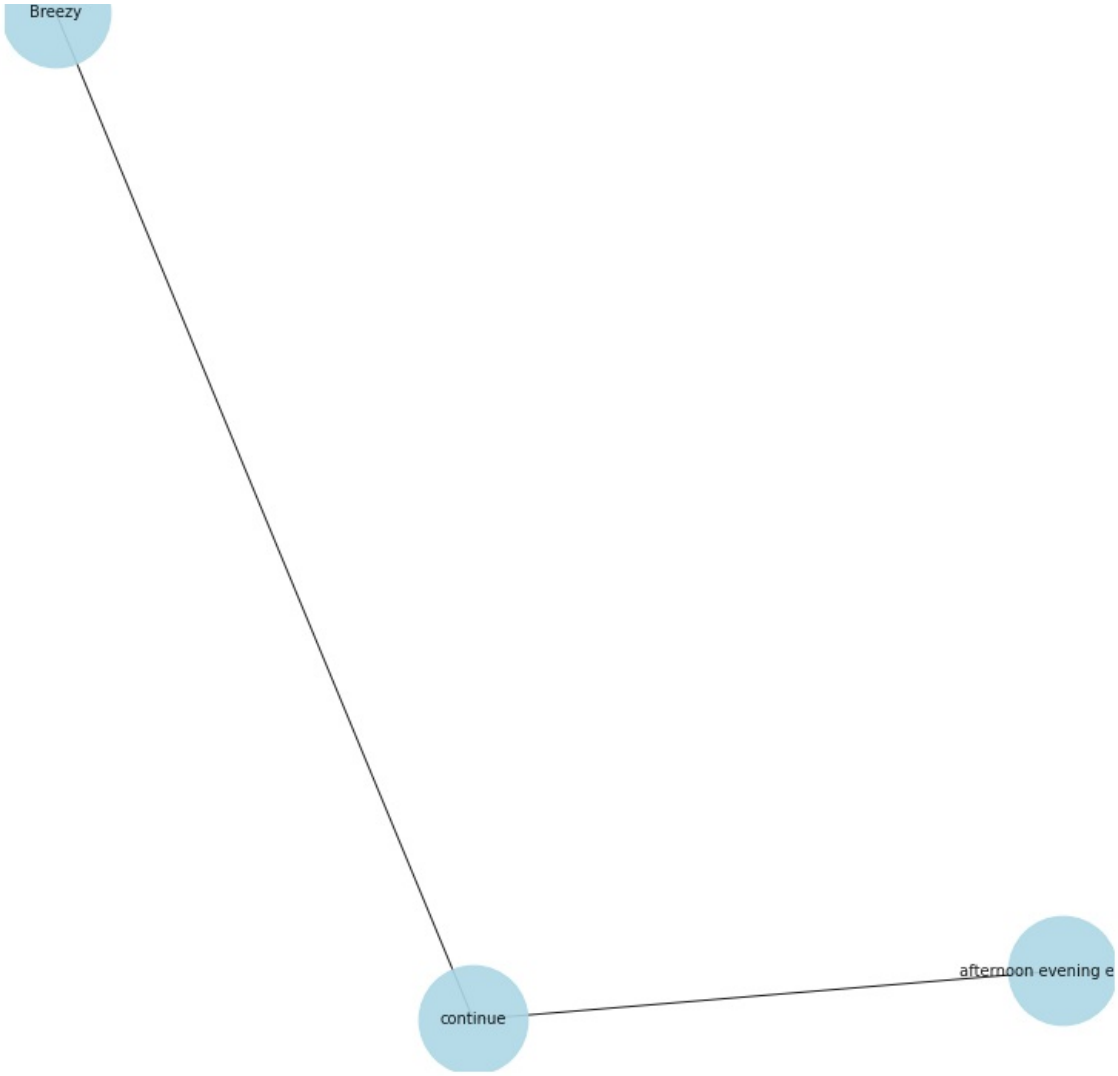




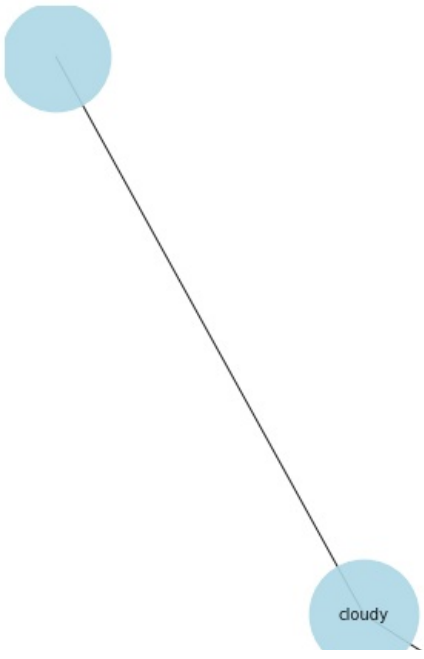
```
Partly cloudy starting overnight and breezy in the afternoon.
Partly -> advmod
cloudy -> amod
starting -> ROOT
overnight -> advmod
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy start , afternoon
```

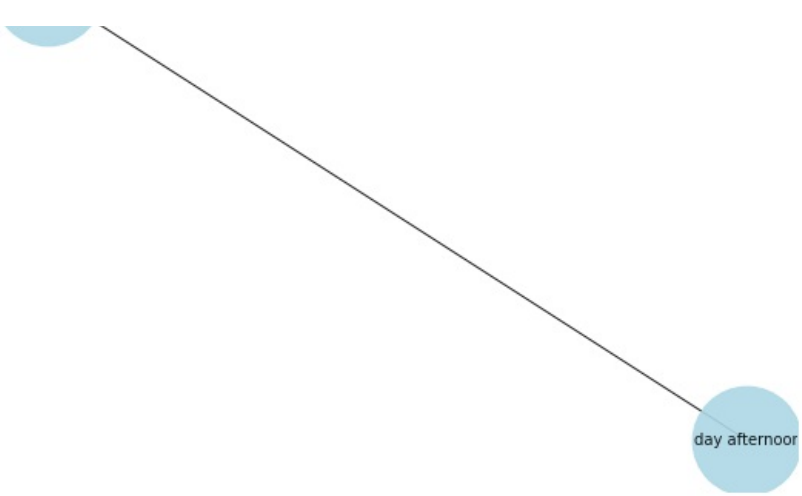



```
Breezy starting in the afternoon continuing until evening and foggy in the evening.
Breezy -> nsubj
starting -> acl
in -> prep
the -> det
afternoon -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
foggy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
Breezy , continue , afternoon evening evening
```

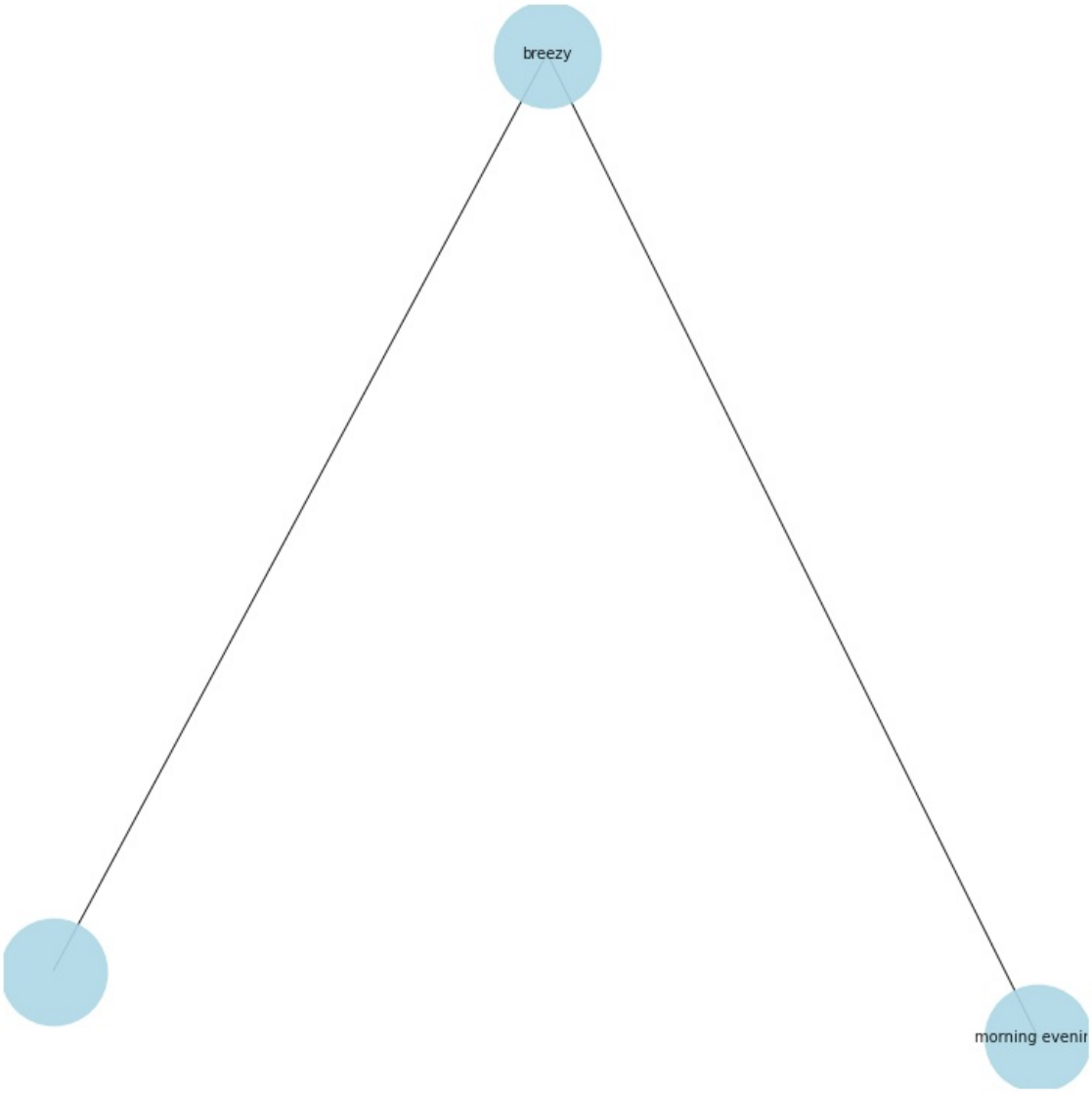



```
Mostly cloudy throughout the day and breezy starting overnight continuing until afternoon.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> advcl
overnight -> advmod
continuing -> xcomp
until -> prep
afternoon -> pobj
. -> punct
 , cloudy , day afternoon
```



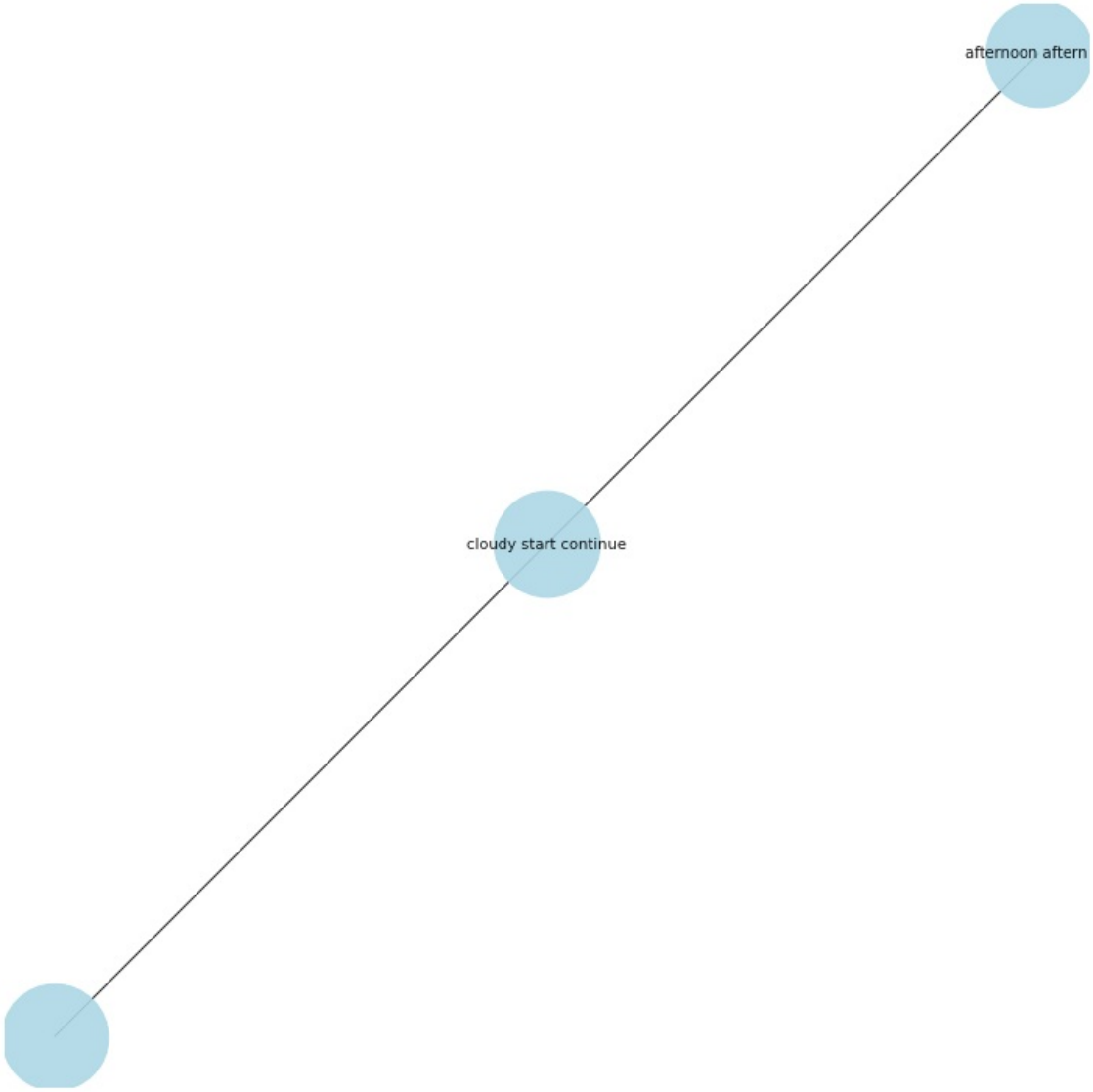



```
Breezy in the morning and foggy in the evening.
Breezy -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
foggy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , breezy , morning evening
```

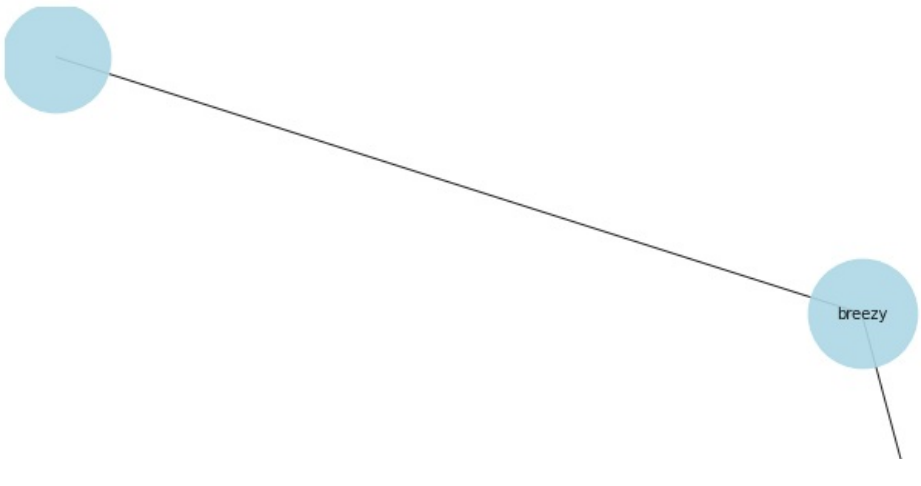



```
Partly cloudy starting overnight continuing until afternoon and breezy in the afternoon.
Partly -> advmod
cloudy -> amod
starting -> amod
overnight -> advmod
continuing -> ROOT
until -> prep
```

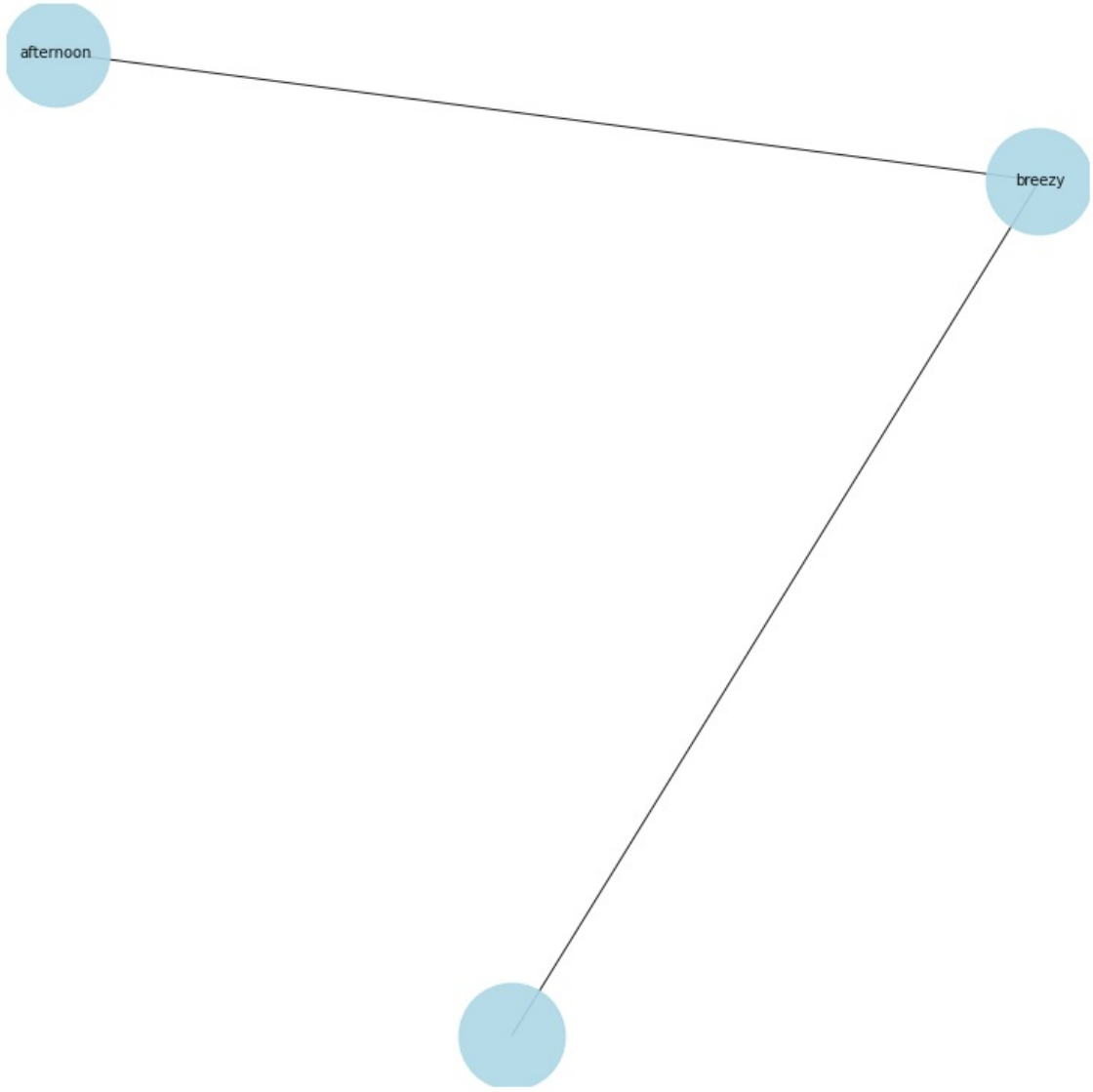
```
afternoon -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy start continue , afternoon afternoon
```




```
Breezy in the morning and mostly cloudy starting in the evening.
Breezy -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
mostly -> advmod
cloudy -> conj
starting -> acl
in -> prep
the -> det
evening -> pobj
. -> punct
 , breezy , morning evening
```

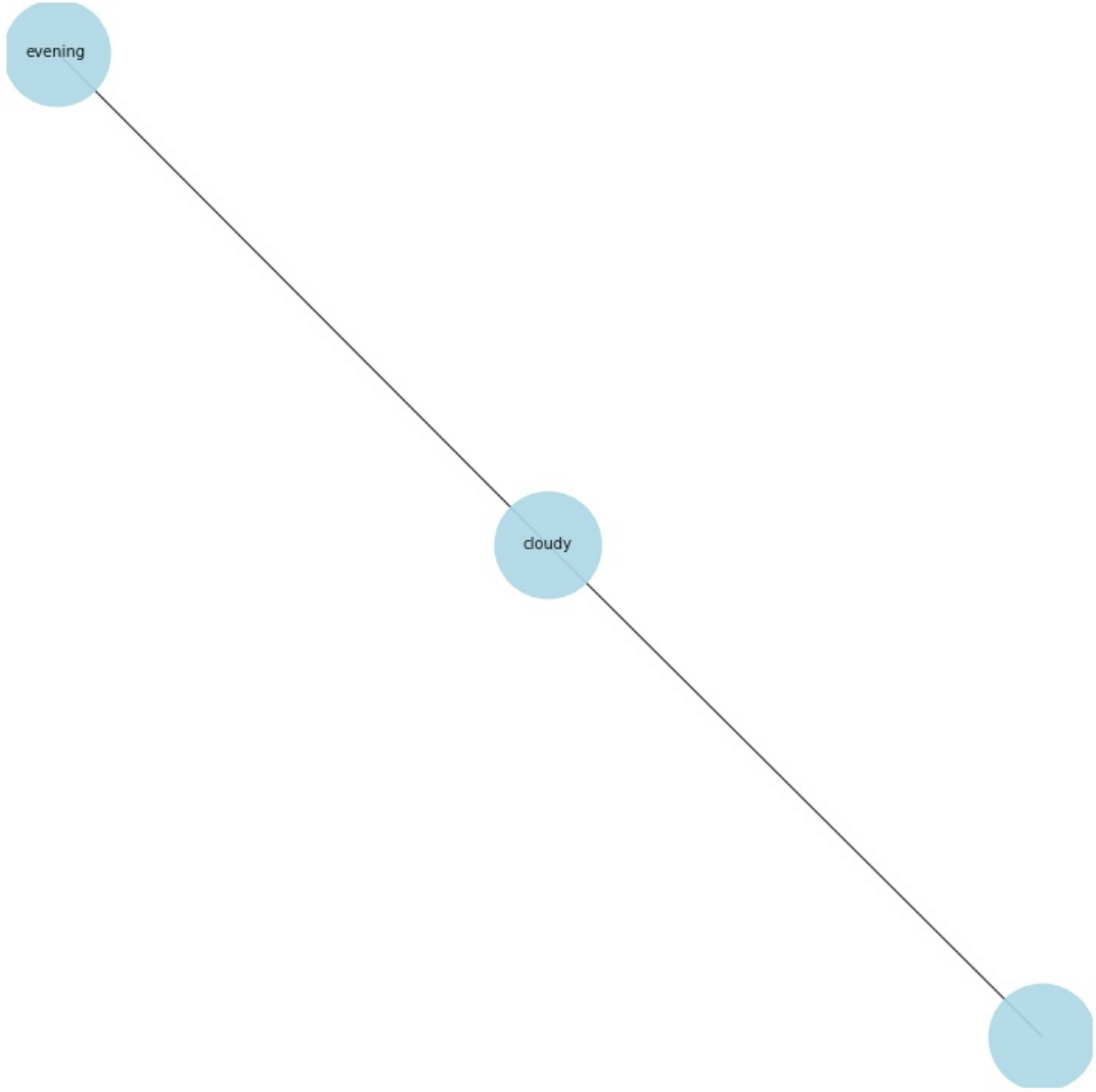



```
Breezy and partly cloudy in the afternoon.
Breezy -> ROOT
and -> cc
partly -> advmod
cloudy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , breezy , afternoon
```

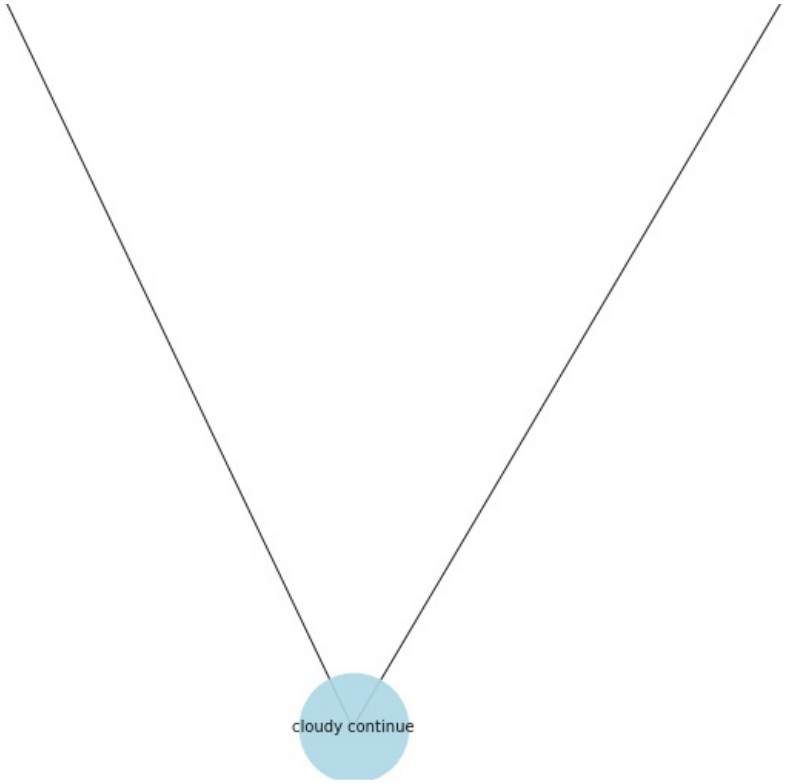



```
Partly cloudy until evening and breezy overnight.
Partly -> advmod
cloudy -> ROOT
```

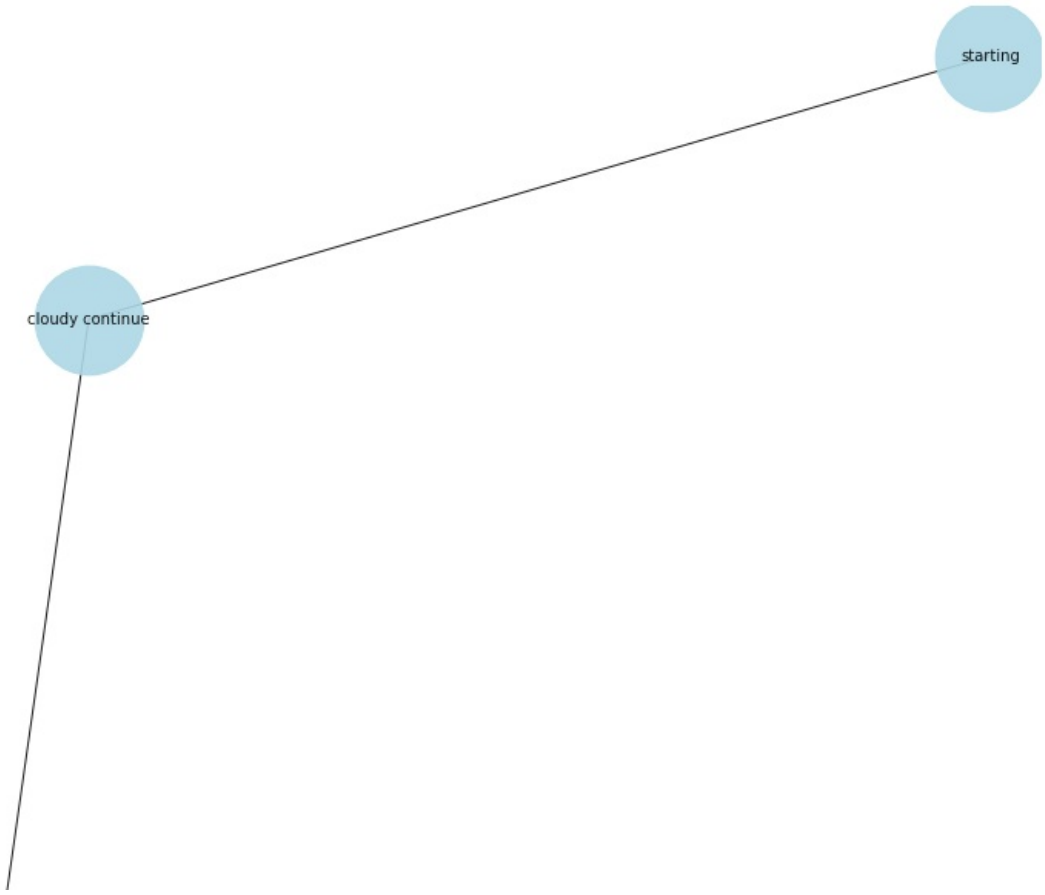
```
until -> prep
evening -> pobj
and -> cc
breezy -> conj
overnight -> advmod
. -> punct
 , cloudy , evening
```



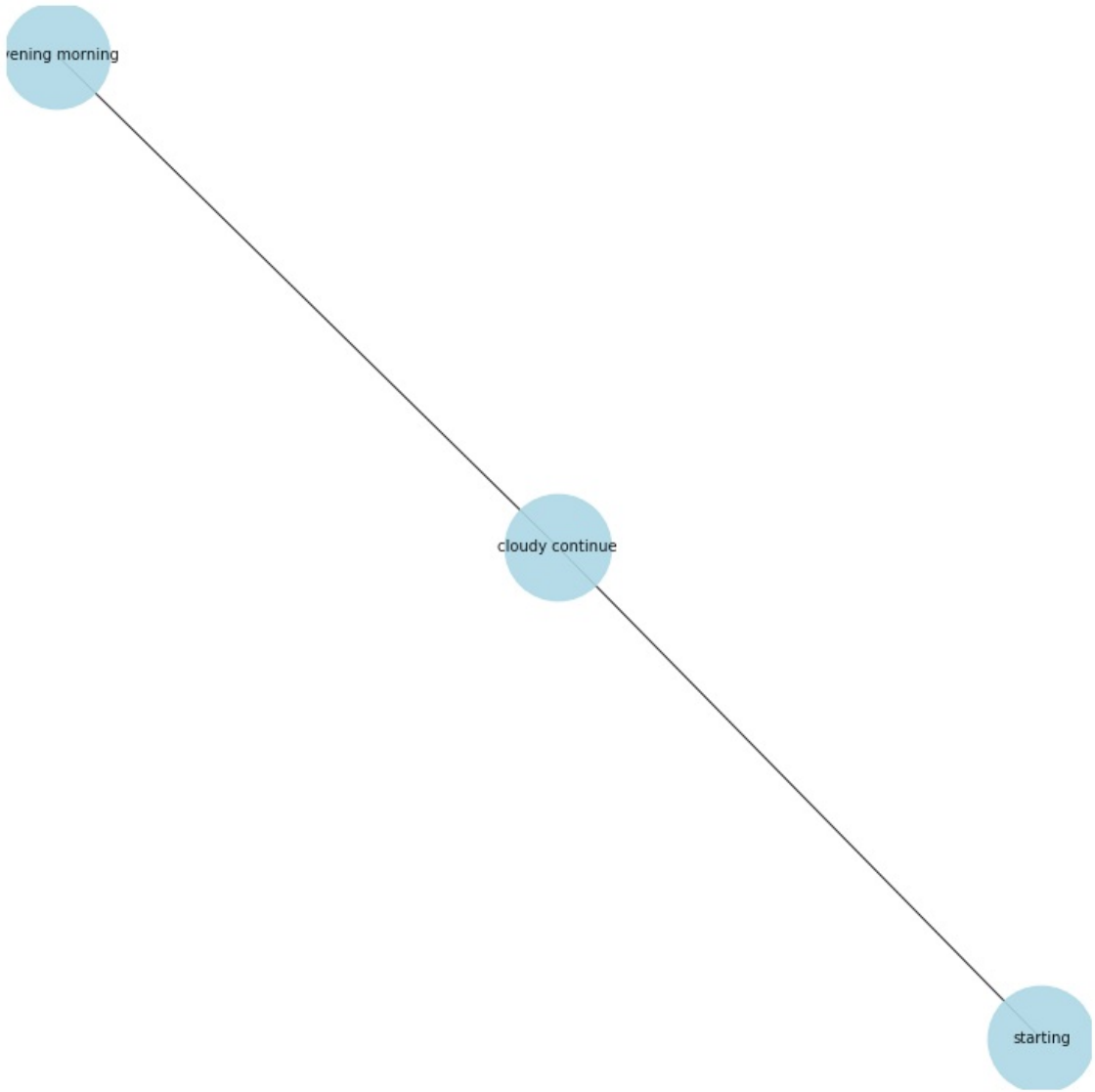

```
Mostly cloudy starting in the morning continuing until night and breezy in the afternoon.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
starting , cloudy continue , morning night afternoon
```



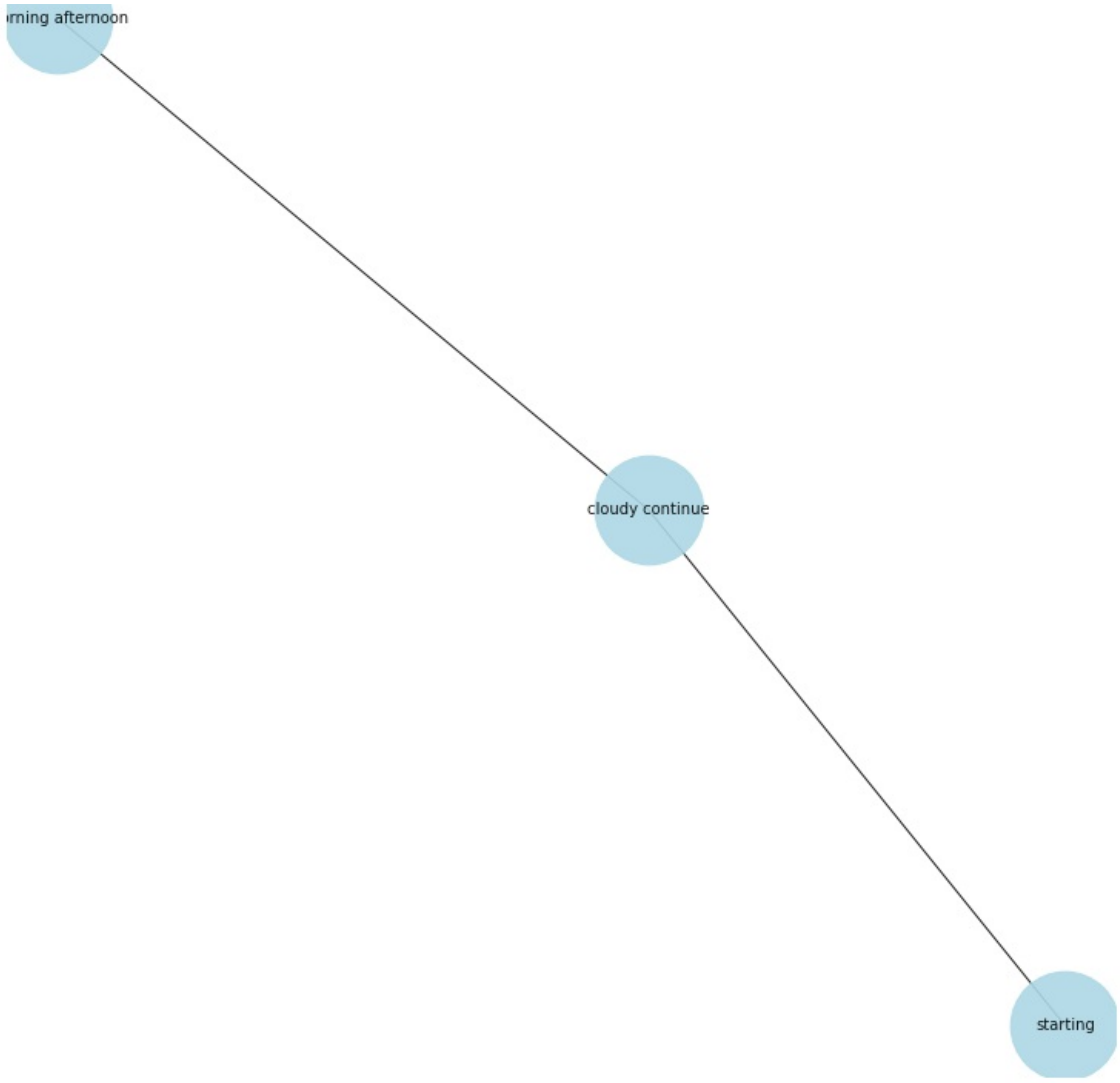

```
Mostly cloudy starting overnight continuing until evening and breezy starting overnight continuing until morning.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
starting -> advcl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
. -> punct
starting , cloudy continue , evening morning
```



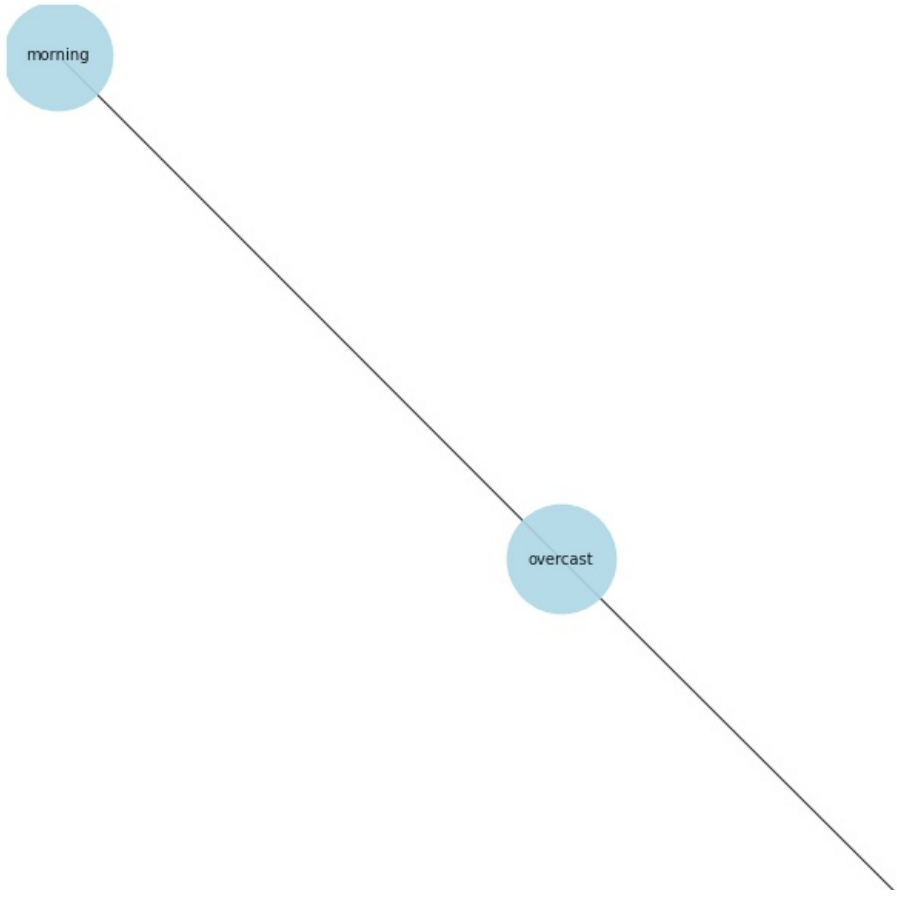

```
Partly cloudy starting overnight continuing until evening and breezy in the morning.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
starting , cloudy continue , evening morning
```



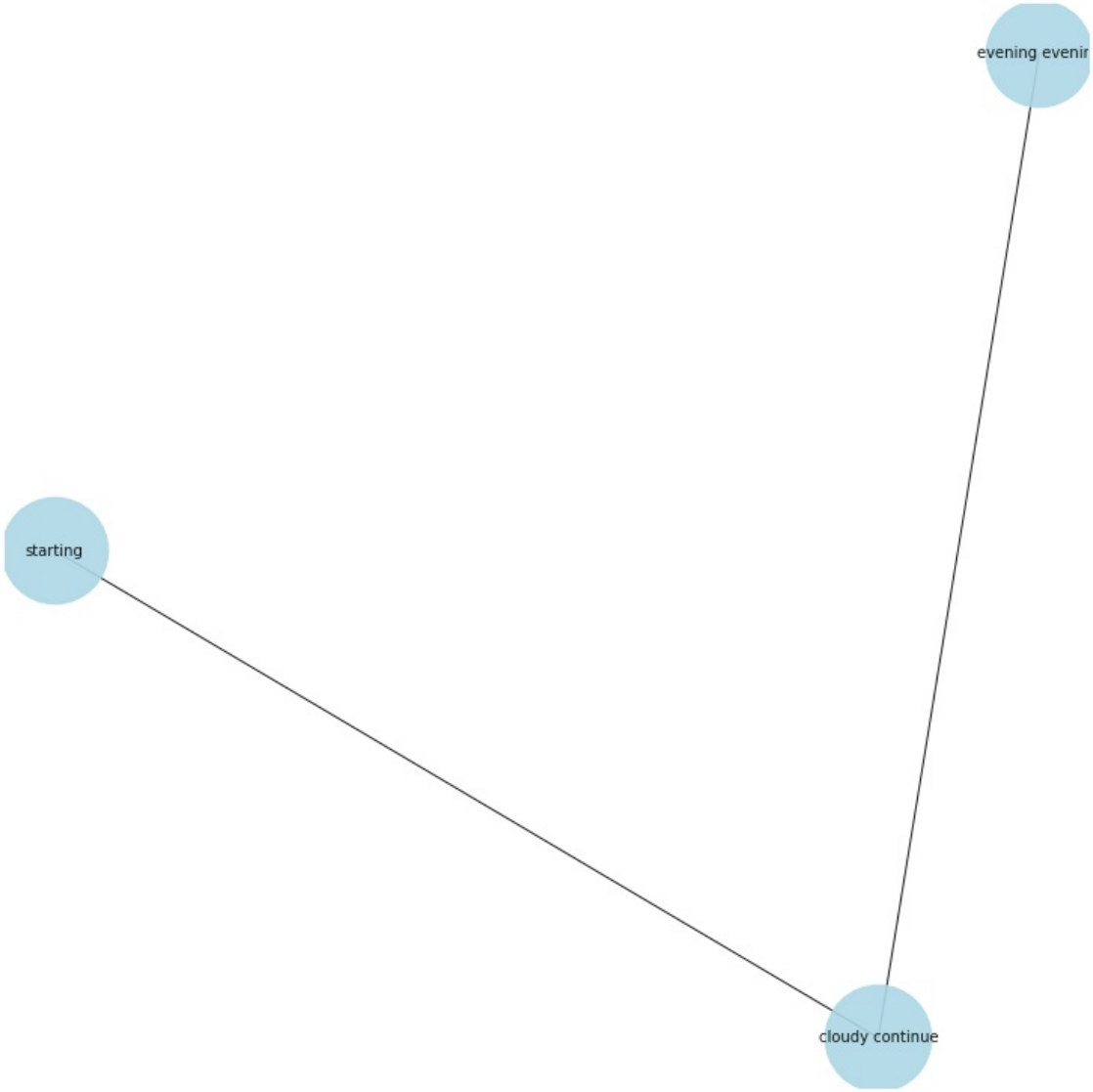

```
Mostly cloudy starting in the morning continuing until afternoon.
Mostly -> advmod
cloudy -> amod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy continue , morning afternoon
```



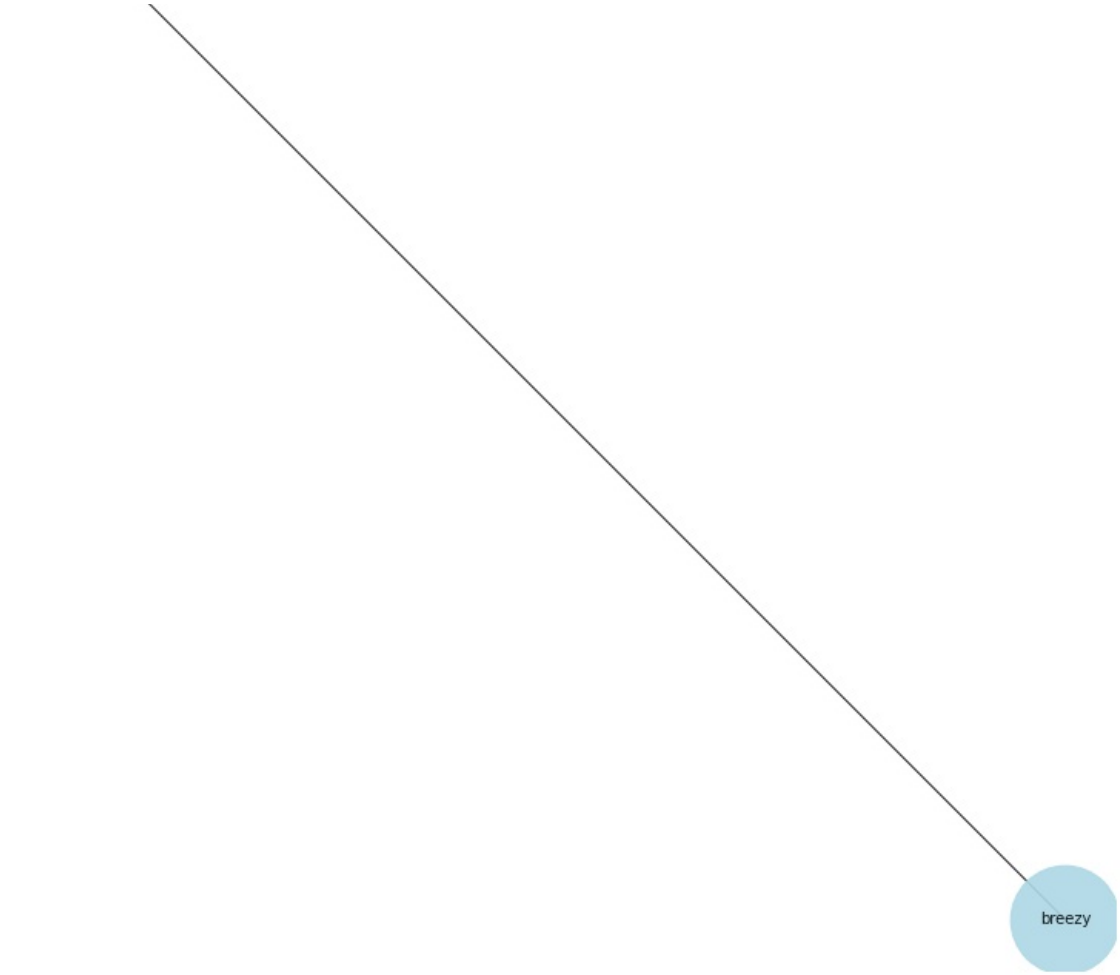

```
Overcast starting in the morning.
Overcast -> ROOT
starting -> acl
in -> prep
the -> det
morning -> pobj
. -> punct
 , overcast , morning
```



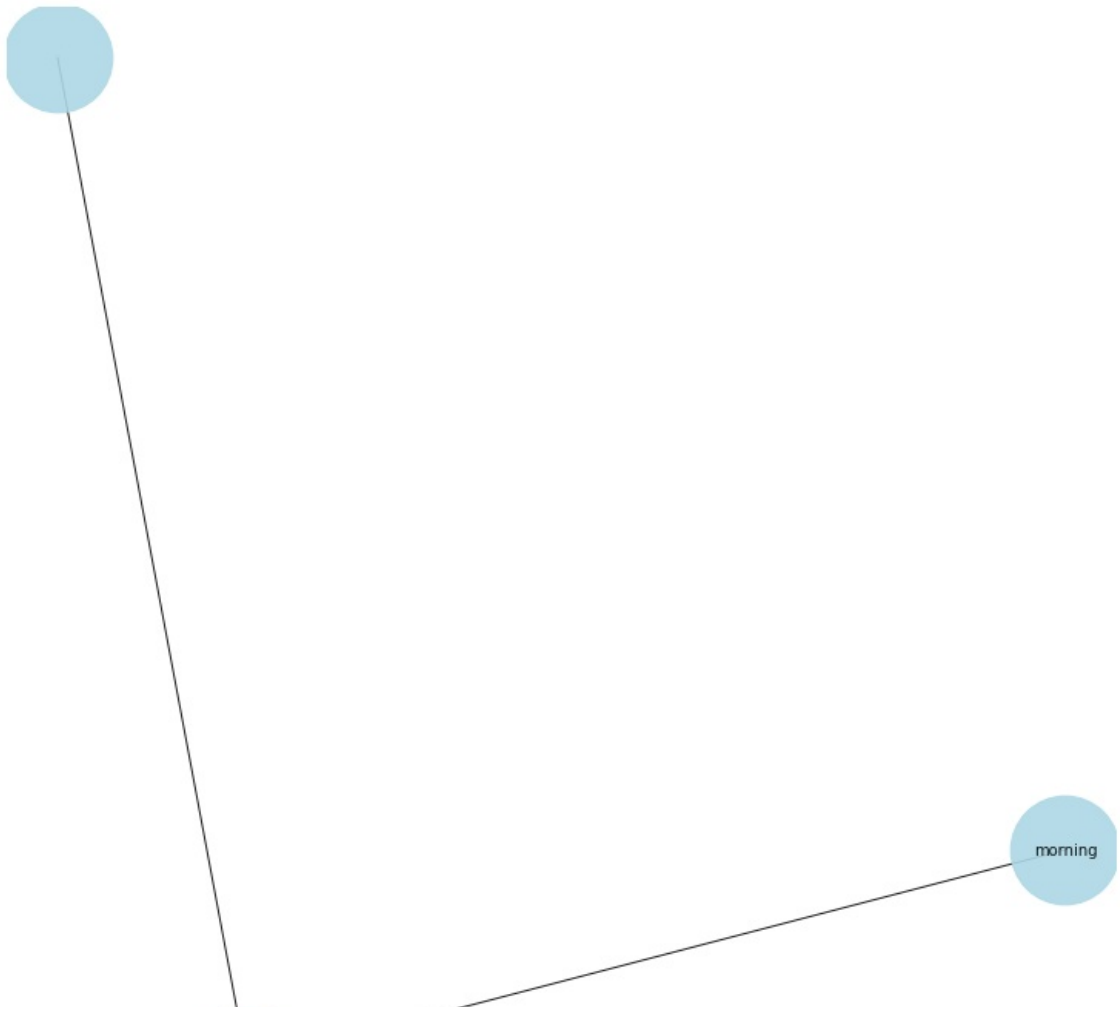
```
Partly cloudy starting overnight continuing until evening and breezy in the evening.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
starting , cloudy continue , evening evening
```

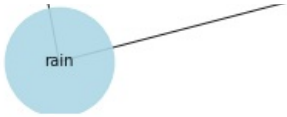



```
Breezy and mostly cloudy overnight.
Breezy -> ROOT
and -> cc
mostly -> advmod
cloudy -> conj
overnight -> advmod
. -> punct
 , breezy ,
```

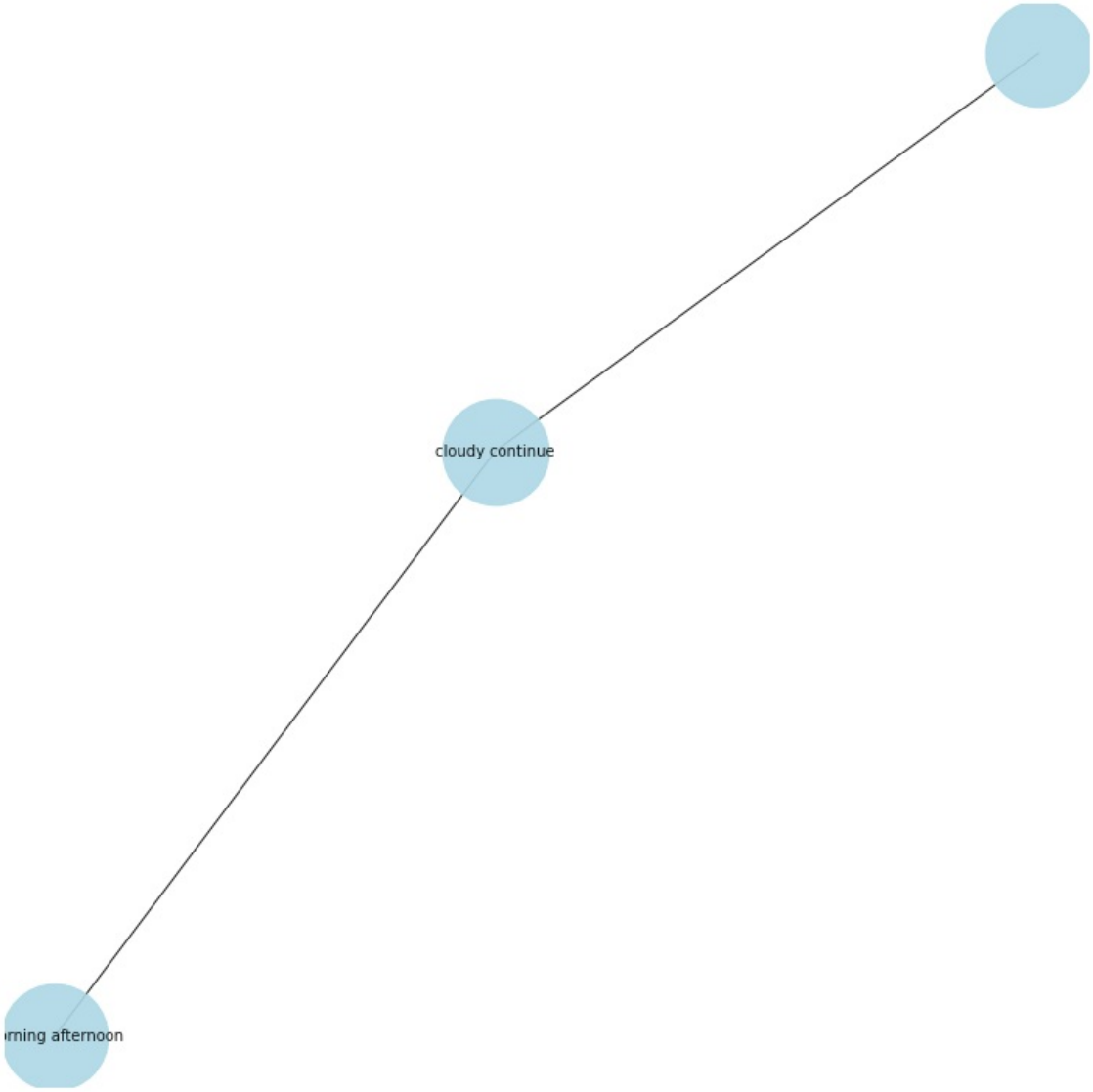


Rain until morning.
Rain -> ROOT
until -> prep
morning -> pobj
. -> punct
 , rain , morning

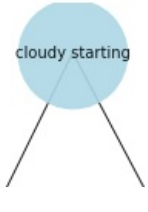


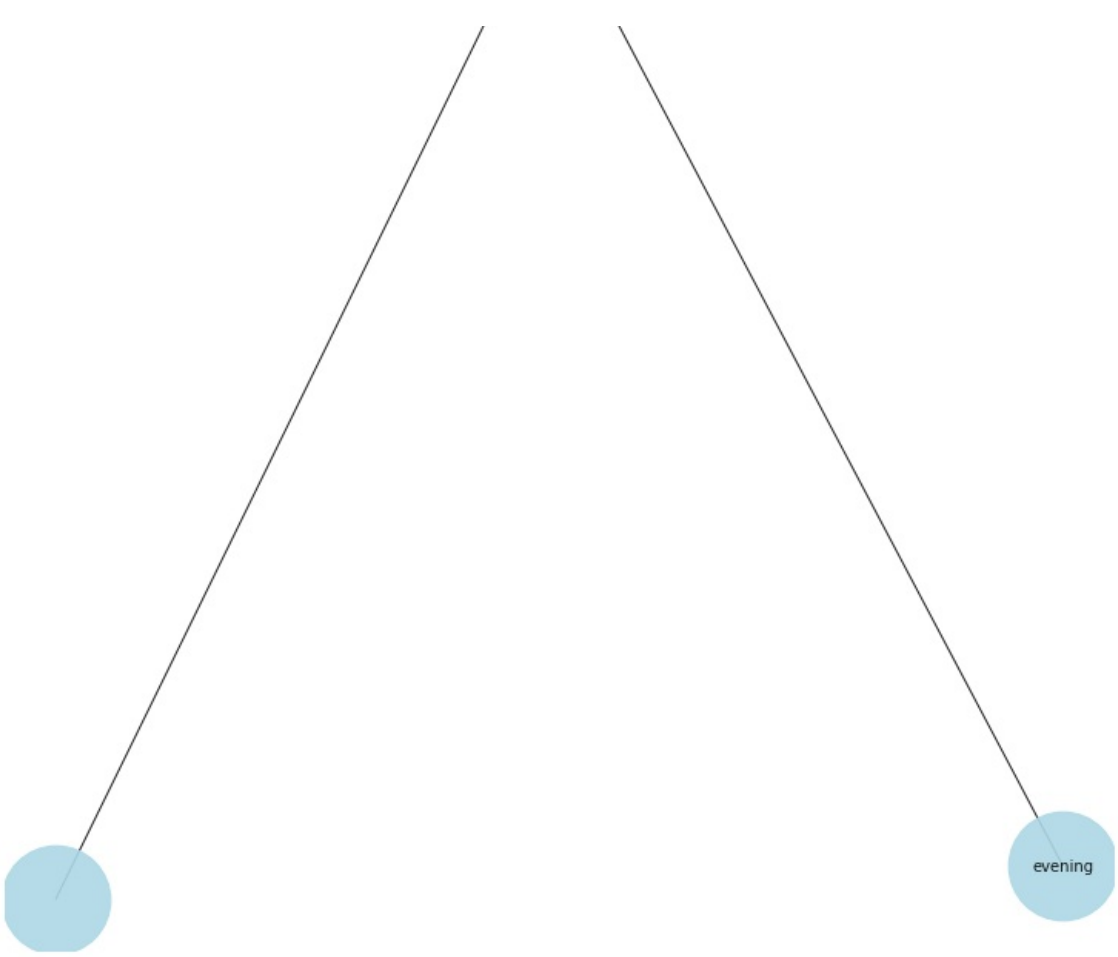



```
Partly cloudy starting overnight and breezy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> amod
starting -> advcl
overnight -> advmod
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
afternoon -> pobj
. -> punct
 , cloudy continue , morning afternoon
```

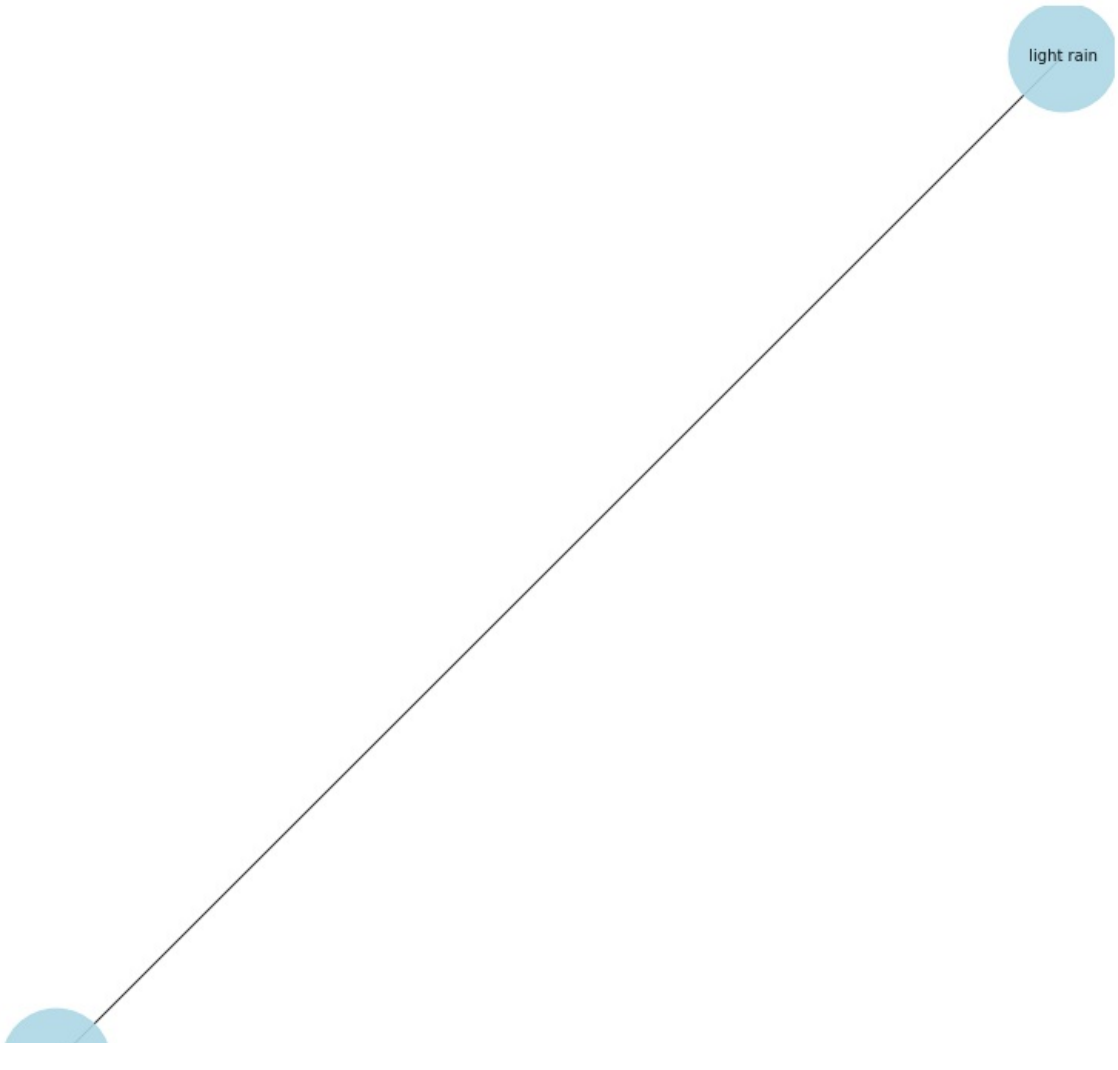


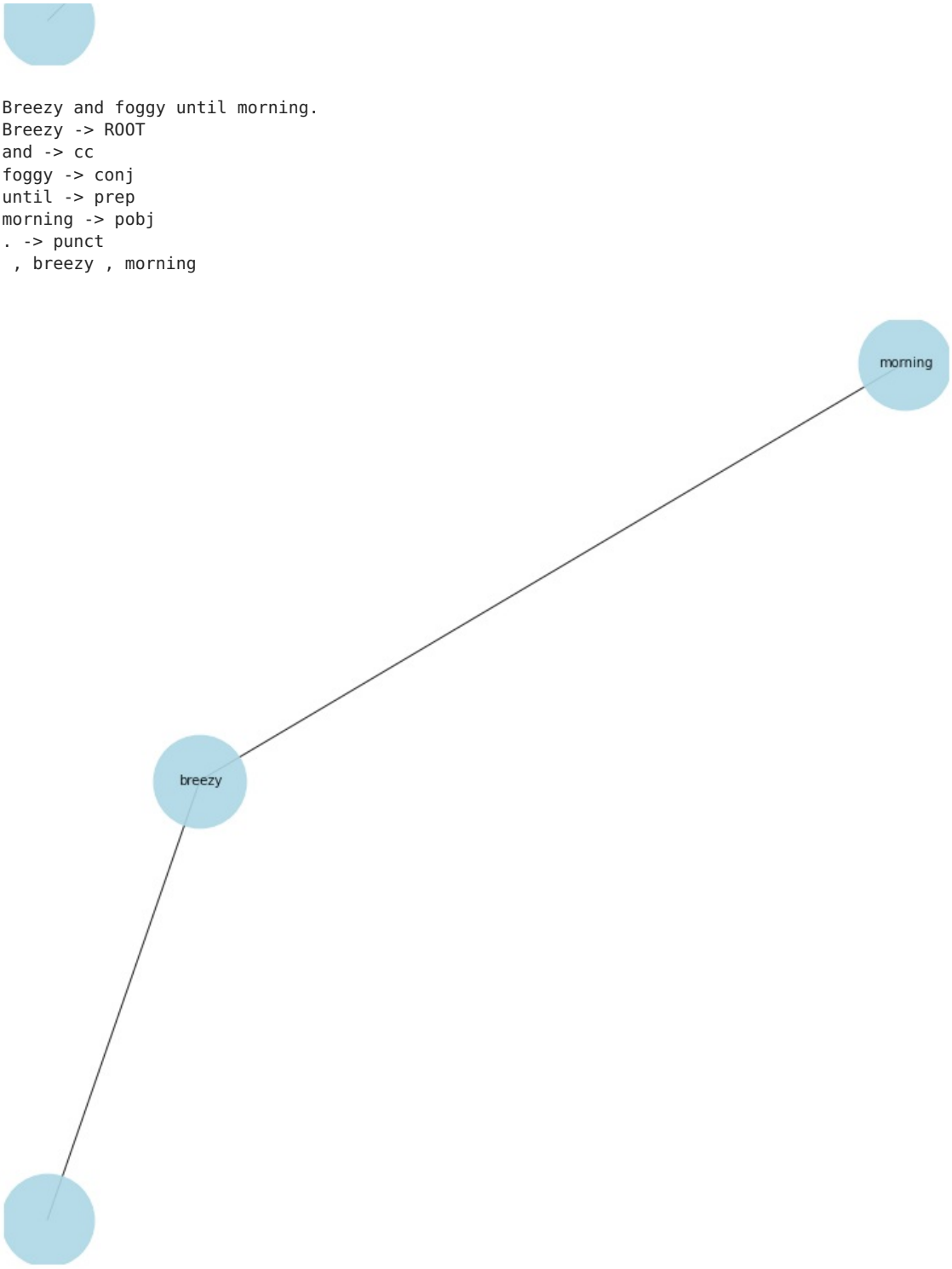

```
Mostly cloudy starting in the evening.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy starting , evening
```



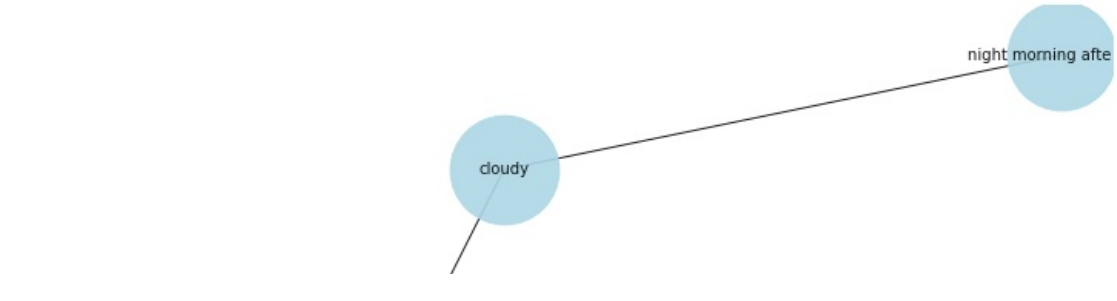



```
Light rain starting overnight.
Light -> amod
rain -> ROOT
starting -> acl
overnight -> advmod
. -> punct
 , light rain ,
```

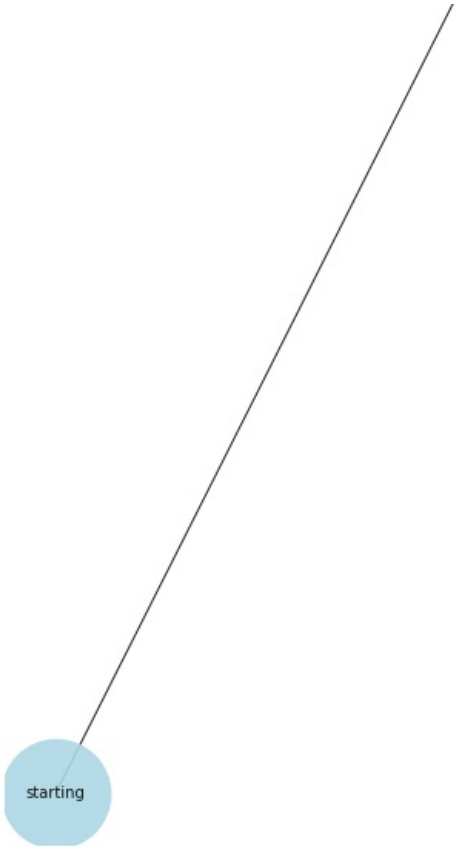




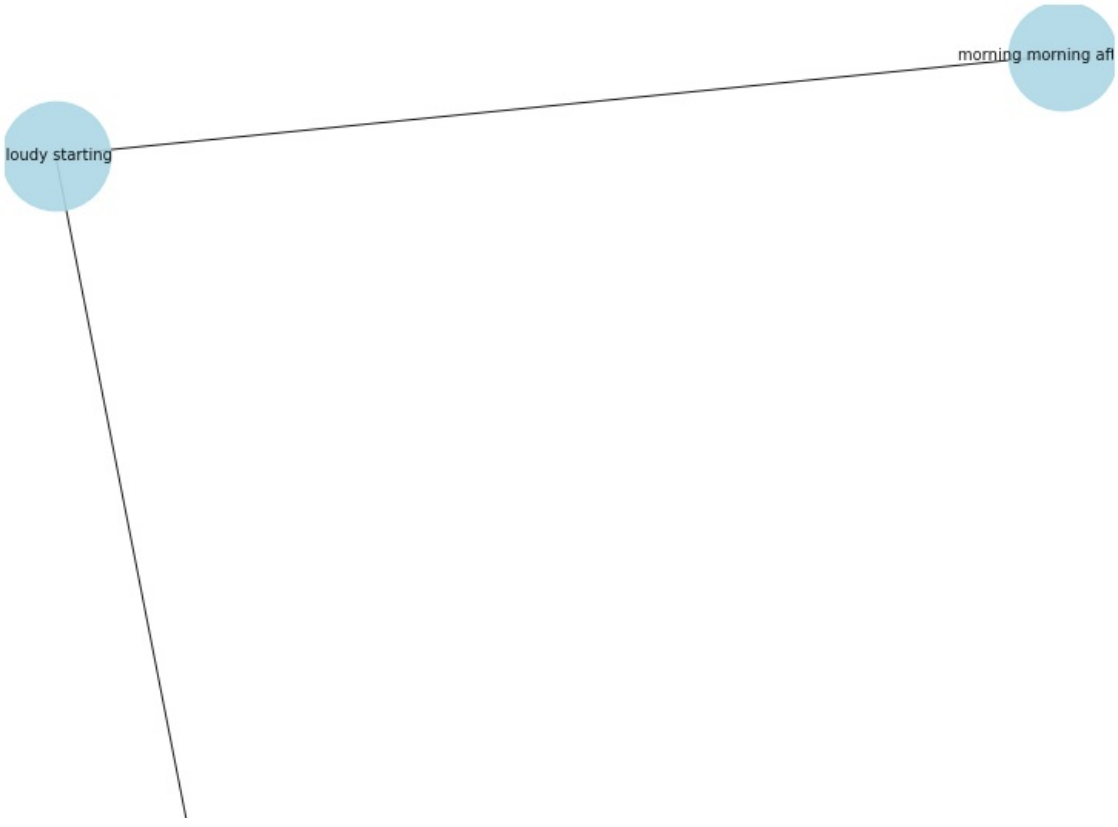
Breezy and foggy until morning.
Breezy -> ROOT
and -> cc
foggy -> conj
until -> prep
morning -> pobj
. -> punct
 , breezy , morning



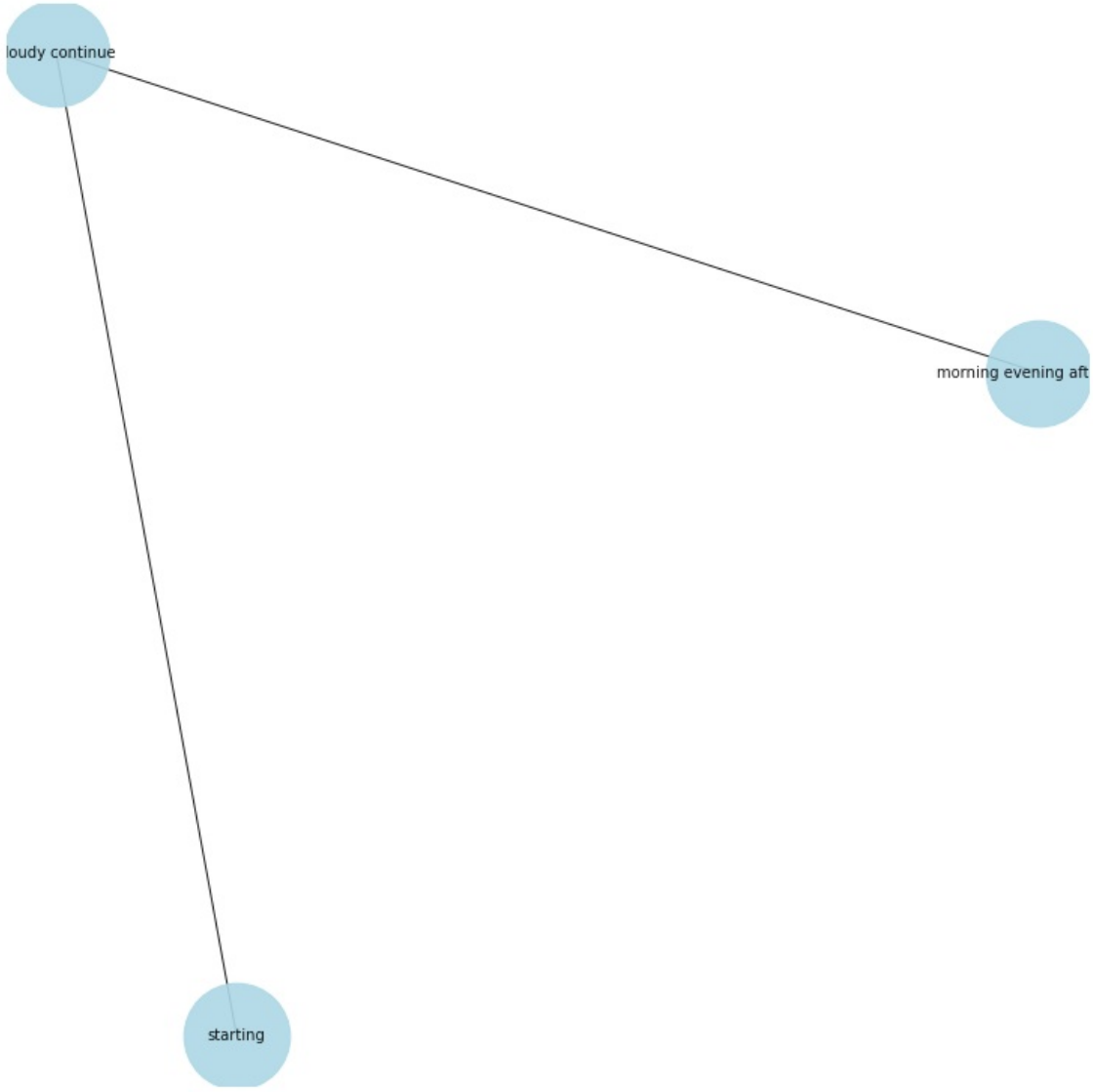
Mostly cloudy until night and breezy starting in the morning continuing until afternoon.
Mostly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy , night morning afternoon




```
Partly cloudy starting in the morning and breezy starting in the morning continuing until afternoon.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> xcomp
until -> prep
afternoon -> pobj
. -> punct
 , cloudy starting , morning morning afternoon
```

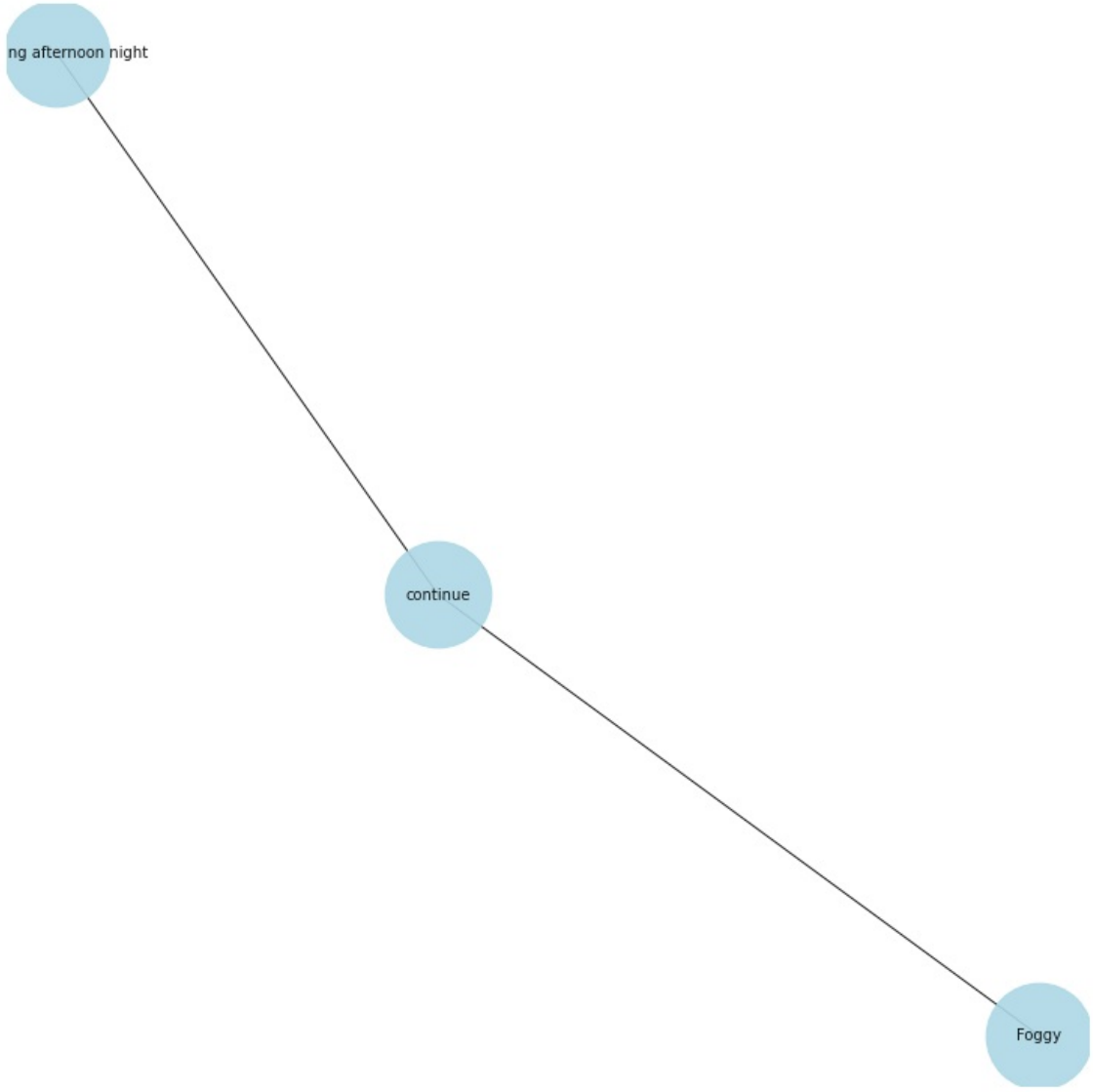


```
Partly cloudy starting in the morning continuing until evening and breezy in the afternoon.
Partly -> advmod
cloudy -> amod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
starting , cloudy continue , morning evening afternoon
```

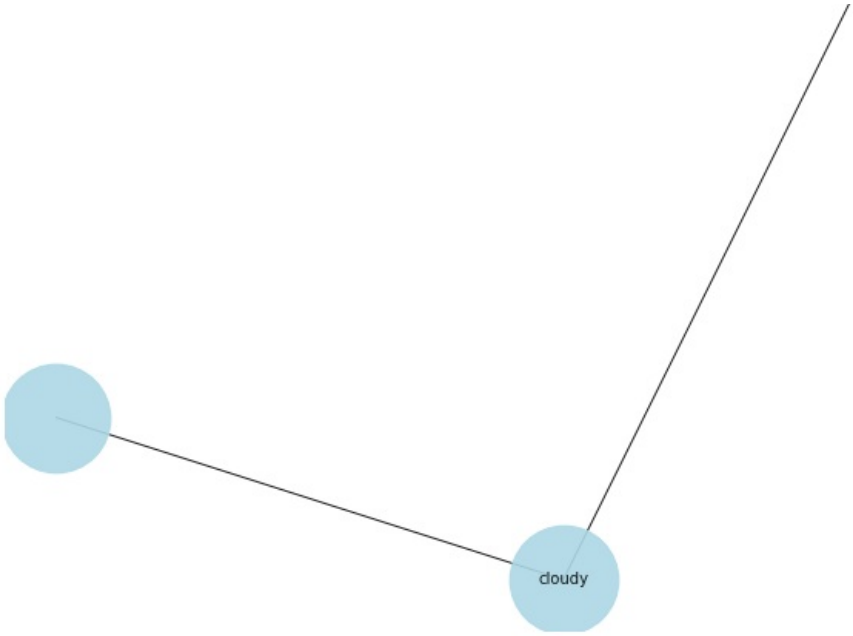



```
Foggy in the morning and breezy starting in the afternoon continuing until night.
Foggy -> nsubj
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
starting -> acl
in -> prep
```

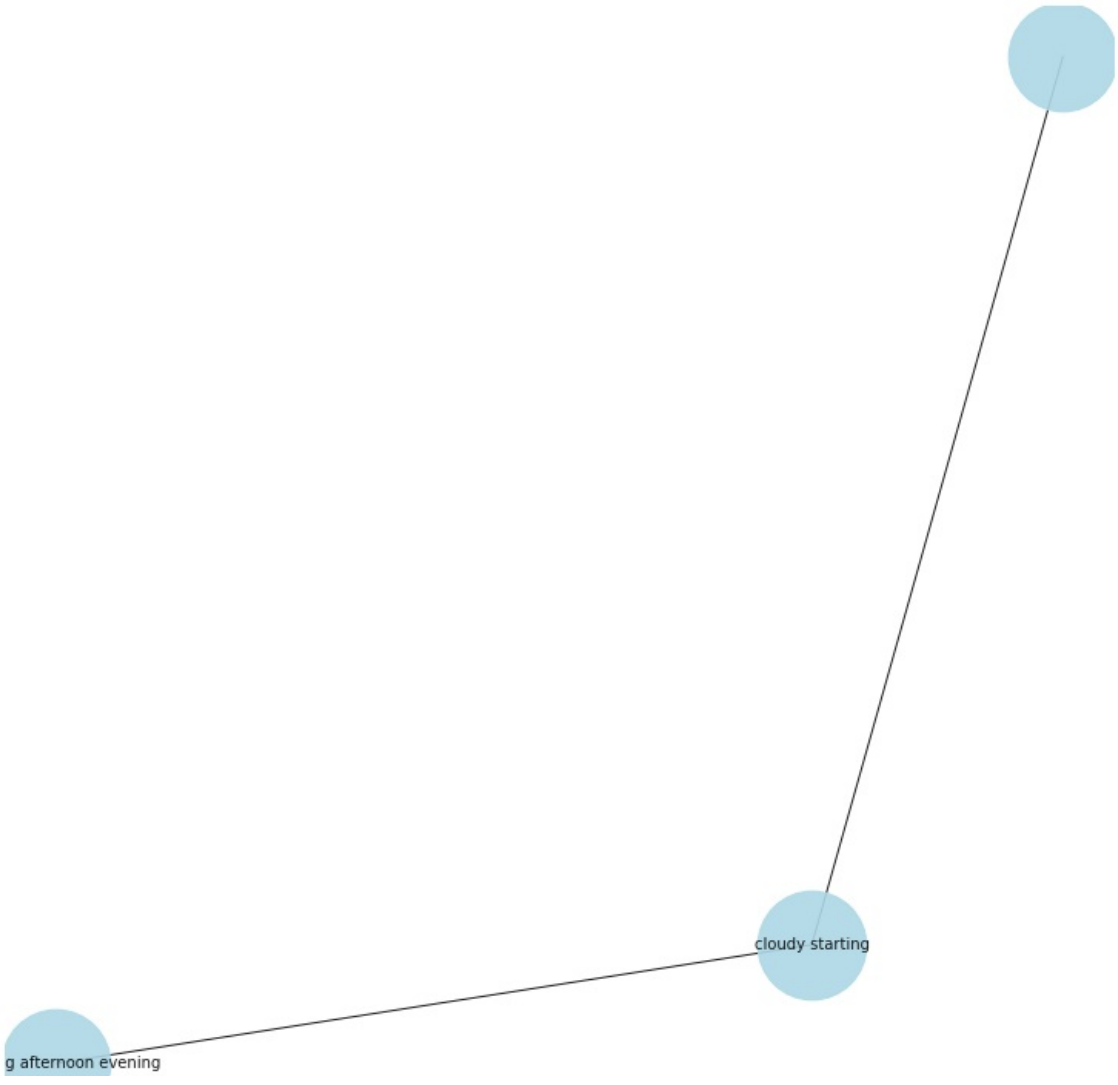
```
the -> det
afternoon -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
Foggy , continue , morning afternoon night
```



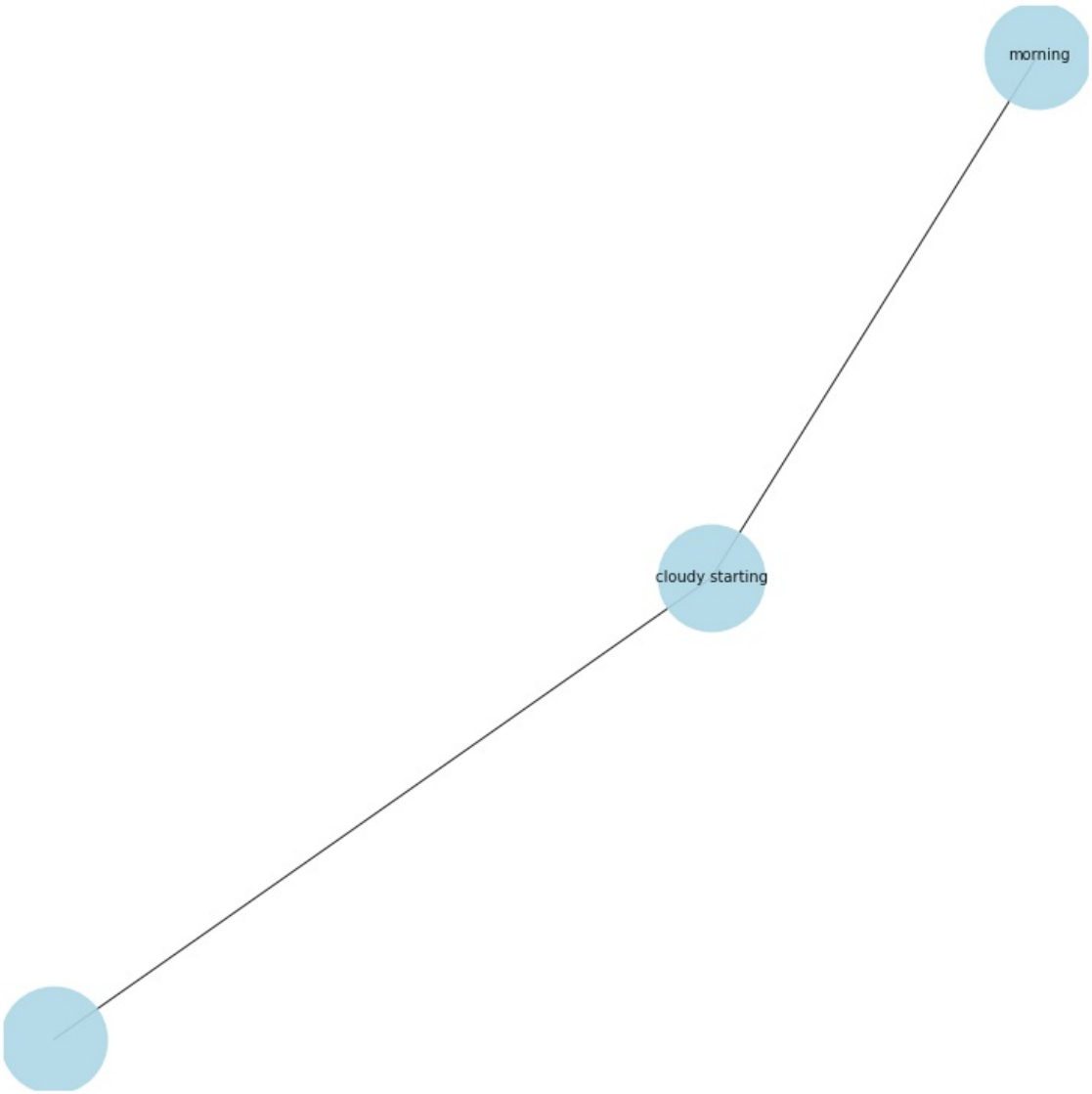

```
Mostly cloudy until night and breezy starting in the afternoon.
Mostly -> advmod
cloudy -> ROOT
until -> mark
night -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , night afternoon
```



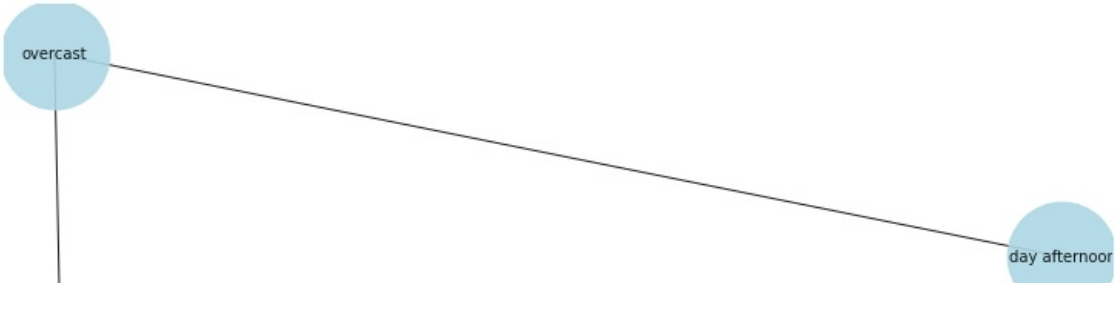

```
Partly cloudy starting in the morning and breezy starting in the afternoon continuing until evening.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> xcomp
until -> prep
evening -> pobj
. -> punct
 , cloudy starting , morning afternoon evening
```



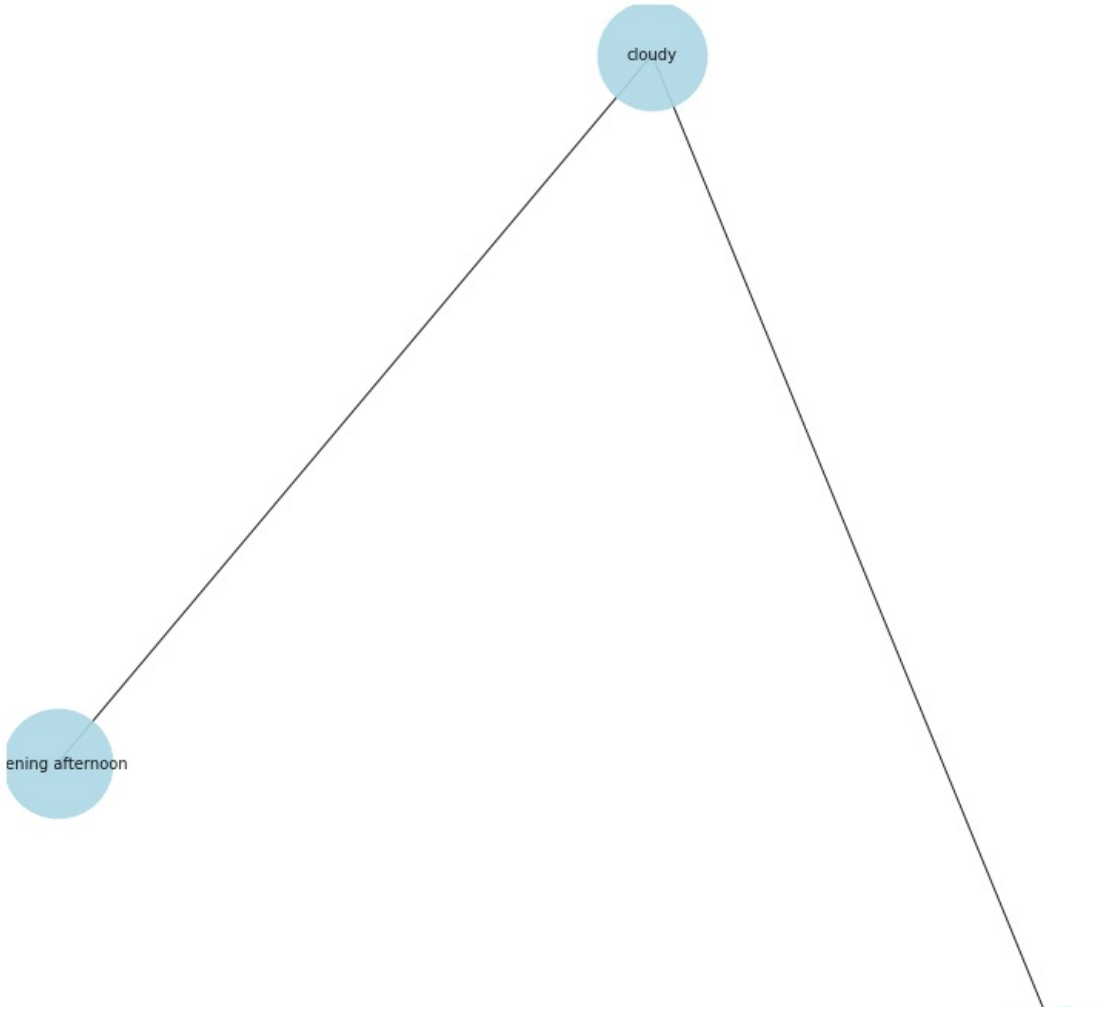
```
Mostly cloudy starting in the morning and breezy overnight.
Mostly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
breezy -> conj
overnight -> advmod
. -> punct
 , cloudy starting , morning
```



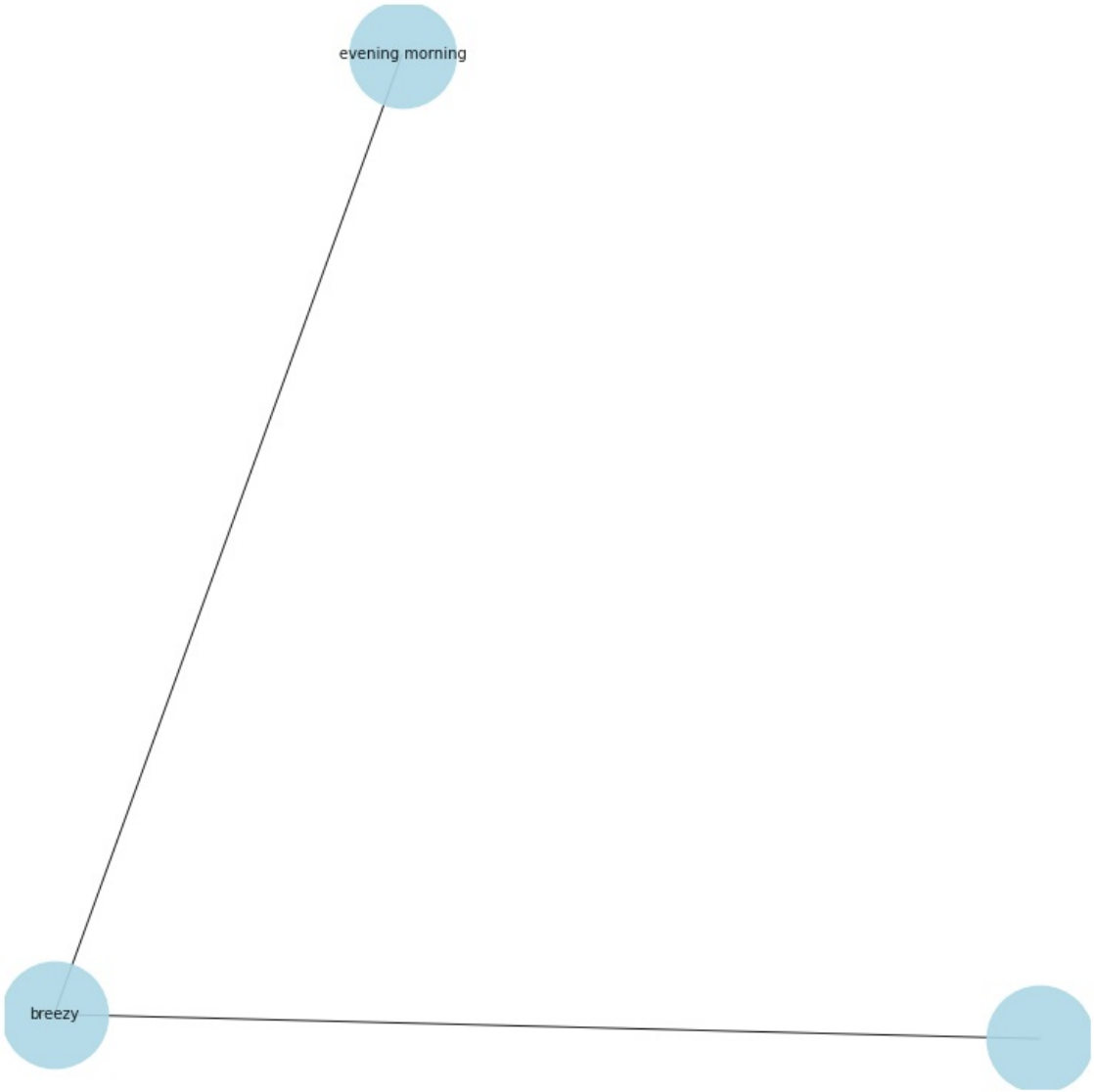

```
Overcast throughout the day and breezy in the afternoon.
Overcast -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , overcast , day afternoon
```



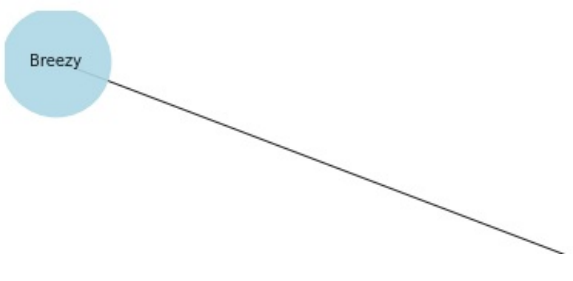
Partly cloudy until evening and breezy in the afternoon.
Partly -> advmod
cloudy -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , evening afternoon

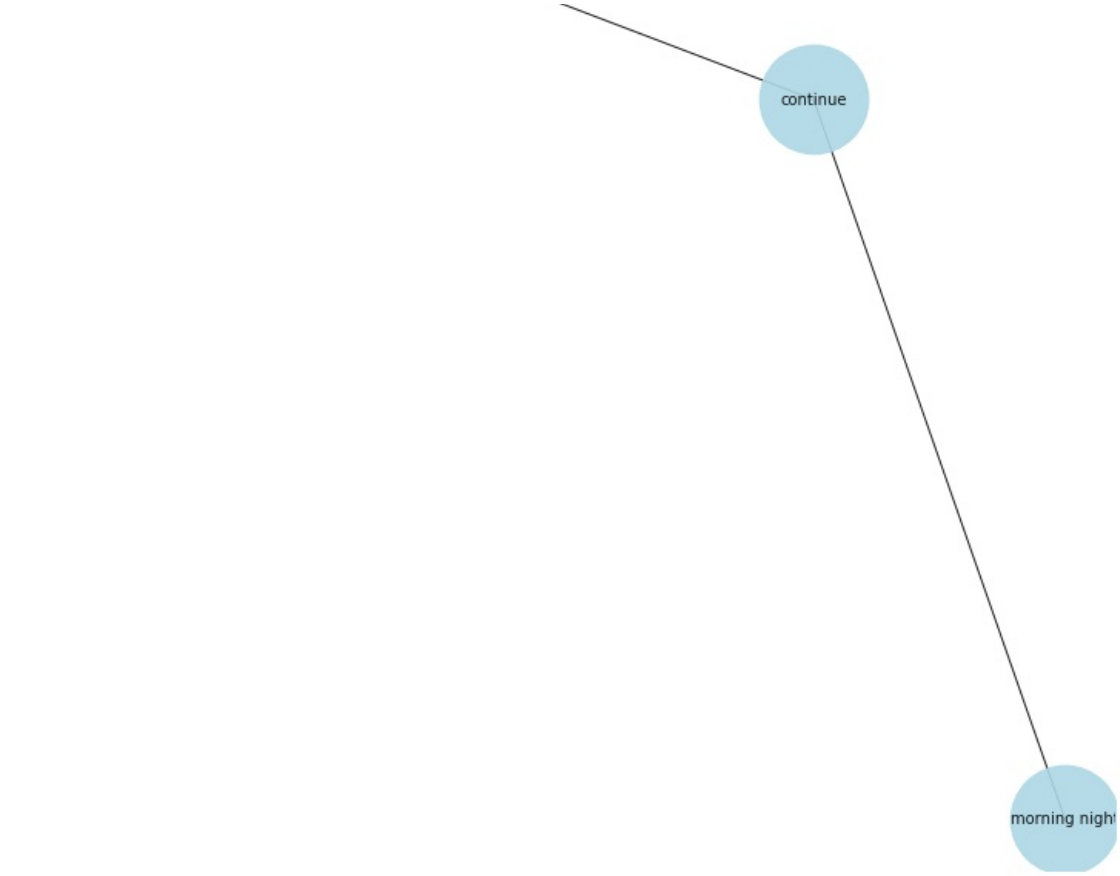


```
Breezy until evening and foggy in the morning.
Breezy -> ROOT
until -> prep
evening -> pobj
and -> cc
foggy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
 , breezy , evening morning
```

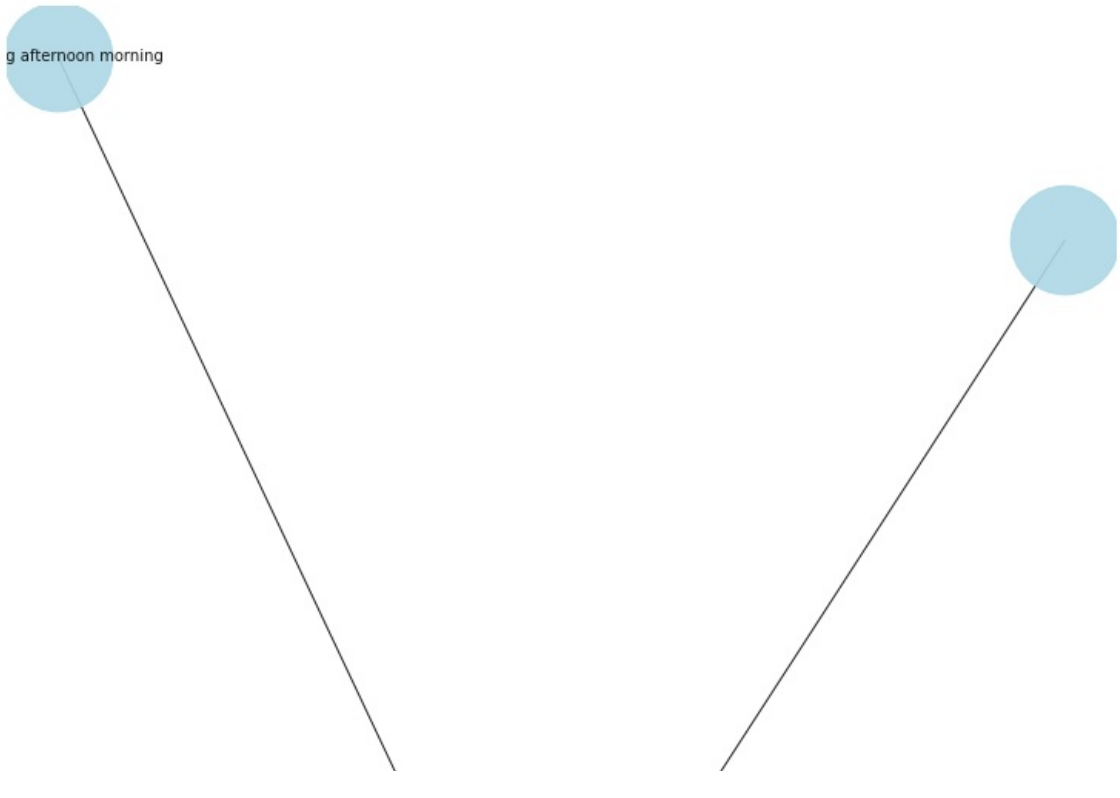


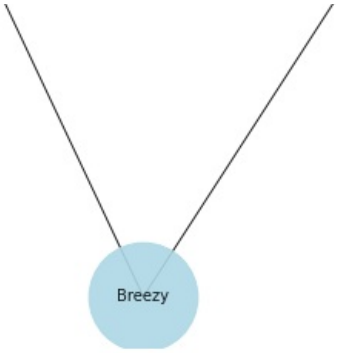

```
Breezy starting in the morning continuing until night.
Breezy -> nsubj
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
night -> pobj
. -> punct
Breezy , continue , morning night
```



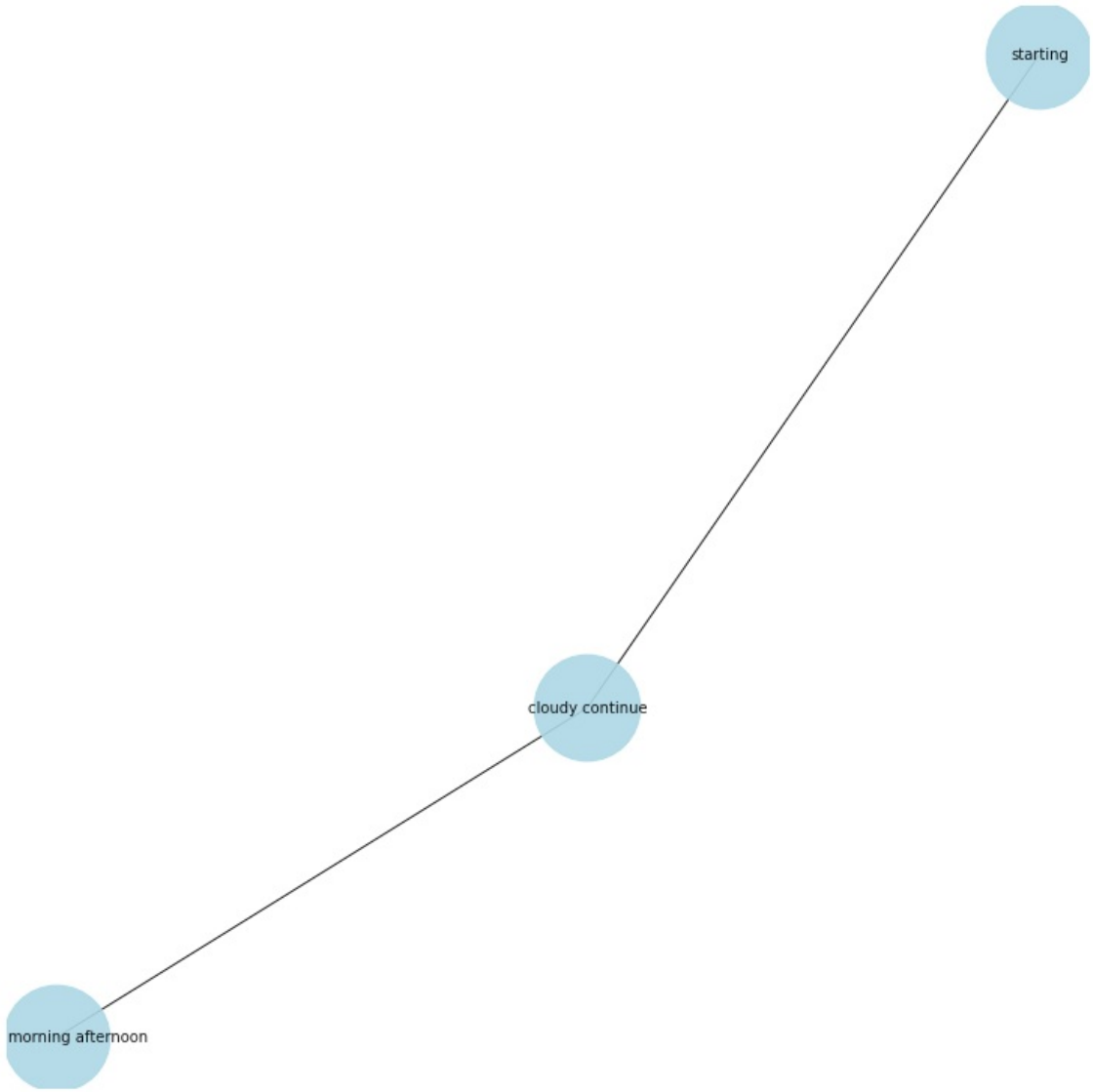


Breezy starting in the morning continuing until afternoon and partly cloudy starting in the morning.
Breezy -> ROOT
starting -> acl
in -> prep
the -> det
morning -> pobj
continuing -> acl
until -> prep
afternoon -> pobj
and -> cc
partly -> advmod
cloudy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
. -> punct
 , Breezy , morning afternoon morning



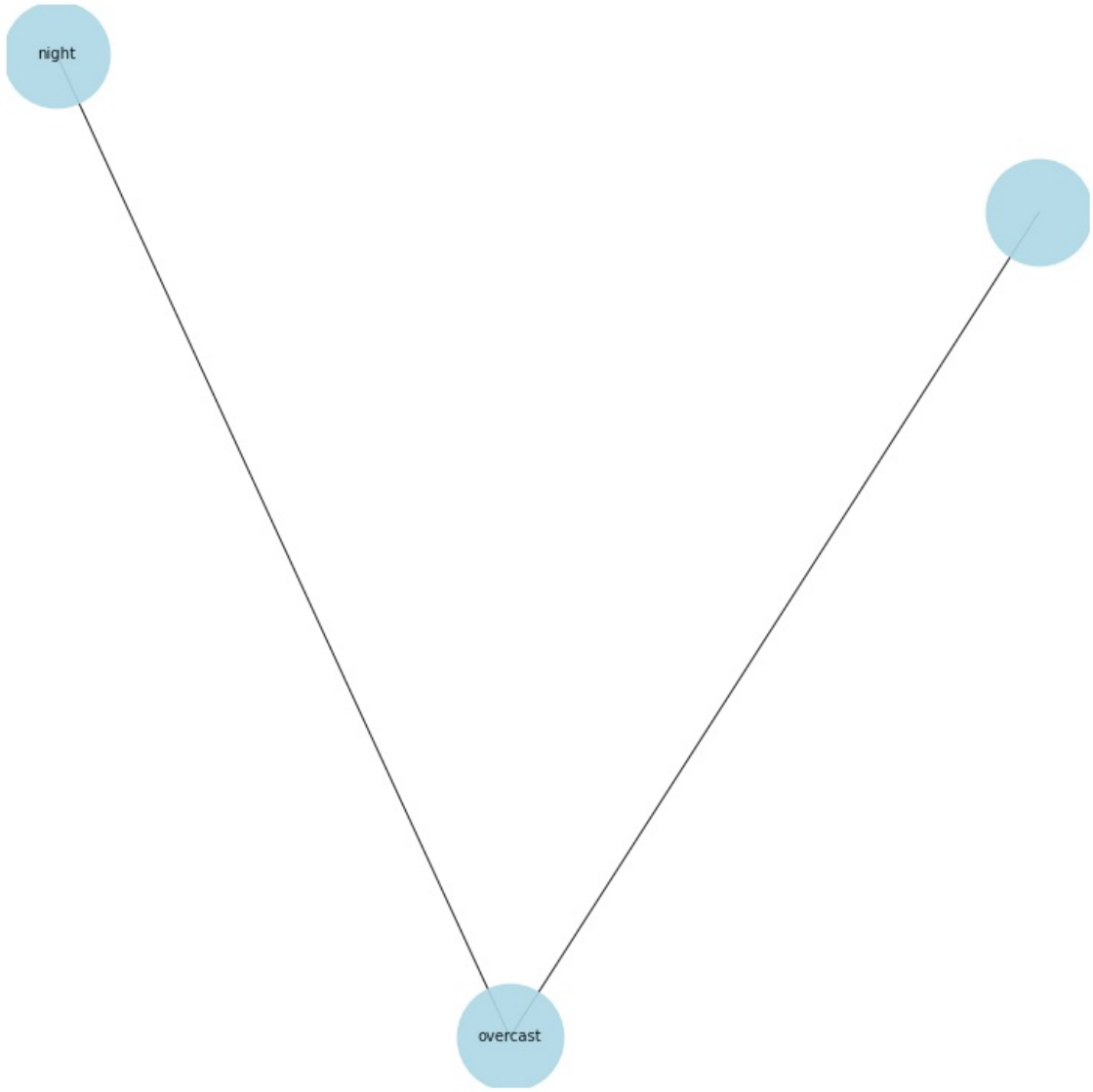



```
Partly cloudy starting overnight continuing until night and windy starting in the morning continuing until aftern
oon.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
night -> pobj
and -> cc
windy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
afternoon -> pobj
. -> punct
starting , cloudy continue , night morning afternoon
```

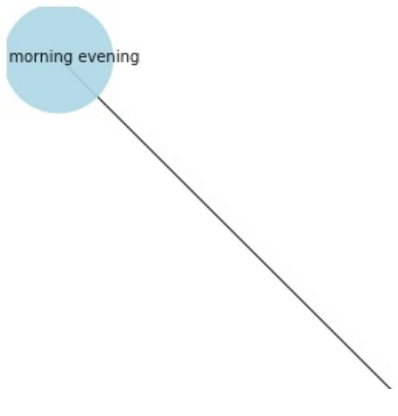


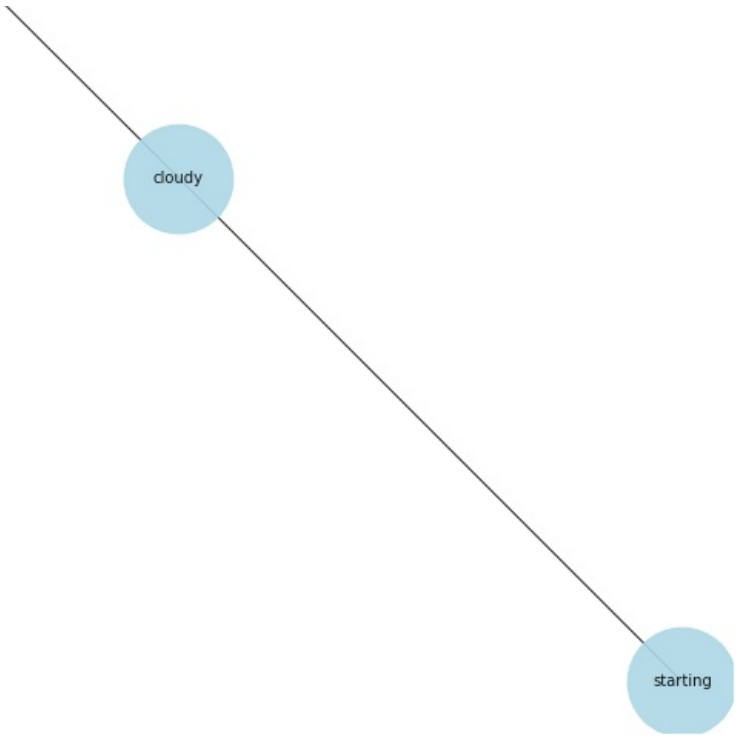

```
Overcast until night and breezy overnight.
Overcast -> ROOT
until -> prep
```

```
night -> pobj
and -> cc
breezy -> conj
overnight -> advmod
. -> punct
 , overcast , night
```

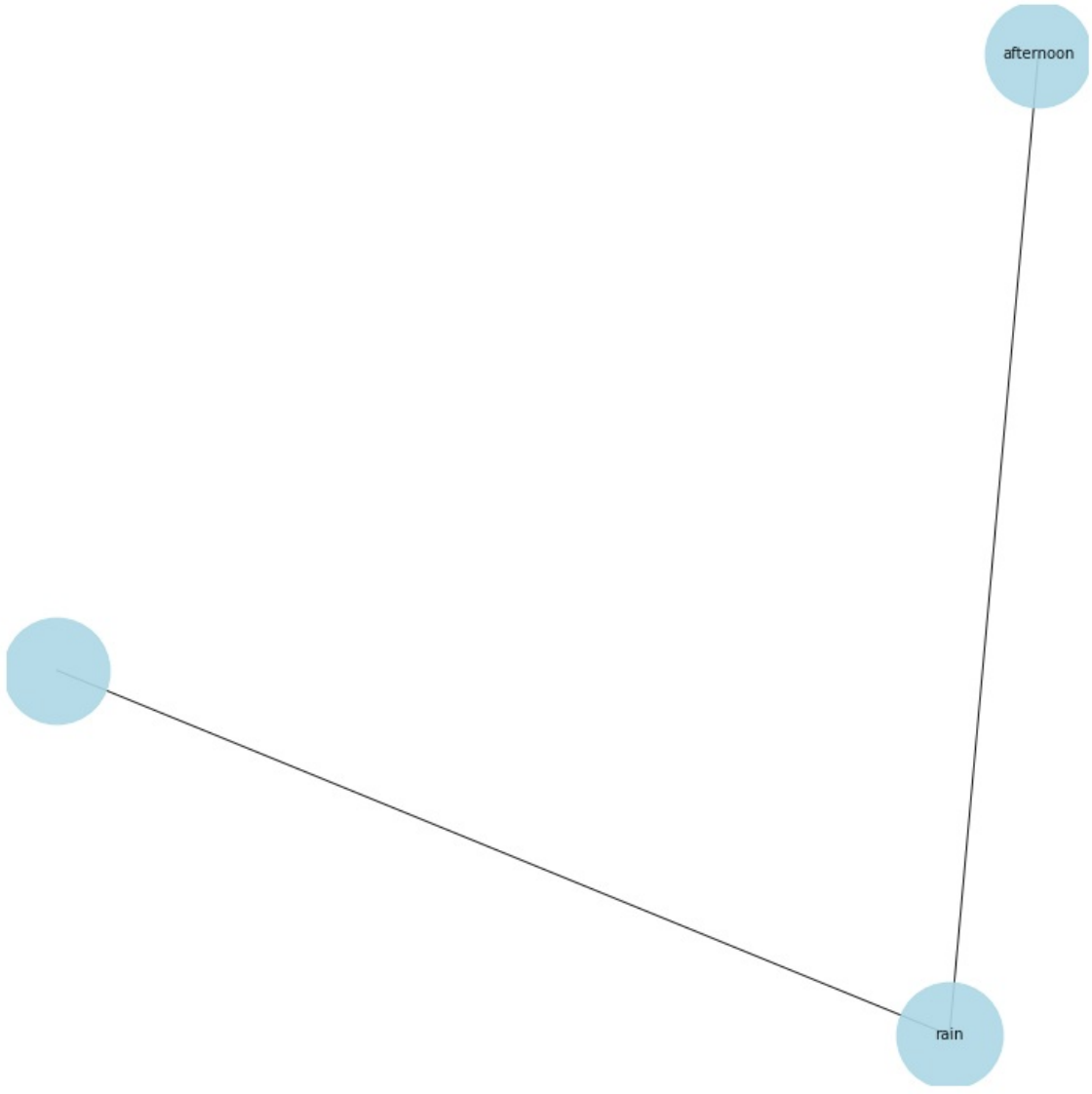



```
Partly cloudy throughout the day and breezy starting in the morning continuing until evening.
Partly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
starting , cloudy , day morning evening
```



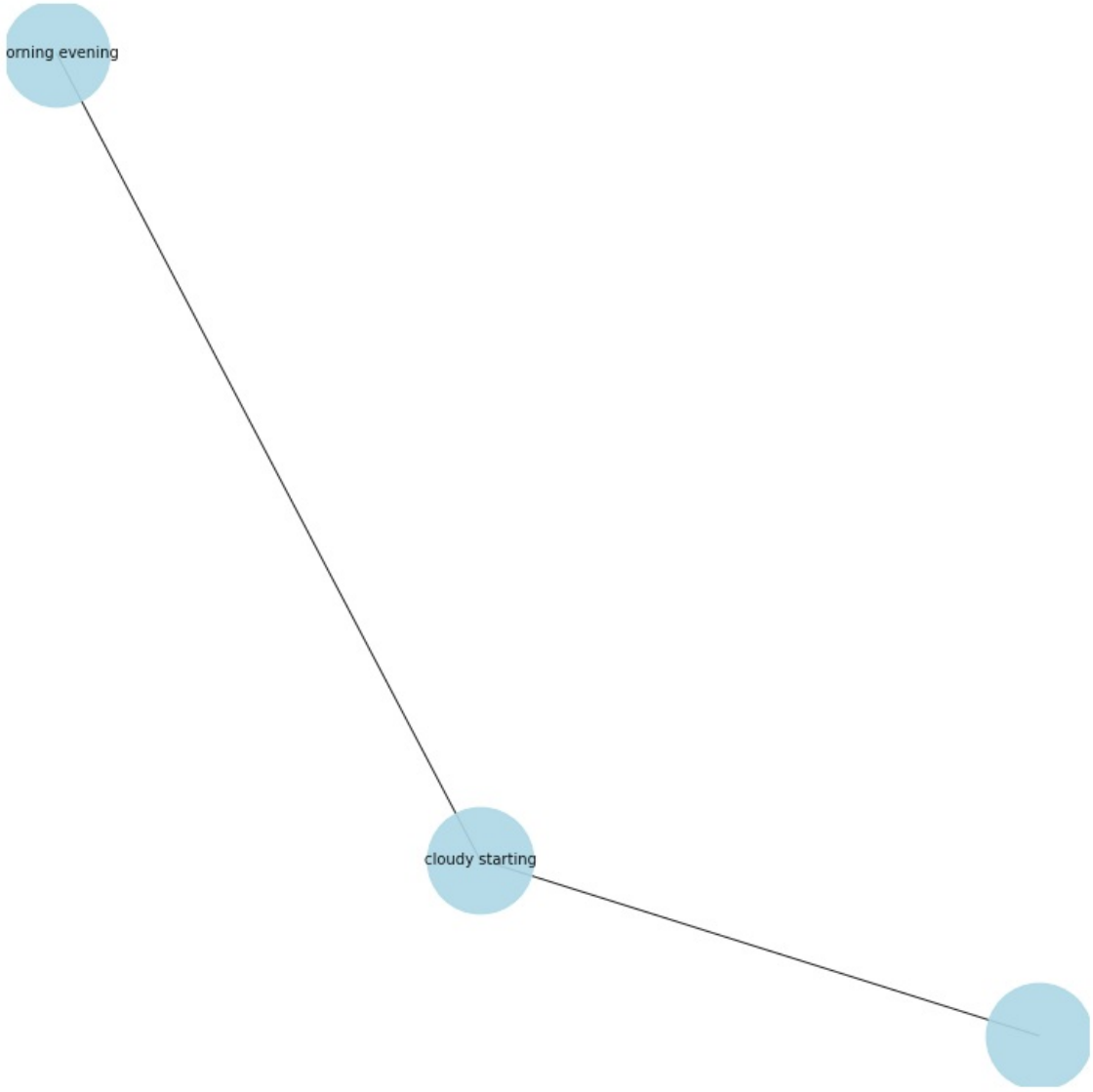



```
Rain until afternoon.
Rain -> ROOT
until -> prep
afternoon -> pobj
. -> punct
 , rain , afternoon
```

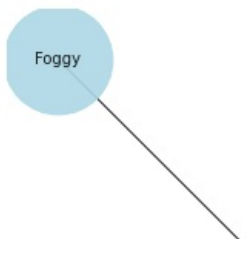


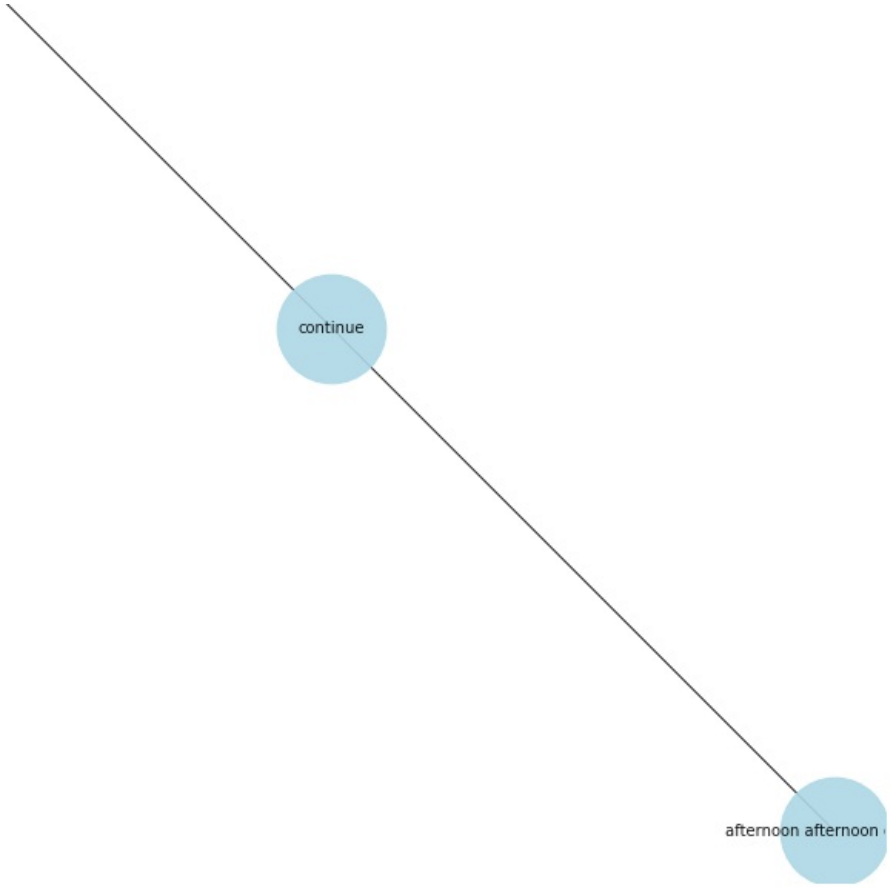

```
Mostly cloudy starting in the morning and windy in the evening.
Mostly -> advmod
cloudy -> amod
```

```
starting -> ROOT
in -> prep
the -> det
morning -> pobj
and -> cc
windy -> conj
in -> prep
the -> det
evening -> pobj
. -> punct
 , cloudy starting , morning evening
```

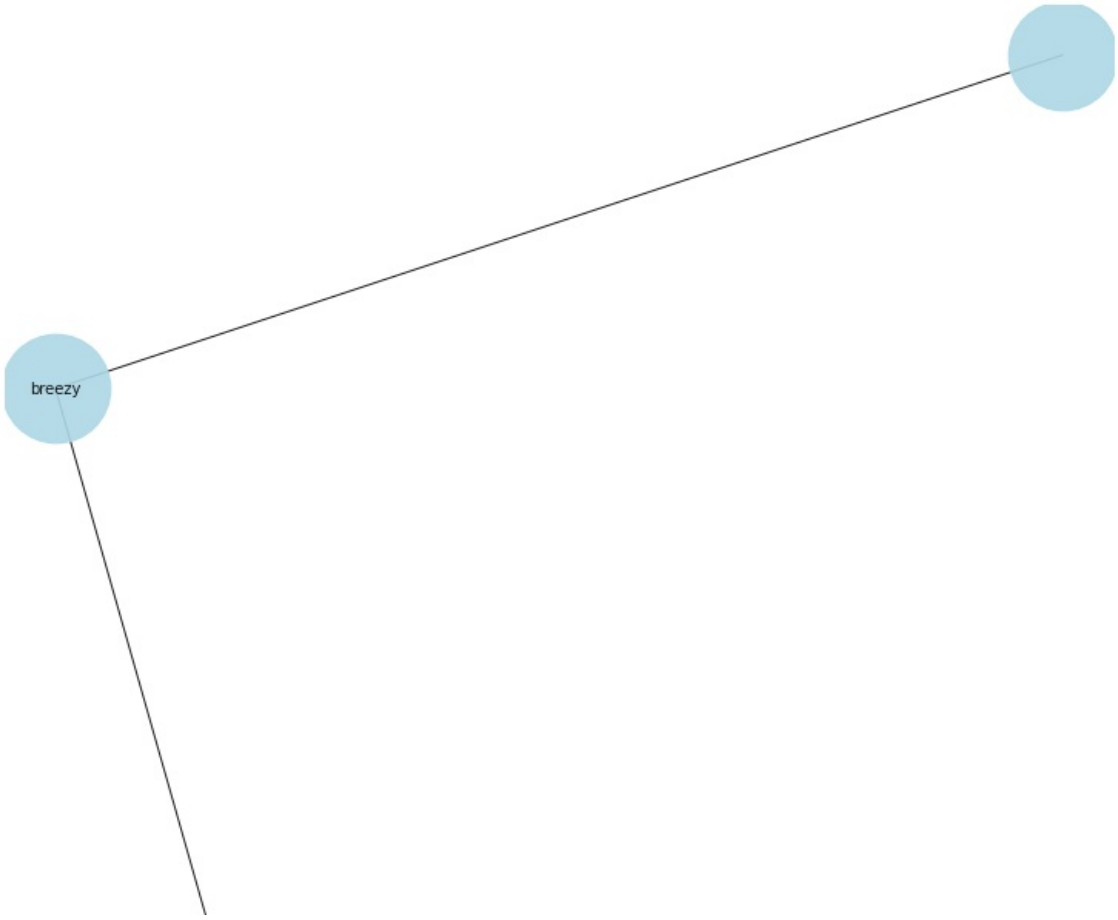



```
Foggy starting in the afternoon and breezy starting in the afternoon continuing until evening.
Foggy -> nsubj
starting -> acl
in -> prep
the -> det
afternoon -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
Foggy , continue , afternoon afternoon evening
```

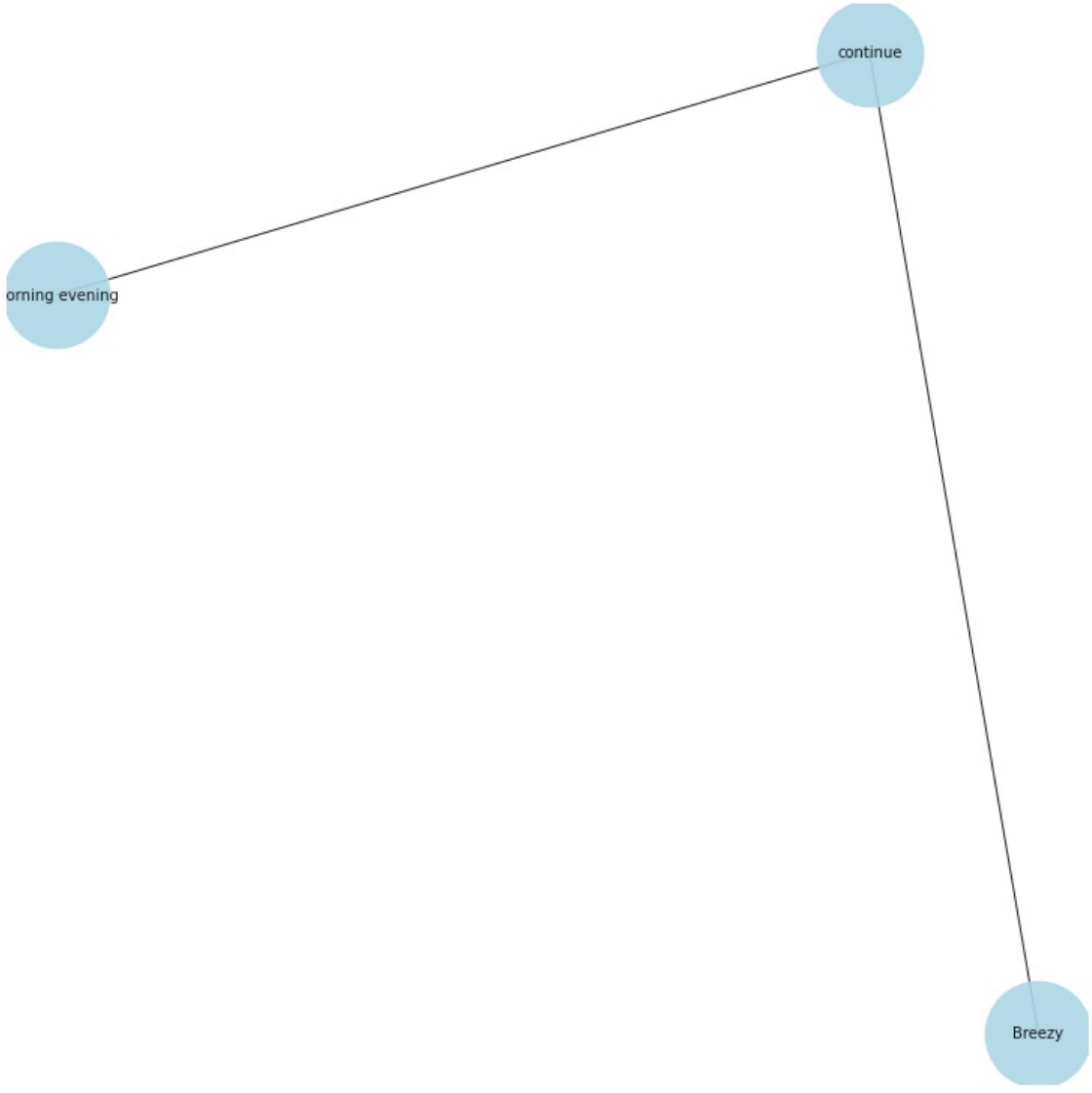




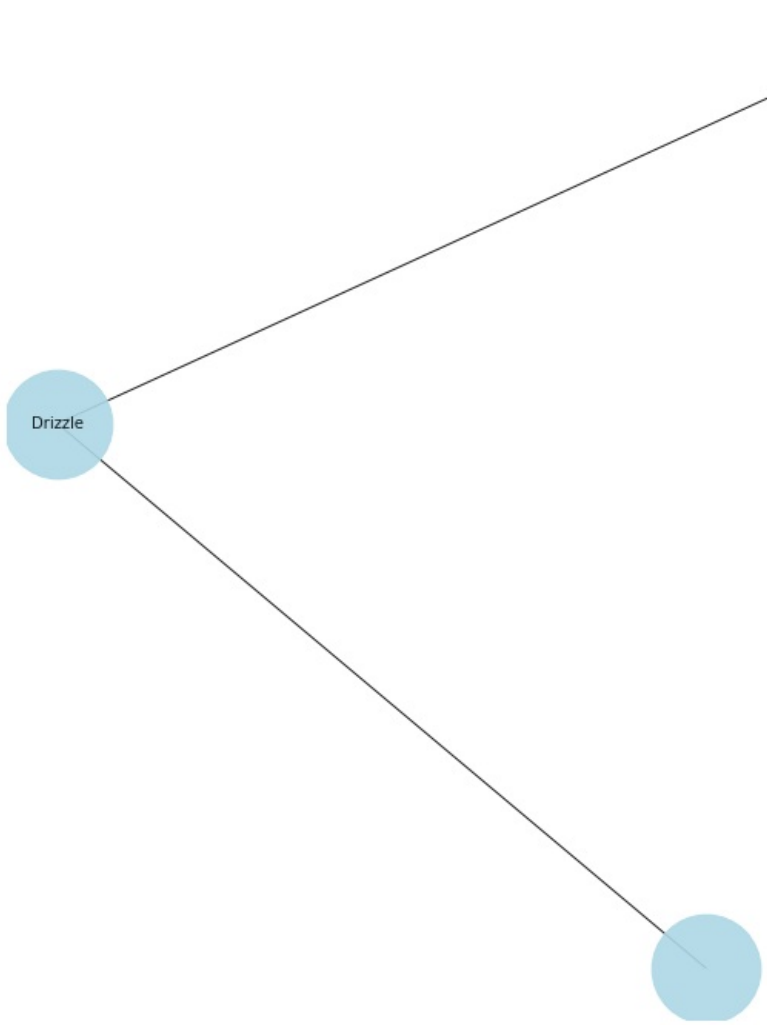

```
Breezy starting overnight continuing until morning and foggy overnight.
Breezy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
foggy -> conj
overnight -> advmod
. -> punct
 , breezy , morning
```



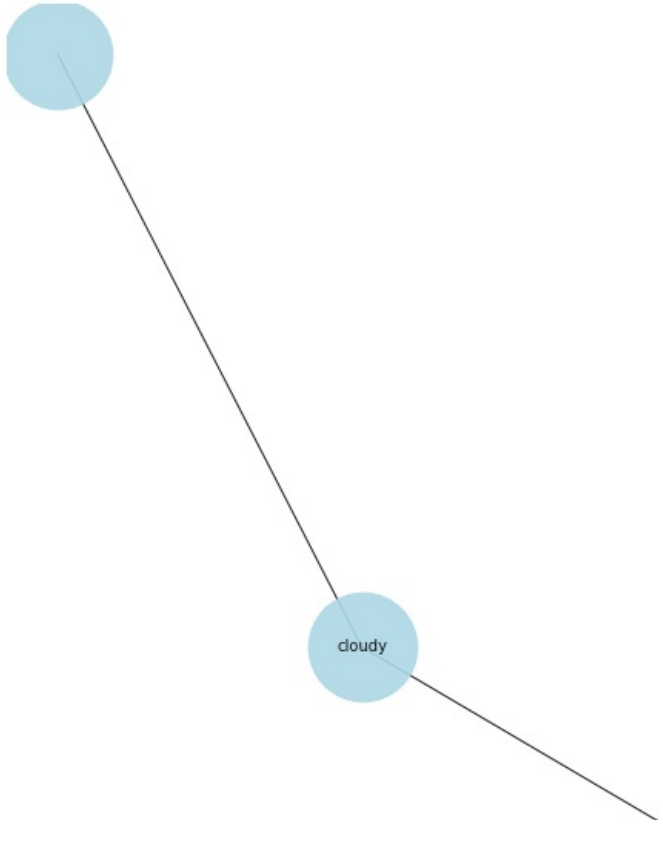

```
Breezy starting overnight continuing until morning and partly cloudy starting overnight continuing until evening.
Breezy -> nsubj
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
partly -> advmod
cloudy -> conj
starting -> dep
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
. -> punct
Breezy , continue , morning evening
```



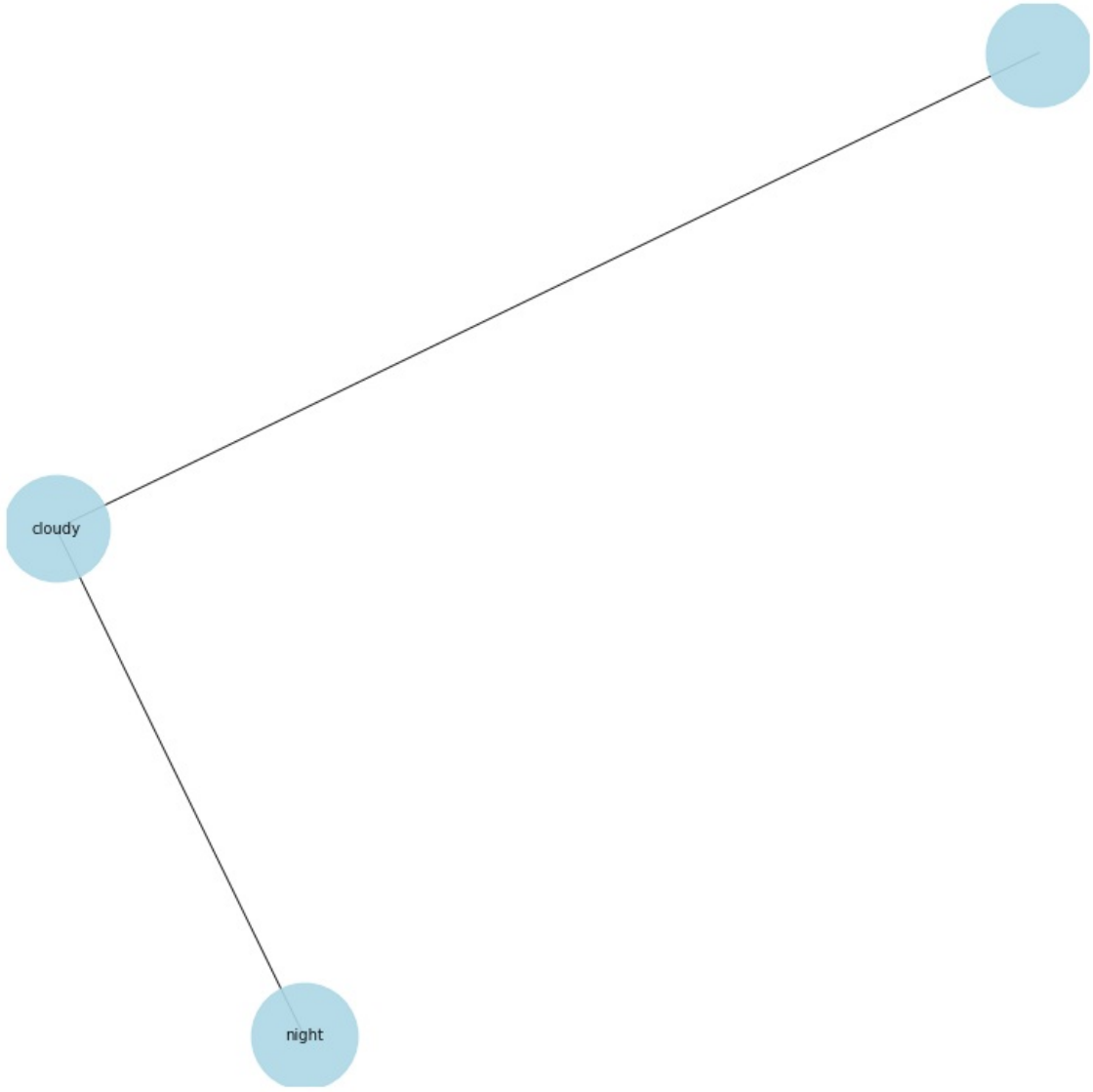

```
Drizzle starting in the evening.
Drizzle -> ROOT
starting -> acl
in -> prep
the -> det
evening -> pobj
. -> punct
 , Drizzle , evening
```




```
Mostly cloudy until evening and breezy in the afternoon.
Mostly -> advmod
cloudy -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , cloudy , evening afternoon
```

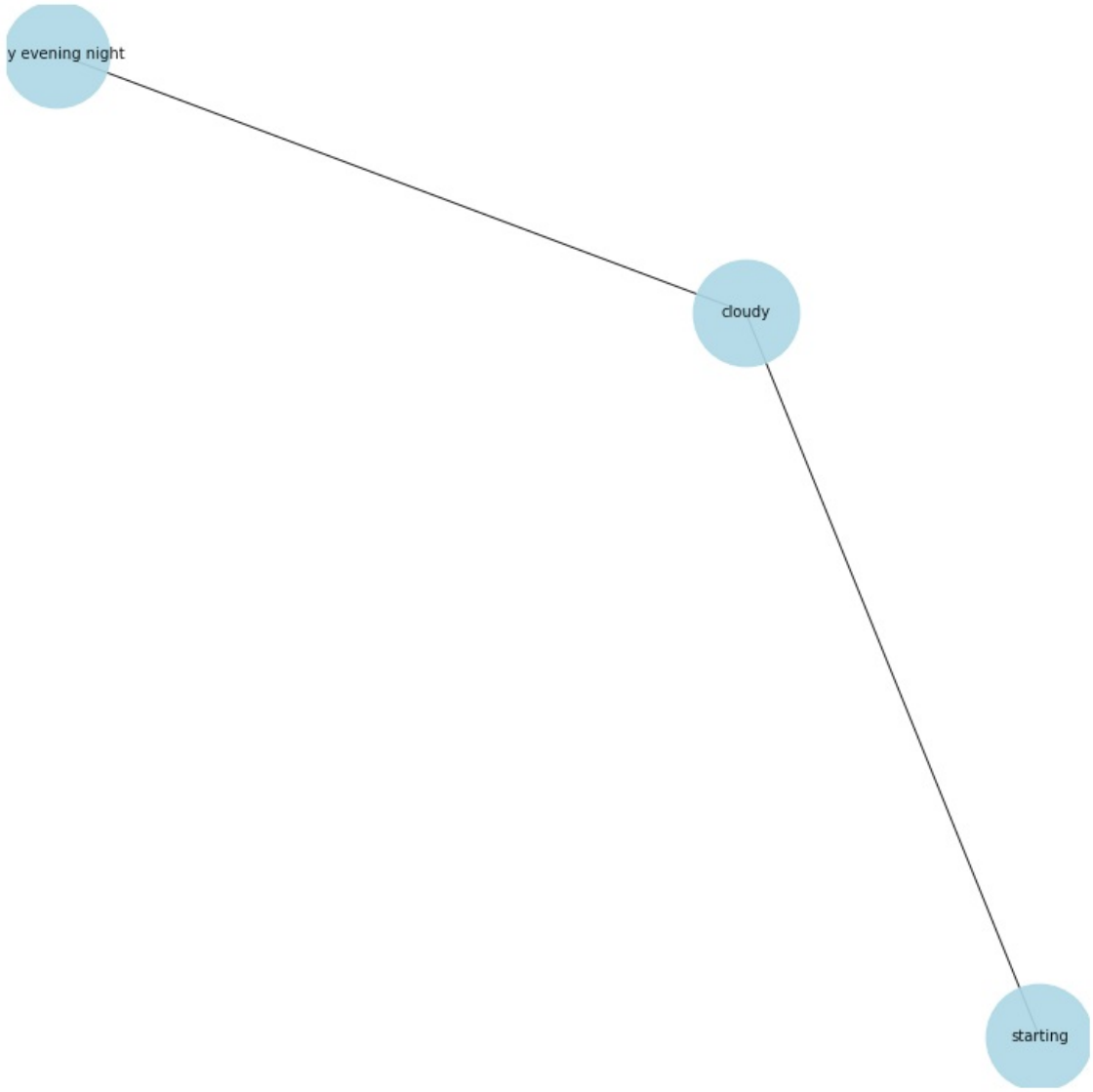



```
Mostly cloudy until night and breezy overnight.
Mostly -> advmod
cloudy -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
overnight -> advmod
. -> punct
 , cloudy , night
```

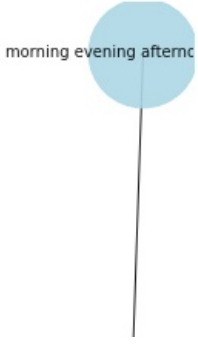



```
Mostly cloudy throughout the day and breezy starting in the evening continuing until night.
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
breezy -> conj
starting -> nsubj
in -> prep
the -> det
evening -> pobj
continuing -> advcl
until -> prep
night -> pobj
```

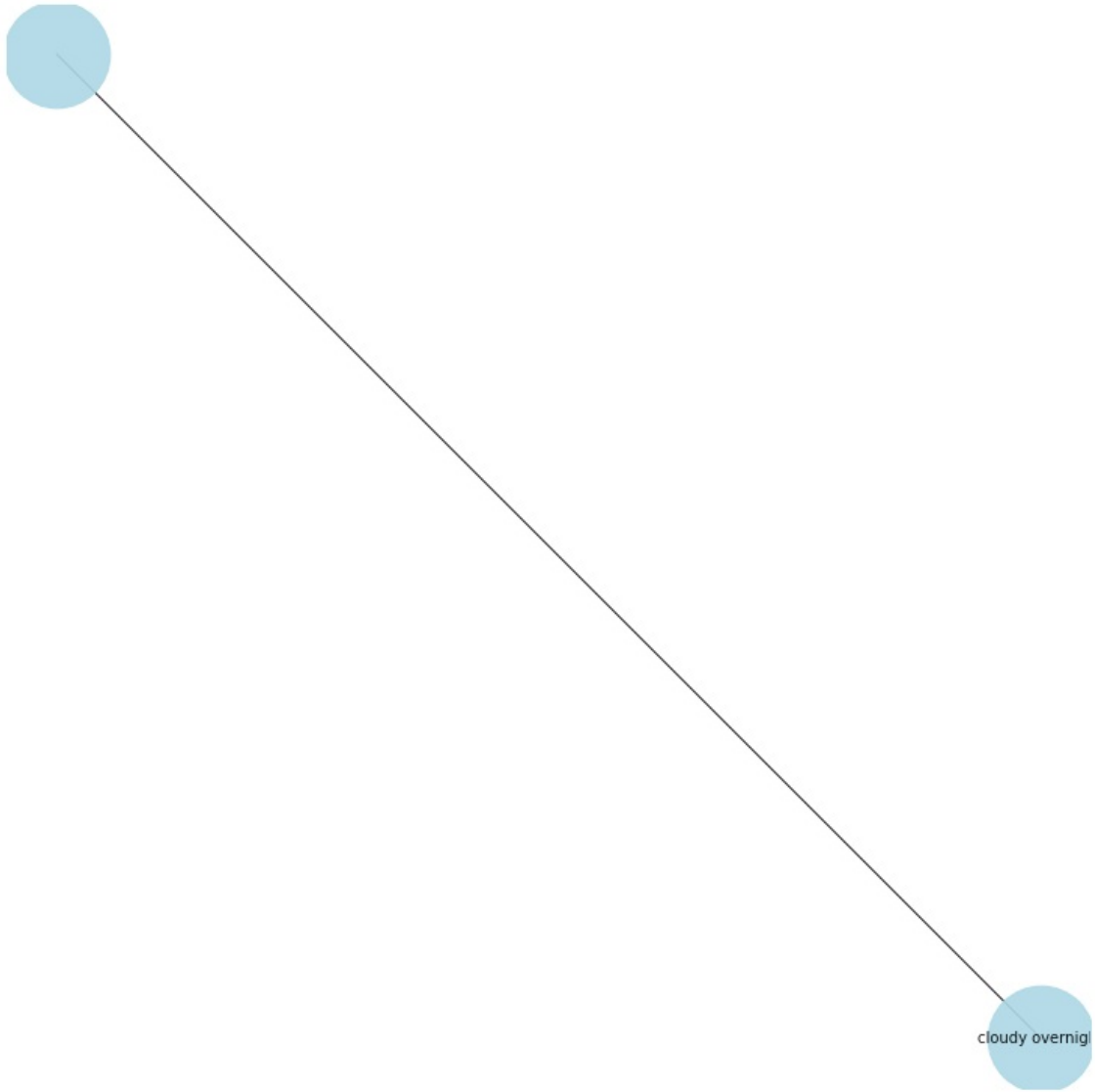
. -> punct
starting , cloudy , day evening night



Partly cloudy starting in the morning continuing until evening and breezy starting in the afternoon continuing un
til evening.
Partly -> advmod
cloudy -> amod
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
afternoon -> pobj
continuing -> xcomp
until -> prep
evening -> pobj
. -> punct
starting , cloudy continue , morning evening afternoon evening

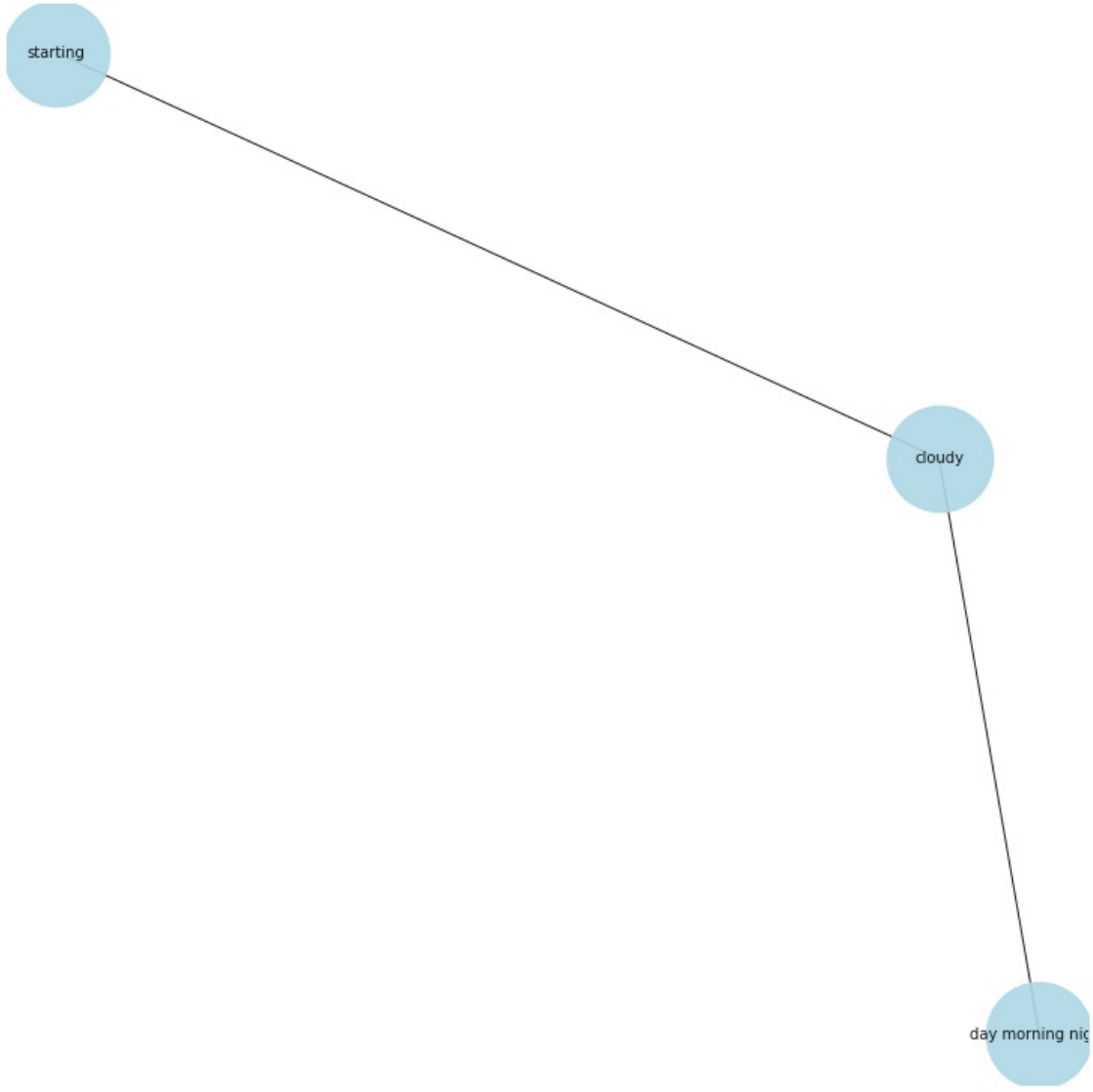



```
Mostly cloudy overnight.
Mostly -> advmod
cloudy -> amod
overnight -> ROOT
. -> punct
 , cloudy overnight ,
```

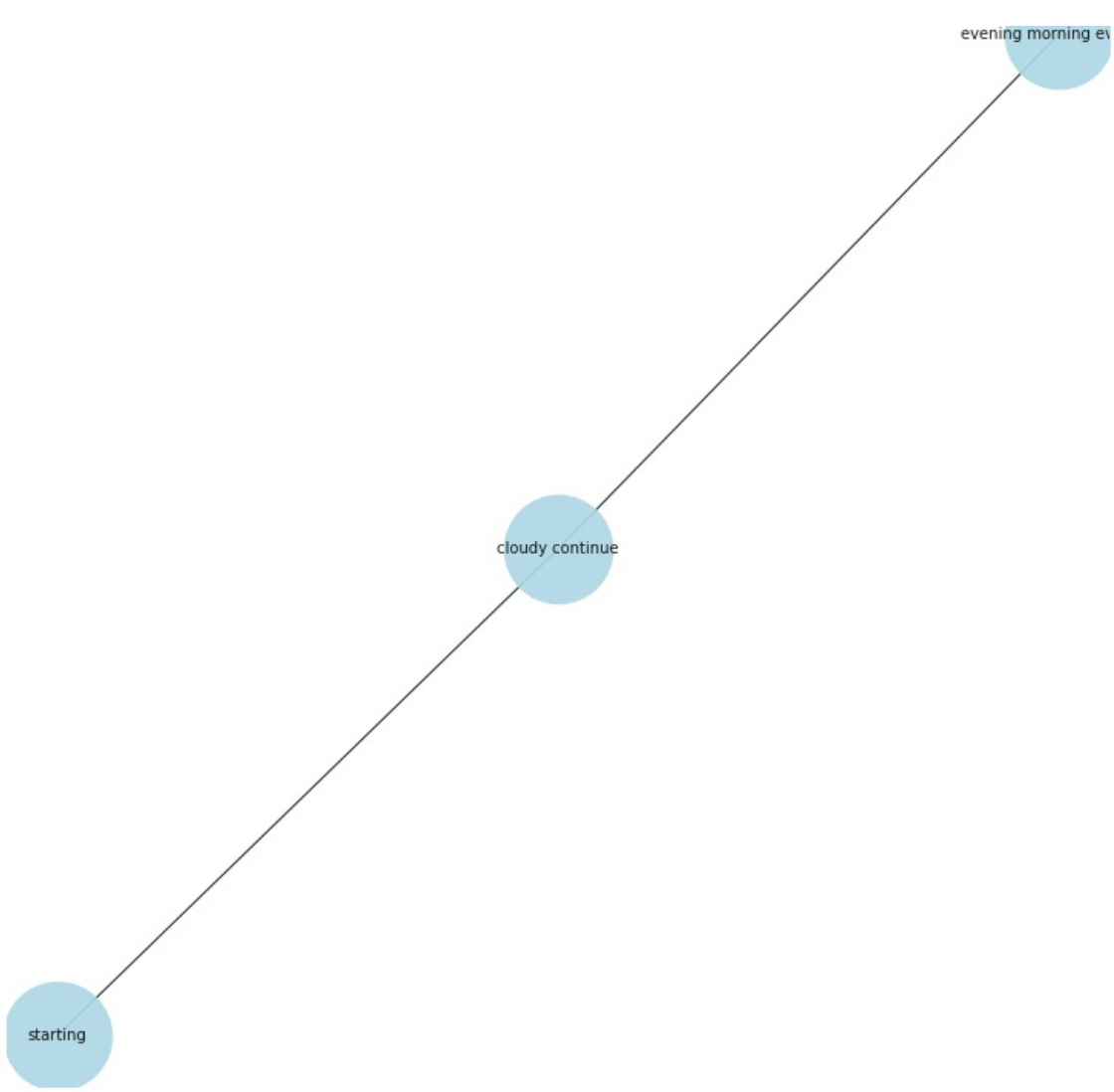


Mostly cloudy throughout the day and windy starting in the morning continuing until night.

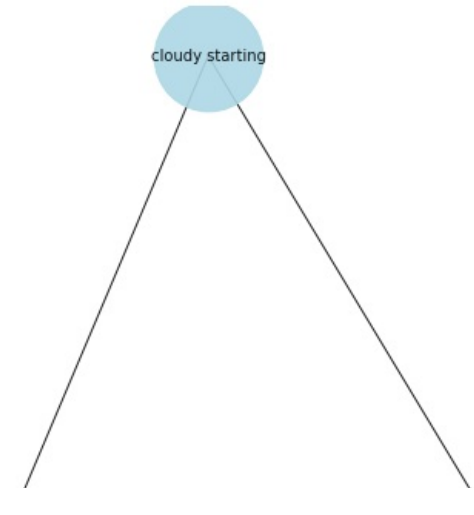
```
Mostly -> advmod
cloudy -> ROOT
throughout -> prep
the -> det
day -> pobj
and -> cc
windy -> conj
starting -> nsubj
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
night -> pobj
. -> punct
starting , cloudy , day morning night
```

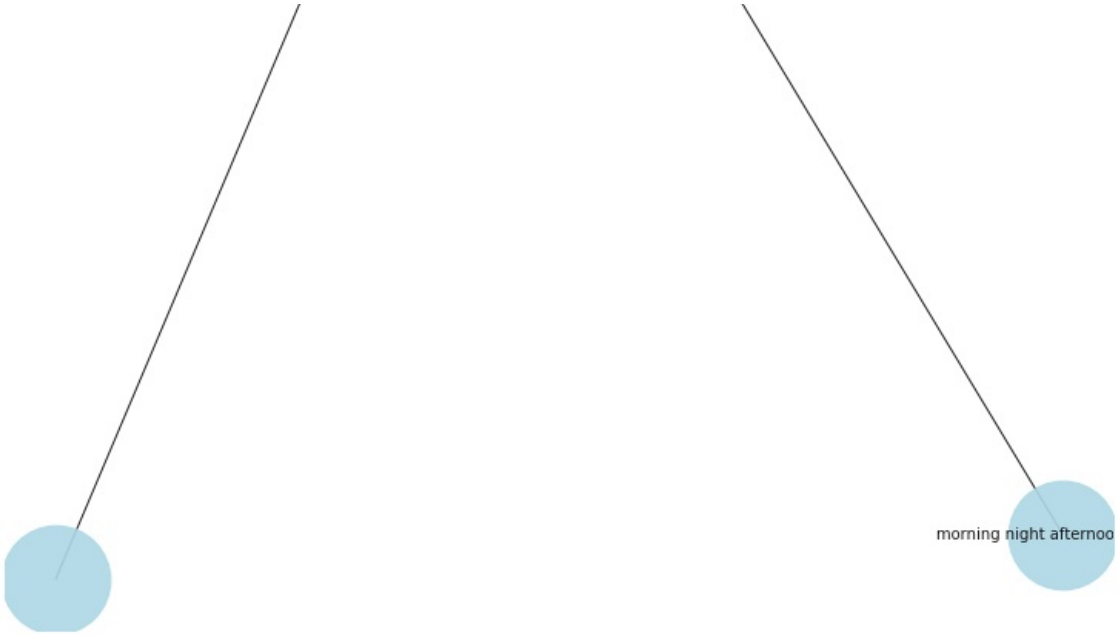



```
Partly cloudy starting overnight continuing until evening and windy starting in the morning continuing until even
ing.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
evening -> pobj
and -> cc
windy -> conj
starting -> advcl
in -> prep
the -> det
morning -> pobj
continuing -> advcl
until -> prep
evening -> pobj
. -> punct
starting , cloudy continue , evening morning evening
```

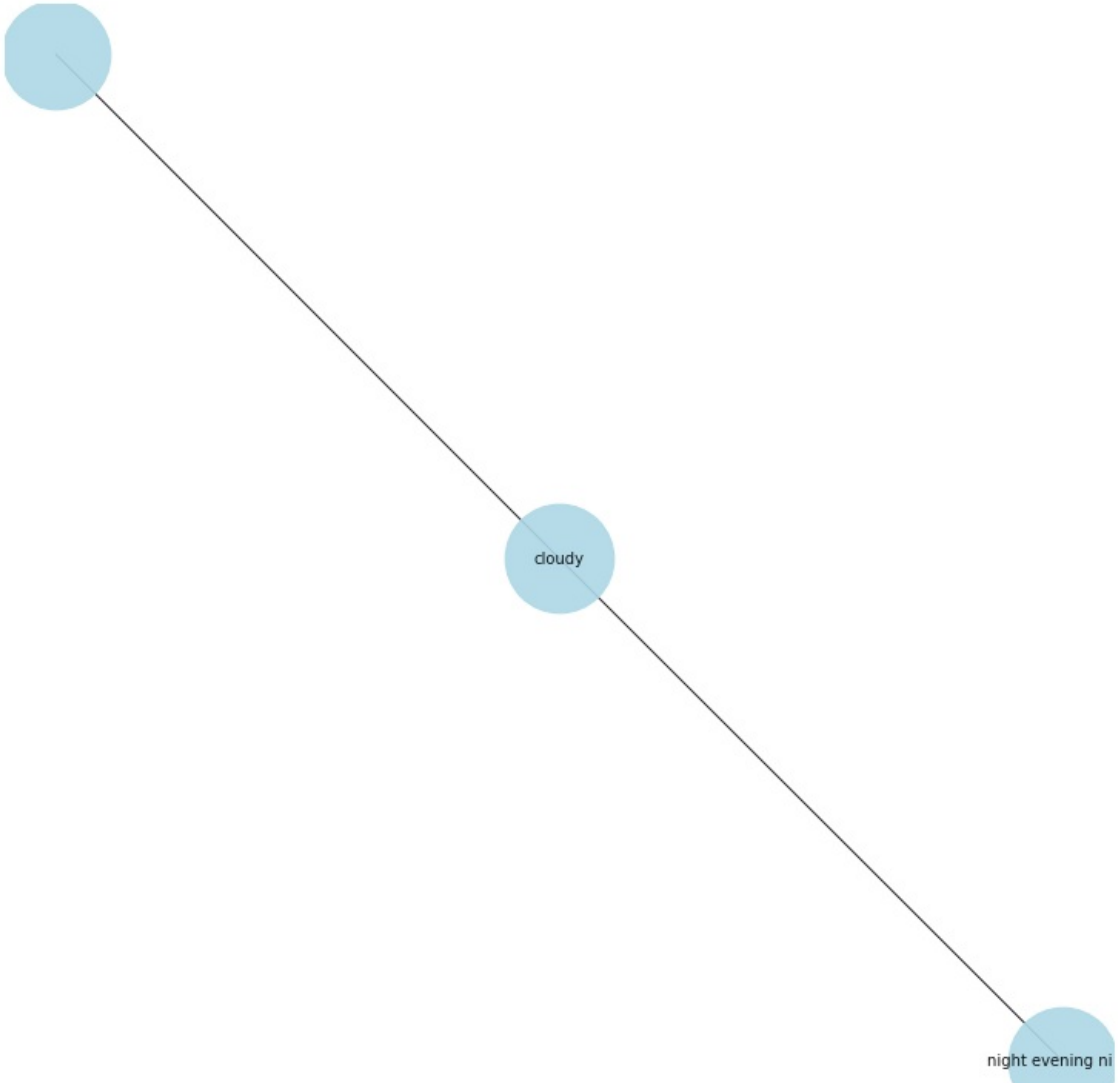



```
Partly cloudy starting in the morning continuing until night and breezy starting in the afternoon continuing unti
l evening.
Partly -> advmod
cloudy -> amod
starting -> ROOT
in -> prep
the -> det
morning -> pobj
continuing -> xcomp
until -> prep
night -> pobj
and -> cc
breezy -> conj
starting -> xcomp
in -> prep
the -> det
afternoon -> pobj
continuing -> xcomp
until -> prep
evening -> pobj
. -> punct
 , cloudy starting , morning night afternoon evening
```

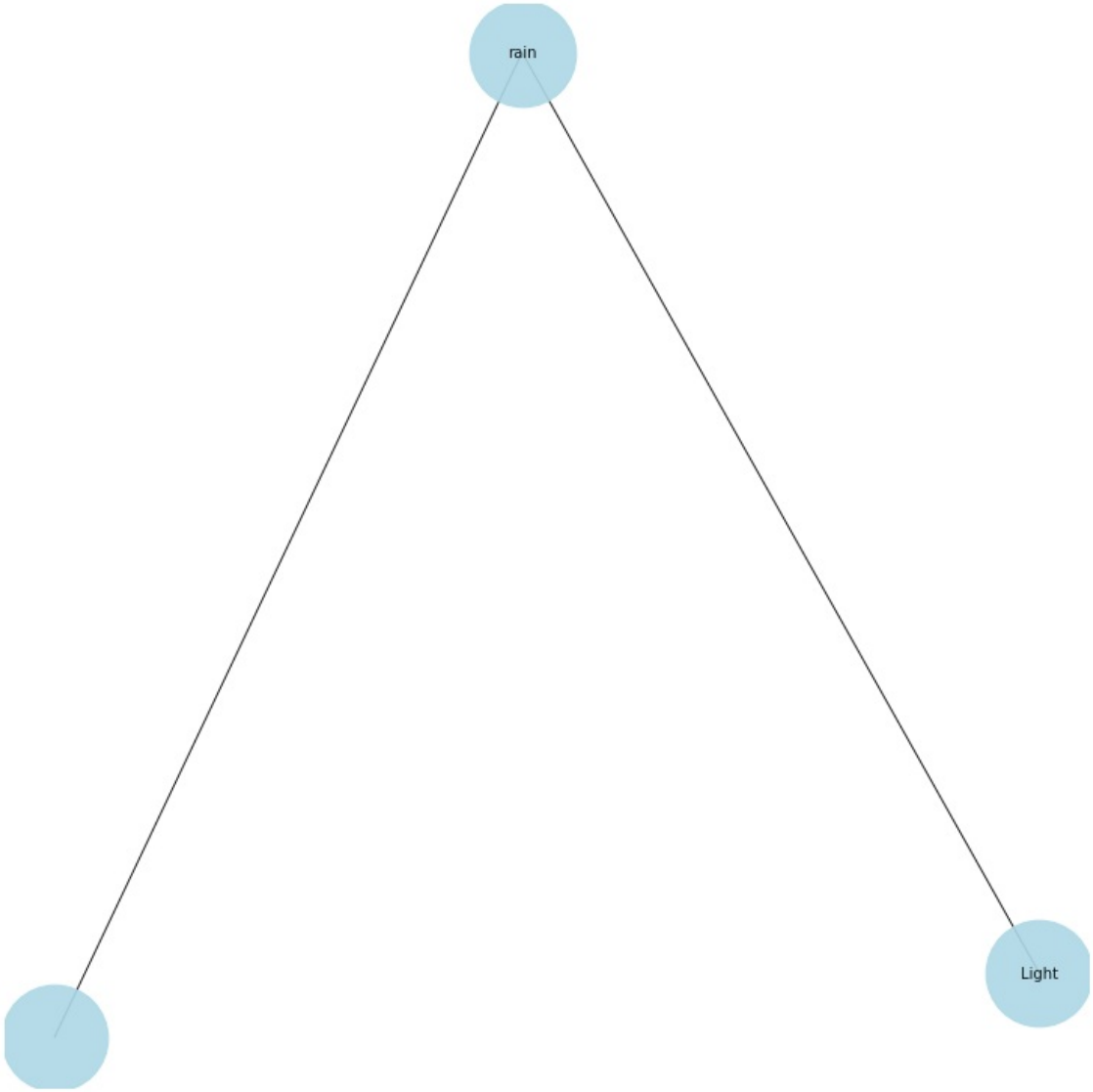




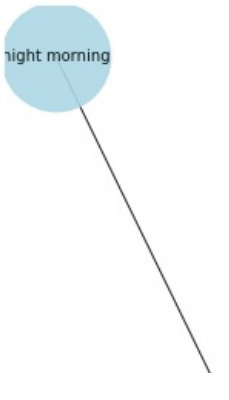

```
Mostly cloudy until night and breezy starting in the evening continuing until night.
Mostly -> advmod
cloudy -> ROOT
until -> mark
night -> pobj
and -> cc
breezy -> conj
starting -> advcl
in -> prep
the -> det
evening -> pobj
continuing -> advcl
until -> prep
night -> pobj
. -> punct
 , cloudy , night evening night
```

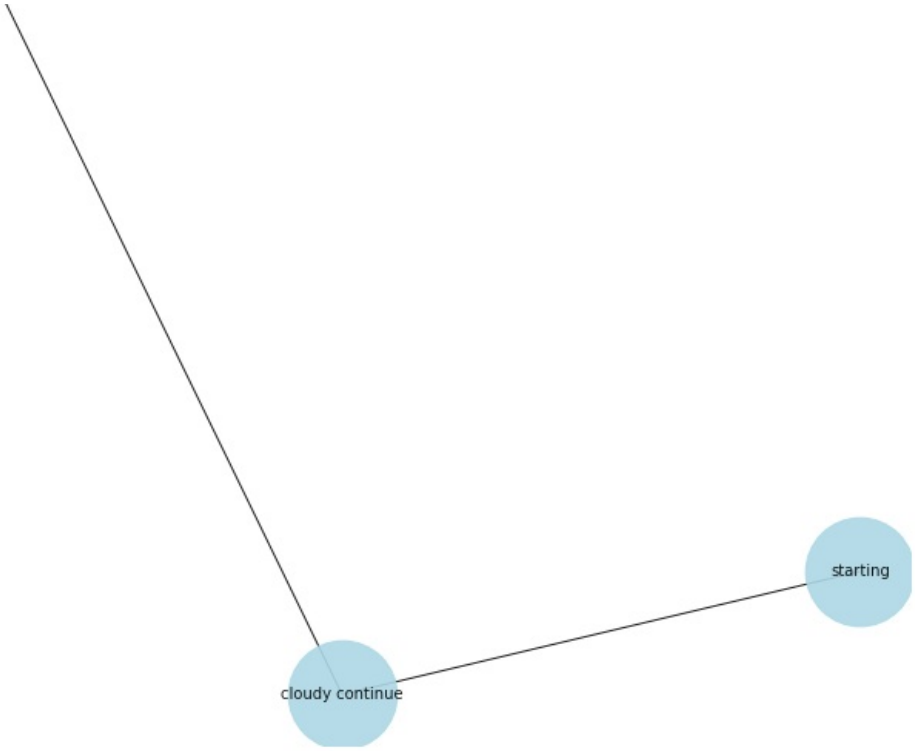


```
Light rain overnight.
Light -> nsubj
rain -> ROOT
overnight -> advmod
. -> punct
Light , rain ,
```

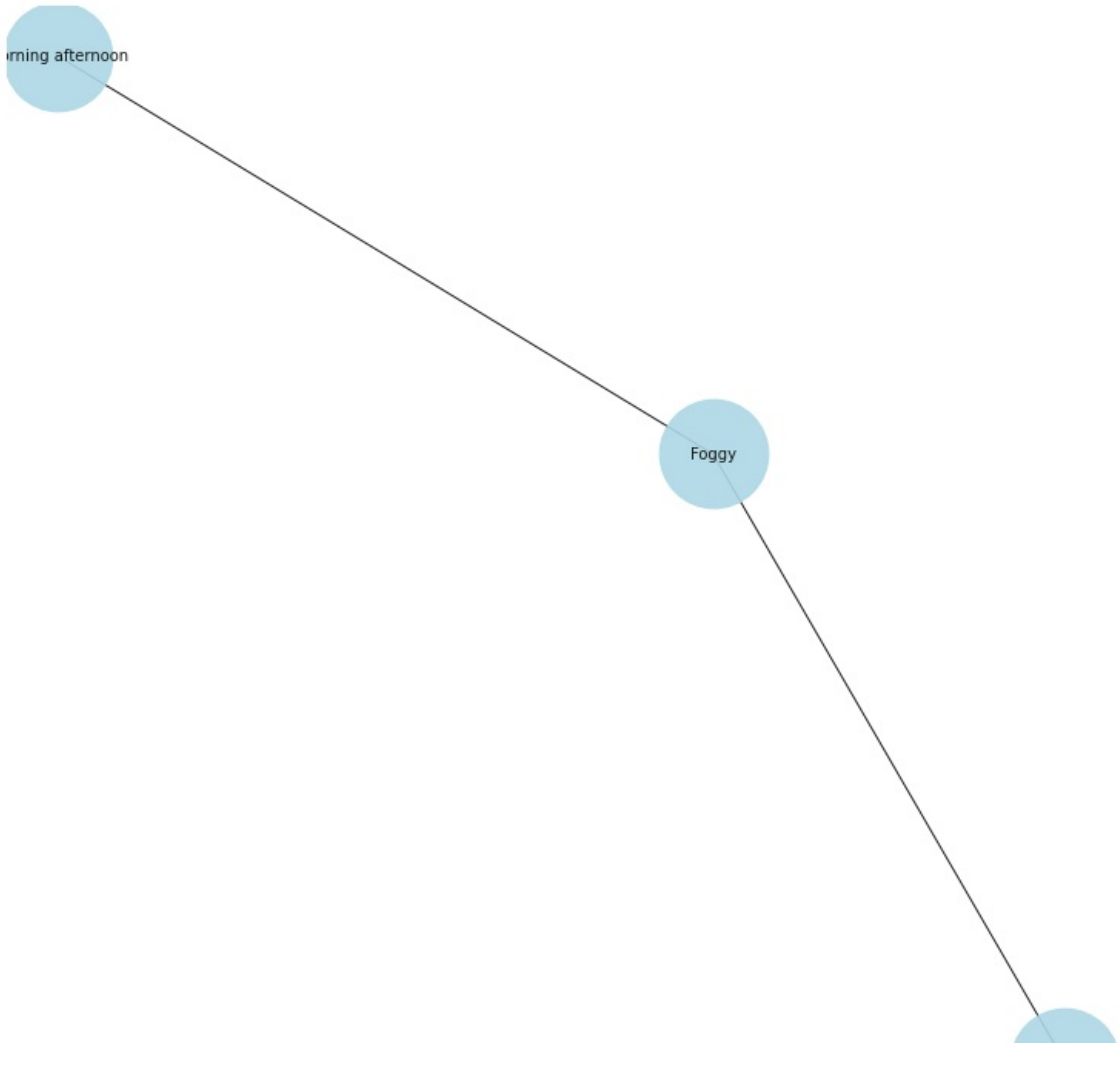



```
Partly cloudy starting overnight continuing until night and breezy in the morning.
Partly -> advmod
cloudy -> amod
starting -> nsubj
overnight -> advmod
continuing -> ROOT
until -> prep
night -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
morning -> pobj
. -> punct
starting , cloudy continue , night morning
```






```
Foggy starting overnight continuing until morning and breezy in the afternoon.
Foggy -> ROOT
starting -> acl
overnight -> advmod
continuing -> xcomp
until -> prep
morning -> pobj
and -> cc
breezy -> conj
in -> prep
the -> det
afternoon -> pobj
. -> punct
 , Foggy , morning afternoon
```



Data Modelling

We have to come conclusion the Data Can be modelled in Two Different Ways:

1. *One is as Time Series Problem*
2. *Two as Classical Machine Learning Problem*

Following is the Pre-Processing For Data Which Has Been Modelled as Time Series

3. Data Pre-processing and cleaning

a. Do the appropriate preprocessing of the data like identifying NULL or Missing Values if any, handling of outliers if present in the dataset, skewed data etc. Apply appropriate feature engineering techniques for them.

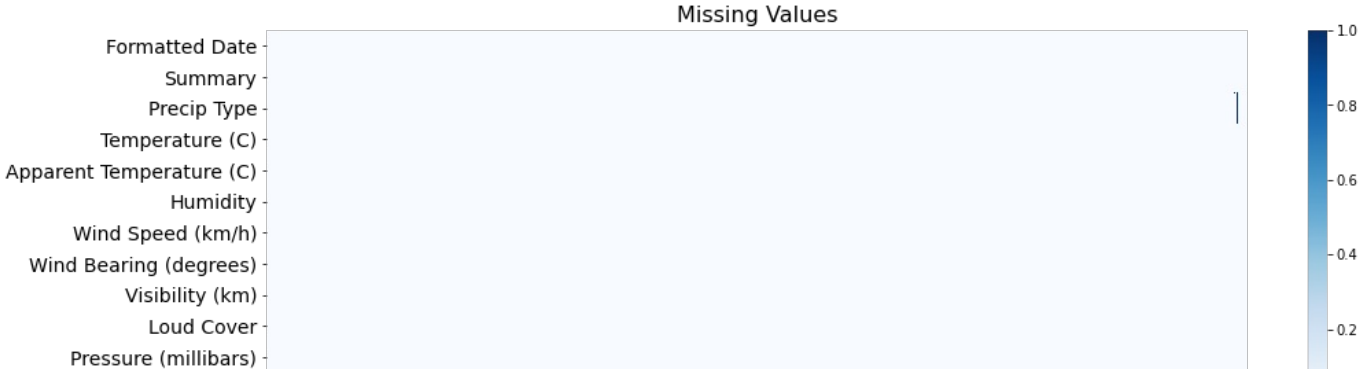
b. Apply the feature transformation techniques like Standardization, Normalization, etc. You are free to apply the appropriate transformations depending upon the structure and the complexity of your dataset.

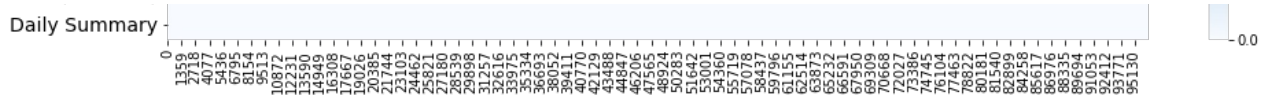
c. Do the correlational analysis on the dataset. Provide a visualization for the same.

In [47]:
```
data.isnull().sum()
```

Out[47]:
```
Formatted Date               0
Summary                      0
Precip Type                517
Temperature (C)              0
Apparent Temperature (C)     0
Humidity                     0
Wind Speed (km/h)            0
Wind Bearing (degrees)       0
Visibility (km)              0
Loud Cover                   0
Pressure (millibars)         0
Daily Summary                0
dtype: int64
```

In [56]:
```
f, ax = plt.subplots(nrows=1, ncols=1, figsize=(16,5))

sns.heatmap(data.T.isna(), cmap='Blues')
ax.set_title('Missing Values', fontsize=16)

for tick in ax.yaxis.get_major_ticks():
    tick.label.set_fontsize(14)
plt.show()
```





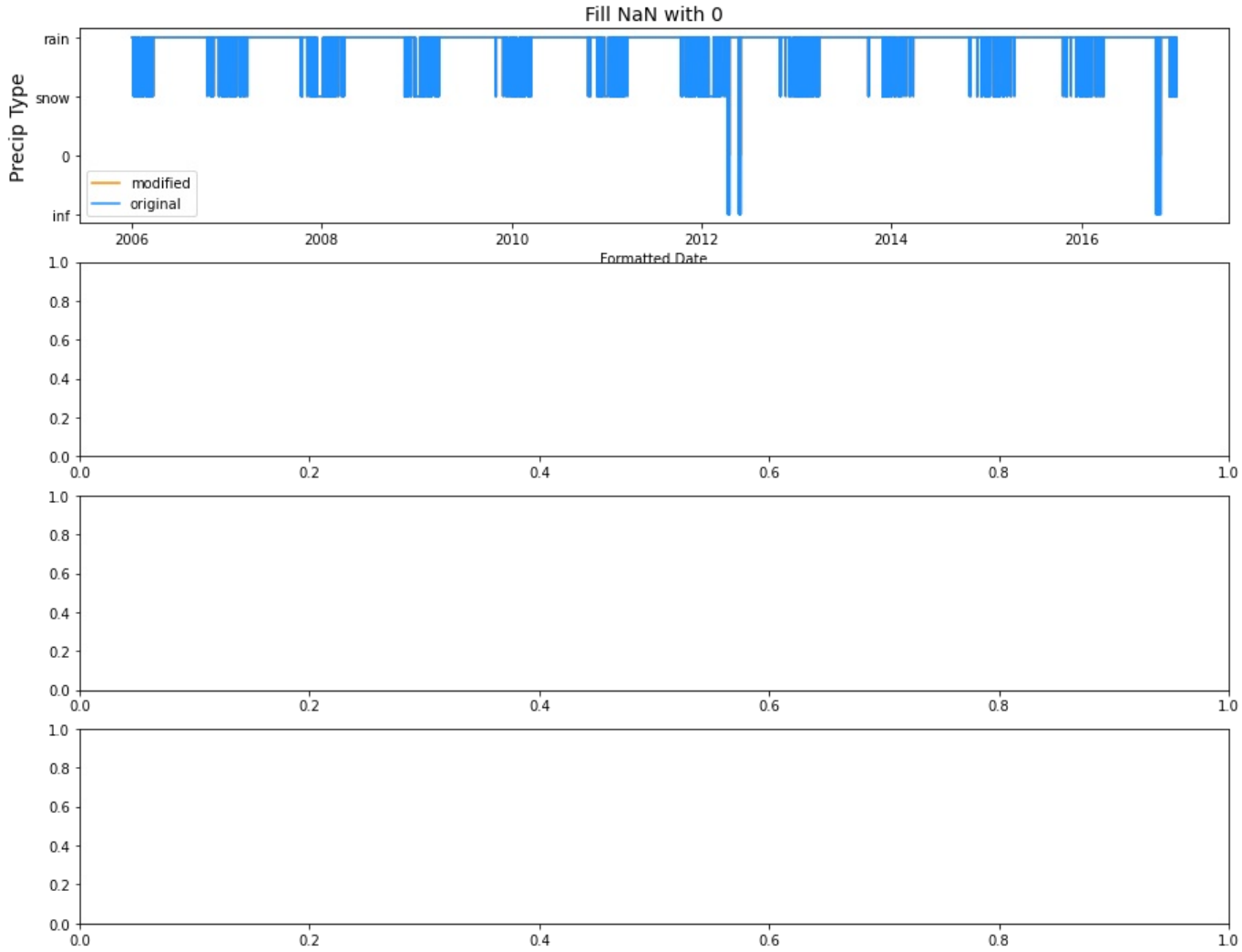
In [62]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96453 entries, 0 to 96452
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Formatted Date            96453 non-null  object 
 1   Summary                   96453 non-null  object 
 2   Precip Type               95936 non-null  object 
 3   Temperature (C)           96453 non-null  float64
 4   Apparent Temperature (C)  96453 non-null  float64
 5   Humidity                  96453 non-null  float64
 6   Wind Speed (km/h)         96453 non-null  float64
 7   Wind Bearing (degrees)    96453 non-null  float64
 8   Visibility (km)           96453 non-null  float64
 9   Loud Cover                96453 non-null  float64
 10  Pressure (millibars)      96453 non-null  float64
 11  Daily Summary             96453 non-null  object 
dtypes: float64(8), object(4)
memory usage: 8.8+ MB
```

In [68]:
```
# f, ax = plt.subplots(nrows=4, ncols=1, figsize=(15, 12))
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna('0'), ax=ax[0], color='darkorange', label =
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna(str(np.inf)), ax=ax[0], color='dodgerblue',
# ax[0].set_title('Fill NaN with 0', fontsize=14)
# ax[0].set_ylabel(ylabel='Precip Type', fontsize=14)
# mean_drainage = data['Precip Type'].mean()
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna(str(mean_drainage)), ax=ax[1], color='darko
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna(str(np.inf)), ax=ax[1], color='dodgerblue',
# ax[1].set_title(f'Fill NaN with Mean Value ({mean_drainage:.0f})', fontsize=14)
# ax[1].set_ylabel(ylabel='Precip Type', fontsize=14)
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].ffill(), ax=ax[2], color='darkorange', label = 'mo
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna(str(np.inf)), ax=ax[2], color='dodgerblue',
# ax[2].set_title(f'FFill', fontsize=14)
# ax[2].set_ylabel(ylabel='Precip Type', fontsize=14)
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].interpolate(), ax=ax[3], color='darkorange', label
# sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna(str(np.inf)), ax=ax[3], color='dodgerblue',
# ax[3].set_title(f'Interpolate', fontsize=14)
# ax[3].set_ylabel(ylabel='Precip Type', fontsize=14)
# plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-68-441d9fc8e6e9> in <module>
      4 ax[0].set_title('Fill NaN with 0', fontsize=14)
      5 ax[0].set_ylabel(ylabel='Precip Type', fontsize=14)
----> 6 mean_drainage = data['Precip Type'].mean()
      7 sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna(str(mean_drainage)), ax=ax[1], color=
'darkorange', label = 'modified')
      8 sns.lineplot(x=data['Formatted Date'], y=data['Precip Type'].fillna(str(np.inf)), ax=ax[1], color='dodger
blue', label = 'original')

~\AppData\Roaming\Python\Python38\site-packages\pandas\core\generic.py in stat_func(self, axis, skipna, level, nu
meric_only, **kwargs)
  11212         if level is not None:
  11213             return self._agg_by_level(name, axis=axis, level=level, skipna=skipna)
> 11214         return self._reduce(
  11215             f, name, axis=axis, skipna=skipna, numeric_only=numeric_only
  11216         )

~\AppData\Roaming\Python\Python38\site-packages\pandas\core\series.py in _reduce(self, op, name, axis, skipna, nu
meric_only, filter_type, **kwds)
   3889                 )
   3890             with np.errstate(all="ignore"):
-> 3891                 return op(delegate, skipna=skipna, **kwds)
   3892
   3893         # TODO(EA) dispatch to Index

~\AppData\Roaming\Python\Python38\site-packages\pandas\core\nanops.py in _f(*args, **kwargs)
     67             try:
     68                 with np.errstate(invalid="ignore"):
---> 69                     return f(*args, **kwargs)
     70             except ValueError as e:
     71                 # we want to transform an object array
```

```
~\AppData\Roaming\Python\Python38\site-packages\pandas\core\nanops.py in f(values, axis, skipna, **kwds)
    123                     result = alt(values, axis=axis, skipna=skipna, **kwds)
    124             else:
--> 125                 result = alt(values, axis=axis, skipna=skipna, **kwds)
    126
    127             return result

~\AppData\Roaming\Python\Python38\site-packages\pandas\core\nanops.py in nanmean(values, axis, skipna, mask)
    540         dtype_count = dtype
    541     count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 542     the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    543
    544     if axis is not None and getattr(the_sum, "ndim", False):

~\anaconda3\lib\site-packages\numpy\core\_methods.py in _sum(a, axis, dtype, out, keepdims, initial, where)
     45 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
     46          initial=_NoValue, where=True):
---> 47     return umr_sum(a, axis, dtype, out, keepdims, initial, where)
     48
     49 def _prod(a, axis=None, dtype=None, out=None, keepdims=False,

TypeError: can only concatenate str (not "int") to str
```

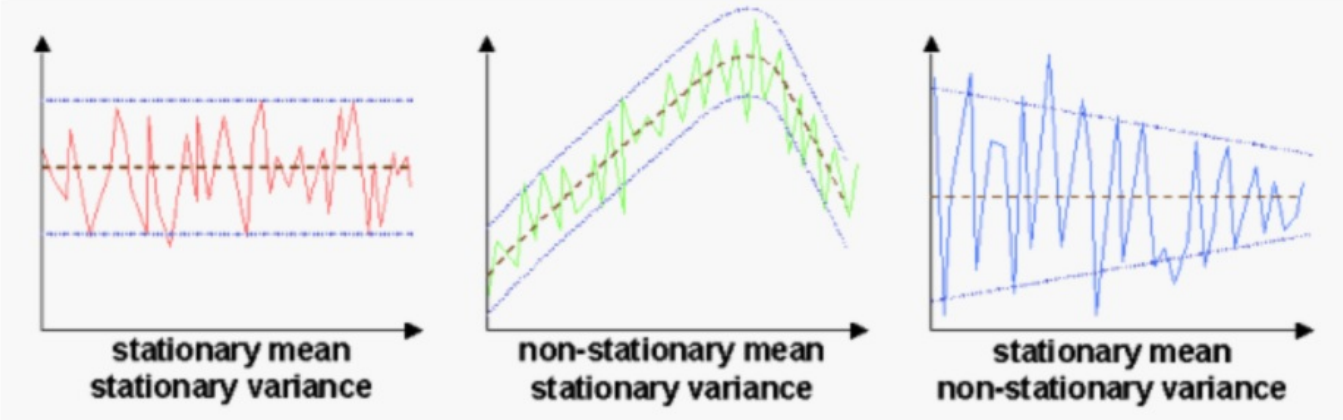


Since we are working with Time Series Analysis, We will have to drop all information related to Categorical Variable like Summary, Daily Summary, Precip-Type

2.3 Stationarity

Some time-series models, such as such as ARIMA, assume that the underlying data is stationary. Stationarity describes that the time-series has

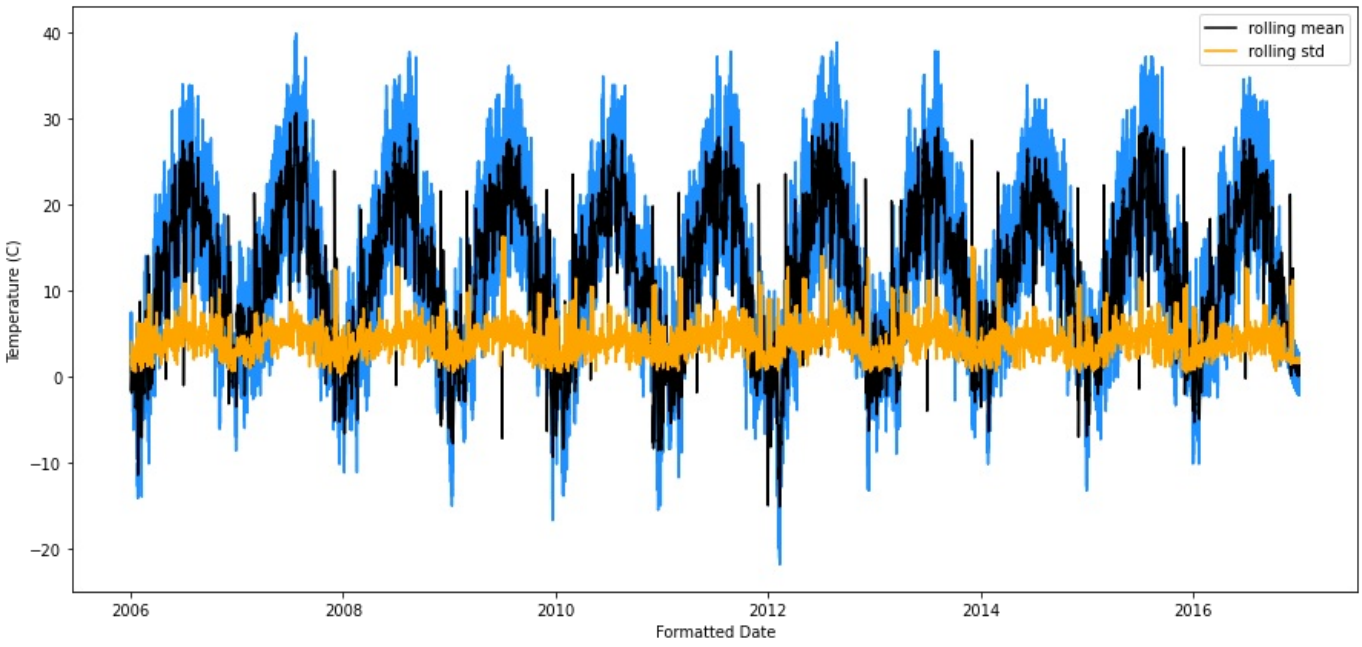
- constant mean and mean is not time-dependent
- constant variance and variance is not time-dependent
- constant covariance and covariance is not time-dependent

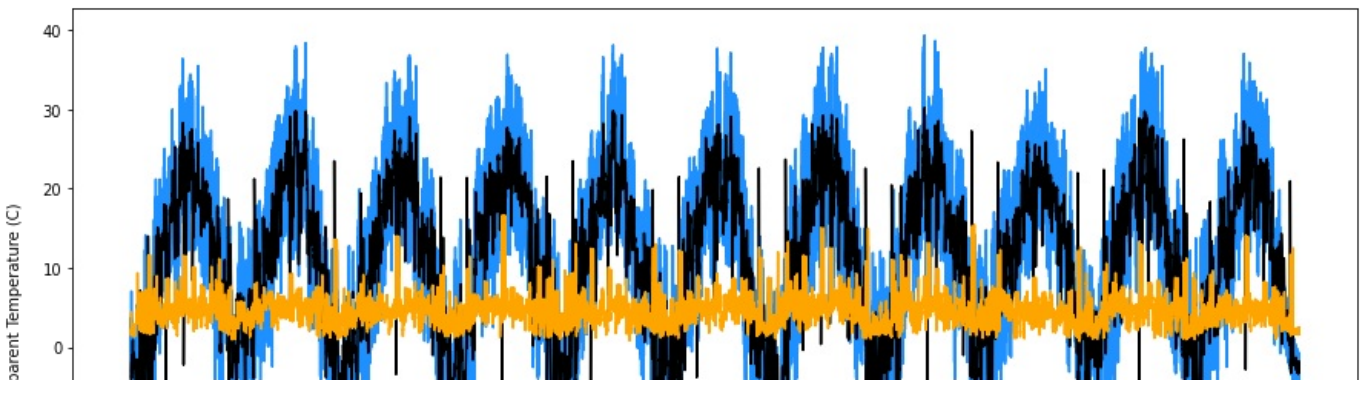


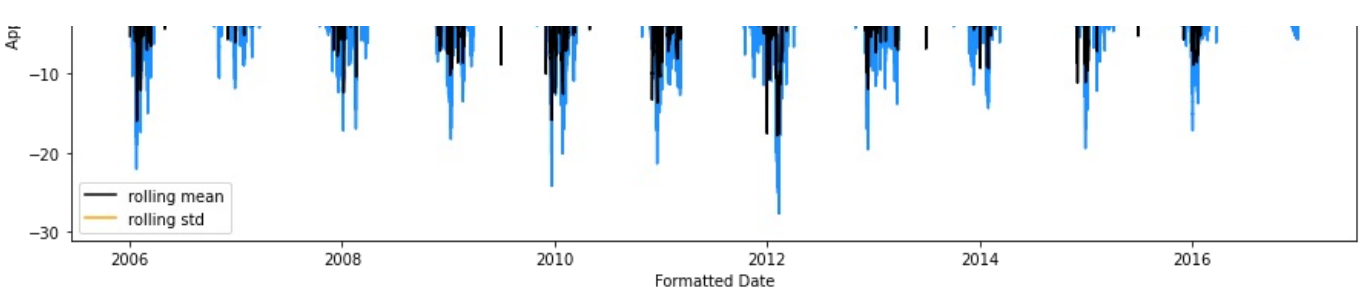
The check for stationarity can be done via three different approaches:

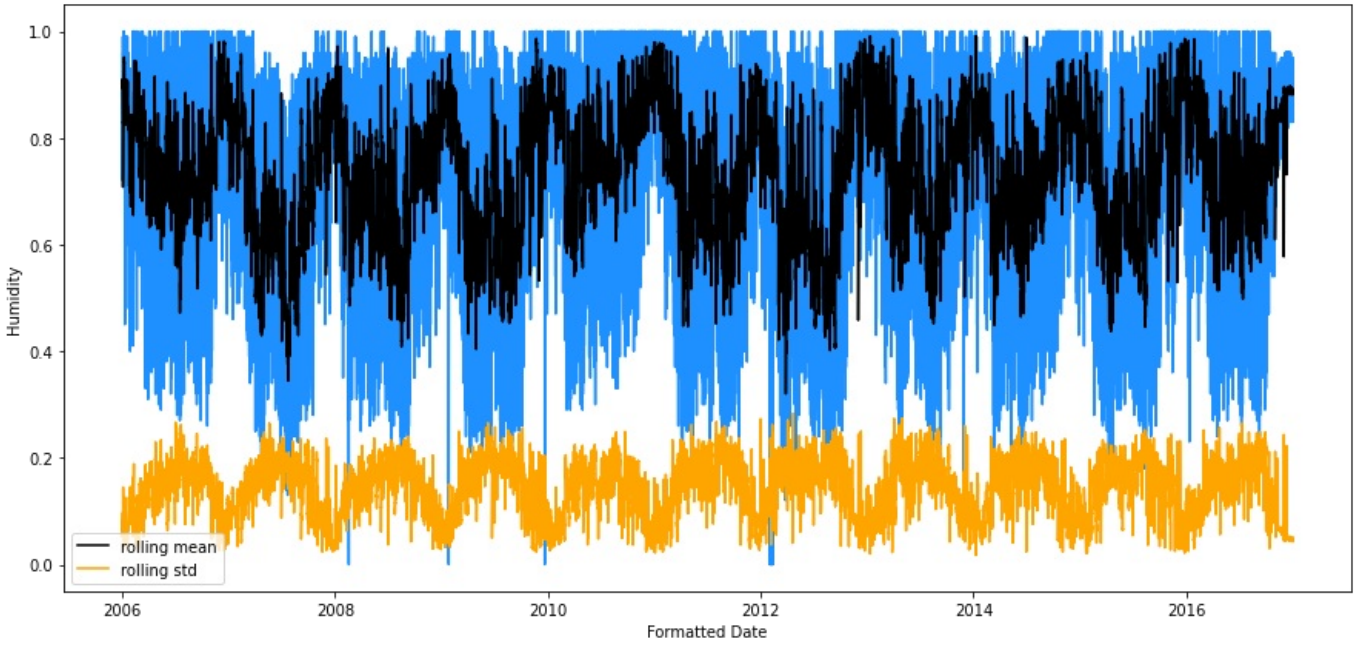
1. **visually**: plot time series and check for trends or seasonality
2. **basic statistics**: split time series and compare the mean and variance of each partition
3. **statistical test**: Augmented Dickey Fuller test

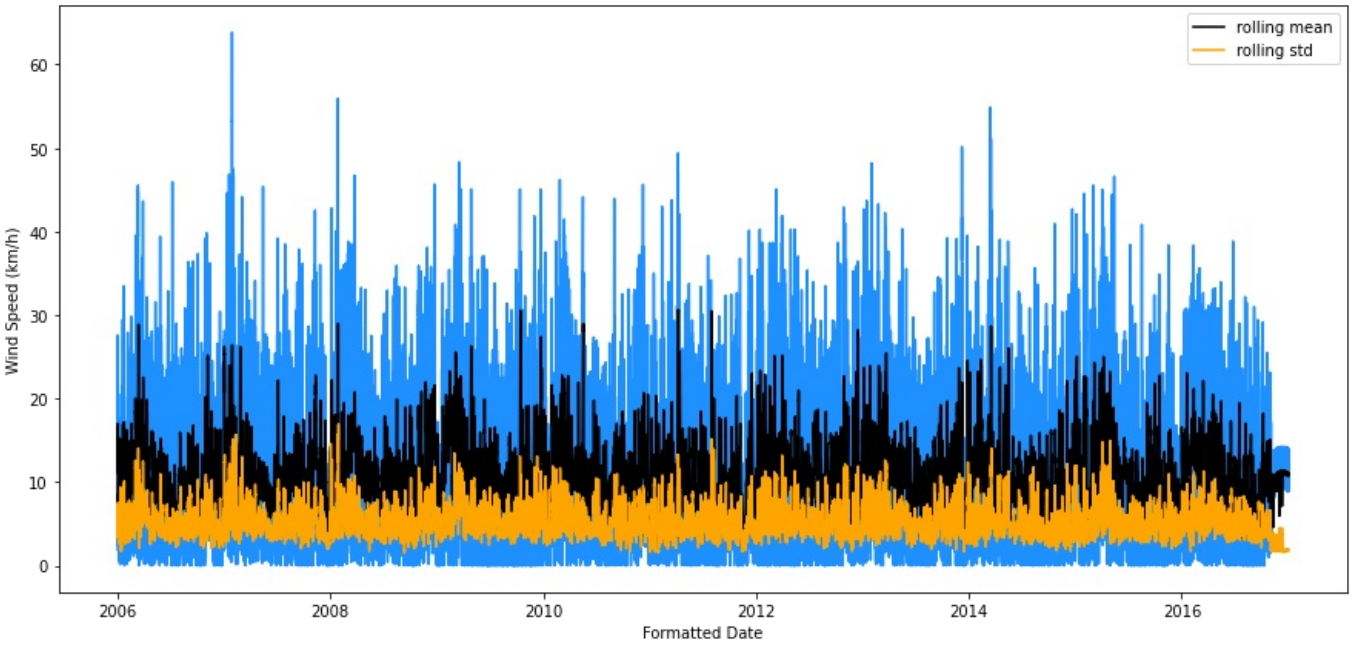
In [75]:
```
# A year has 52 weeks (52 weeks * 7 days per week) aporx.
for col in numerical_feat:
    rolling_window = 52
    plt.figure(figsize=(15, 7))
    sns.lineplot(x=data['Formatted Date'], y=data[col],  color='dodgerblue')
    sns.lineplot(x=data['Formatted Date'], y=data[col].rolling(rolling_window).mean(), color='black', label='roll
    sns.lineplot(x=data['Formatted Date'], y=data[col].rolling(rolling_window).std(),  color='orange', label='rol

#     sns.lineplot(x=df['date'], y=df['temperature'], ax=ax[1], color='dodgerblue')
#     sns.lineplot(x=df['date'], y=df['temperature'].rolling(rolling_window).mean(), ax=ax[1], color='black', lab
#     sns.lineplot(x=df['date'], y=df['temperature'].rolling(rolling_window).std(), ax=ax[1], color='orange', lab
#     ax[1].set_title('Temperature: Non-stationary \nvariance is time-dependent (seasonality)', fontsize=14)
#     ax[1].set_ylabel(ylabel=col, fontsize=14)

    plt.show()
```

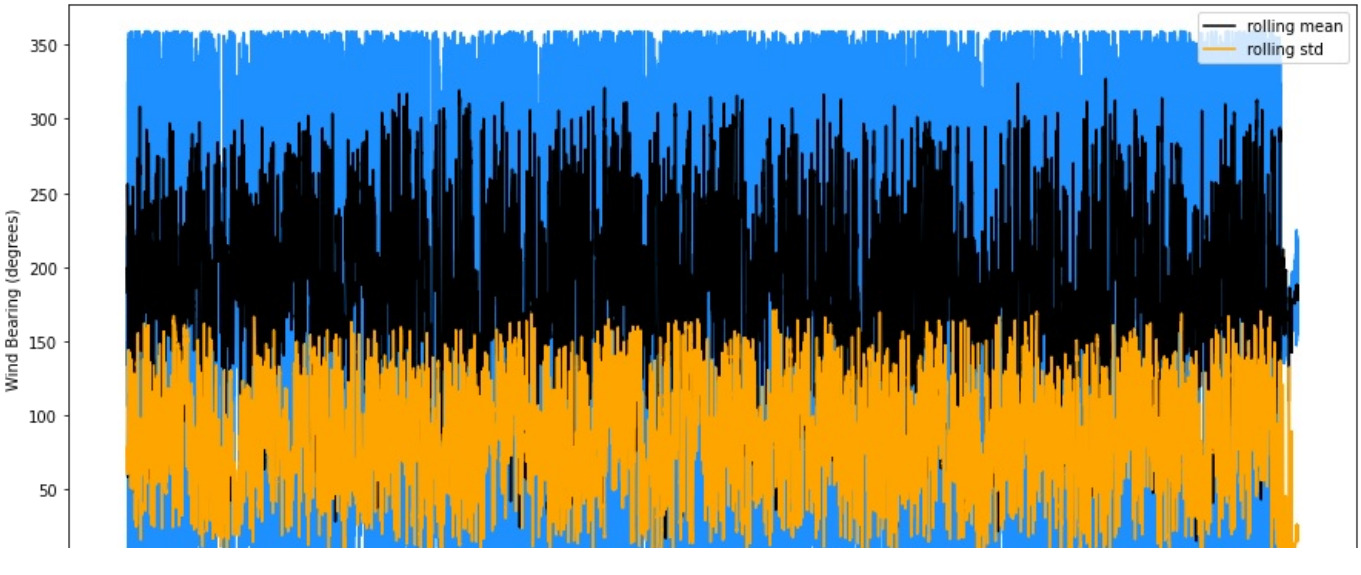


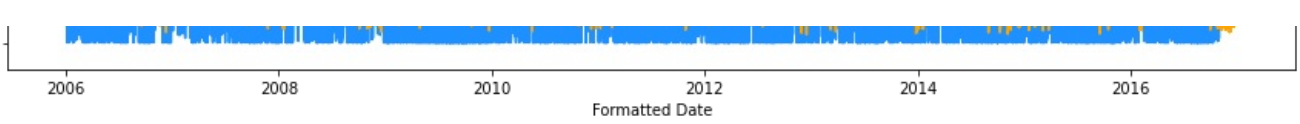


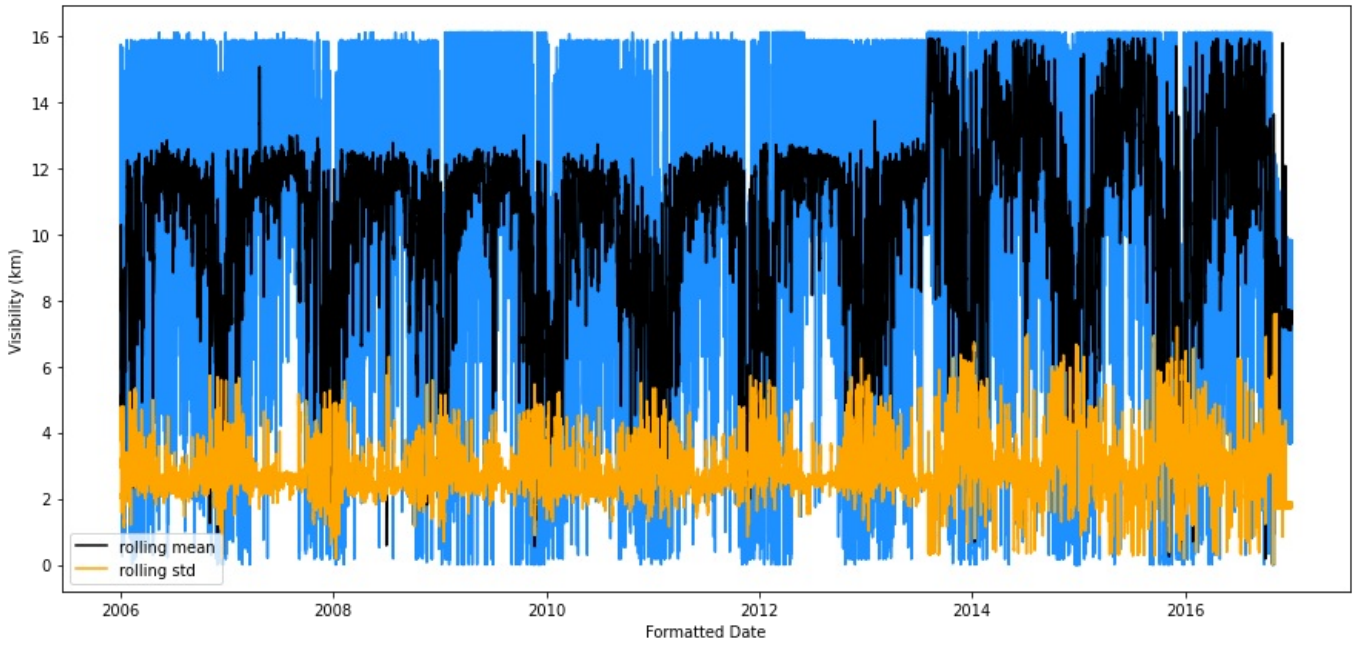

−10
−20
−30
rolling mean
rolling std
2006
2008
2010
2012
2014
2016
Formatted Date

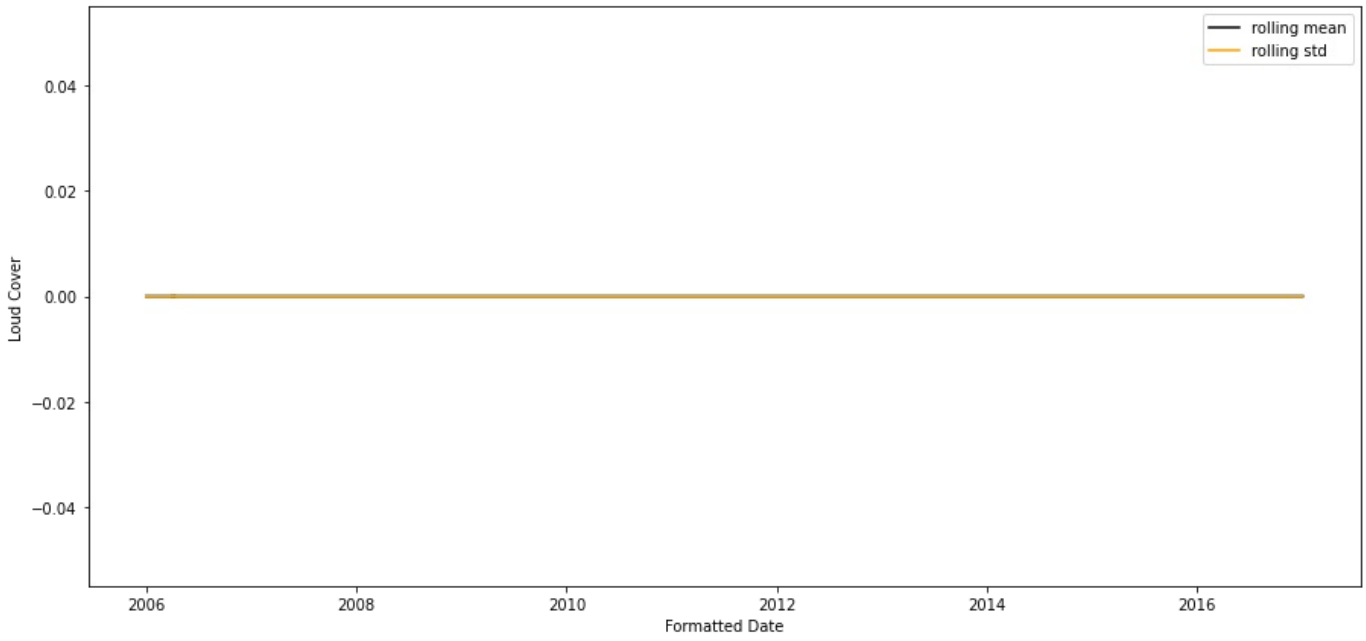

Humidity
1.0
0.8
0.6
0.4
0.2
0.0
rolling mean
rolling std
2006
2008
2010
2012
2014
2016
Formatted Date

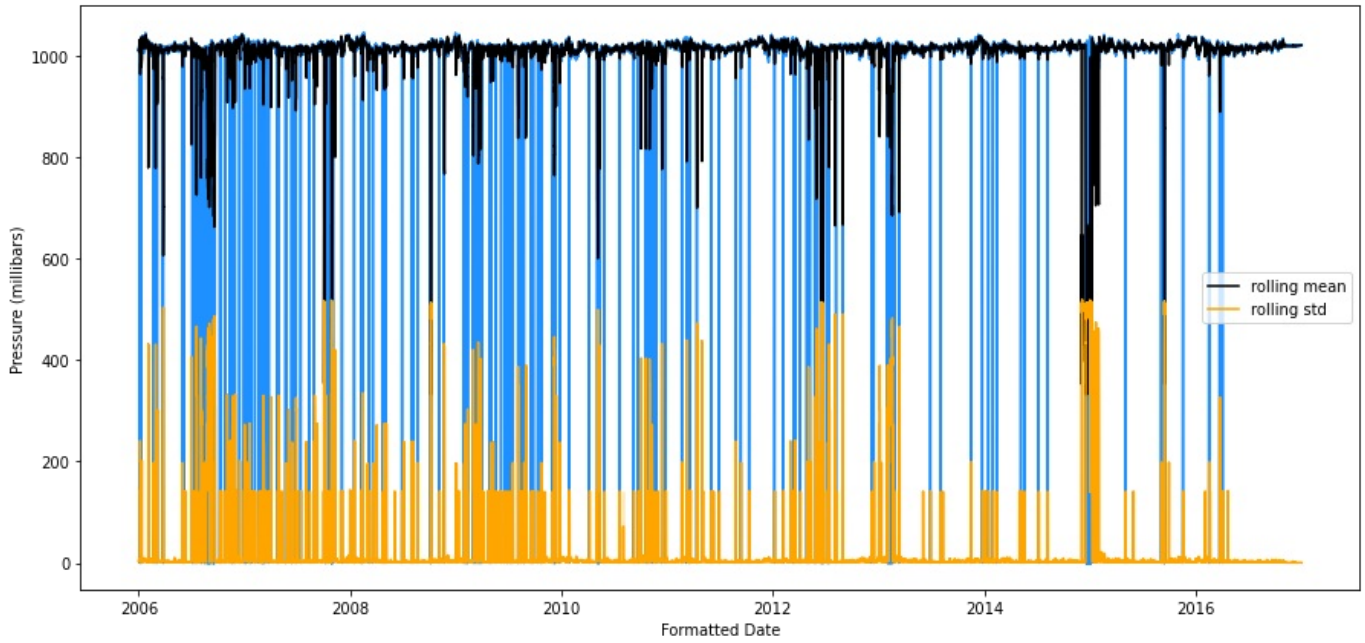

Wind Speed (km/h)
60
50
40
30
20
10
0
rolling mean
rolling std
2006
2008
2010
2012
2014
2016
Formatted Date


Wind Bearing (degrees)
350
300
250
200
150
100
50
rolling mean
rolling std


0
2006
2008
2010
2012
2014
2016
Formatted Date


Visibility (km)
16
14
12
10
8
6
4
2
0
rolling mean
rolling std
2006
2008
2010
2012
2014
2016
Formatted Date


Loud Cover
0.04
0.02
0.00
−0.02
−0.04
rolling mean
rolling std
2006
2008
2010
2012
2014
2016
Formatted Date


Pressure (millibars)
1000
800
600
400
200
0
rolling mean
rolling std
2006
2008
2010
2012
2014
2016
Formatted Date

Unit Root Test

Unit root is a characteristic of a time series that makes it non-stationary. And ADF test belong to the unit root test. Technically , a unit root is said to exist in a time series of value of alpha =1 in below equation.

$$Y_t = \alpha Y_{t-1} + \beta X_e + \epsilon$$

where Yt is value of the time series at time 't' and Xe is an exogenous variable .

The presence of a unit root means the time series is non-stationary.

2.3.1 Augmented Dickey-Fuller (ADF)

Augmented Dickey-Fuller (ADF) test is a type of statistical test called a unit root test. Unit roots are a cause for non-stationarity.

- **Null Hypothesis (H0)**: Time series has a unit root. (Time series is not stationary).
- **Alternate Hypothesis (H1)**: Time series has no unit root (Time series is stationary).

If the null hypothesis can be rejected, we can conclude that the time series is stationary.

There are two ways to rejects the null hypothesis:

On the one hand, the null hypothesis can be rejected if the p-value is below a set significance level. The defaults significance level is 5%

- **p-value > significance level (default: 0.05)**: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.
- **p-value <= significance level (default: 0.05)**: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

On the other hand, the null hypothesis can be rejects if the test statistic is less than the critical value.

- **ADF statistic > critical value**: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.
- **ADF statistic < critical value**: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

Removing The 0 value column, Loud Cover as it contains all values as 0

In [79]:
```
numerical_feat.remove('Loud Cover')
```

In [80]:
```
numerical_feat
```

Out[80]:
```
['Temperature (C)',
 'Apparent Temperature (C)',
 'Humidity',
 'Wind Speed (km/h)',
 'Wind Bearing (degrees)',
 'Visibility (km)',
 'Pressure (millibars)']
```

In [91]:
```
def adfuller_test(series, signif=0.05, name='', verbose=False):
    r = adfuller(series, autolag='AIC')
    output = {'test_statistic':round(r[0], 4), 'pvalue':round(r[1], 4), 'n_lags':round(r[2], 4), 'n_obs':r[3]}
    p_value = output['pvalue']
    def adjust(val, length= 6): return str(val).ljust(length)

    # Print Summary
    print(f'    Augmented Dickey-Fuller Test on "{name}"', "\n   ", '-'*47)
    print(f' Null Hypothesis: Data has unit root. Non-Stationary.')
    print(f' Significance Level    = {signif}')
    print(f' Test Statistic        = {output["test_statistic"]}')
    print(f' No. Lags Chosen       = {output["n_lags"]}')

    for key,val in r[4].items():
        print(f' Critical value {adjust(key)} = {round(val, 3)}')
```

```
    if p_value <= signif:
        print(f" => P-Value = {p_value}. Rejecting Null Hypothesis.")
        print(f" => Series is Stationary.")
    else:
        print(f" => P-Value = {p_value}. Weak evidence to reject the Null Hypothesis.")
        print(f" => Series is Non-Stationary.")
for name, column in data[numerical_feat].iteritems():
    adfuller_test(column, name=column.name)
    print('\n')
```

```
   Augmented Dickey-Fuller Test on "Temperature (C)"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -10.1401
No. Lags Chosen       = 67
Critical value 1%     = -3.43
Critical value 5%     = -2.862
Critical value 10%    = -2.567
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Apparent Temperature (C)"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -10.0578
No. Lags Chosen       = 67
Critical value 1%     = -3.43
Critical value 5%     = -2.862
Critical value 10%    = -2.567
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Humidity"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -15.7959
No. Lags Chosen       = 67
Critical value 1%     = -3.43
Critical value 5%     = -2.862
Critical value 10%    = -2.567
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Wind Speed (km/h)"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -27.9511
No. Lags Chosen       = 67
Critical value 1%     = -3.43
Critical value 5%     = -2.862
Critical value 10%    = -2.567
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Wind Bearing (degrees)"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -33.7924
No. Lags Chosen       = 54
Critical value 1%     = -3.43
Critical value 5%     = -2.862
Critical value 10%    = -2.567
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Visibility (km)"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -18.8941
No. Lags Chosen       = 66
Critical value 1%     = -3.43
Critical value 5%     = -2.862
Critical value 10%    = -2.567
```

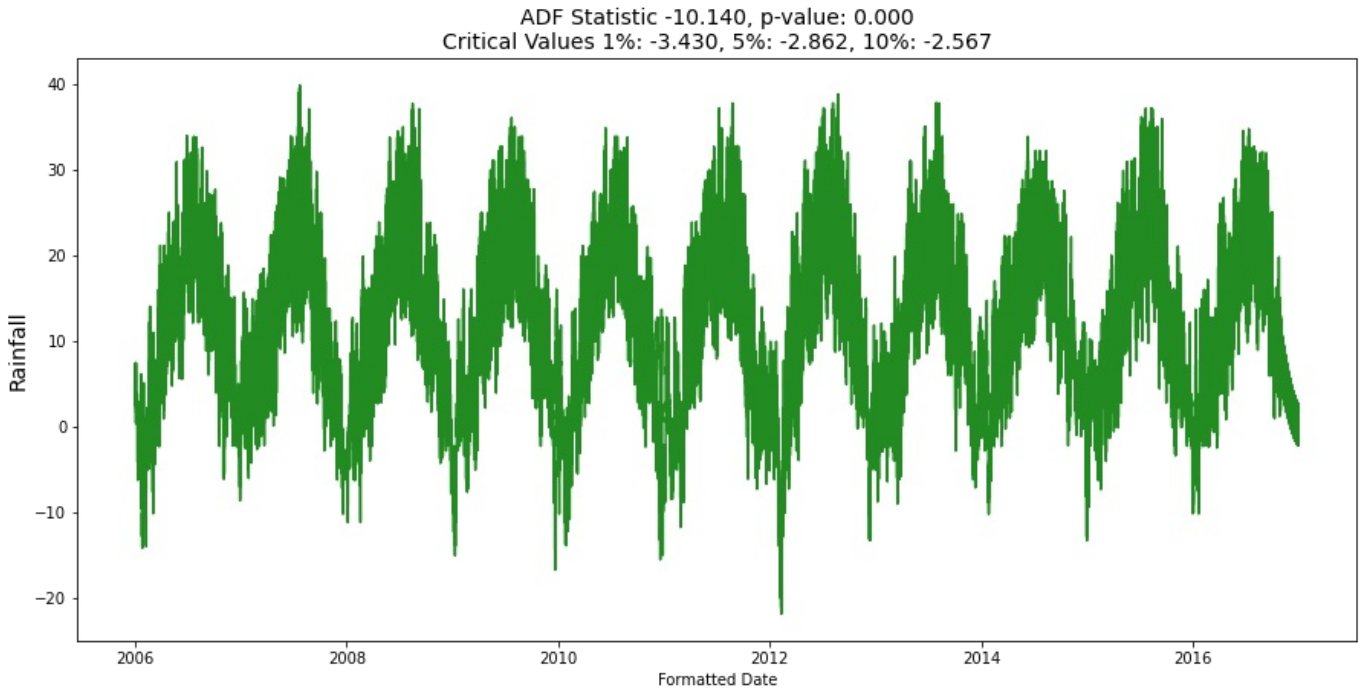
```
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.


   Augmented Dickey-Fuller Test on "Pressure (millibars)"
   -----------------------------------------------
Null Hypothesis: Data has unit root. Non-Stationary.
Significance Level    = 0.05
Test Statistic        = -22.6526
No. Lags Chosen       = 66
Critical value 1%     = -3.43
Critical value 5%     = -2.862
Critical value 10%    = -2.567
=> P-Value = 0.0. Rejecting Null Hypothesis.
=> Series is Stationary.
```

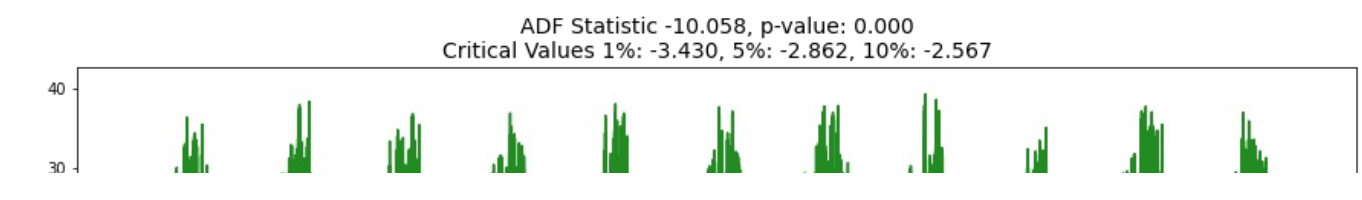
Now, we are going to check for each variable:

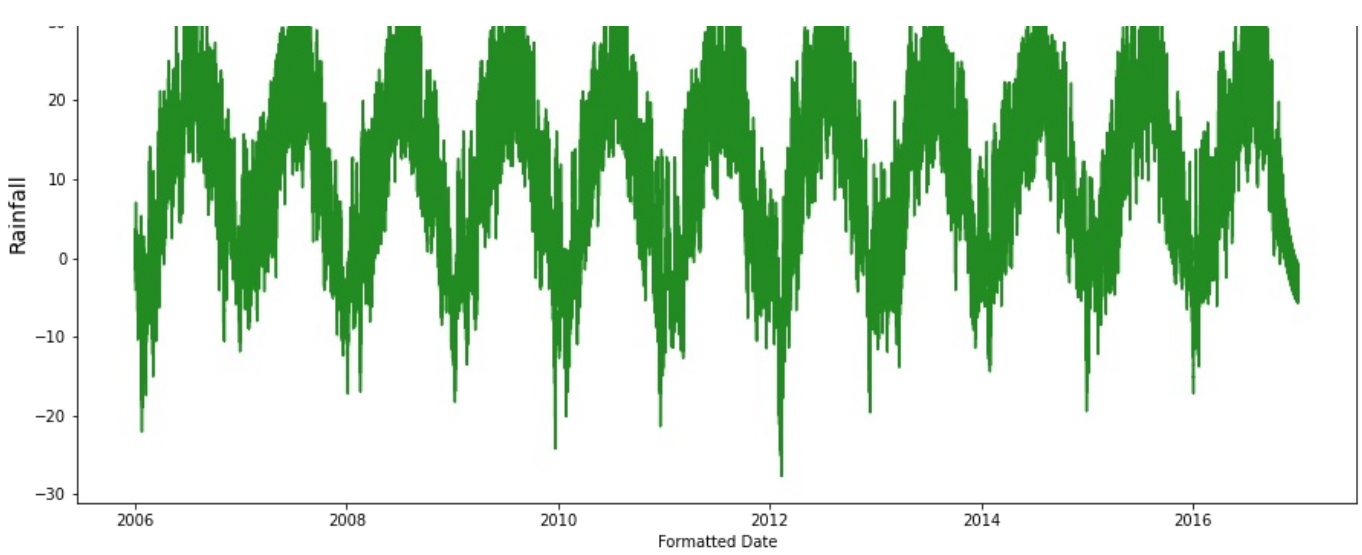
- The p-value is less than 0.05
- Check the range of the ADF statistic compared with critical_values

In [85]:

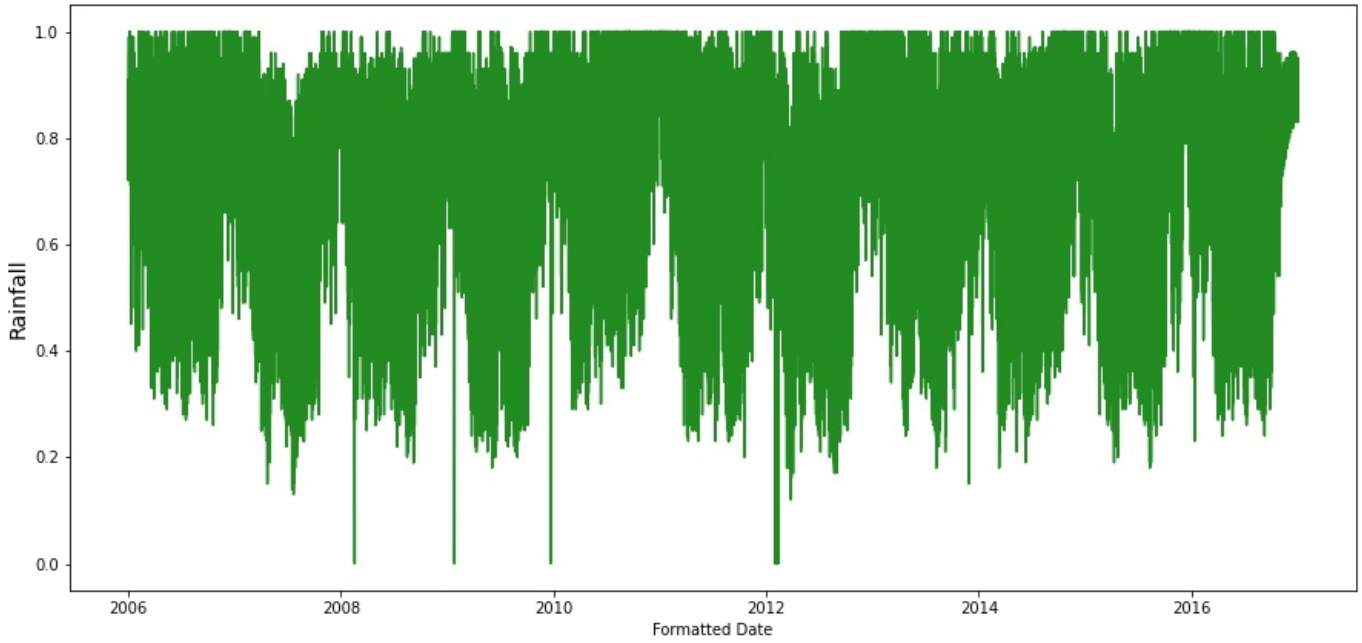
```
#f, ax = plt.subplots(nrows=3, ncols=2, figsize=(15, 9))
def visualize_adfuller_results(series, title):
    plt.figure(figsize=(15,7))
    result = adfuller(series)
    significance_level = 0.05
    adf_stat = result[0]
    p_val = result[1]
    crit_val_1 = result[4]['1%']
    crit_val_5 = result[4]['5%']
    crit_val_10 = result[4]['10%']

    if (p_val < significance_level) & ((adf_stat < crit_val_1)):
        linecolor = 'forestgreen'
    elif (p_val < significance_level) & (adf_stat < crit_val_5):
        linecolor = 'orange'
    elif (p_val < significance_level) & (adf_stat < crit_val_10):
        linecolor = 'red'
    else:
        linecolor = 'purple'
    sns.lineplot(x=data['Formatted Date'], y=series, color=linecolor)
    plt.title(f'ADF Statistic {adf_stat:0.3f}, p-value: {p_val:0.3f}\nCritical Values 1%: {crit_val_1:0.3f}, 5%:
    plt.ylabel(ylabel=title, fontsize=14)
    plt.show()
for col in numerical_feat:
    visualize_adfuller_results(data[col].values, 'Rainfall')
```



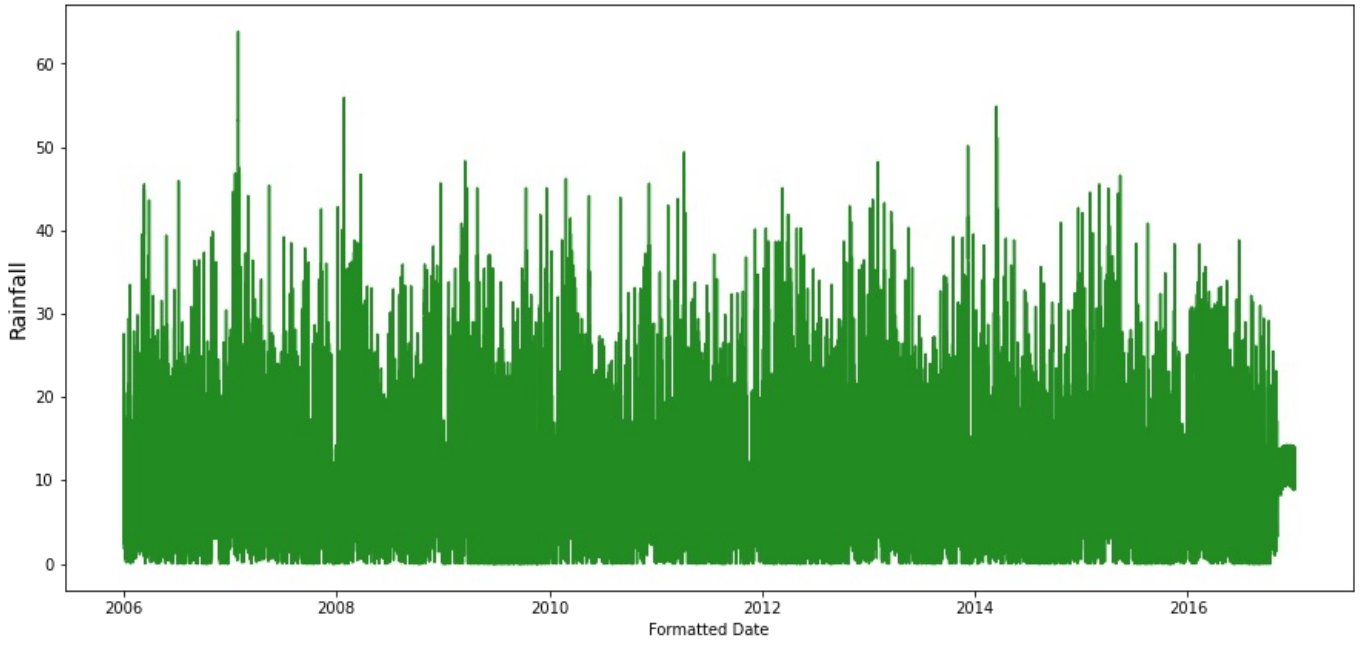




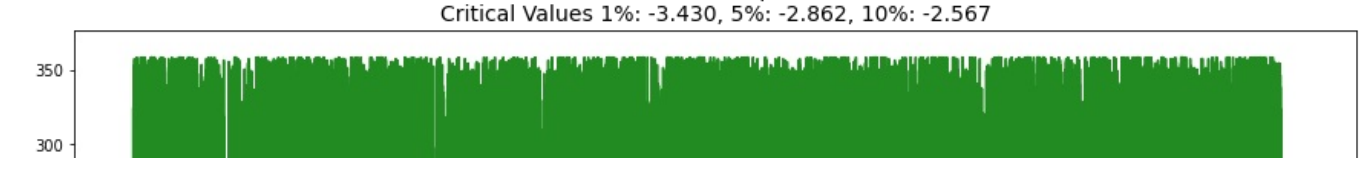
ADF Statistic -15.796, p-value: 0.000
Critical Values 1%: -3.430, 5%: -2.862, 10%: -2.567

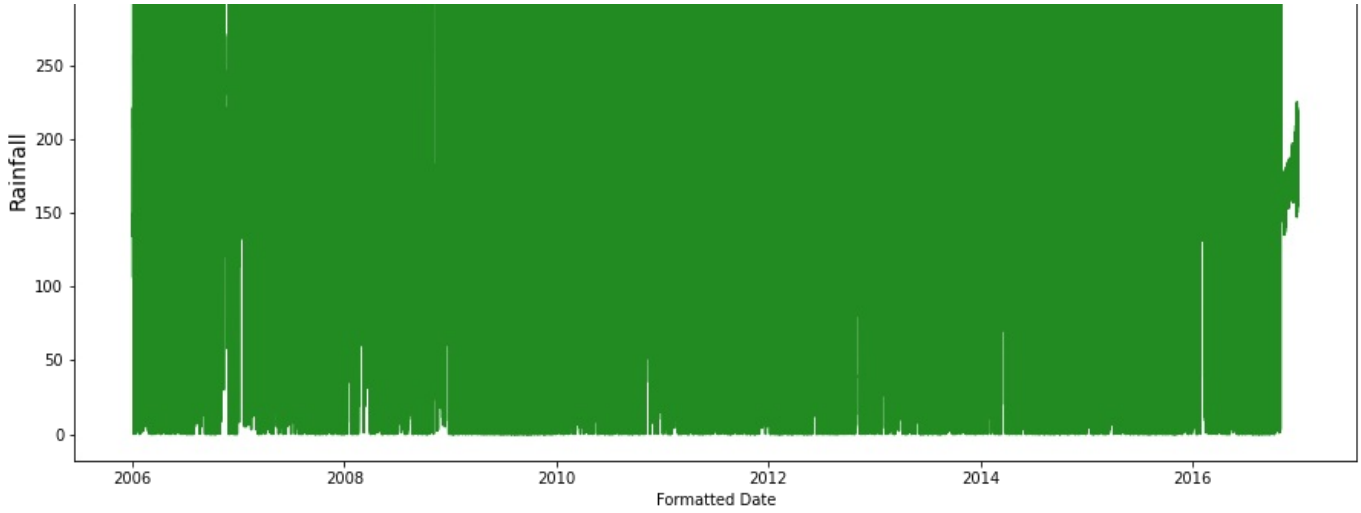


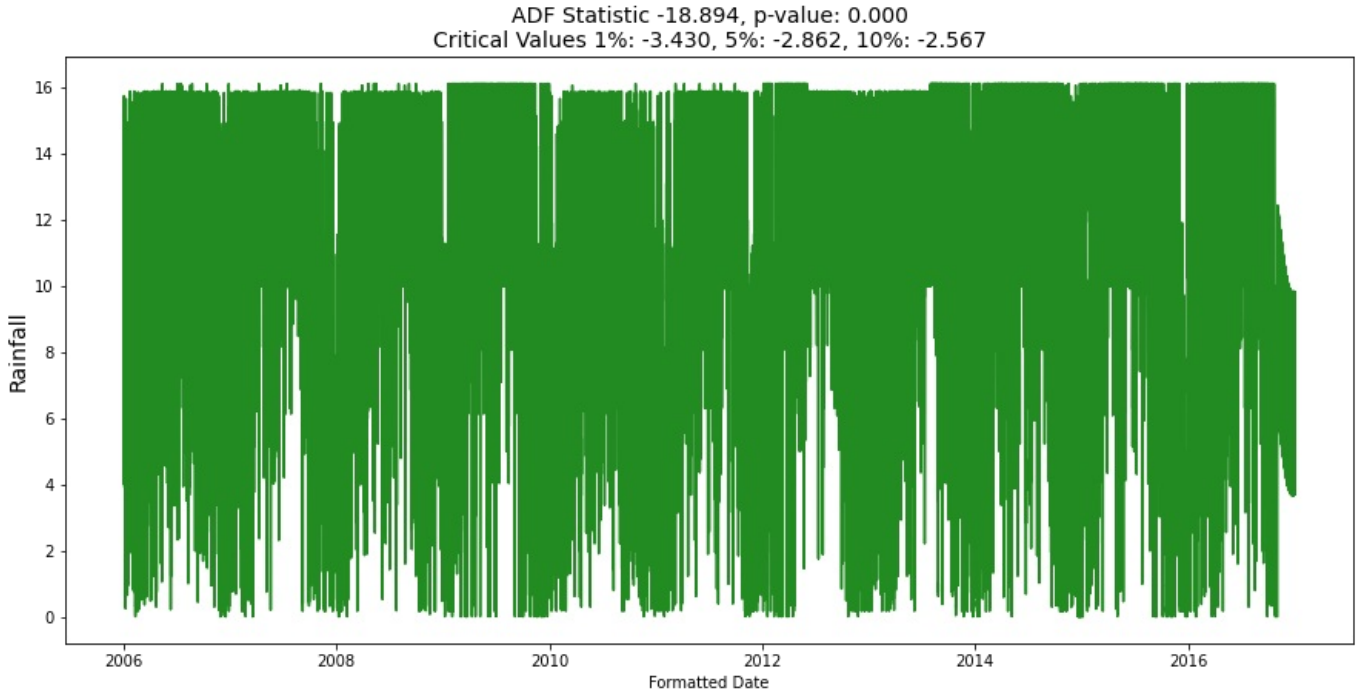
ADF Statistic -27.951, p-value: 0.000
Critical Values 1%: -3.430, 5%: -2.862, 10%: -2.567

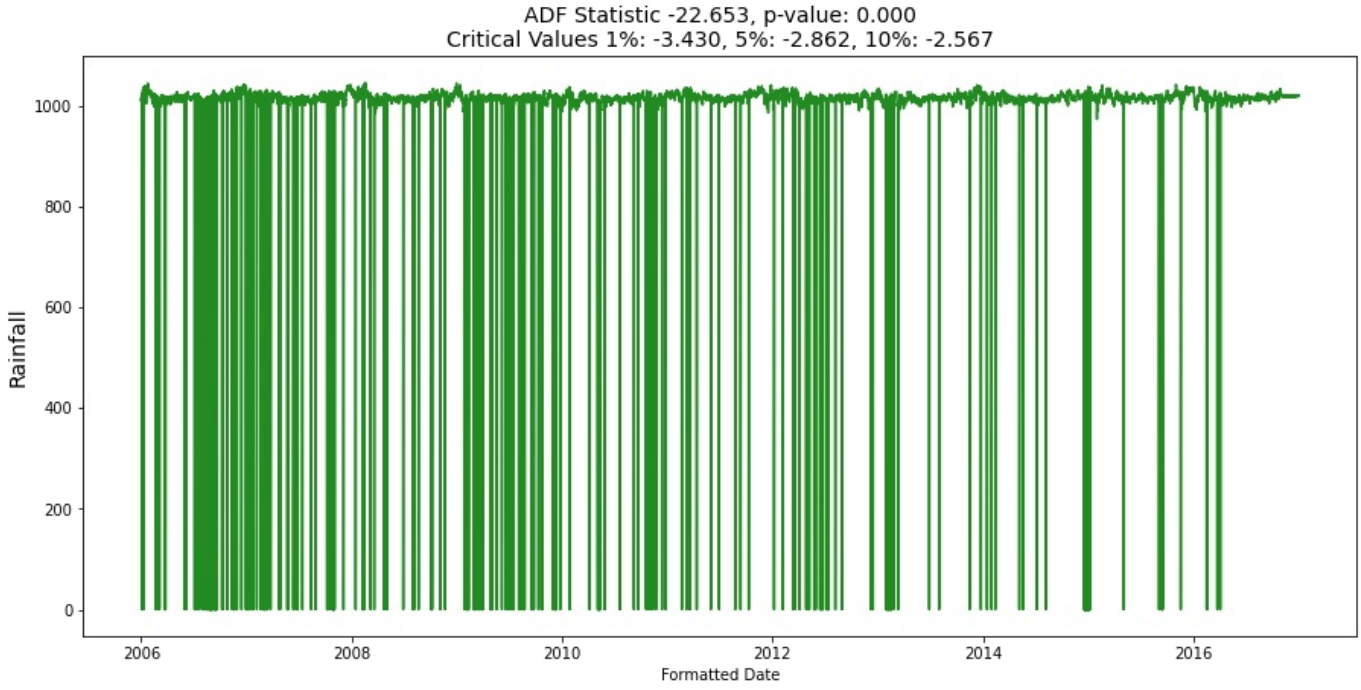


ADF Statistic -33.792, p-value: 0.000
Critical Values 1%: -3.430, 5%: -2.862, 10%: -2.567









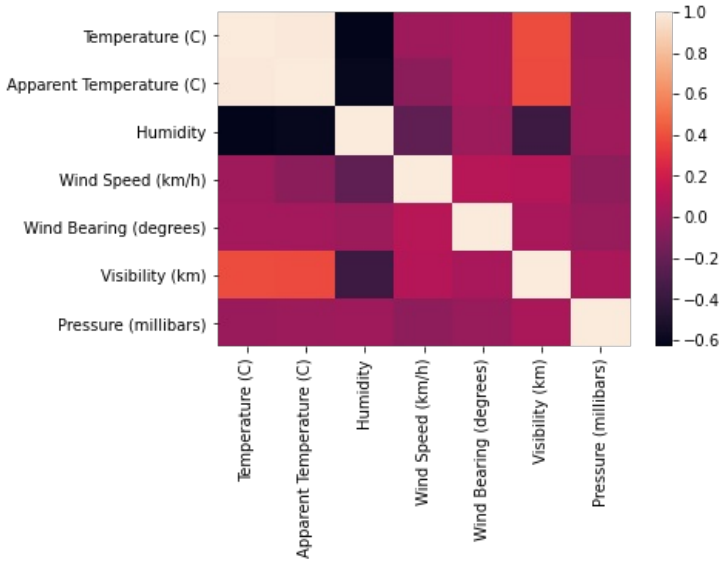
Clearly The Null Hypothesis has to be rejected with these Value of the Adfuller Test,

That is all the Time Series in this Dataset are Stationary.

Correlational Analysis

In [88]:
```
sns.heatmap(data[numerical_feat].corr())
```

Out[88]: <AxesSubplot:>



Grangercausality test

The grangercausality test used to find the determinant between two variable in the series. It uses prior datasets to find the correlation. i.e., X is cause of Y or Y is cause of X. It uses Bottom up/top down approach to see if the variables are generated independently or not from each other. The test gives value as null hypothesis i.e, variation in y does not interrupted by x. Grangercausality test used to find the dependencies between the variable in particular instantaneous time.

In [94]:
```
from statsmodels.tsa.stattools import grangercausalitytests
maxlag=12
test = 'ssr_chi2test'
def grangers_causation_matrix(data, variables, test='ssr_chi2test', verbose=False):

    df = pd.DataFrame(np.zeros((len(variables), len(variables))), columns=variables, index=variables)
    for c in df.columns:
        for r in df.index:
            test_result = grangercausalitytests(data[[r, c]], maxlag=maxlag, verbose=False)
            p_values = [round(test_result[i+1][0][test][1],4) for i in range(maxlag)]
            if verbose: print(f'Y = {r}, X = {c}, P Values = {p_values}')
            min_p_value = np.min(p_values)
            df.loc[r, c] = min_p_value
    df.columns = [var + '_x' for var in variables]
    df.index = [var + '_y' for var in variables]
    return df

df= grangers_causation_matrix(data[numerical_feat], variables = data[numerical_feat].columns)
```

```
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:

covariance of constraints does not have full rank. The number of constraints is 2, but rank is 1

C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:

covariance of constraints does not have full rank. The number of constraints is 3, but rank is 1

C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:

covariance of constraints does not have full rank. The number of constraints is 4, but rank is 1

C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:

covariance of constraints does not have full rank. The number of constraints is 5, but rank is 1

C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:

covariance of constraints does not have full rank. The number of constraints is 6, but rank is 1

C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:

covariance of constraints does not have full rank. The number of constraints is 7, but rank is 1
```

```
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 8, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 9, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 10, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 11, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 12, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 2, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 3, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 4, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 5, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 6, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 7, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 8, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 9, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 10, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 11, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 2, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 3, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 4, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 5, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 6, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
```

```
covariance of constraints does not have full rank. The number of constraints is 7, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 8, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 9, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 10, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 11, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 12, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 2, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 3, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 4, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 5, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 6, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 7, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 8, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 9, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 10, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 11, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 12, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 2, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 3, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 4, but rank is 2
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 5, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
```

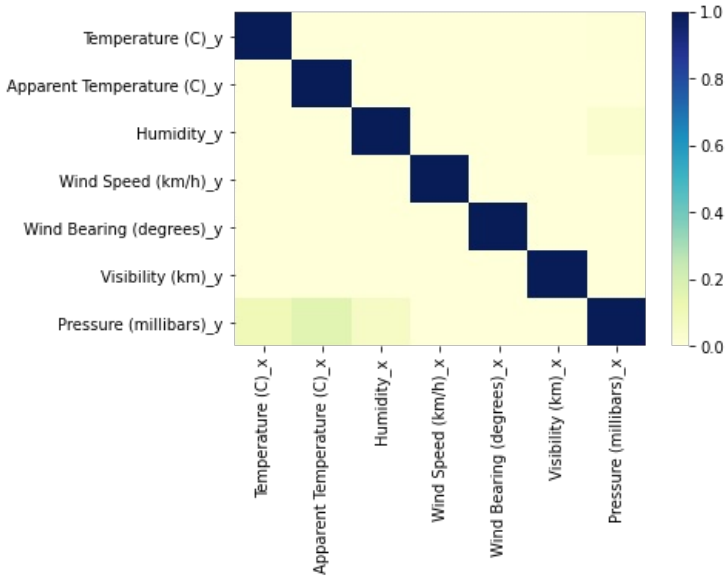
```
covariance of constraints does not have full rank. The number of constraints is 6, but rank is 2
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 7, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 8, but rank is 5
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 9, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 10, but rank is 2
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 11, but rank is 2
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 12, but rank is 1
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\model.py:1832: ValueWarning:
covariance of constraints does not have full rank. The number of constraints is 2, but rank is 1
```

In [98]:
```
df['Humidity_x']
```

Out[98]:
```
Temperature (C)_y             0.0000
Apparent Temperature (C)_y    0.0000
Humidity_y                    1.0000
Wind Speed (km/h)_y           0.0000
Wind Bearing (degrees)_y      0.0007
Visibility (km)_y             0.0000
Pressure (millibars)_y        0.0655
Name: Humidity_x, dtype: float64
```

In [100…
```
sns.heatmap(df,  cmap="YlGnBu")
```

Out[100… <AxesSubplot:>



All Variables Pass the Granger Causality Test.

In [15]:
```
data.corr()
```

Out[15]:

	Temperature					Visibility	Loud	Pressure

	(C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bearing (degrees)	(km)	Cover	(millibars)
Temperature (C)	1.000000	0.992629	-0.632255	0.008957	0.029988	0.392847	NaN	-0.005447
Apparent Temperature (C)	0.992629	1.000000	-0.602571	-0.056650	0.029031	0.381718	NaN	-0.000219
Humidity	-0.632255	-0.602571	1.000000	-0.224951	0.000735	-0.369173	NaN	0.005454
Wind Speed (km/h)	0.008957	-0.056650	-0.224951	1.000000	0.103822	0.100749	NaN	-0.049263
Wind Bearing (degrees)	0.029988	0.029031	0.000735	0.103822	1.000000	0.047594	NaN	-0.011651
Visibility (km)	0.392847	0.381718	-0.369173	0.100749	0.047594	1.000000	NaN	0.059818
Loud Cover	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Pressure (millibars)	-0.005447	-0.000219	0.005454	-0.049263	-0.011651	0.059818	NaN	1.000000

Consider Dropping Apparent Temperature(C) in favor of Temperature as the Correlation value is too high. Same With Loud Cover as it doesn't correlate with any other variable.

4. Data Preparation

a. Do the final feature selection and extract them into Column X and the class label into Column into Y.

b. Split the dataset into training and test sets.

Features we are taking for Column X for Time Series Model are:

1. Temperature (C)
2. Wind Speed (km/h)
3. Wind Bearing (degrees)
4. Visibility (km)
5. Pressure (millibars)

And the Column Y will be:- Humidity

In [294…
```
x=data[[col for col in data.columns if data[col].dtype ==float and col not in ['Apparent Temperature (C)','Loud (
x= (x-x.min())/(x.max()-x.min())
data = pd.concat([x,data[['Summary', 'Precip Type']]],axis=1)
```

In [302…
```
dataset = data[['Temperature (C)', 'Humidity',
       'Wind Speed (km/h)', 'Wind Bearing (degrees)', 'Visibility (km)',
       'Pressure (millibars)']].reindex(columns=['Temperature (C)',
       'Wind Speed (km/h)', 'Wind Bearing (degrees)', 'Visibility (km)',
       'Pressure (millibars)','Humidity'])
```

In [303…
```
dataset.columns
```

Out[303…
```
Index(['Temperature (C)', 'Wind Speed (km/h)', 'Wind Bearing (degrees)',
       'Visibility (km)', 'Pressure (millibars)', 'Humidity'],
      dtype='object')
```

In [304…
```
data_train = dataset[:int(0.8*(len(dataset)))]
data_test = dataset[int(0.8*(len(dataset))):]
```

In [305…
```
print(data_train.shape)
print(data_test.shape)
```

```
(77162, 6)
(19291, 6)
```

In [306…
```
test =data_test[['Humidity']]
```

Part B (12 marks)

1. Model Building

a. Perform Model Development using at least three models, separately. You are free to apply any Machine Learning Models on the dataset. Deep Learning Models are strictly not allowed.

b. Train the model and print the training accuracy and loss values.

2. Performance Evaluation

a. Print the confusion matrix. Provide appropriate analysis for the same.
b. Do the prediction for the test data and display the results for the inference.

In [307…
```
from statsmodels.tsa.vector_ar.var_model import VAR
```

In [308…
```
model = VAR(data_train)
for i in [1,2,3,4,5]:
    result = model.fit(i)
    print('Lag Order =', i)
    print('AIC : ', result.aic)
    print('BIC : ', result.bic)
    print('FPE : ', result.fpe)
    print('HQIC: ', result.hqic, '\n')
```

```
Lag Order = 1
AIC :  -30.328143574667024
BIC :  -30.32310666125306
FPE :  6.73991778819472e-14
HQIC:  -30.326596962162743

Lag Order = 2
AIC :  -30.681842482091035
BIC :  -30.67248810619125
FPE :  4.7320041344855075e-14
HQIC:  -30.678970166829277

Lag Order = 3
AIC :  -31.13026402907274
BIC :  -31.116592090870054
FPE :  3.022025412265245e-14
HQIC:  -31.126065978839925

Lag Order = 4
AIC :  -31.18272075257663
BIC :  -31.164731152250315
FPE :  2.867585967374926e-14
HQIC:  -31.177196935157973

Lag Order = 5
AIC :  -31.200972973504673
BIC :  -31.178665611230365
FPE :  2.815720921127625e-14
HQIC:  -31.19412335668418
```

In [309…
```
x = model.select_order(maxlags=5)
x.summary()
```

Out[309…

VAR Order Selection (* highlights the minimums)

	AIC	BIC	FPE	HQIC
0	-21.63	-21.63	4.042e-10	-21.63
1	-30.33	-30.32	6.740e-14	-30.33
2	-30.68	-30.67	4.732e-14	-30.68
3	-31.13	-31.12	3.022e-14	-31.13
4	-31.18	-31.16	2.868e-14	-31.18
5	-31.20*	-31.18*	2.816e-14*	-31.19*

In [310…
```
model_fitted = model.fit()
model_fitted.summary()
```

Out[310…
```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Sun, 29, Aug, 2021
Time:                     16:29:43
--------------------------------------------------------------------
No. of Equations:         6.00000    BIC:                   -30.3231
Nobs:                     77161.0    HQIC:                  -30.3266
Log likelihood:           513197.    FPE:                6.73992e-14
AIC:                     -30.3281    Det(Omega_mle):     6.73625e-14
```

```
---------------------------------------------------------------------
Results for equation Temperature (C)
=====================================================================================================
                              coefficient       std. error           t-stat             prob
-----------------------------------------------------------------------------------------------------
const                           -0.006661         0.001164           -5.722            0.000
L1.Temperature (C)               1.003530         0.000796         1260.807            0.000
L1.Wind Speed (km/h)             0.003257         0.000886            3.674            0.000
L1.Wind Bearing (degrees)       -0.001735         0.000311           -5.586            0.000
L1.Visibility (km)              -0.010019         0.000405          -24.719            0.000
L1.Pressure (millibars)          0.001172         0.000781            1.501            0.133
L1.Humidity                      0.014106         0.000638           22.104            0.000
=====================================================================================================

Results for equation Wind Speed (km/h)
=====================================================================================================
                              coefficient       std. error           t-stat             prob
-----------------------------------------------------------------------------------------------------
const                            0.036911         0.002737           13.487            0.000
L1.Temperature (C)              -0.010413         0.001871           -5.565            0.000
L1.Wind Speed (km/h)             0.834478         0.002084          400.413            0.000
L1.Wind Bearing (degrees)       -0.001019         0.000730           -1.395            0.163
L1.Visibility (km)              -0.000784         0.000953           -0.822            0.411
L1.Pressure (millibars)          0.000770         0.001836            0.420            0.675
L1.Humidity                     -0.003784         0.001500           -2.522            0.012
=====================================================================================================

Results for equation Wind Bearing (degrees)
=====================================================================================================
                              coefficient       std. error           t-stat             prob
-----------------------------------------------------------------------------------------------------
const                            0.133461         0.010782           12.378            0.000
L1.Temperature (C)               0.049928         0.007372            6.773            0.000
L1.Wind Speed (km/h)             0.093837         0.008210           11.429            0.000
L1.Wind Bearing (degrees)        0.603017         0.002877          209.566            0.000
L1.Visibility (km)               0.027981         0.003754            7.453            0.000
L1.Pressure (millibars)         -0.018468         0.007234           -2.553            0.011
L1.Humidity                      0.043692         0.005911            7.392            0.000
=====================================================================================================

Results for equation Visibility (km)
=====================================================================================================
                              coefficient       std. error           t-stat             prob
-----------------------------------------------------------------------------------------------------
const                            0.133046         0.006085           21.866            0.000
L1.Temperature (C)               0.100105         0.004160           24.064            0.000
L1.Wind Speed (km/h)             0.037237         0.004633            8.037            0.000
L1.Wind Bearing (degrees)        0.004597         0.001624            2.831            0.005
L1.Visibility (km)               0.805215         0.002119          380.086            0.000
L1.Pressure (millibars)         -0.050114         0.004082          -12.277            0.000
L1.Humidity                     -0.034629         0.003335          -10.382            0.000
=====================================================================================================

Results for equation Pressure (millibars)
=====================================================================================================
                              coefficient       std. error           t-stat             prob
-----------------------------------------------------------------------------------------------------
const                            0.639537         0.005034          127.043            0.000
L1.Temperature (C)              -0.011652         0.003442           -3.385            0.001
L1.Wind Speed (km/h)            -0.030198         0.003833           -7.878            0.000
L1.Wind Bearing (degrees)       -0.002990         0.001343           -2.225            0.026
L1.Visibility (km)               0.000076         0.001753            0.043            0.965
L1.Pressure (millibars)          0.352229         0.003377          104.294            0.000
L1.Humidity                     -0.009091         0.002760           -3.294            0.001
=====================================================================================================

Results for equation Humidity
=====================================================================================================
                              coefficient       std. error           t-stat             prob
-----------------------------------------------------------------------------------------------------
const                            0.086101         0.002945           29.237            0.000
L1.Temperature (C)              -0.059661         0.002013          -29.631            0.000
L1.Wind Speed (km/h)            -0.036175         0.002243          -16.131            0.000
L1.Wind Bearing (degrees)        0.001667         0.000786            2.121            0.034
L1.Visibility (km)               0.025487         0.001025           24.856            0.000
L1.Pressure (millibars)         -0.005439         0.001976           -2.753            0.006
L1.Humidity                      0.918969         0.001614          569.238            0.000
=====================================================================================================

Correlation matrix of residuals
                        Temperature (C)  Wind Speed (km/h)  Wind Bearing (degrees)  Visibility (km)  Pressure (
millibars)  Humidity
Temperature (C)                1.000000           0.178792               -0.019204         0.098316
```

```
-0.005206 -0.636186
Wind Speed (km/h)                0.178792          1.000000               0.067928         0.067508
-0.022845 -0.226101
Wind Bearing (degrees)          -0.019204          0.067928               1.000000        -0.000493
0.003291  0.021720
Visibility (km)                  0.098316          0.067508              -0.000493         1.000000
0.100965 -0.153889
Pressure (millibars)            -0.005206         -0.022845               0.003291         0.100965
1.000000  0.004739
Humidity                        -0.636186         -0.226101               0.021720        -0.153889
0.004739  1.000000
```

In [312…
```
# Get the lag order
lag_order = model_fitted.k_ar
print(lag_order)  #> 4
```

```
1
```

In [313…
```
# Input data for forecasting
forecast_input = data_train.values[-lag_order:]
forecast_input
```

Out[313…
```
array([[0.33822338, 0.05042864, 0.        , 0.017     , 0.98101072,
        0.93      ]])
```

In [314…
```
nobs= data_test.shape[0]
nobs
```

Out[314…
```
19291
```

In [315…
```
# Forecast
fc = model_fitted.forecast(y=forecast_input,steps= nobs)
df_forecast = pd.DataFrame(fc, index=dataset.index[-nobs:], columns=dataset.columns + '_2d')
df_forecast
```

Out[315…

	Temperature (C)_2d	Wind Speed (km/h)_2d	Wind Bearing (degrees)_2d	Visibility (km)_2d	Pressure (millibars)_2d	Humidity_2d
77162	0.347019	0.072693	0.178071	0.101103	0.971160	0.913837
77163	0.354526	0.090987	0.289809	0.172406	0.966537	0.900148
77164	0.361013	0.106053	0.360762	0.232472	0.964064	0.888487
77165	0.366679	0.118481	0.406503	0.282902	0.962561	0.878501
77166	0.371679	0.128743	0.436537	0.325171	0.961548	0.869907
...	...	...	...	...	...	...
96448	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576
96449	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576
96450	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576
96451	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576
96452	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576

19291 rows × 6 columns

In [325…
```
def invert_transformation(df_train, df_forecast, second_diff=False):
    df_fc = df_forecast.copy()
    columns = df_train.columns
    for col in columns:
        # Roll back 2nd Diff
        if second_diff:
            df_fc[str(col)+'_1d'] = (df_train[col].iloc[-1]-df_train[col].iloc[-2]) + df_fc[str(col)+'_2d'].cumsu
        # Roll back 1st Diff
        df_fc[str(col)+'_forecast'] = df_train[col].iloc[-1] + df_fc[str(col)+'_2d'].cumsum()
    return df_fc
```

In [326…
```
df_results = invert_transformation(data_train, df_forecast, second_diff=False)
```

In [327…
```
df_results
```

Out[327…

	Wind	Wind

	Temperature (C)_2d	Speed (km/h)_2d	Bearing (degrees)_2d	Visibility (km)_2d	Pressure (millibars)_2d	Humidity_2d	Temperature (C)_forecast	Wind Speed (km/h)_forecast	Wind Bearing (degrees)_forecast	Visib (km)_fore
77162	0.347019	0.072693	0.178071	0.101103	0.971160	0.913837	0.685242	0.123122	0.178071	0.118
77163	0.354526	0.090987	0.289809	0.172406	0.966537	0.900148	1.039768	0.214109	0.467880	0.290
77164	0.361013	0.106053	0.360762	0.232472	0.964064	0.888487	1.400781	0.320163	0.828643	0.522
77165	0.366679	0.118481	0.406503	0.282902	0.962561	0.878501	1.767460	0.438644	1.235146	0.805
77166	0.371679	0.128743	0.436537	0.325171	0.961548	0.869907	2.139139	0.567387	1.671683	1.131
...	...	...	...	...	...	...	...	...	...	
96448	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576	10500.569821	3283.295689	10131.430132	12179.033
96449	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576	10501.114981	3283.465911	10131.955559	12179.665
96450	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576	10501.660141	3283.636133	10132.480986	12180.297
96451	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576	10502.205300	3283.806355	10133.006412	12180.929
96452	0.545160	0.170222	0.525427	0.632070	0.956943	0.730576	10502.750460	3283.976577	10133.531839	12181.561

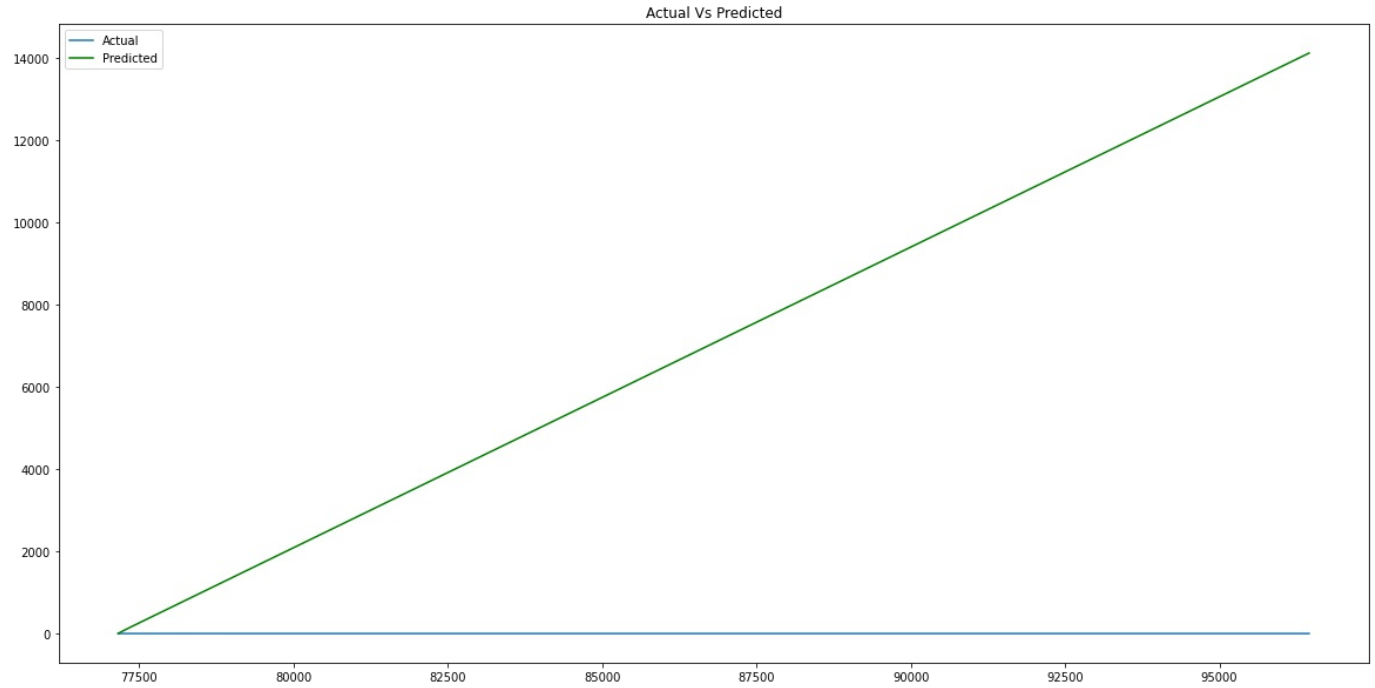
19291 rows × 12 columns

In [328…
```
predict = df_results['Humidity_forecast']
```

In [329…
```
import math
from sklearn.metrics import mean_squared_error
math.sqrt( mean_squared_error(test,predict))
```

Out[329… 8145.121720978112

In [333…
```
import matplotlib.pyplot as plt
# zoom plot
plt.figure(figsize=(20,10))
plt.plot(test, label='Actual')
plt.plot(predict, color='green',label='Predicted')
plt.title('Actual Vs Predicted')
plt.legend()
plt.show()
```



In [341…
```
model = VAR(endog=data_train)
model_fit = model.fit()

# make prediction on validation
prediction = model_fit.forecast(model_fit.y, steps=len(data_test))
```

```
C:\Users\Rishabh\anaconda3\lib\site-packages\statsmodels\base\wrapper.py:34: FutureWarning:

y is a deprecated alias for endog, will be removed in version 0.11.0
```

In [339…
```
pred = pd.DataFrame(index=range(0,len(prediction)),columns=[dataset.columns])
# for j in range(0,13):
#     for i in range(0, len(prediction)):
#         pred.iloc[i][j] = prediction[i][j]

# #check rmse
# for i in cols:
#     print('rmse value for', i, 'is : ', sqrt(mean_squared_error(pred[i], valid[i])))
```

In [343…
```
pred =df_forecast['Humidity_2d']
```

In [344…
```
import math
from sklearn.metrics import mean_squared_error
math.sqrt( mean_squared_error(test,pred))
```

Out[344… 0.19091683639808885

In [184…
```
len(test)
```

Out[184… 19291

In [196…
```
test.index = data['Formatted Date'][-nobs:]
```

In [197…
```
pred_d = pred[['Humidity']]
```

In [198…
```
pred_d.index = data['Formatted Date'][-nobs:]
```

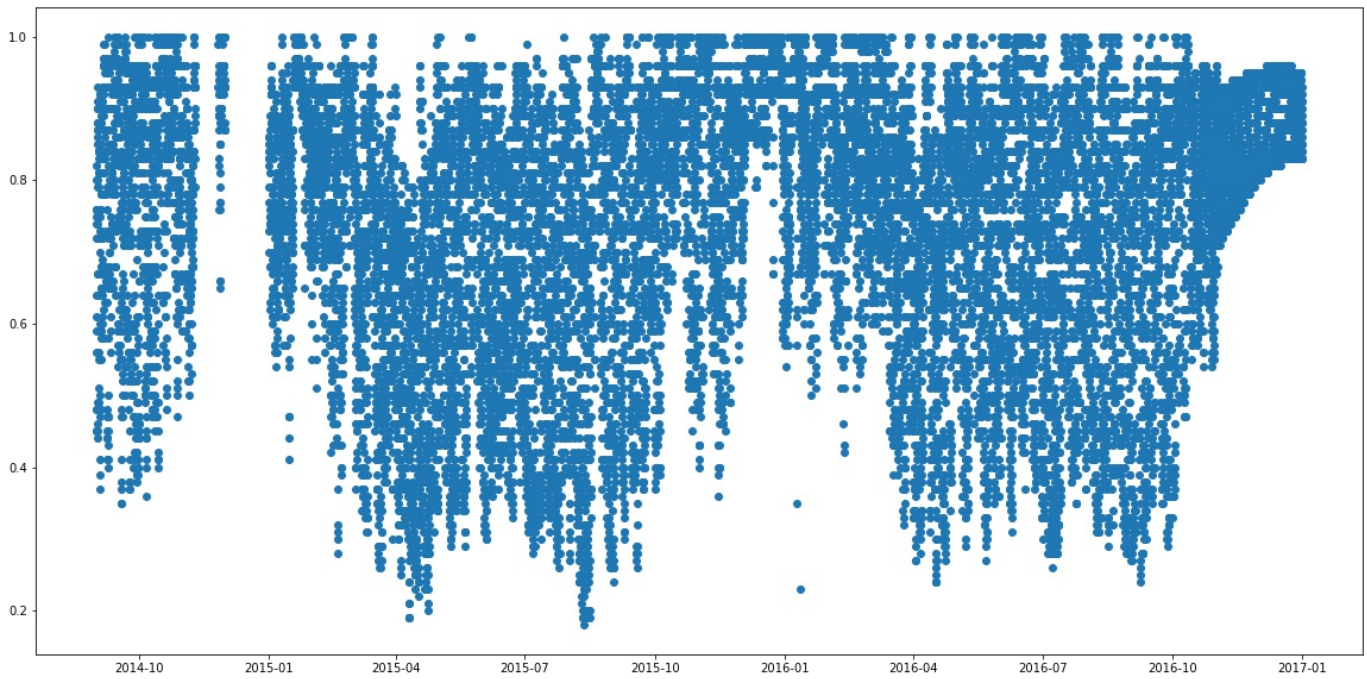
In [186…
```
len(pred[['Humidity']])
```

Out[186… 19291

In [206…
```
# import matplotlib.pyplot as plt
# # zoom plot
# plt.figure(figsize=(20,10))
# plt.plot_date(x= data['Formatted Date'][-nobs:],y= test['Humidity'], label='Actual')
# plt.plot_date(x=data['Formatted Date'][-nobs:] ,y =pred_d[['Humidity']], color='green',label='Predicted')
# plt.title('Actual Vs Predicted')
# plt.legend()
# plt.show()
```

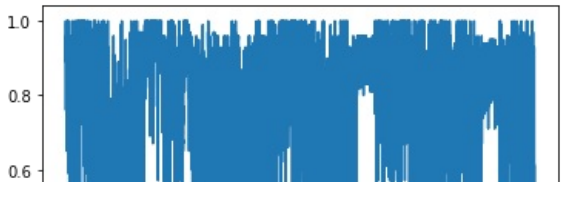
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-206-ff06b5358641> in <module>
      3 plt.figure(figsize=(20,10))
      4 plt.plot_date(x= data['Formatted Date'][-nobs:],y= test['Humidity'], label='Actual')
----> 5 plt.plot_date(x=data['Formatted Date'][-nobs:] ,y =pred_d[['Humidity']], color='green',label='Predicted')
      6 plt.title('Actual Vs Predicted')
      7 plt.legend()

~\AppData\Roaming\Python\Python38\site-packages\matplotlib\pyplot.py in plot_date(x, y, fmt, tz, xdate, ydate, da
ta, **kwargs)
   2848         x, y, fmt='o', tz=None, xdate=True, ydate=False, *,
   2849         data=None, **kwargs):
-> 2850     return gca().plot_date(
   2851         x, y, fmt=fmt, tz=tz, xdate=xdate, ydate=ydate,
   2852         **({"data": data} if data is not None else {}), **kwargs)

~\AppData\Roaming\Python\Python38\site-packages\matplotlib\__init__.py in inner(ax, data, *args, **kwargs)
   1436     def inner(ax, *args, data=None, **kwargs):
   1437         if data is None:
-> 1438             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1439
   1440         bound = new_sig.bind(ax, *args, **kwargs)

~\AppData\Roaming\Python\Python38\site-packages\matplotlib\axes\_axes.py in plot_date(self, x, y, fmt, tz, xdate,
ydate, **kwargs)
   1811         if ydate:
   1812             self.yaxis_date(tz)
-> 1813         return self.plot(x, y, fmt, **kwargs)
   1814
   1815     # @_preprocess_data() # let 'plot' do the unpacking..

~\AppData\Roaming\Python\Python38\site-packages\matplotlib\axes\_axes.py in plot(self, scalex, scaley, data, *arg
s, **kwargs)
```

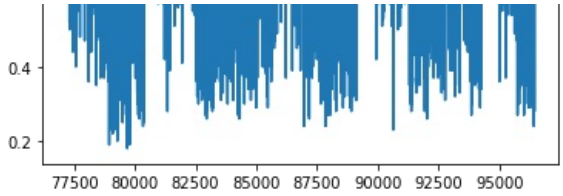
```
   1741         """
   1742         kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1743         lines = [*self._get_lines(*args, data=data, **kwargs)]
   1744         for line in lines:
   1745             self.add_line(line)

~\AppData\Roaming\Python\Python38\site-packages\matplotlib\axes\_base.py in __call__(self, data, *args, **kwargs)
    271                 this += args[0],
    272                 args = args[1:]
--> 273             yield from self._plot_args(this, kwargs)
    274
    275     def get_next_color(self):

~\AppData\Roaming\Python\Python38\site-packages\matplotlib\axes\_base.py in _plot_args(self, tup, kwargs)
    387         if len(tup) == 2:
    388             x = _check_1d(tup[0])
--> 389             y = _check_1d(tup[-1])
    390         else:
    391             x, y = index_of(tup[-1])

~\AppData\Roaming\Python\Python38\site-packages\matplotlib\cbook\__init__.py in _check_1d(x)
   1316                     message='Support for multi-dimensional indexing')
   1317
-> 1318                 ndim = x[:, None].ndim
   1319                 # we have definitely hit a pandas index or series object
   1320                 # cast to a numpy array.

~\AppData\Roaming\Python\Python38\site-packages\pandas\core\frame.py in __getitem__(self, key)
   2798             if self.columns.nlevels > 1:
   2799                 return self._getitem_multilevel(key)
-> 2800             indexer = self.columns.get_loc(key)
   2801             if is_integer(indexer):
   2802                 indexer = [indexer]

~\AppData\Roaming\Python\Python38\site-packages\pandas\core\indexes\multi.py in get_loc(self, key, method)
   2665         keylen = len(key)
   2666         if self.nlevels < keylen:
-> 2667             raise KeyError(
   2668                 f"Key length ({keylen}) exceeds index depth ({self.nlevels})"
   2669             )

KeyError: 'Key length (2) exceeds index depth (1)'
```

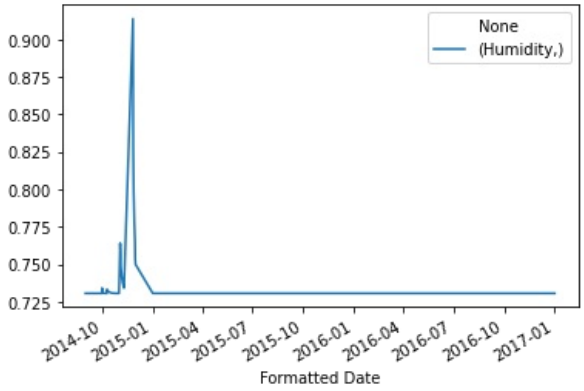


In [331…
```
test['Humidity'].plot()
pred_d[['Humidity']].plot()
```

Out[331…
```
<AxesSubplot:xlabel='Formatted Date'>
```







In [332…

```
from sklearn.metrics import mean_squared_error
( mean_squared_error(test,pred[['Humidity']]))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-332-a69b6d067253> in <module>
      1 from sklearn.metrics import mean_squared_error
----> 2 ( mean_squared_error(test,pred[['Humidity']]))

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
     71                           FutureWarning)
     72         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 73         return f(**kwargs)
     74     return inner_f
     75

~\anaconda3\lib\site-packages\sklearn\metrics\_regression.py in mean_squared_error(y_true, y_pred, sample_weight,
multioutput, squared)
    251
    252     """
--> 253     y_type, y_true, y_pred, multioutput = _check_reg_targets(
    254         y_true, y_pred, multioutput)
    255     check_consistent_length(y_true, y_pred, sample_weight)

~\anaconda3\lib\site-packages\sklearn\metrics\_regression.py in _check_reg_targets(y_true, y_pred, multioutput, d
type)
     84     check_consistent_length(y_true, y_pred)
     85     y_true = check_array(y_true, ensure_2d=False, dtype=dtype)
---> 86     y_pred = check_array(y_pred, ensure_2d=False, dtype=dtype)
     87
     88     if y_true.ndim == 1:

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
     71                           FutureWarning)
     72         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 73         return f(**kwargs)
     74     return inner_f
     75

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in check_array(array, accept_sparse, accept_large_spars
e, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_features, estimator)
    643
    644         if force_all_finite:
--> 645             _assert_all_finite(array,
    646                                allow_nan=force_all_finite == 'allow-nan')
    647

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in _assert_all_finite(X, allow_nan, msg_dtype)
     95                 not allow_nan and not np.isfinite(X).all()):
     96             type_err = 'infinity' if allow_nan else 'NaN, infinity'
---> 97             raise ValueError(
     98                     msg_err.format
     99                     (type_err,

ValueError: Input contains NaN, infinity or a value too large for dtype('float64').
```

Conclusions

1. RMSE for VAR: 0.1909
2. MSE for VAR: 0.03645

Following is Modeling for Data which as been modelled for Classical Machine Learning

3. Data Pre-processing and cleaning

a. Do the appropriate preprocessing of the data like identifying NULL or Missing Values if any, handling of outliers if present in the dataset, skewed data etc. Apply appropriate feature engineering techniques for them.

b. Apply the feature transformation techniques like Standardization, Normalization, etc. You are free to apply the appropriate transformations depending upon the structure and the complexity of your dataset.

c. Do the correlational analysis on the dataset. Provide a visualization for the same.

In [223…
```
data = pd.read_csv('weatherHistory.csv')
```

In [228…
```
data['Humidity'].describe()
```

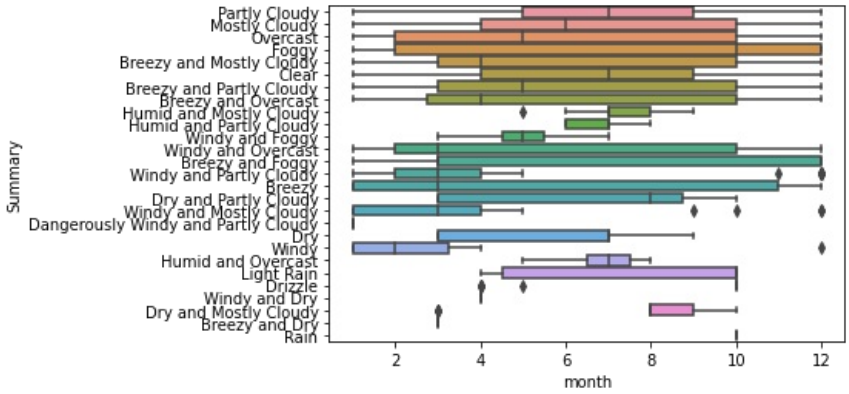
Out[228…
```
count    96453.000000
mean         0.734899
std          0.195473
min          0.000000
25%          0.600000
50%          0.780000
75%          0.890000
max          1.000000
Name: Humidity, dtype: float64
```

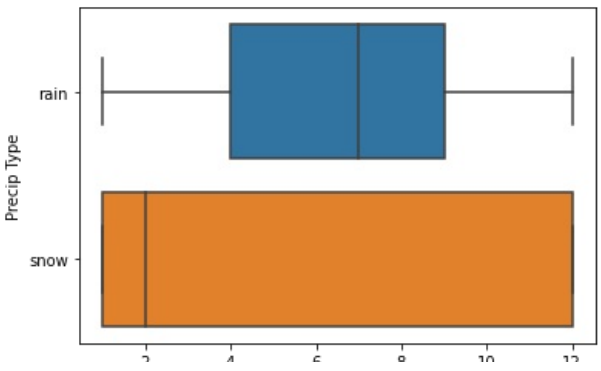
In [235…
```
data['Date'] = pd.to_datetime(data['Formatted Date'],utc= True)
```

In [236…
```
data['month'] = data['Date'].dt.month
```

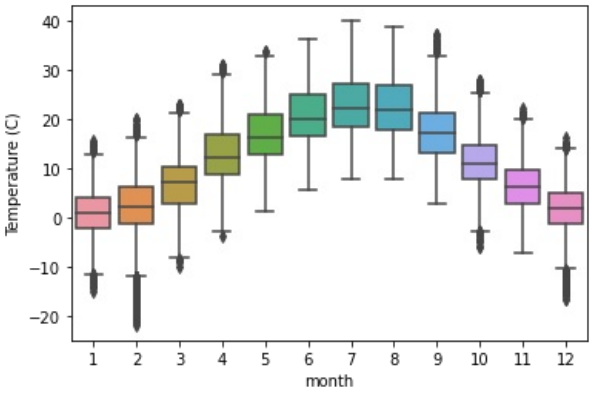
In [247…
```
data['year'] = data['Date'].dt.year
```

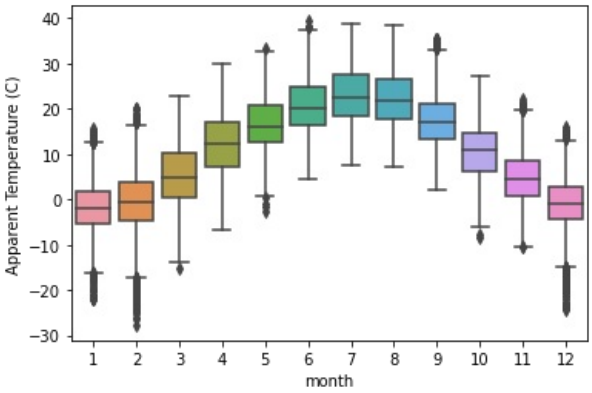
In [244…
```
for col in [col for col in data.columns if col not in ['Date', 'Formatted Date', 'Daily Summary', 'month']]:
    sns.boxplot(data=data,y=col, x='month')
    plt.show()
```

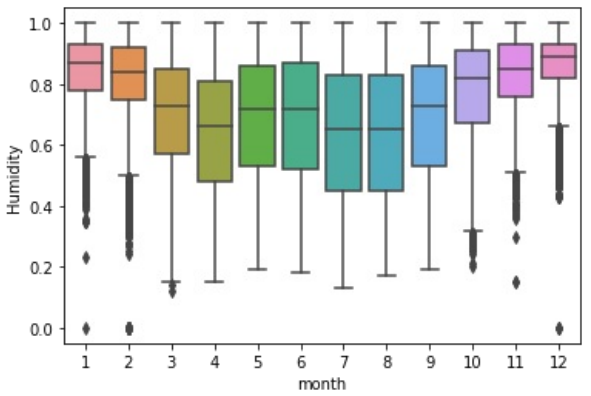


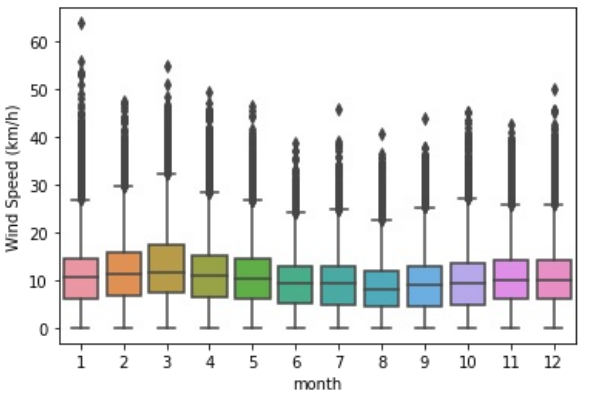


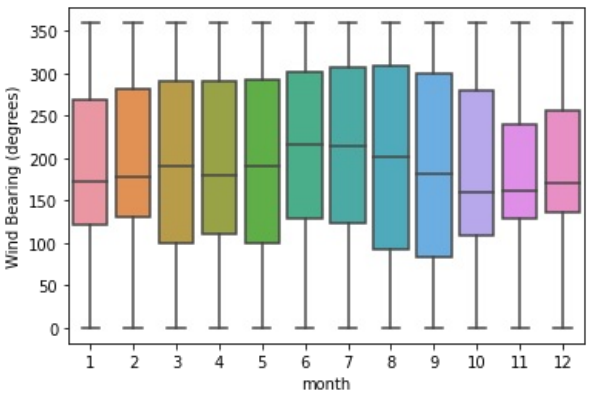

month


Temperature (C)
month

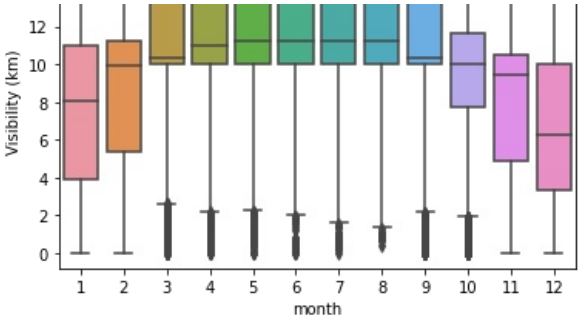

Apparent Temperature (C)
month

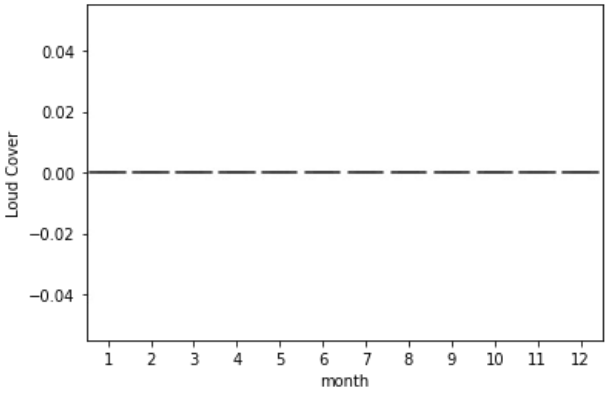

Humidity
month

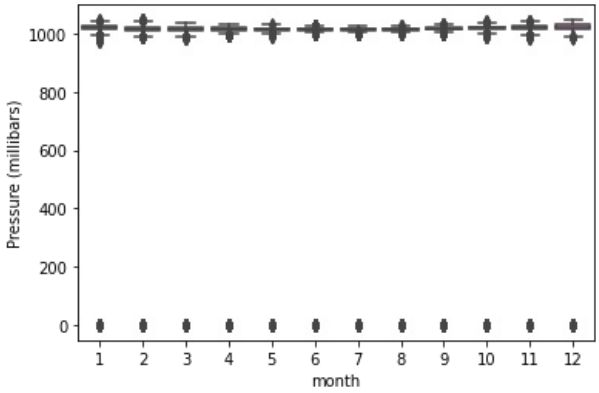

Wind Speed (km/h)
month


Wind Bearing (degrees)
month


16
14

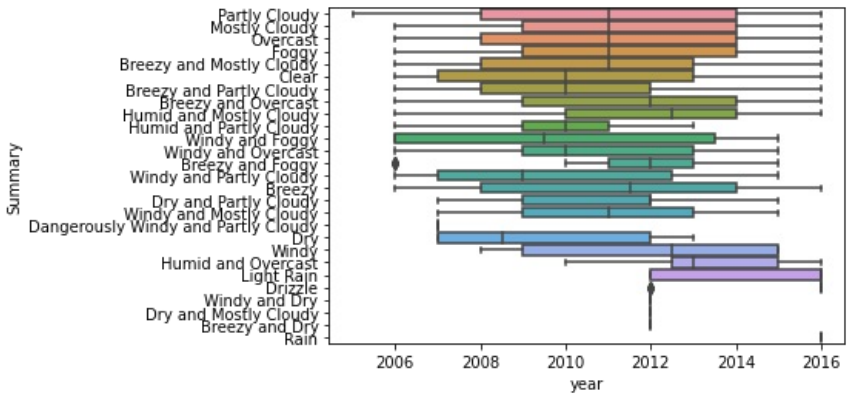


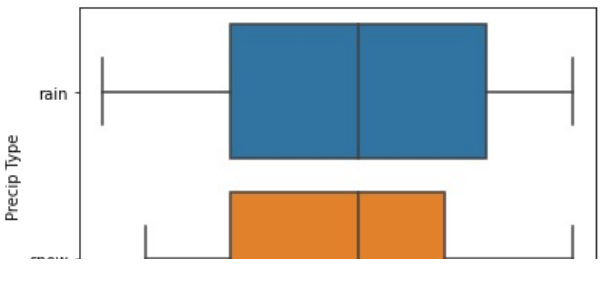


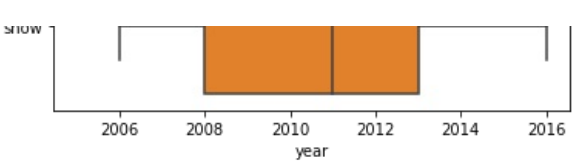


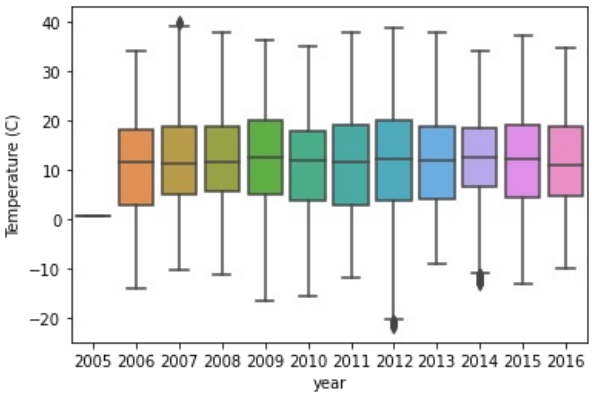
In [248…

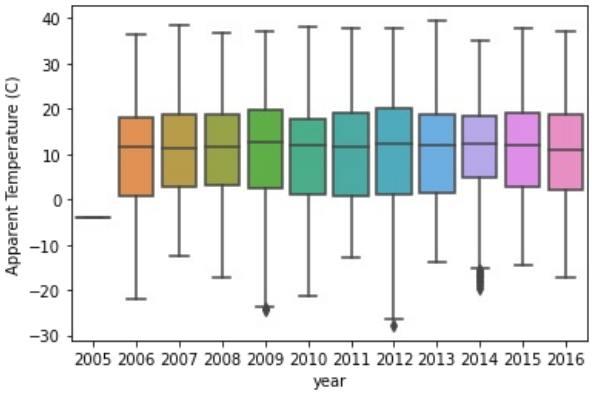
```
for col in [col for col in data.columns if col not in ['Date', 'Formatted Date', 'Daily Summary', 'month','year']]:
    sns.boxplot(data=data,y=col, x='year')
    plt.show()
```

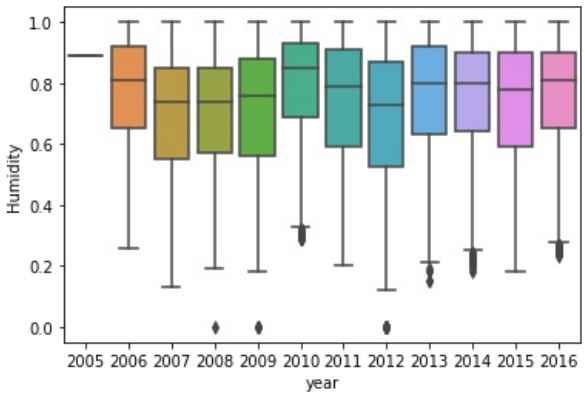


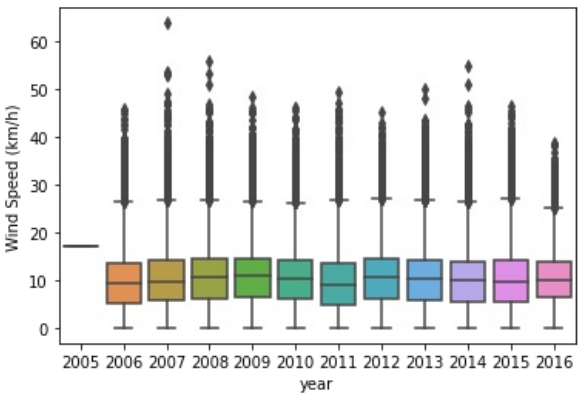


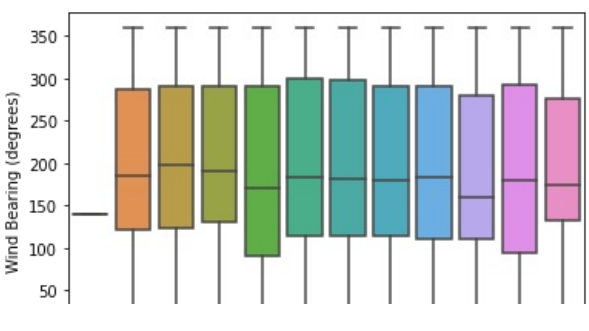

snow
2006
2008
2010
2012
2014
2016
year

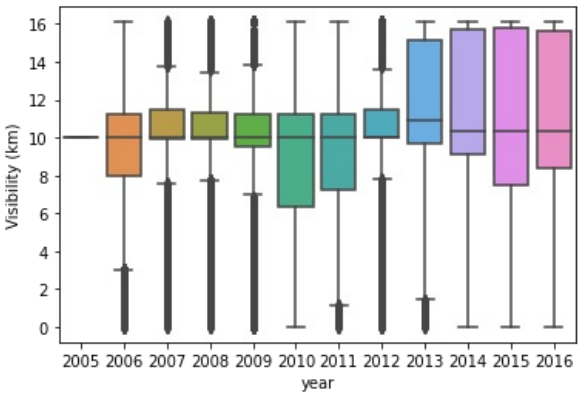

Temperature (C)
40
30
20
10
0
−10
−20
2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016
year

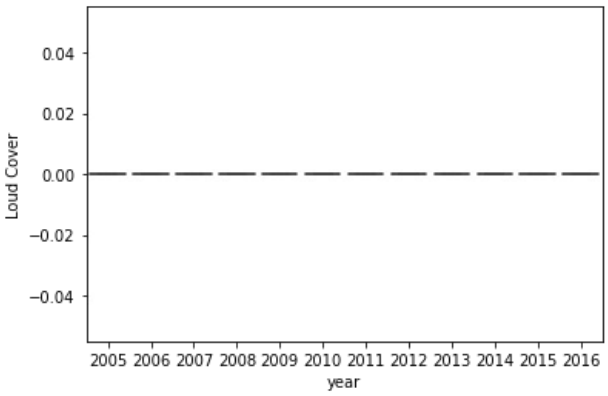

Apparent Temperature (C)
40
30
20
10
0
−10
−20
−30
2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016
year

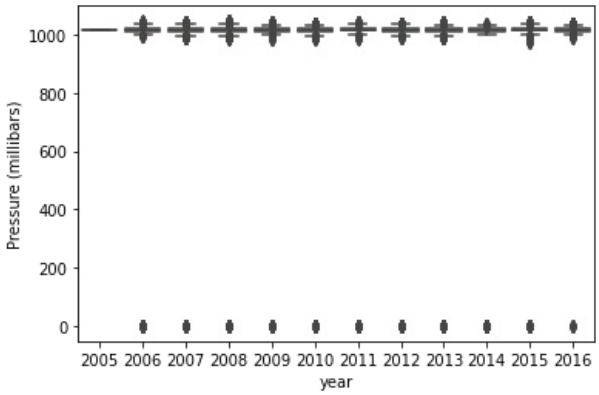

Humidity
1.0
0.8
0.6
0.4
0.2
0.0
2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016
year


Wind Speed (km/h)
60
50
40
30
20
10
0
2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016
year


Wind Bearing (degrees)
350
300
250
200
150
100
50







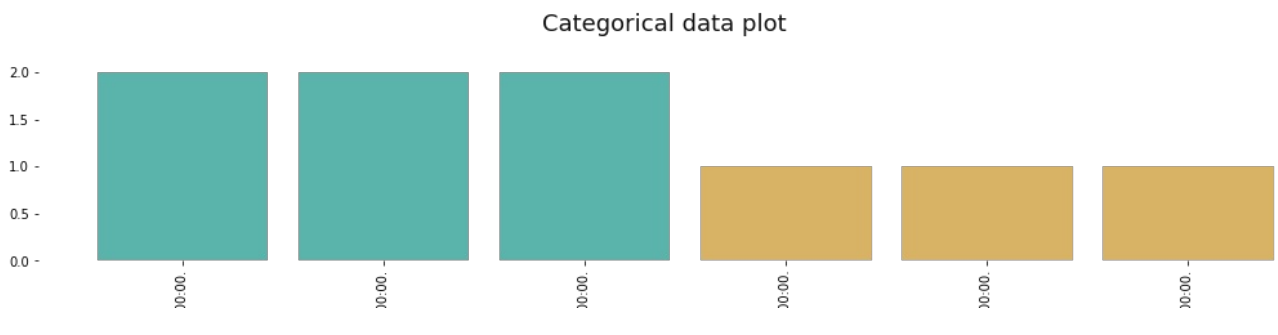
Loading Libraries and Dataset

In [1]:
```
import pandas as pd
import klib
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

In [2]:
```
data = pd.read_csv('weatherHistory.csv')
```

In [76]:
```
klib.cat_plot(data)
```

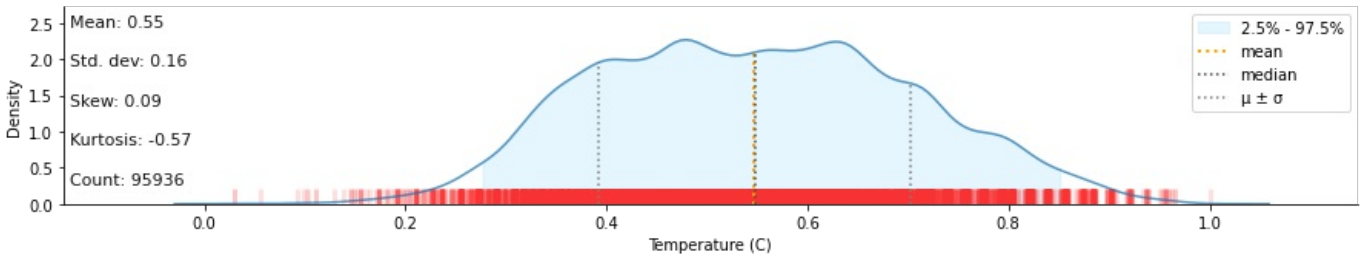
Out[76]:
```
GridSpec(6, 1)
```

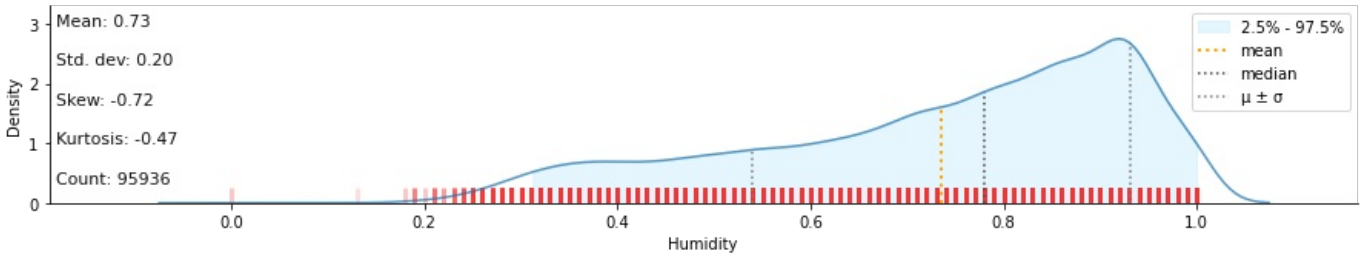


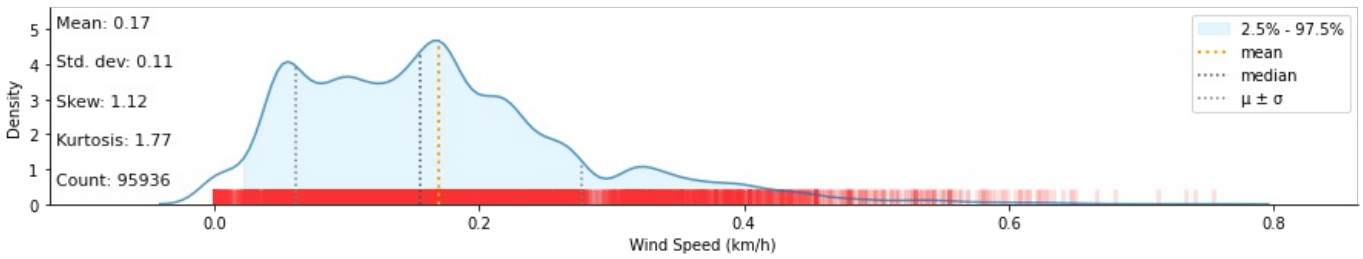
In [78]:
```
klib.dist_plot(data)
```

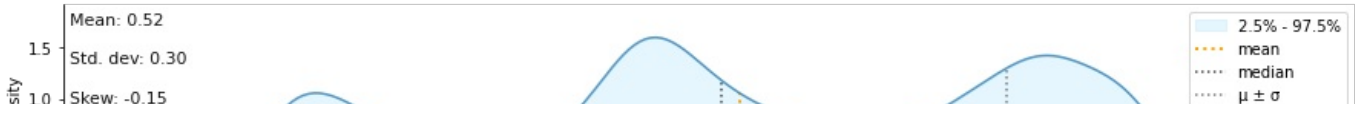
```
Large dataset detected, using 10000 random samples for the plots. Summary statistics are still based on the entir
e dataset.
```

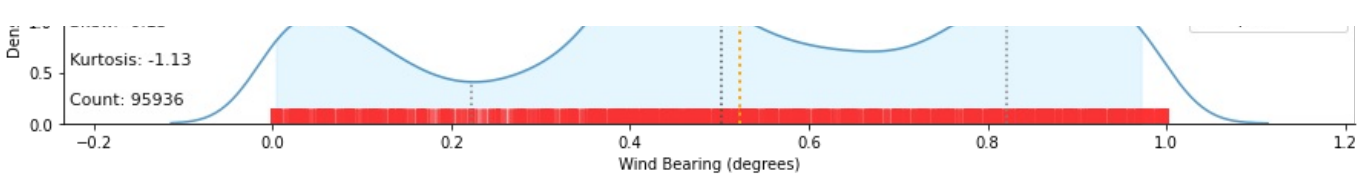
Out[78]: <AxesSubplot:xlabel='Summary', ylabel='Density'>

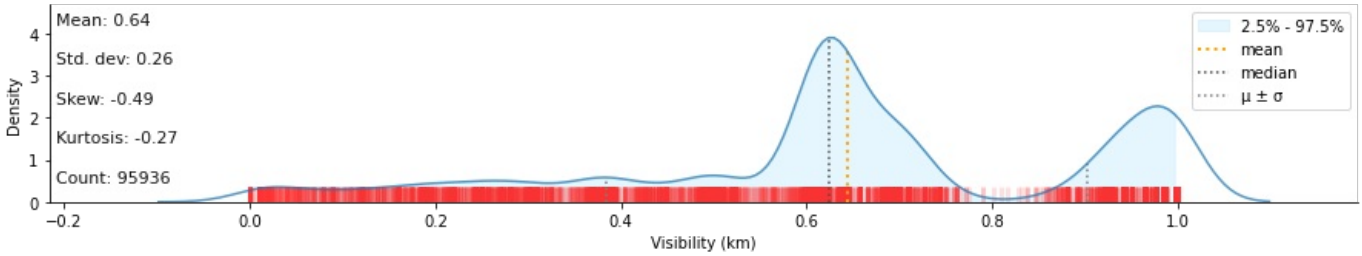


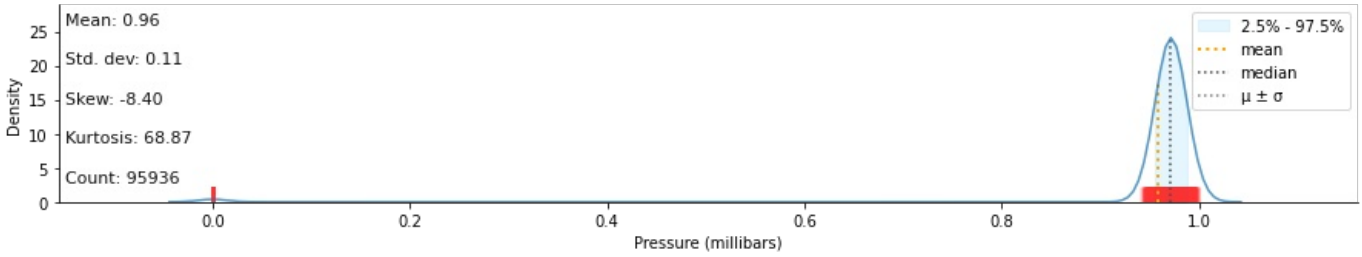


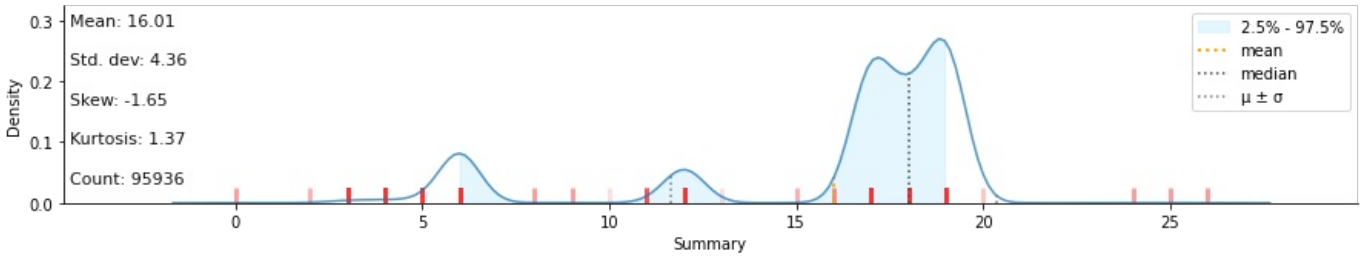






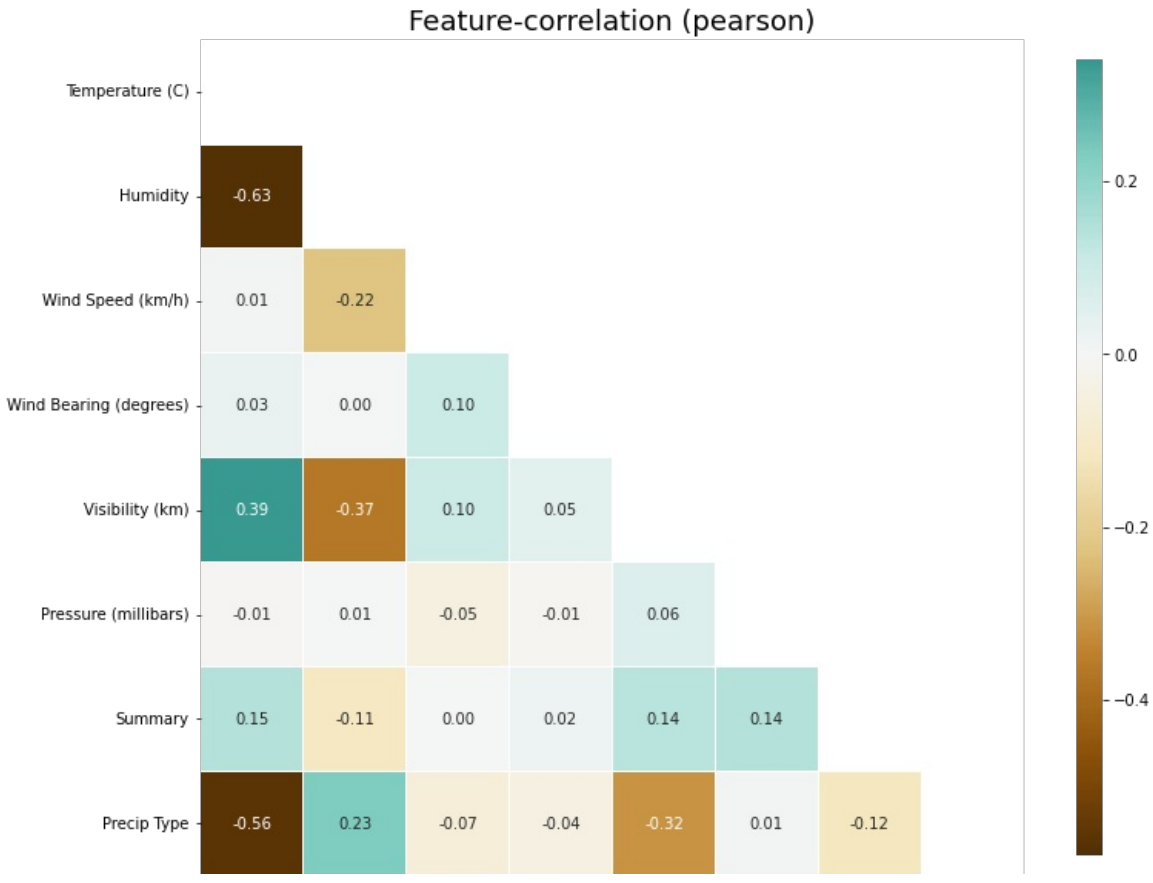






In [79]:
```
klib.corr_plot(data)
```

Out[79]: <AxesSubplot:title={'center':'Feature-correlation (pearson)'}>



Normalization And Stadardization

In [3]:
```
x=data[[col for col in data.columns if data[col].dtype ==float and col not in ['Apparent Temperature (C)','Loud (
x= (x-x.min())/(x.max()-x.min())
data = pd.concat([x,data[['Summary', 'Precip Type','Formatted Date']]],axis=1)
```

In [7]:
```
data['Formatted Time'] = pd.to_datetime(data['Formatted Date'],utc=True)
```

In [13]:
```
[col for col in data.columns if data[col].dtypes =='O']
```

Out[13]:
```
['Precip Type', 'Formatted Date']
```

Imputation

In [16]:
```
def label(x):
    if x=='snow':
        return 2
    elif x=='rain':
        return 1
    return x
```

In [20]:
```
data['Precip Type'] = data['Precip Type'].apply(label)
```

In [19]:
```
data['Precip Type'].value_counts()
```

Out[19]:
```
rain    85224
snow    10712
Name: Precip Type, dtype: int64
```

In [42]:
```
data['Precip Type']
```

Out[42]:
```
0        1.0
1        1.0
2        1.0
3        1.0
4        1.0
        ...
96448    1.0
96449    1.0
96450    1.0
96451    1.0
96452    1.0
Name: Precip Type, Length: 96453, dtype: float64
```

In [46]:
```
data ['Date'] = data['Formatted Time'].dt.day
data['Year']=data['Formatted Time'].dt.year
data['Month']=data['Formatted Time'].dt.month
data['Time']=data['Formatted Time'].dt.time
```

In [70]:
```
from statistics import mode
df= pd.pivot_table(data, values = 'Precip Type', index=['Month','Date','Time'], columns = 'Year').reset_index()
del df
```

In [66]:
```
## Complex Filter which uses Neighboring Date Time Of different years to impute values
for month in data['Month'][data['Precip Type'].isnull()]:
    #First Choose Month
    for day in data['Date'].loc[(data['Month']==month)&(data['Precip Type'].isnull())]:
        #Than The Day, like 1st
        for time in data['Time'].loc[(data['Month']==month)&(data['Precip Type'].isnull())&(data['Date']==day)]:
            #Than Time
            data.loc[(data['Month']==month)&(data['Date']==day)&(data['Time']==time)&(data['Precip Type'].isnull(
            #Than Write the Query To Get all the Same Date, Month And Time but with Different Years(which have Va
```

In [71]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96453 entries, 0 to 96452
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Temperature (C)         95936 non-null  float64
 1   Humidity                95936 non-null  float64
 2   Wind Speed (km/h)       95936 non-null  float64
 3   Wind Bearing (degrees)  95936 non-null  float64
 4   Visibility (km)         95936 non-null  float64
 5   Pressure (millibars)    95936 non-null  float64
 6   Summary                 95936 non-null  float64
 7   Precip Type             95936 non-null  float64
 8   Formatted Date          95936 non-null  object 
 9   Formatted Time          95936 non-null  object 
 10  Date                    95936 non-null  float64
 11  Year                    95936 non-null  float64
 12  Month                   95936 non-null  float64
 13  Time                    95936 non-null  object 
dtypes: float64(11), object(3)
memory usage: 10.3+ MB
```

In [72]:
```
data.drop(columns = ['Year','Month','Time','Date'],axis=1,inplace=True)
```

In [74]:
```
data.drop(columns=['Formatted Time'],axis=1,inplace=True)
```

In [75]:
```
data.columns
```

Out[75]:
```
Index(['Temperature (C)', 'Humidity', 'Wind Speed (km/h)',
       'Wind Bearing (degrees)', 'Visibility (km)', 'Pressure (millibars)',
       'Summary', 'Precip Type', 'Formatted Date'],
      dtype='object')
```

4. Data Preparation

a. Do the final feature selection and extract them into Column X and the class label into Column into Y.

b. Split the dataset into training and test sets.

In [88]:
```
pd.concat([data,data.shift(1)],axis=1)
X= data[[col for col in data.columns if col not in ['Formatted Date','Humidity']]]
x=X.copy()
shift_list=[1,3,7,14,30,60]
for shift in shift_list:
    new_x= X.shift(shift)
    for col in new_x:
        new_x.rename(columns = {col:col +'_shift_by_'+str(shift)+'_days'}, inplace = True)
    x=pd.concat([x,new_x],axis=1)
print(x)
```

```
       Temperature (C)  Wind Speed (km/h)  Wind Bearing (degrees)  \
0             0.506975           0.221130                0.699164   
1             0.505085           0.223399                0.721448   
2             0.505445           0.061523                0.568245   
3             0.487805           0.220877                0.749304   
4             0.495365           0.172970                0.721448   
...                ...                ...                     ...   
96448         0.774998           0.172214                0.086351   
96449         0.751778           0.158094                0.055710   
96450         0.710557           0.140696                0.083565   
96451         0.702187           0.164902                0.055710   
96452         0.684637           0.092032                0.108635   

       Visibility (km)  Pressure (millibars)  Summary  Precip Type  \
0                0.983              0.970135     19.0          1.0   
1                0.983              0.970613     19.0          1.0   
2                0.929              0.970909     17.0          1.0   
3                0.983              0.971358     19.0          1.0   
4                0.983              0.971454     17.0          1.0   
...                ...                   ...      ...          ...   
96448            1.000              0.969399     19.0          1.0   
96449            0.966              0.970164     19.0          1.0   
96450            1.000              0.970642     19.0          1.0   
```

```
96451           1.000             0.970919     19.0          1.0
96452           0.964             0.971119     19.0          1.0

       Temperature (C)_shift_by_1_days  Wind Speed (km/h)_shift_by_1_days  \
0                                  NaN                                NaN
1                             0.506975                           0.221130
2                             0.505085                           0.223399
3                             0.505445                           0.061523
4                             0.487805                           0.220877
...                                ...                                ...
96448                         0.820718                           0.191125
96449                         0.774998                           0.172214
96450                         0.751778                           0.158094
96451                         0.710557                           0.140696
96452                         0.702187                           0.164902

       Wind Bearing (degrees)_shift_by_1_days  ...  \
0                                         NaN  ...
1                                    0.699164  ...
2                                    0.721448  ...
3                                    0.568245  ...
4                                    0.749304  ...
...                                       ...  ...
96448                                0.058496  ...
96449                                0.086351  ...
96450                                0.055710  ...
96451                                0.083565  ...
96452                                0.055710  ...

       Pressure (millibars)_shift_by_30_days  Summary_shift_by_30_days  \
0                                        NaN                       NaN
1                                        NaN                       NaN
2                                        NaN                       NaN
3                                        NaN                       NaN
4                                        NaN                       NaN
...                                      ...                       ...
96448                               0.971358                      19.0
96449                               0.970737                      19.0
96450                               0.970231                      19.0
96451                               0.969925                      19.0
96452                               0.969409                      19.0

       Precip Type_shift_by_30_days  Temperature (C)_shift_by_60_days  \
0                               NaN                               NaN
1                               NaN                               NaN
2                               NaN                               NaN
3                               NaN                               NaN
4                               NaN                               NaN
...                             ...                               ...
96448                           1.0                          0.631446
96449                           1.0                          0.642066
96450                           1.0                          0.677167
96451                           1.0                          0.721447
96452                           1.0                          0.740437

       Wind Speed (km/h)_shift_by_60_days  \
0                                     NaN
1                                     NaN
2                                     NaN
3                                     NaN
4                                     NaN
...                                   ...
96448                            0.202723
96449                            0.190368
96450                            0.179022
96451                            0.174231
96452                            0.163641

       Wind Bearing (degrees)_shift_by_60_days  \
0                                          NaN
1                                          NaN
2                                          NaN
3                                          NaN
4                                          NaN
...                                        ...
96448                                 0.058496
96449                                 0.061281
96450                                 0.089136
96451                                 0.089136
96452                                 0.192201

       Visibility (km)_shift_by_60_days  \
0                                   NaN
```

```
1                                  NaN
2                                  NaN
3                                  NaN
4                                  NaN
...                                ...
96448                            0.920
96449                            0.853
96450                            0.744
96451                            0.983
96452                            0.941

       Pressure (millibars)_shift_by_60_days  Summary_shift_by_60_days  \
0                                        NaN                       NaN
1                                        NaN                       NaN
2                                        NaN                       NaN
3                                        NaN                       NaN
4                                        NaN                       NaN
...                                      ...                       ...
96448                               0.976099                      19.0
96449                               0.976089                      19.0
96450                               0.976079                      19.0
96451                               0.976089                      19.0
96452                               0.976079                      19.0

       Precip Type_shift_by_60_days
0                               NaN
1                               NaN
2                               NaN
3                               NaN
4                               NaN
...                             ...
96448                           1.0
96449                           1.0
96450                           1.0
96451                           1.0
96452                           1.0

[96453 rows x 49 columns]
```

In [90]:
```
[col for col in x.columns if 'Summary' in col]
```

Out[90]:
```
['Summary',
 'Summary_shift_by_1_days',
 'Summary_shift_by_3_days',
 'Summary_shift_by_7_days',
 'Summary_shift_by_14_days',
 'Summary_shift_by_30_days',
 'Summary_shift_by_60_days']
```

In [93]:
```
x.drop([col for col in x.columns if 'Summary' in col and len(col)>len('Summary')],axis=1,inplace=True)
```

In [97]:
```
data= pd.concat([x,data['Humidity']],axis=1)
data.dropna(inplace=True)
```

In [99]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
data['Summary'] = le.fit_transform(data['Summary'])
```

In [100…
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 95323 entries, 60 to 96452
Data columns (total 44 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Temperature (C)                         95323 non-null  float64
 1   Wind Speed (km/h)                       95323 non-null  float64
 2   Wind Bearing (degrees)                  95323 non-null  float64
 3   Visibility (km)                         95323 non-null  float64
 4   Pressure (millibars)                    95323 non-null  float64
 5   Summary                                 95323 non-null  int64
 6   Precip Type                             95323 non-null  float64
 7   Temperature (C)_shift_by_1_days         95323 non-null  float64
 8   Wind Speed (km/h)_shift_by_1_days       95323 non-null  float64
 9   Wind Bearing (degrees)_shift_by_1_days  95323 non-null  float64
 10  Visibility (km)_shift_by_1_days         95323 non-null  float64
 11  Pressure (millibars)_shift_by_1_days    95323 non-null  float64
 12  Precip Type_shift_by_1_days             95323 non-null  float64
```

```
 13  Temperature (C)_shift_by_3_days           95323 non-null  float64
 14  Wind Speed (km/h)_shift_by_3_days         95323 non-null  float64
 15  Wind Bearing (degrees)_shift_by_3_days    95323 non-null  float64
 16  Visibility (km)_shift_by_3_days           95323 non-null  float64
 17  Pressure (millibars)_shift_by_3_days      95323 non-null  float64
 18  Precip Type_shift_by_3_days               95323 non-null  float64
 19  Temperature (C)_shift_by_7_days           95323 non-null  float64
 20  Wind Speed (km/h)_shift_by_7_days         95323 non-null  float64
 21  Wind Bearing (degrees)_shift_by_7_days    95323 non-null  float64
 22  Visibility (km)_shift_by_7_days           95323 non-null  float64
 23  Pressure (millibars)_shift_by_7_days      95323 non-null  float64
 24  Precip Type_shift_by_7_days               95323 non-null  float64
 25  Temperature (C)_shift_by_14_days          95323 non-null  float64
 26  Wind Speed (km/h)_shift_by_14_days        95323 non-null  float64
 27  Wind Bearing (degrees)_shift_by_14_days   95323 non-null  float64
 28  Visibility (km)_shift_by_14_days          95323 non-null  float64
 29  Pressure (millibars)_shift_by_14_days     95323 non-null  float64
 30  Precip Type_shift_by_14_days              95323 non-null  float64
 31  Temperature (C)_shift_by_30_days          95323 non-null  float64
 32  Wind Speed (km/h)_shift_by_30_days        95323 non-null  float64
 33  Wind Bearing (degrees)_shift_by_30_days   95323 non-null  float64
 34  Visibility (km)_shift_by_30_days          95323 non-null  float64
 35  Pressure (millibars)_shift_by_30_days     95323 non-null  float64
 36  Precip Type_shift_by_30_days              95323 non-null  float64
 37  Temperature (C)_shift_by_60_days          95323 non-null  float64
 38  Wind Speed (km/h)_shift_by_60_days        95323 non-null  float64
 39  Wind Bearing (degrees)_shift_by_60_days   95323 non-null  float64
 40  Visibility (km)_shift_by_60_days          95323 non-null  float64
 41  Pressure (millibars)_shift_by_60_days     95323 non-null  float64
 42  Precip Type_shift_by_60_days              95323 non-null  float64
 43  Humidity                                  95323 non-null  float64
dtypes: float64(43), int64(1)
memory usage: 32.7 MB
```

Part B (12 marks)

1. **Model Building**
   a. Perform Model Development using at least three models, separately. You are free to apply any Machine Learning Models on the dataset. Deep Learning Models are strictly not allowed.
   b. Train the model and print the training accuracy and loss values.

2. **Performance Evaluation**
   a. Print the confusion matrix. Provide appropriate analysis for the same.
   b. Do the prediction for the test data and display the results for the inference.

Sliding Window

Expanding Window

Expanding Window Strategy

Run-1

The Train is from 0 to 80% of dataset
The Test is from 80 to 100%

In [104…
```
data_train = data[:int(0.8*(len(data)))]
data_test = data[int(0.8*(len(data))):]
```

In [116…
```
from pygam import LinearGAM, s, f
```

```
gam = LinearGAM(s(0) + s(3) + f(2))
gam.gridsearch(data_train[[col for col in data_train if col!='Humidity']].values,data_train['Humidity'])
```

```
100% (11 of 11) |########################| Elapsed Time: 0:01:06 Time:  0:01:06
```

Out[116…

```
LinearGAM(callbacks=[Deviance(), Diffs()], fit_intercept=True,
   max_iter=100, scale=None, terms=s(0) + s(3) + f(2) + intercept,
   tol=0.0001, verbose=False)
```

In [119…

```
gam.summary()
```

```
LinearGAM
============================================= ============================================================
Distribution:                        NormalDist Effective DoF:                                     218.0052
Link Function:                     IdentityLink Log Likelihood:                                -2250760.9916
Number of Samples:                        76258 AIC:                                             4501959.9936
                                                AICc:                                            4501961.2609
                                                GCV:                                                0.0153
                                                Scale:                                              0.0152
                                                Pseudo R-Squared:                                   0.6084
============================================================================================================
Feature Function                  Lambda               Rank         EDoF         P > x        Sig. Code
================================= ==================== ============ ============ ============ ============
s(0)                              [0.0158]             20           19.1         1.11e-16     ***
s(3)                              [0.0158]             20           19.0         1.11e-16     ***
f(2)                              [0.0158]             360          180.0        1.11e-16     ***
intercept                                              1            0.0          1.11e-16     ***
============================================================================================================
Significance codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

WARNING: Fitting splines and a linear function to a feature introduces a model identifiability problem
         which can cause p-values to appear significant when they are not.

WARNING: p-values calculated in this manner behave correctly for un-penalized models or models with
         known smoothing parameters, but when smoothing parameters have been estimated, the p-values
         are typically lower than they should be, meaning that the tests reject the null too readily.
```

```
<ipython-input-119-dec6a6acdaaa>:1: UserWarning: KNOWN BUG: p-values computed in this summary are likely much sma
ller than they should be.

Please do not make inferences based on these values!

Collaborate on a solution, and stay up to date at:
github.com/dswah/pyGAM/issues/163

  gam.summary()
```

In [120…

```
from sklearn.metrics import mean_squared_error
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
```

```
0.016292757404520387
0.127643086003592
```

In [121…

```
lams = np.random.rand(100, 11)
lams = lams * 11 - 3
lams = np.exp(lams)
print(lams.shape)
gam = LinearGAM(n_splines=10).gridsearch(data_train[[col for col in data_train if col!='Humidity']].values,data_t
```

```
N/A% (0 of 11) |                        | Elapsed Time: 0:00:00 ETA:  --:--:--
(100, 11)
100% (11 of 11) |########################| Elapsed Time: 0:02:34 Time:  0:02:34
```

In [122…

```
gam.summary()
```

```
LinearGAM
============================================= ============================================================
Distribution:                        NormalDist Effective DoF:                                     287.6482
Link Function:                     IdentityLink Log Likelihood:                                -4039551.0448
```

```
Number of Samples:                        76258 AIC:                                    8079679.3859
                                                AICc:                                   8079681.587
                                                GCV:                                         0.0088
                                                Scale:                                       0.0088
                                                Pseudo R-Squared:                            0.7744
==========================================================================================================
Feature Function                  Lambda               Rank         EDoF         P > x        Sig. Code
================================= ==================== ============ ============ ============ ============
s(0)                              [0.001]              10           10.0         1.11e-16     ***
s(1)                              [0.001]              10           8.6          1.11e-16     ***
s(2)                              [0.001]              10           9.0          2.22e-16     ***
s(3)                              [0.001]              10           9.0          1.11e-16     ***
s(4)                              [0.001]              10           3.6          1.05e-09     ***
s(5)                              [0.001]              10           9.0          1.11e-16     ***
s(6)                              [0.001]              10           1.0          7.56e-01
s(7)                              [0.001]              10           8.9          4.23e-11     ***
s(8)                              [0.001]              10           8.5          1.11e-16     ***
s(9)                              [0.001]              10           9.0          7.00e-08     ***
s(10)                             [0.001]              10           9.0          1.11e-16     ***
s(11)                             [0.001]              10           2.9          1.11e-16     ***
s(12)                             [0.001]              10           1.0          9.04e-01
s(13)                             [0.001]              10           9.0          2.18e-08     ***
s(14)                             [0.001]              10           8.5          1.11e-16     ***
s(15)                             [0.001]              10           9.0          2.88e-11     ***
s(16)                             [0.001]              10           9.0          1.11e-16     ***
s(17)                             [0.001]              10           3.0          1.11e-16     ***
s(18)                             [0.001]              10           1.0          9.90e-01
s(19)                             [0.001]              10           9.0          1.11e-16     ***
s(20)                             [0.001]              10           8.5          1.14e-13     ***
s(21)                             [0.001]              10           9.0          2.79e-12     ***
s(22)                             [0.001]              10           9.0          1.11e-16     ***
s(23)                             [0.001]              10           3.1          1.11e-16     ***
s(24)                             [0.001]              10           1.0          4.51e-02     *
s(25)                             [0.001]              10           9.0          1.11e-16     ***
s(26)                             [0.001]              10           8.5          5.69e-12     ***
s(27)                             [0.001]              10           9.0          1.11e-16     ***
s(28)                             [0.001]              10           9.0          1.11e-16     ***
s(29)                             [0.001]              10           3.1          1.11e-16     ***
s(30)                             [0.001]              10           1.0          4.75e-01
s(31)                             [0.001]              10           9.0          1.11e-16     ***
s(32)                             [0.001]              10           8.5          9.49e-02     .
s(33)                             [0.001]              10           9.0          1.11e-16     ***
s(34)                             [0.001]              10           9.0          1.11e-16     ***
s(35)                             [0.001]              10           3.2          1.11e-16     ***
s(36)                             [0.001]              10           1.0          3.45e-02     *
s(37)                             [0.001]              10           9.0          1.11e-16     ***
s(38)                             [0.001]              10           8.5          2.73e-02     *
s(39)                             [0.001]              10           9.0          1.11e-16     ***
s(40)                             [0.001]              10           9.0          1.11e-16     ***
s(41)                             [0.001]              10           3.3          1.11e-16     ***
s(42)                             [0.001]              10           1.0          3.21e-01
intercept                                              1            0.0          1.11e-16     ***
==========================================================================================================
Significance codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

WARNING: Fitting splines and a linear function to a feature introduces a model identifiability problem
         which can cause p-values to appear significant when they are not.

WARNING: p-values calculated in this manner behave correctly for un-penalized models or models with
         known smoothing parameters, but when smoothing parameters have been estimated, the p-values
         are typically lower than they should be, meaning that the tests reject the null too readily.
```

```
<ipython-input-122-dec6a6acdaaa>:1: UserWarning: KNOWN BUG: p-values computed in this summary are likely much sma
ller than they should be.

Please do not make inferences based on these values!

Collaborate on a solution, and stay up to date at:
github.com/dswah/pyGAM/issues/163

  gam.summary()
```

In [134…

```
print('MSE is : '+ str(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),
print('RMSE is : '+ str(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']])
```

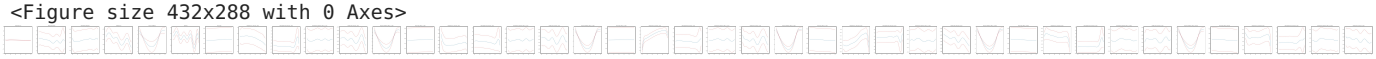
```
MSE is : 0.010126697791491363
RMSE is : 0.10063149502760735
```

In [131…

```
titles = data_train.columns[0:41]
```

```
titles = data_train.columns[0:41]
plt.figure()
fig, axs = plt.subplots(1,41,figsize=(400, 8))
for i, ax in enumerate(axs):
    XX = gam.generate_X_grid(term=i)
    ax.plot(XX[:, i], gam.partial_dependence(term=i, X=XX))
    ax.plot(XX[:, i], gam.partial_dependence(term=i, X=XX,   width=.95)[1], c='r', ls='--')
    if i == 0:
        ax.set_ylim(-30,30)
    ax.set_title(titles[i])
```

```
<Figure size 432x288 with 0 Axes>
```



2nd Run

Now the Train Dataset size increased from 80 to 90

And the Datset is from 90 to the end of dataset

In [136…

```
data_train = data[:int(0.9*(len(data)))]
data_test = data[int(0.9*(len(data))):]
lams = np.random.rand(200, 42)
lams = lams * 42 - 3
lams = np.exp(lams)
print(lams.shape)
gam = LinearGAM(n_splines=10).gridsearch(data_train[[col for col in data_train if col!='Humidity']].values,data_t
from sklearn.metrics import mean_squared_error
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
```

```
N/A% (0 of 11) |                        | Elapsed Time: 0:00:00 ETA:  --:--:--
```

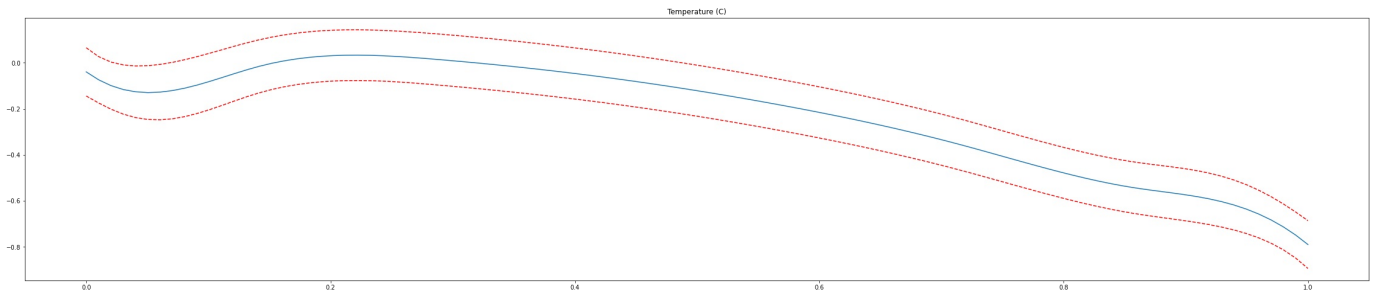
```
(200, 42)
```

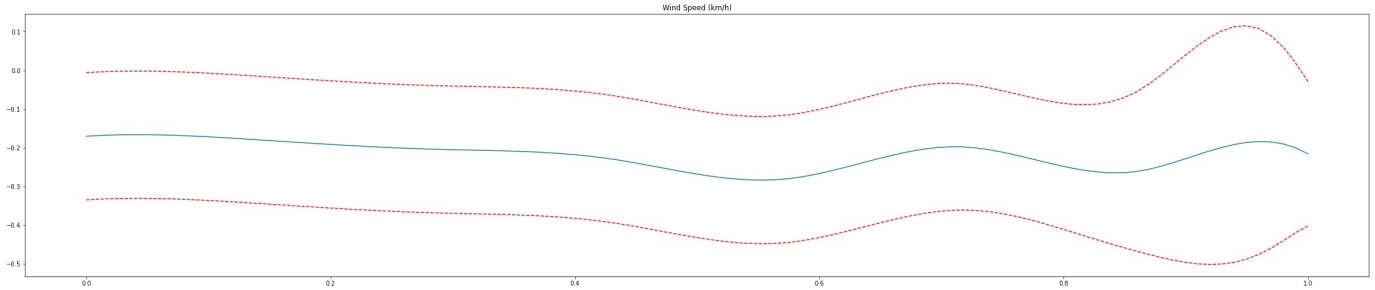
```
100% (11 of 11) |########################| Elapsed Time: 0:02:58 Time:  0:02:58
```

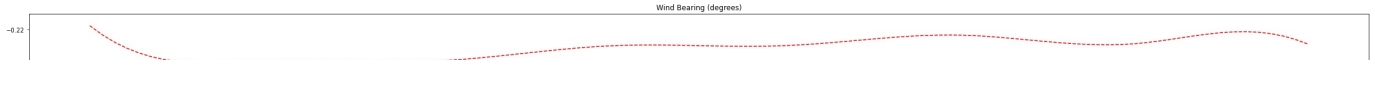
```
0.009349094752144484
0.09669071699053888
```

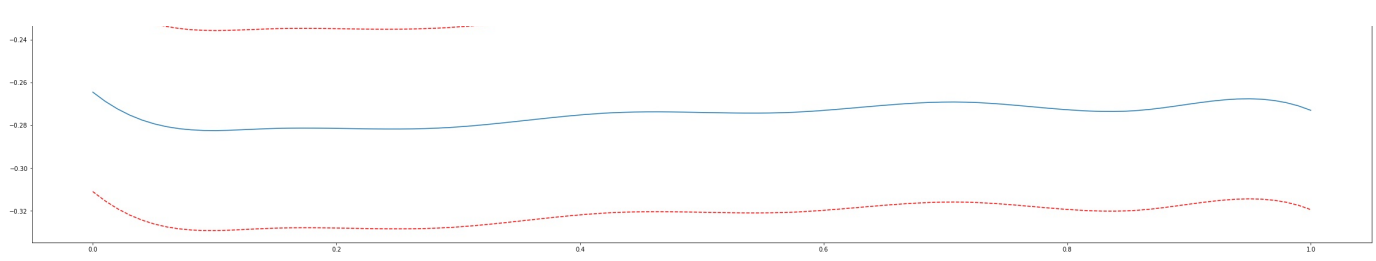
In [137…

```
titles = data_train.columns[0:41]

for i in range(0,41):
    plt.figure(figsize=(40,8))
    XX = gam.generate_X_grid(term=i)
    plt.plot(XX[:, i], gam.partial_dependence(term=i, X=XX))
    plt.plot(XX[:, i], gam.partial_dependence(term=i, X=XX,   width=.95)[1], c='r', ls='--')

    plt.title(titles[i])
    plt.show()
```

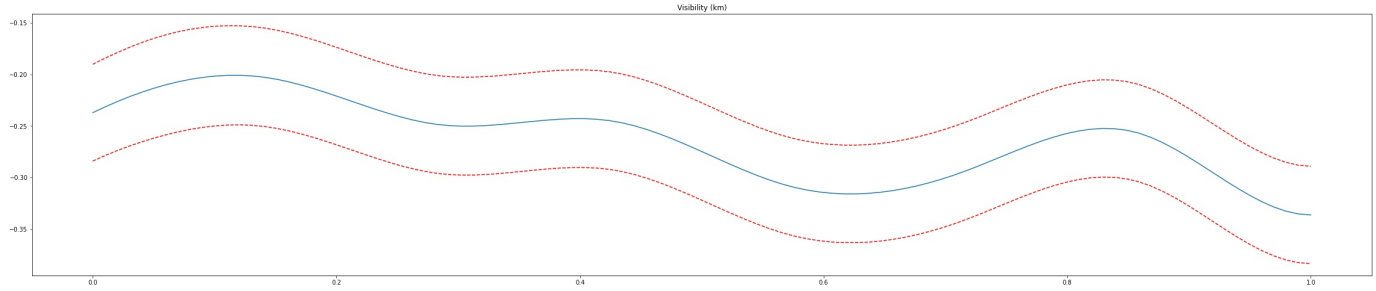




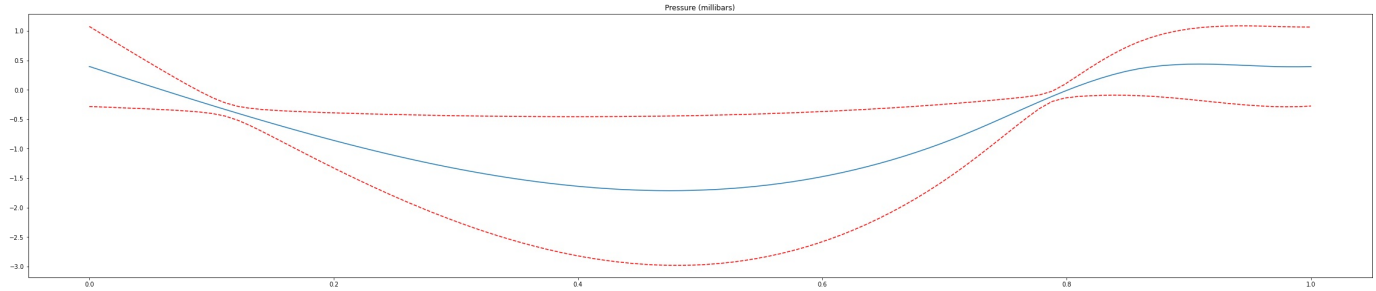




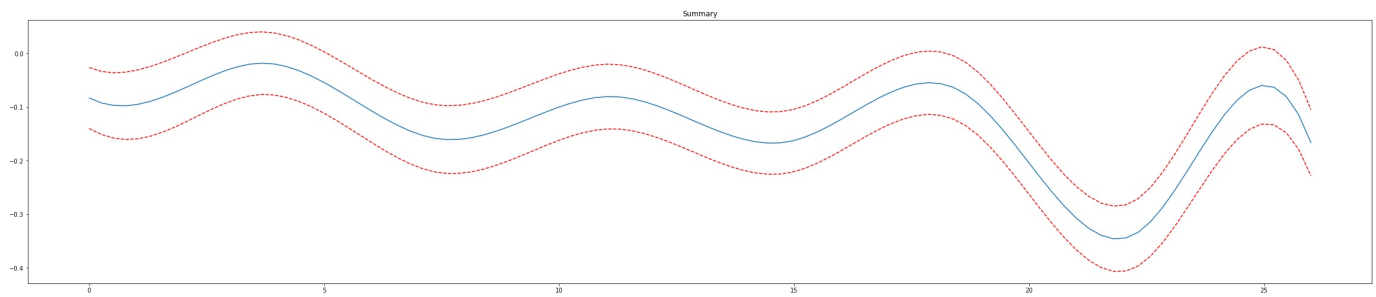
Visibility (km)



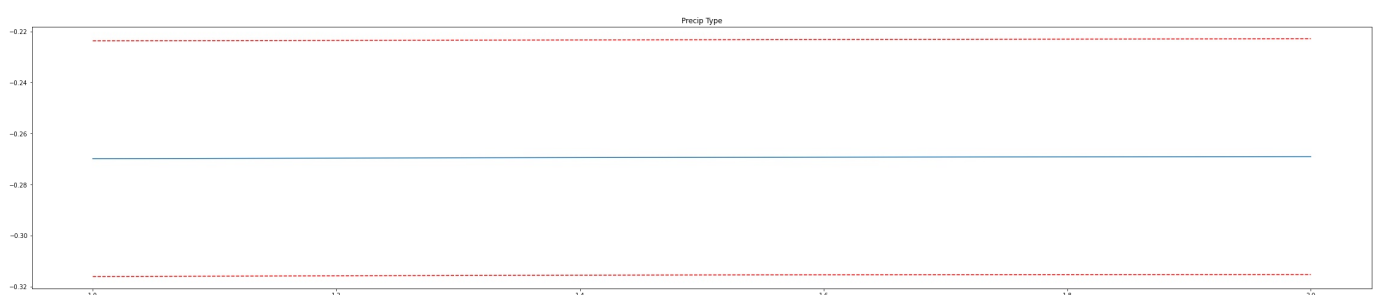
Pressure (millibars)



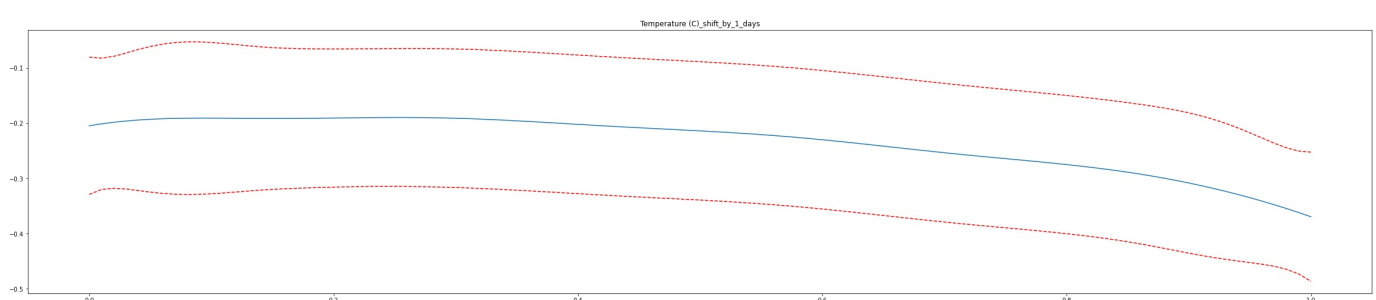
Summary



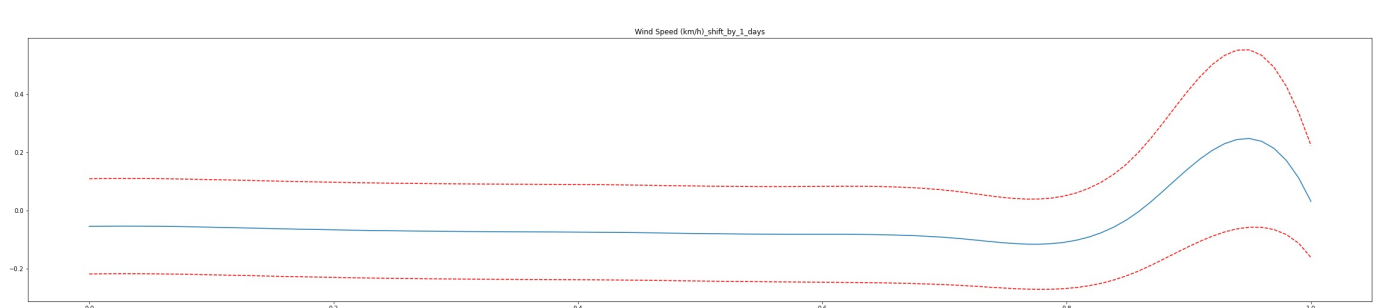
Precip Type



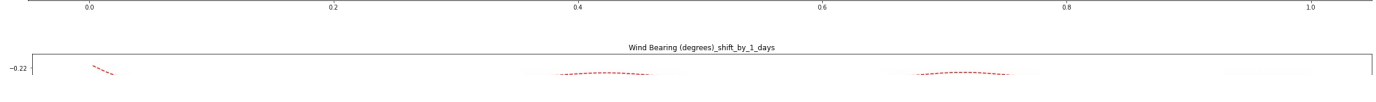
Temperature (C)_shift_by_1_days

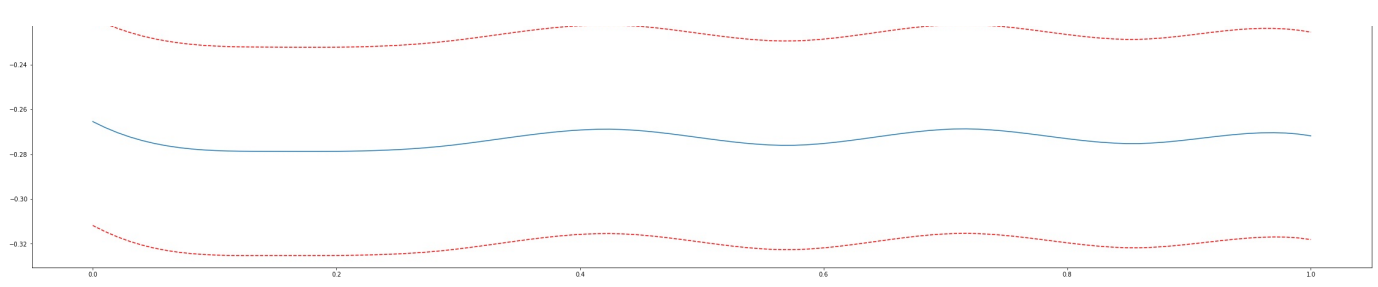


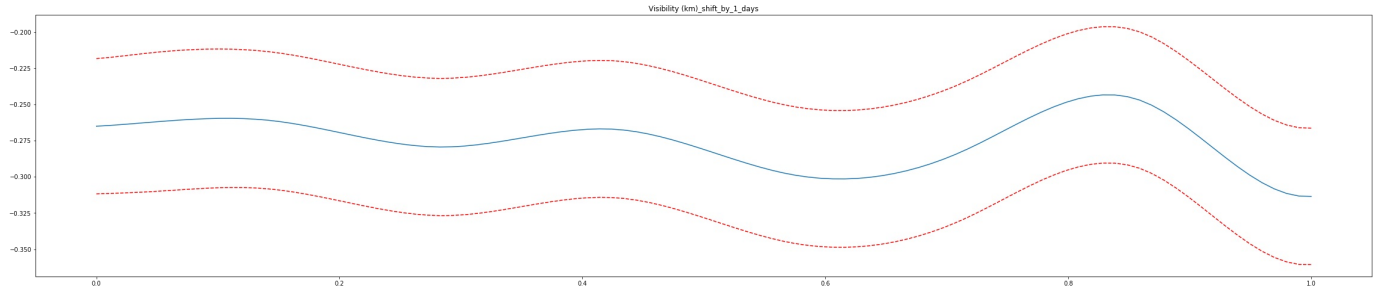
Wind Speed (km/h)_shift_by_1_days

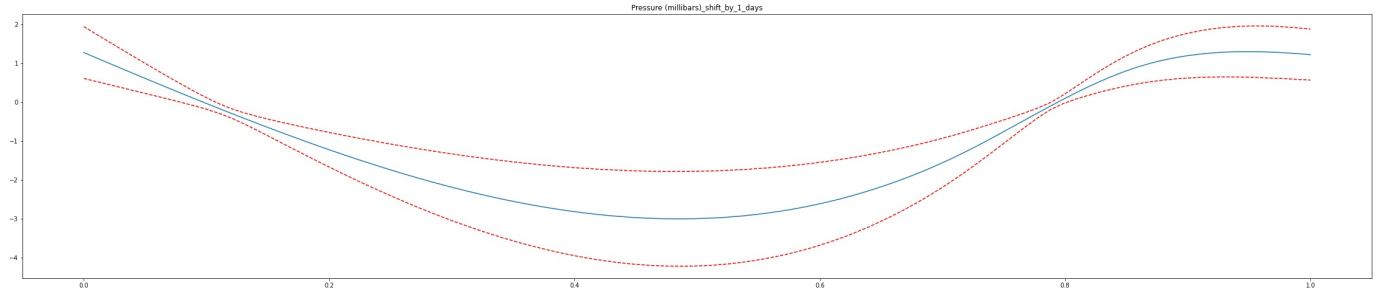


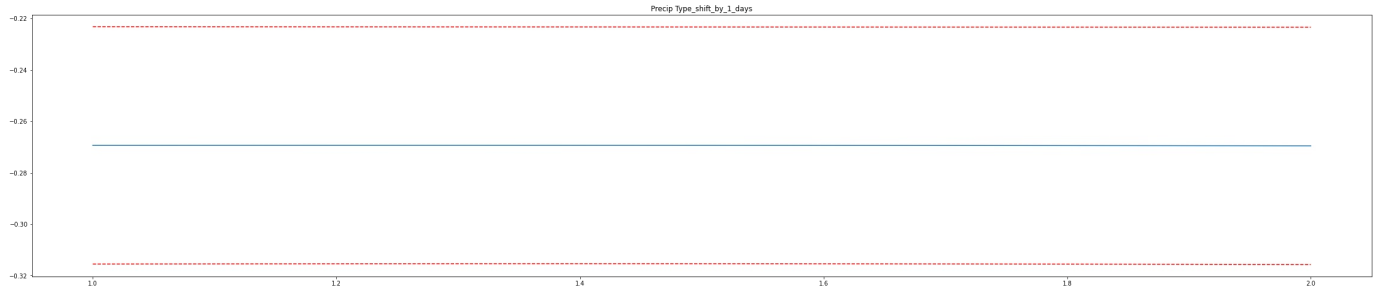
Wind Bearing (degrees)_shift_by_1_days

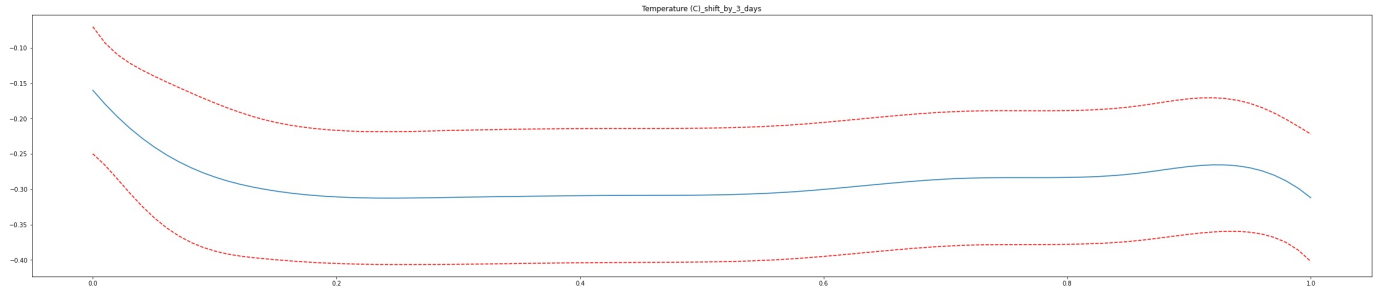


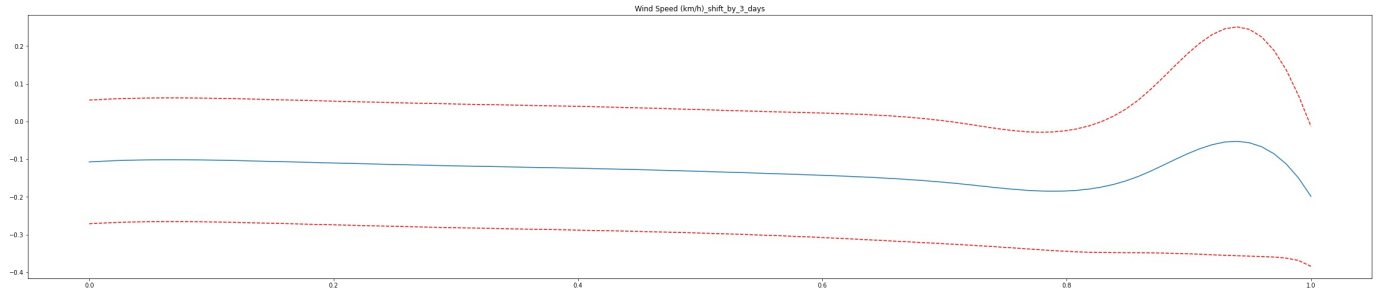


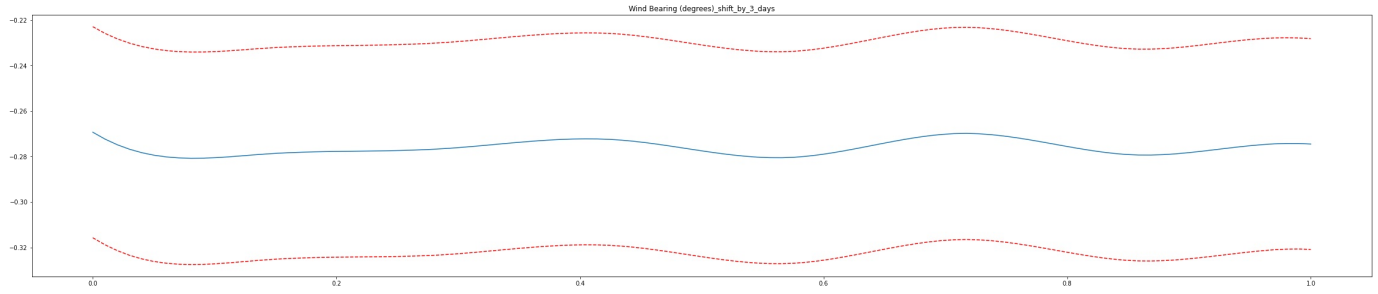




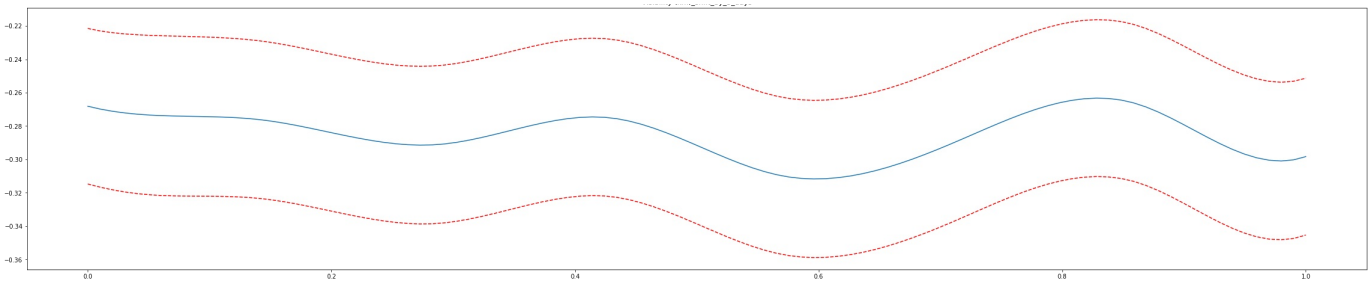


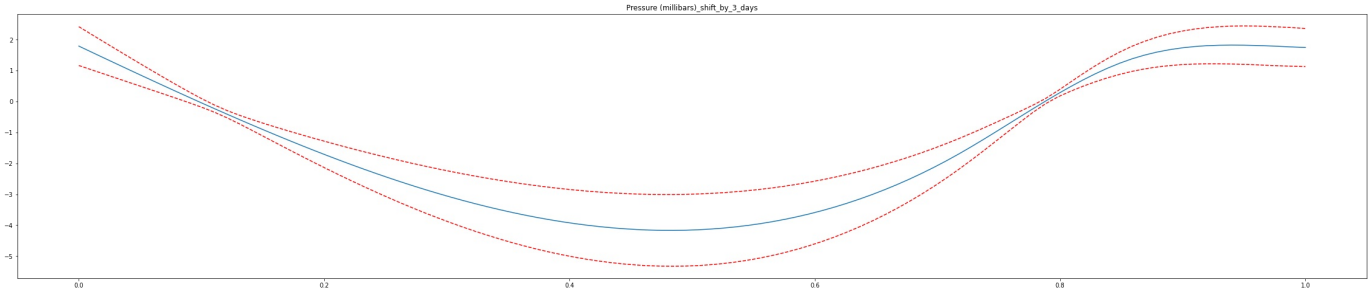


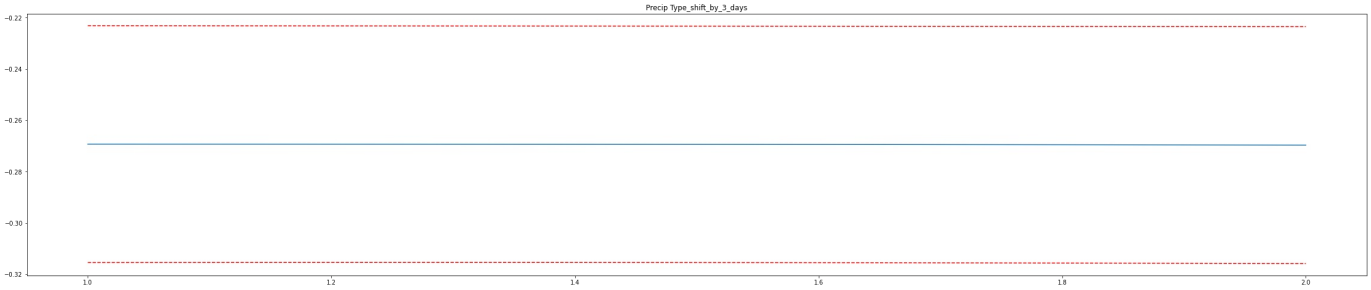


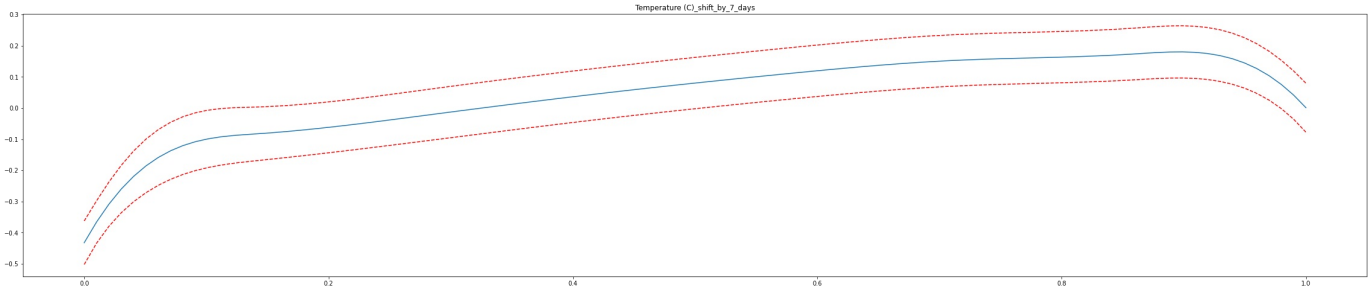


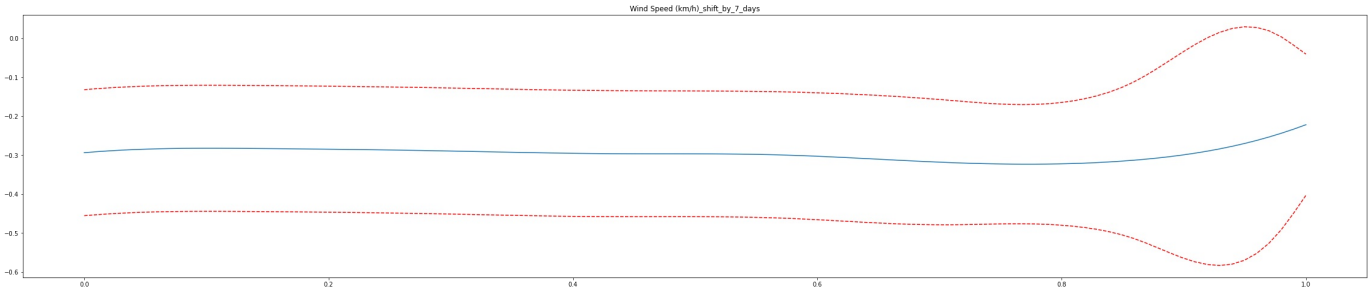
Visibility (km)_shift_by_3_days

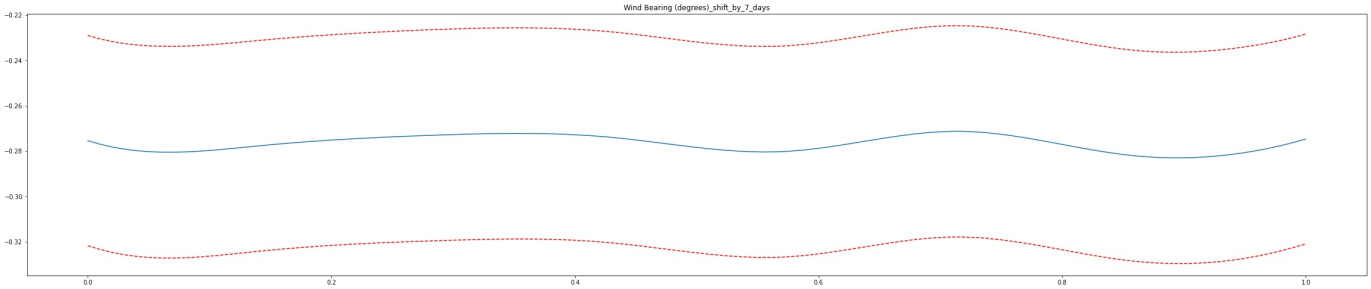


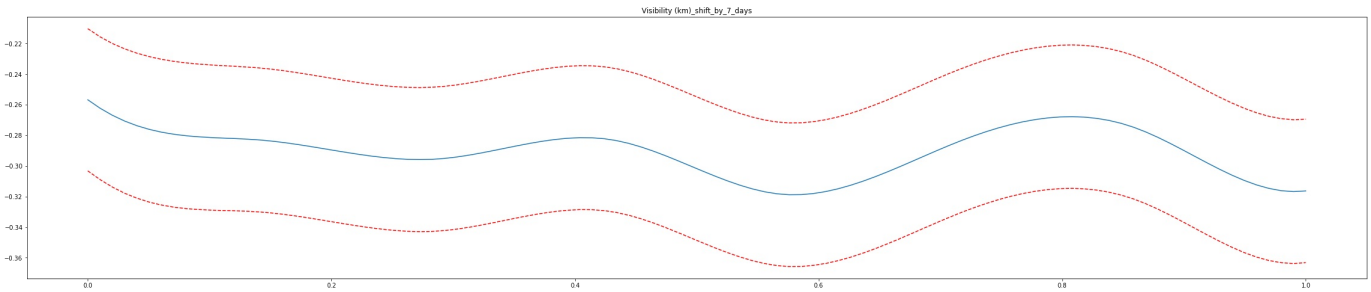

Pressure (millibars)_shift_by_3_days

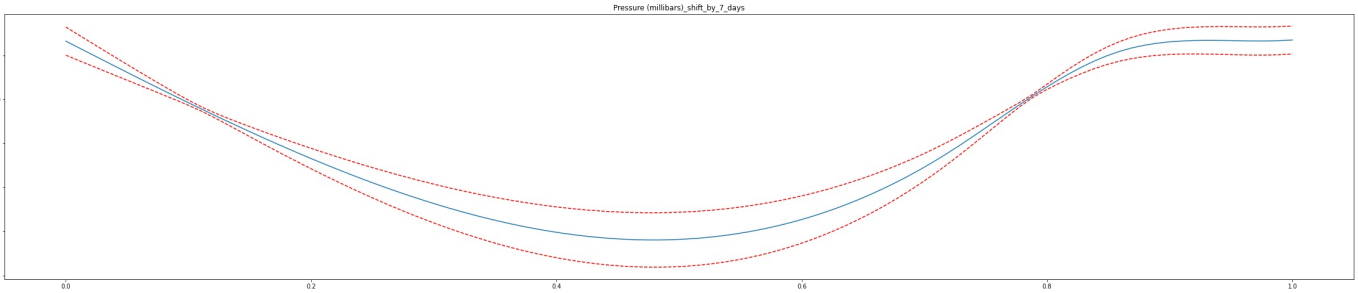

Precip Type_shift_by_3_days

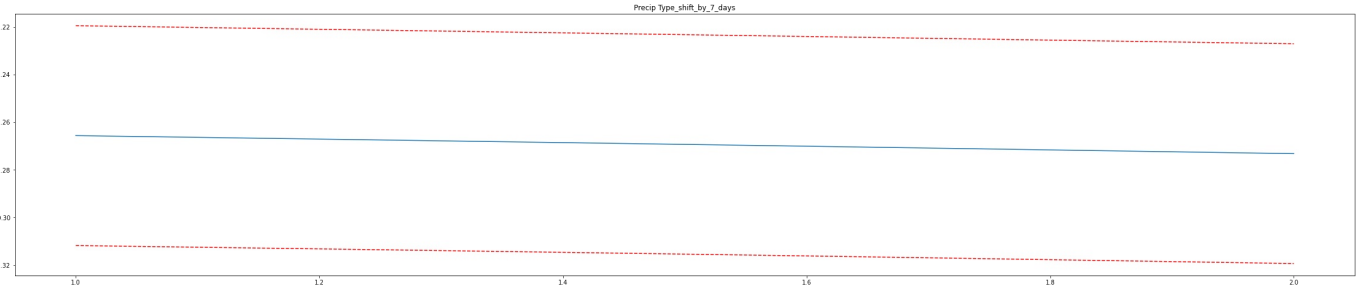

Temperature (C)_shift_by_7_days

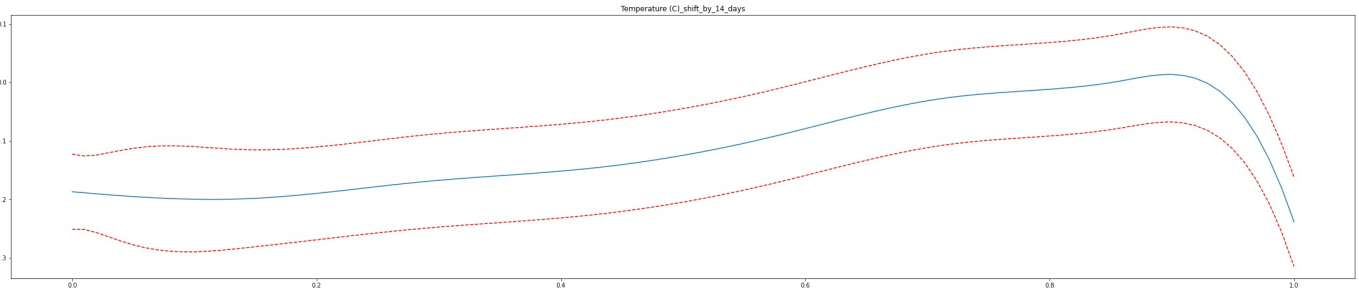

Wind Speed (km/h)_shift_by_7_days

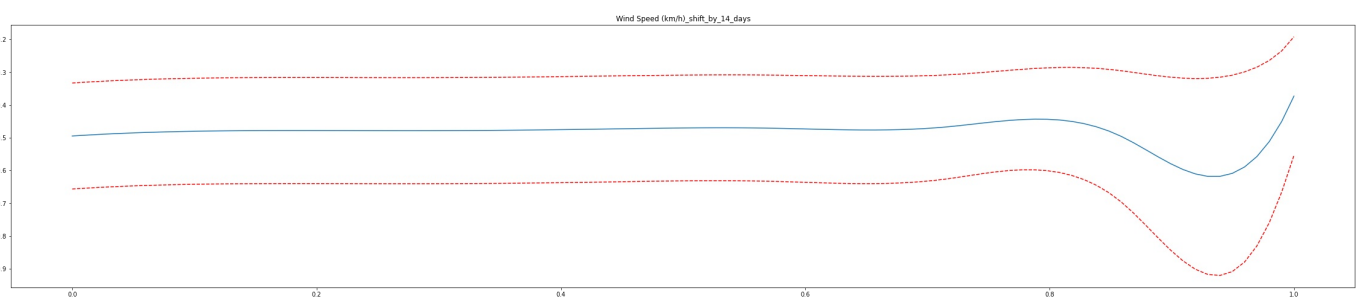

Wind Bearing (degrees)_shift_by_7_days

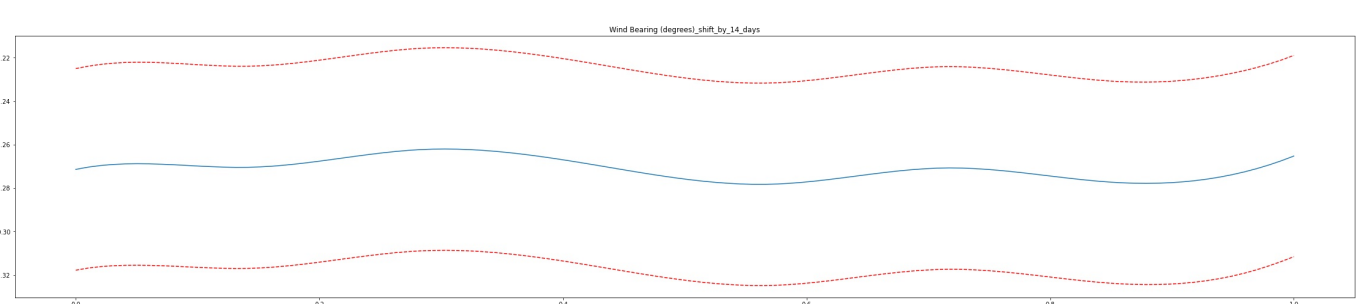

Visibility (km)_shift_by_7_days

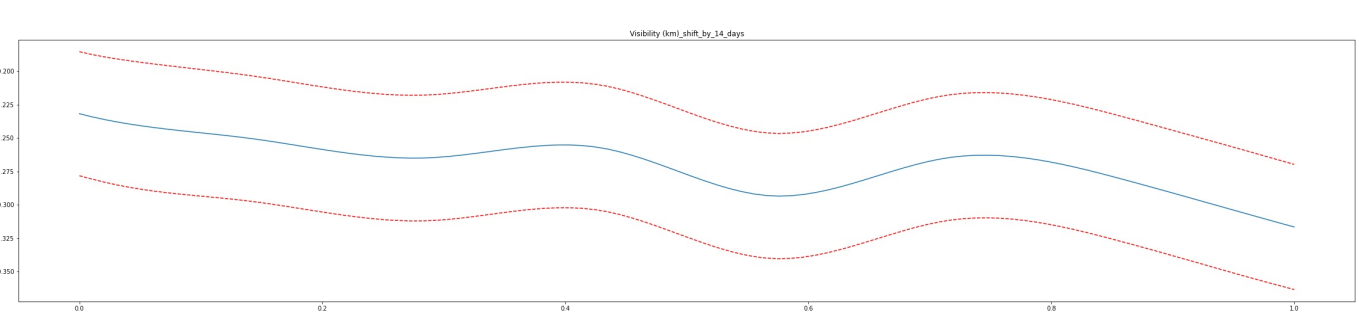

Pressure (millibars)_shift_by_7_days

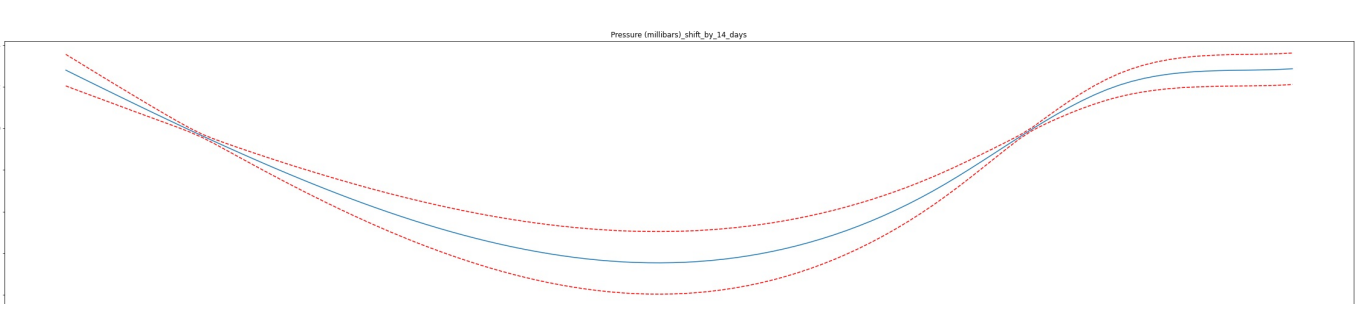

Precip Type_shift_by_7_days

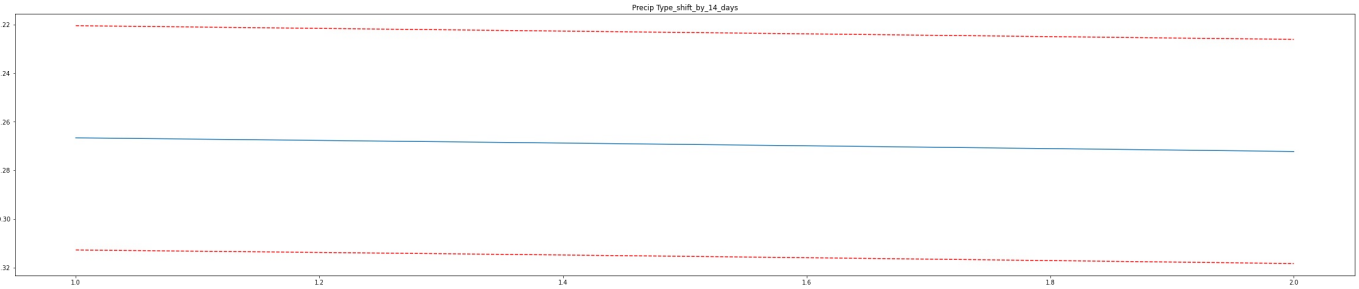

Temperature (C)_shift_by_14_days

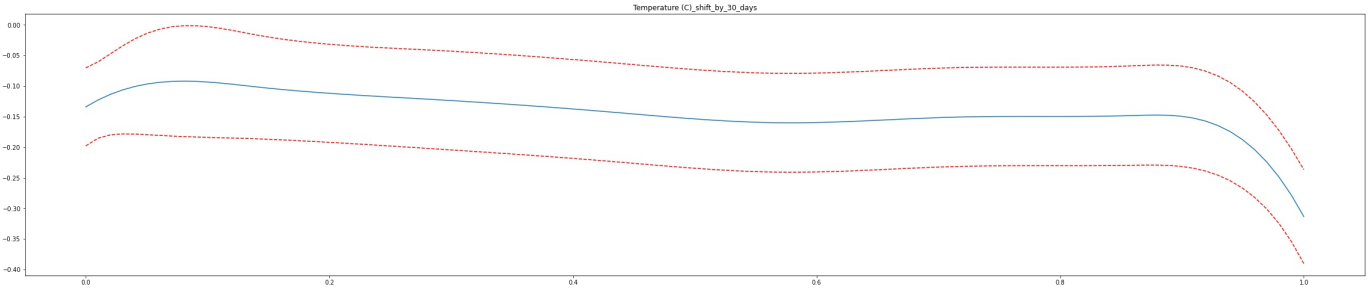

Wind Speed (km/h)_shift_by_14_days

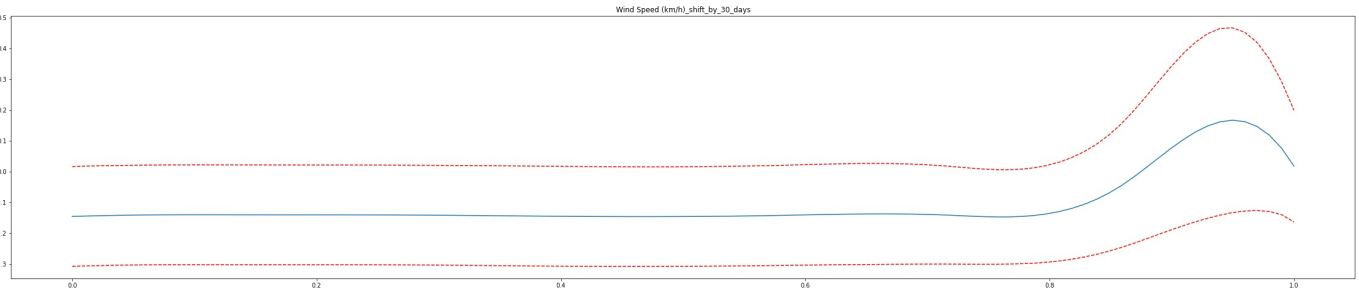

Wind Bearing (degrees)_shift_by_14_days

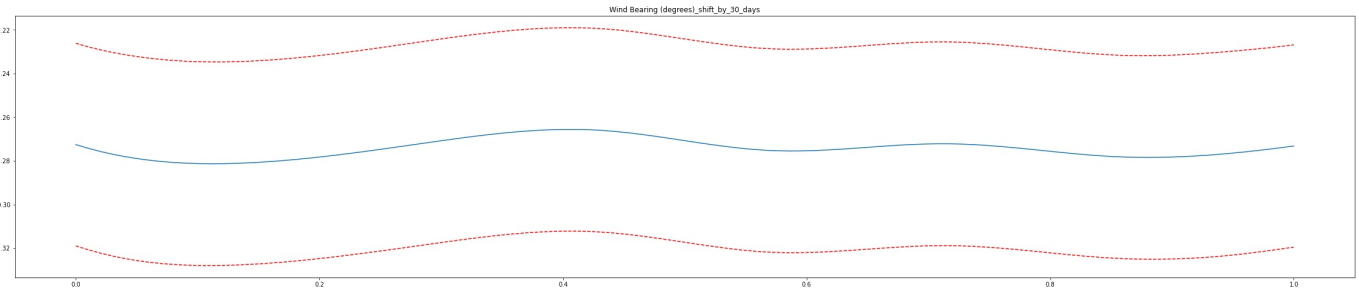

Visibility (km)_shift_by_14_days

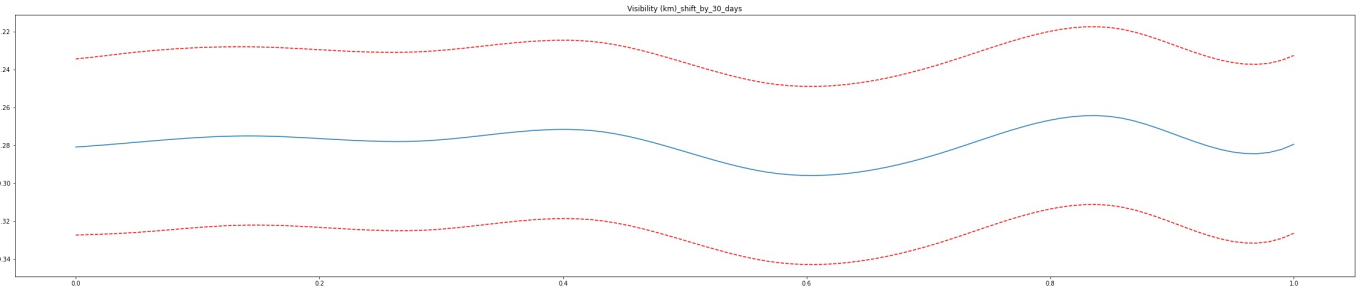

Pressure (millibars)_shift_by_14_days

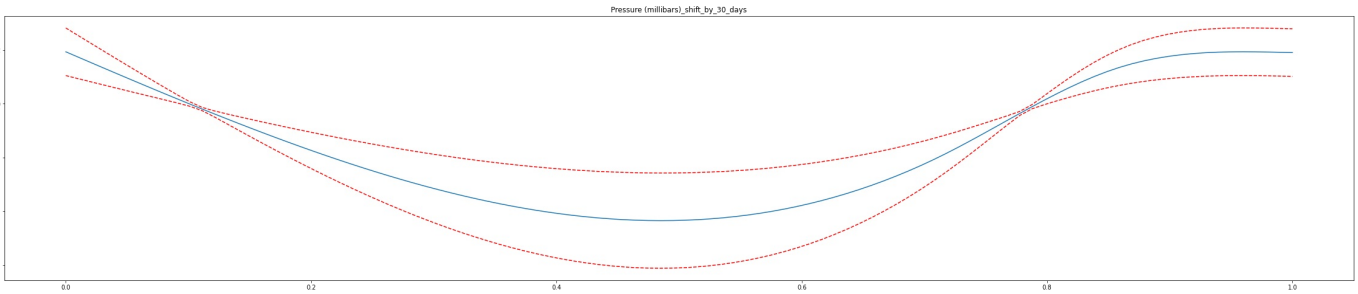

Precip Type_shift_by_14_days

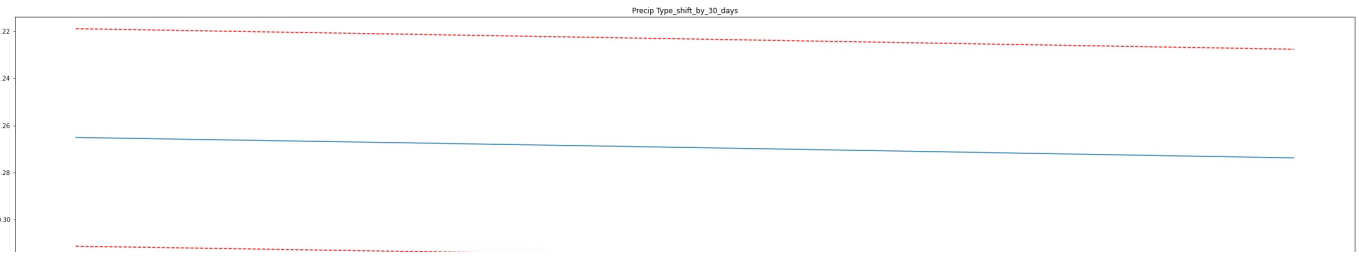

Temperature (C)_shift_by_30_days

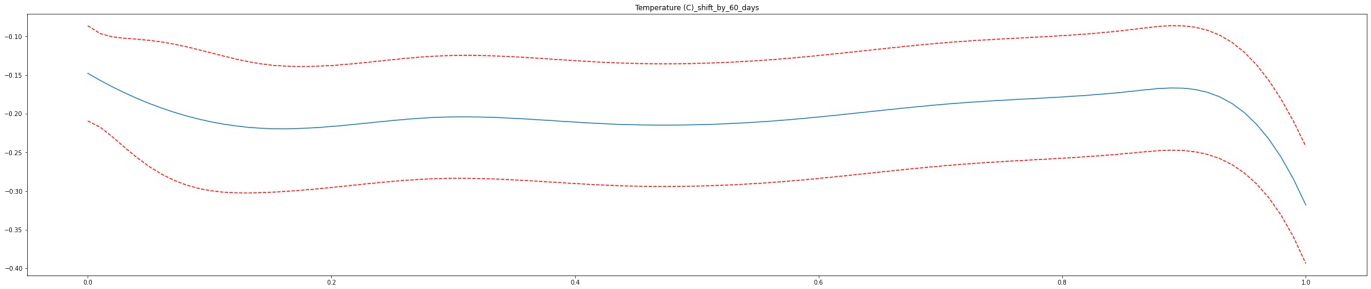

Wind Speed (km/h)_shift_by_30_days

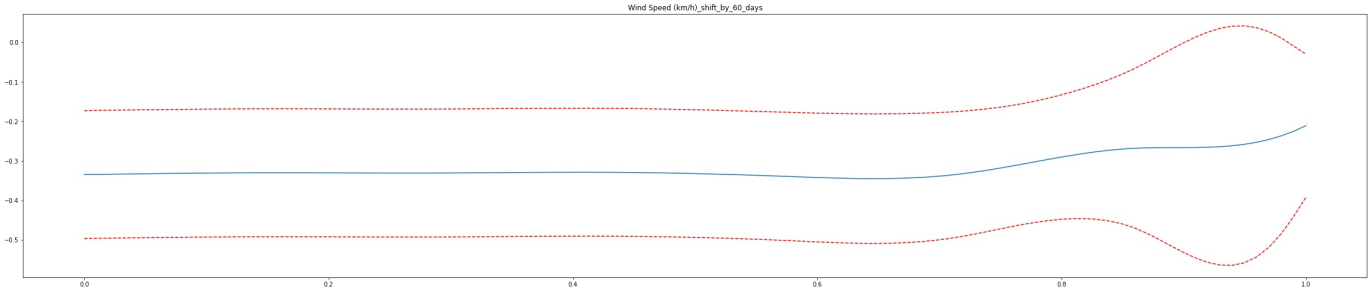

Wind Bearing (degrees)_shift_by_30_days

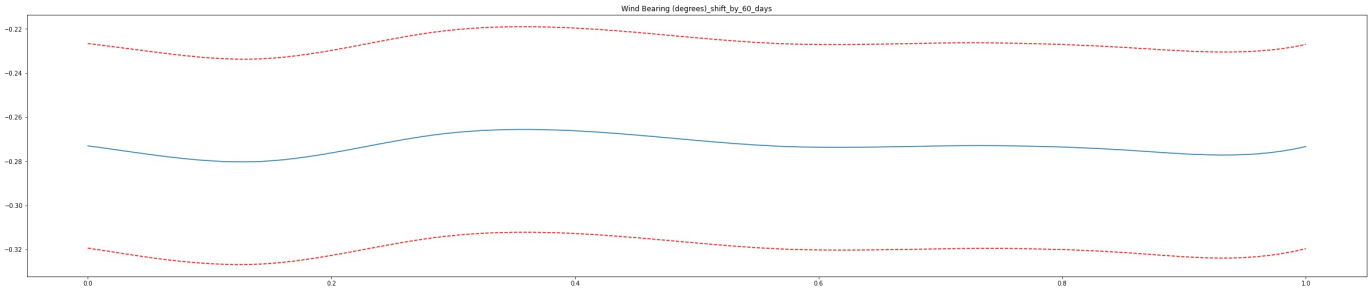

Visibility (km)_shift_by_30_days

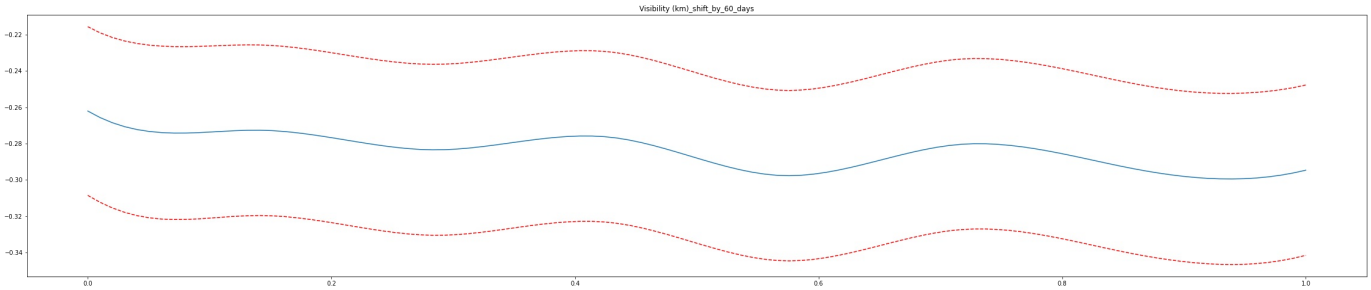

Pressure (millibars)_shift_by_30_days


Precip Type_shift_by_30_days









Sliding Window

Run-1

The Train Dataset is 0 to 50%

The Test Dataset is from 50 to 60% of Total

In [138…

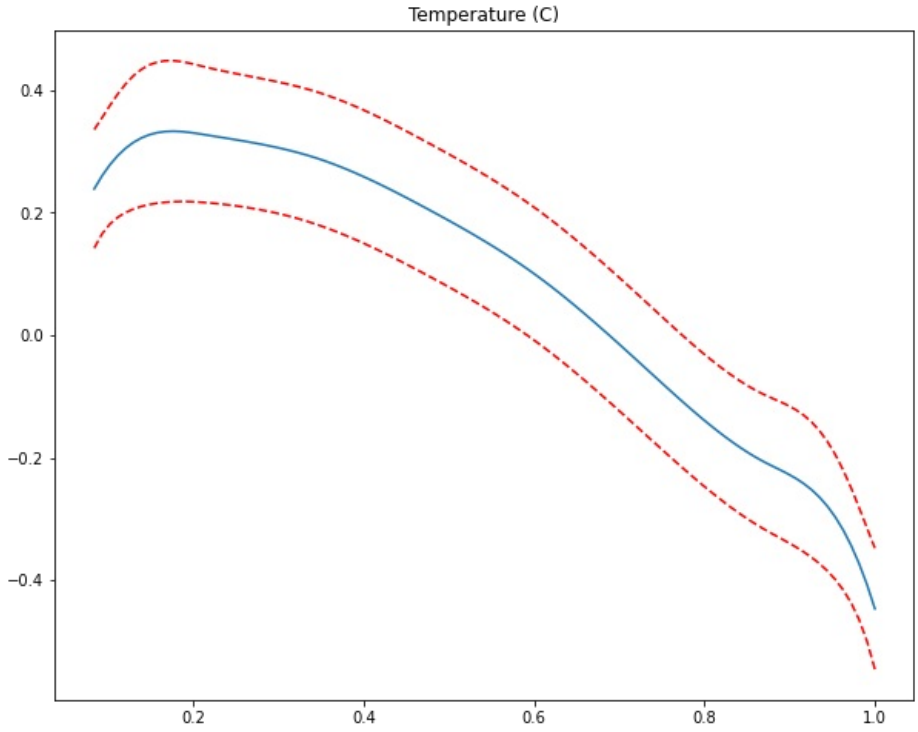
```
data_train = data[:int(0.5*(len(data)))]
data_test = data[int(0.5*(len(data))):int(0.6*(len(data)))]
lams = np.random.rand(400, 42)
lams = lams * 42 - 3
lams = np.exp(lams)
print(lams.shape)
gam = LinearGAM(n_splines=10).gridsearch(data_train[[col for col in data_train if col!='Humidity']].values,data_t
from sklearn.metrics import mean_squared_error
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
titles = data_train.columns[0:41]

for i in range(0,41):
    plt.figure(figsize=(10,8))
    XX = gam.generate_X_grid(term=i)
    plt.plot(XX[:, i], gam.partial_dependence(term=i, X=XX))
    plt.plot(XX[:, i], gam.partial_dependence(term=i, X=XX,   width=.95)[1], c='r', ls='--')

    plt.title(titles[i])
    plt.show()
```

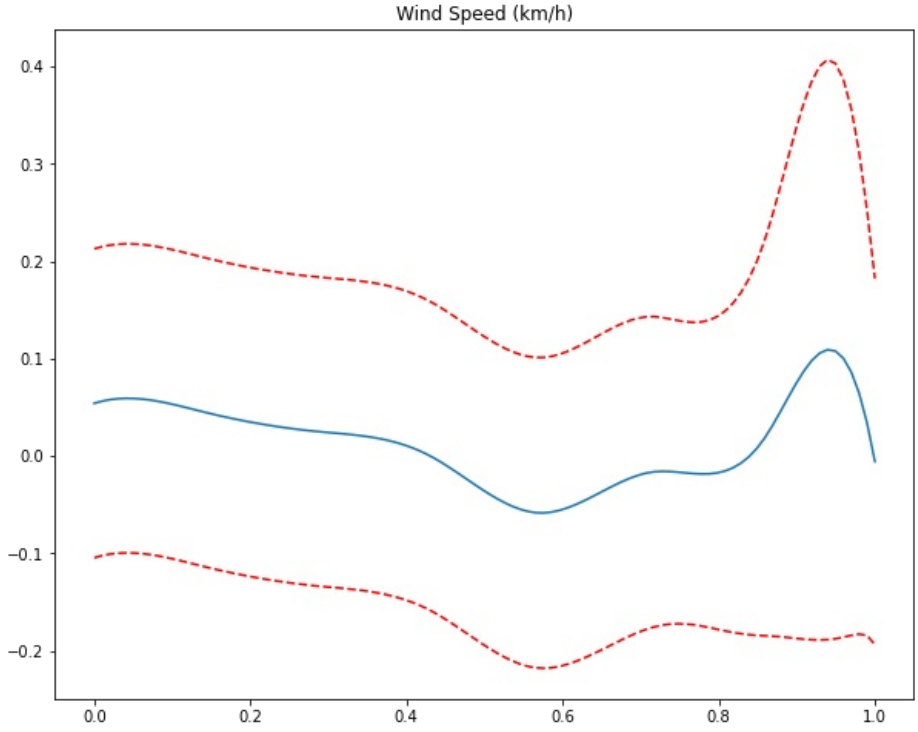
```
N/A% (0 of 11) |                        | Elapsed Time: 0:00:00 ETA:  --:--:--
```

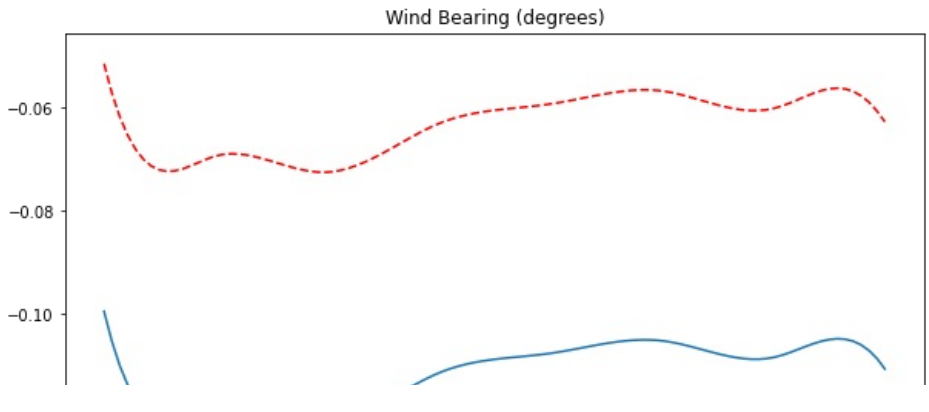
(400, 42)

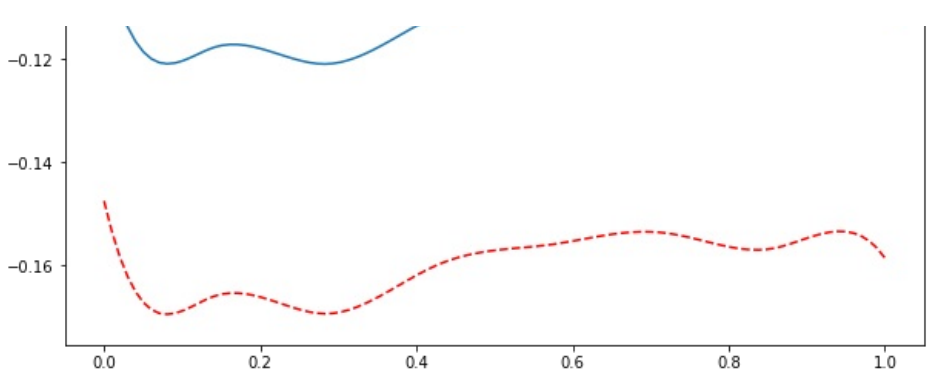
100% (11 of 11) |########################| Elapsed Time: 0:01:32 Time: 0:01:32

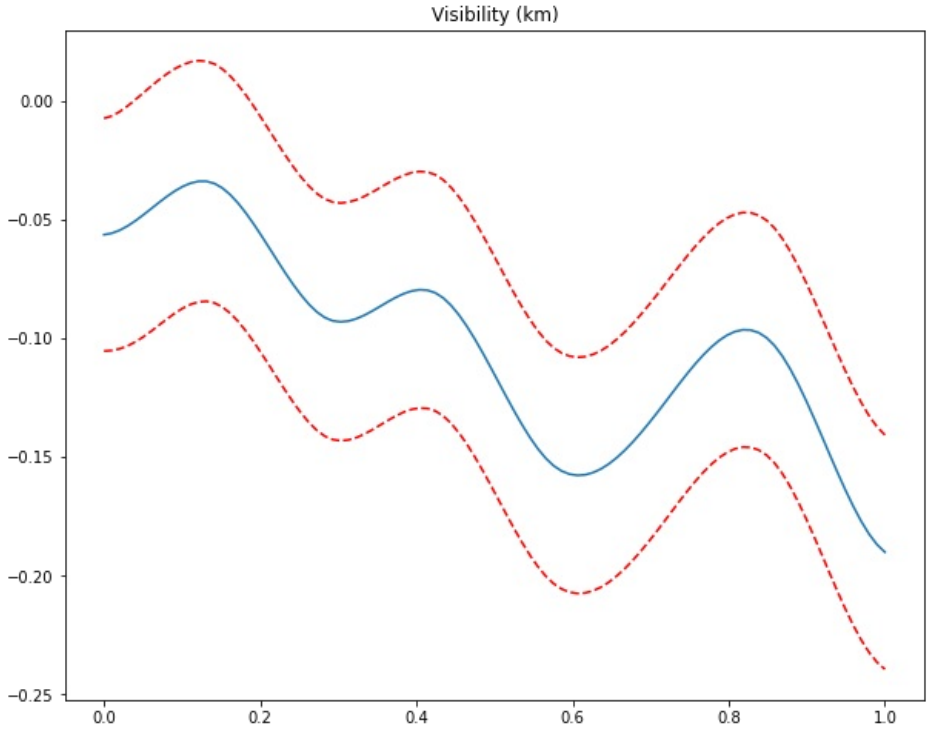
0.010478222509640828
0.1023631892314851

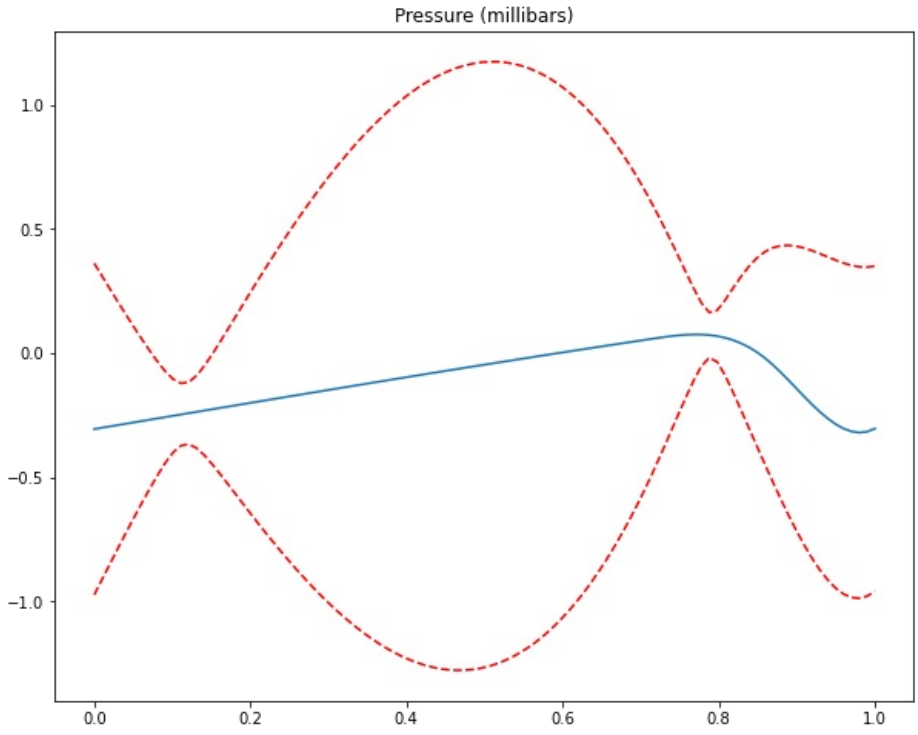


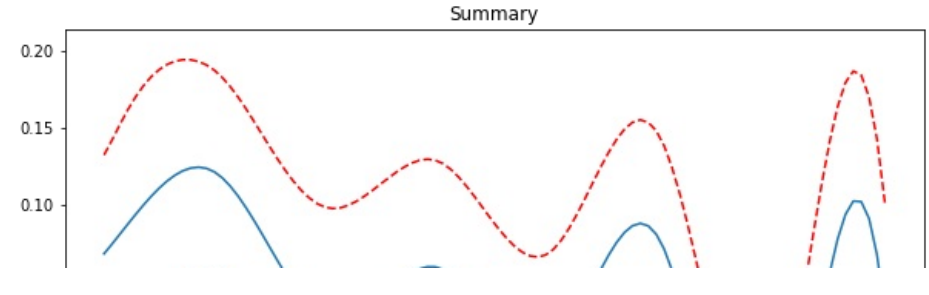


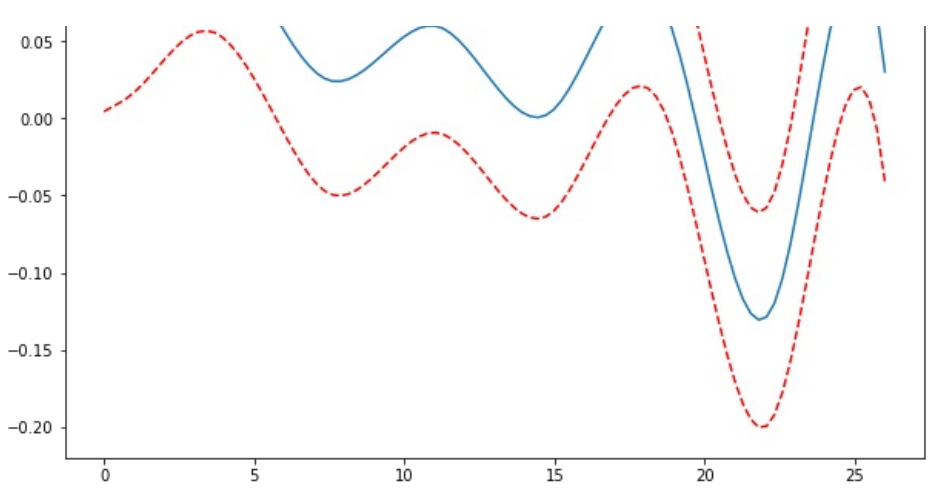


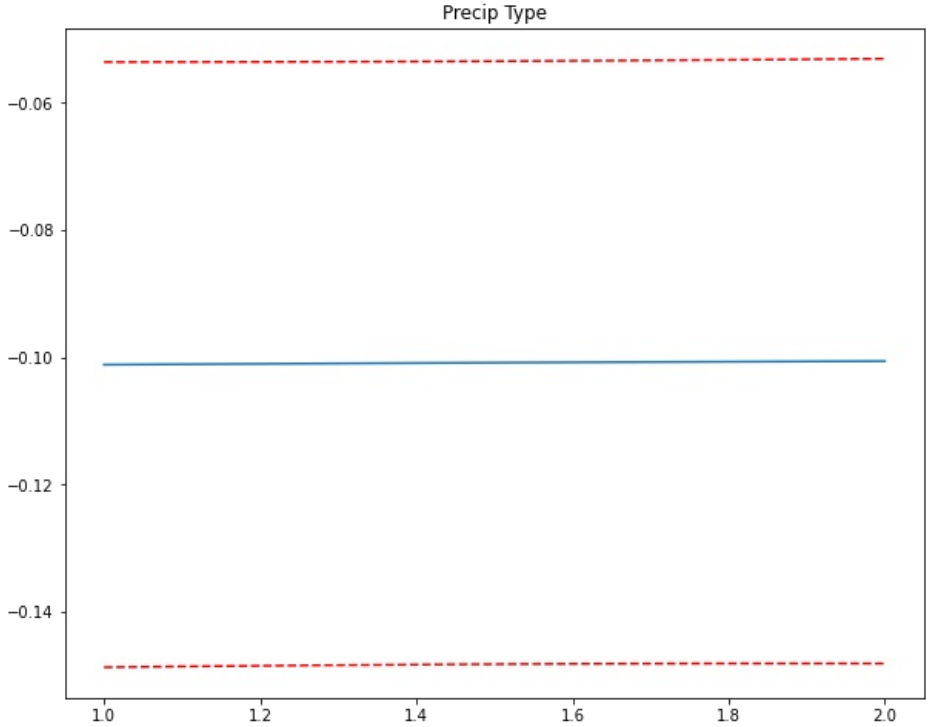

−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

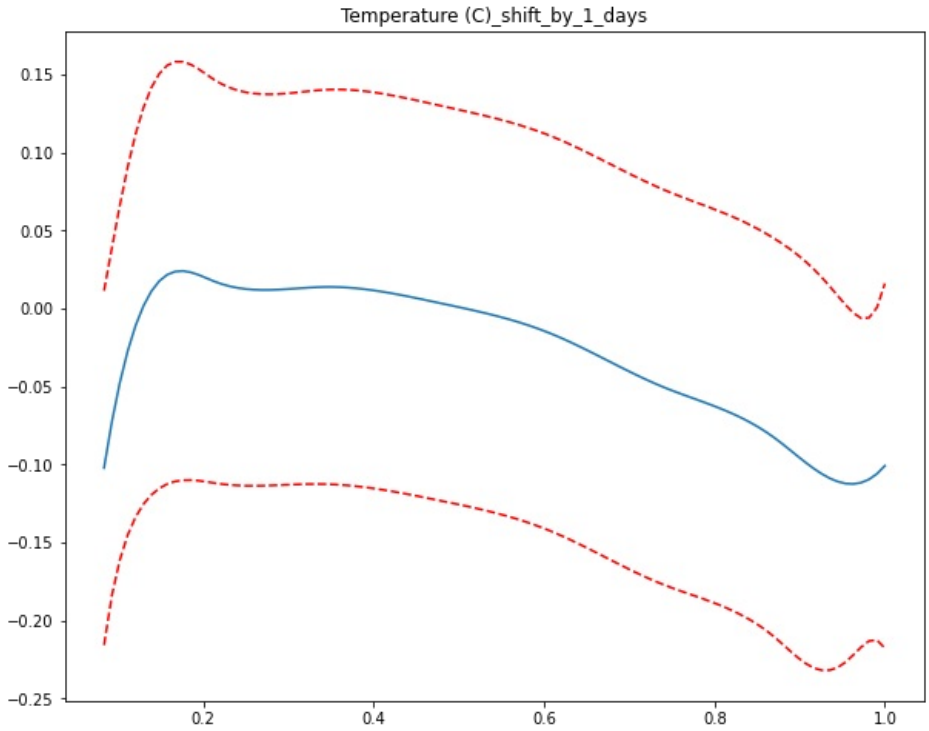

Visibility (km)
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
0.0
0.2
0.4
0.6
0.8
1.0

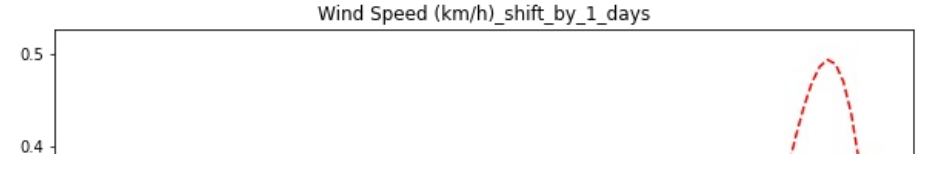

Pressure (millibars)
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

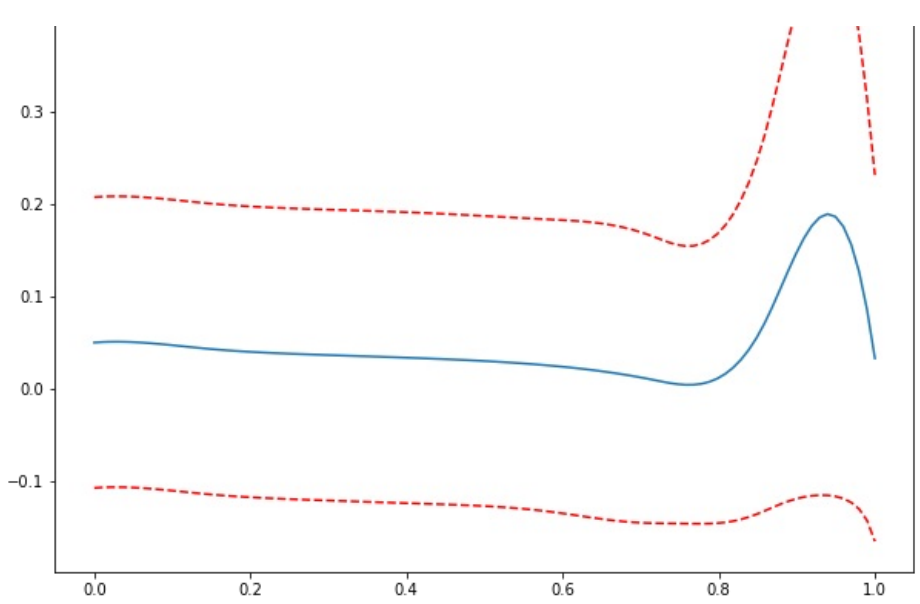

Summary
0.20
0.15
0.10

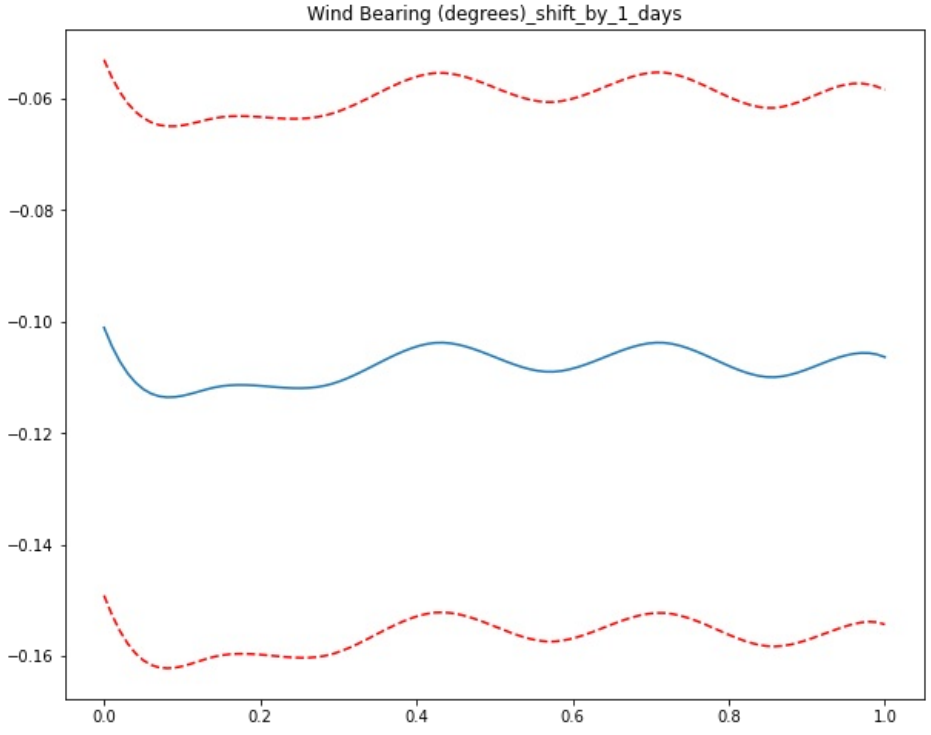

0.05
0.00
−0.05
−0.10
−0.15
−0.20
0
5
10
15
20
25

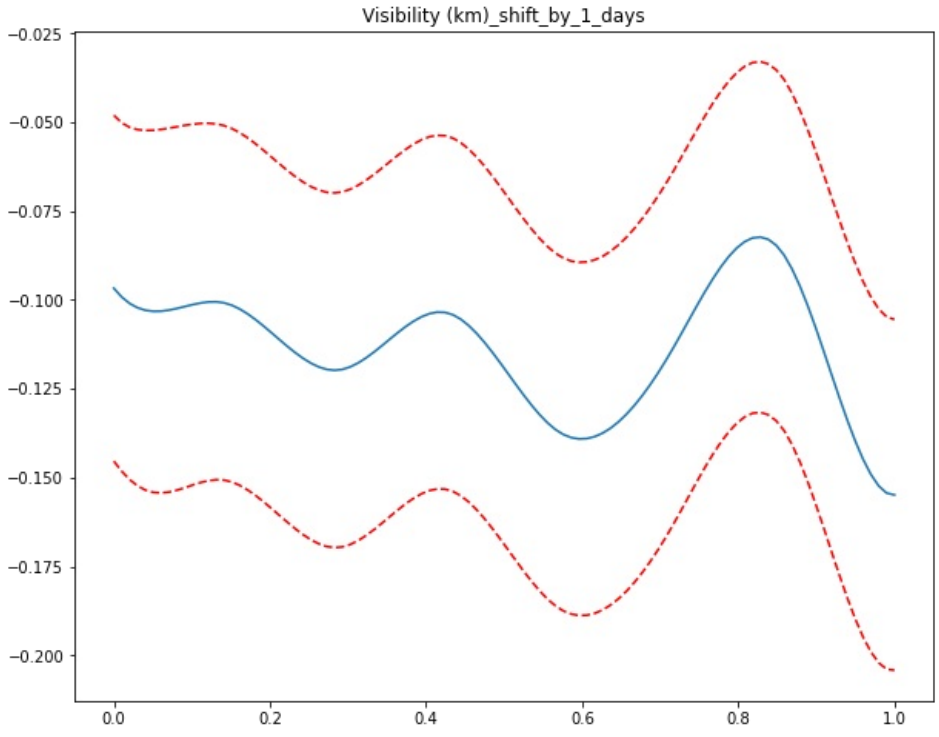

Precip Type
−0.06
−0.08
−0.10
−0.12
−0.14
1.0
1.2
1.4
1.6
1.8
2.0

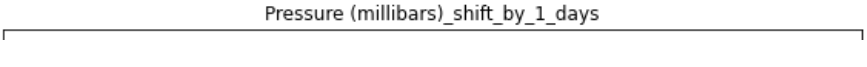

Temperature (C)_shift_by_1_days
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
0.2
0.4
0.6
0.8
1.0

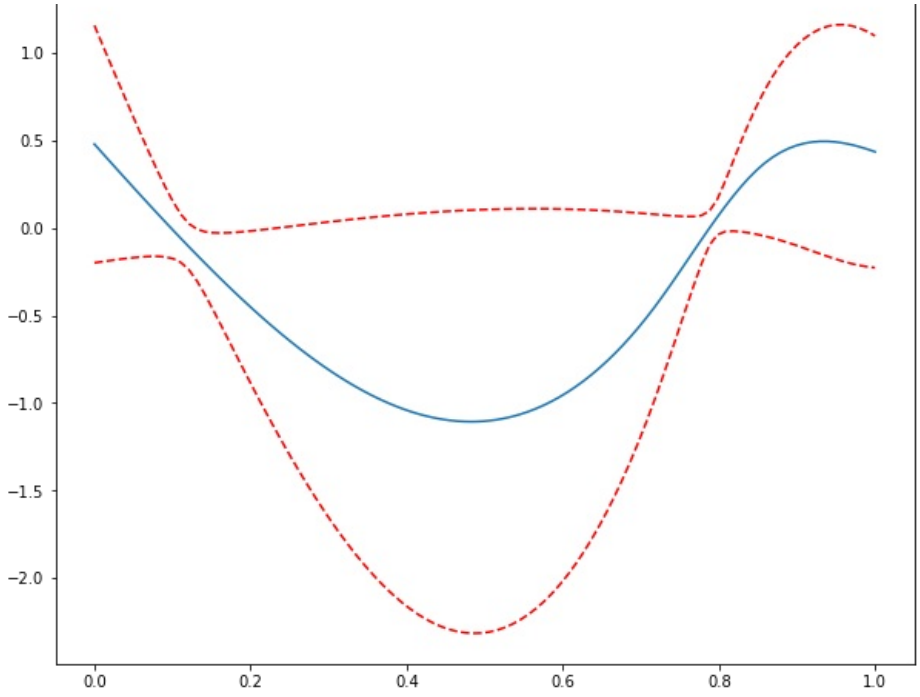

Wind Speed (km/h)_shift_by_1_days
0.5
0.4

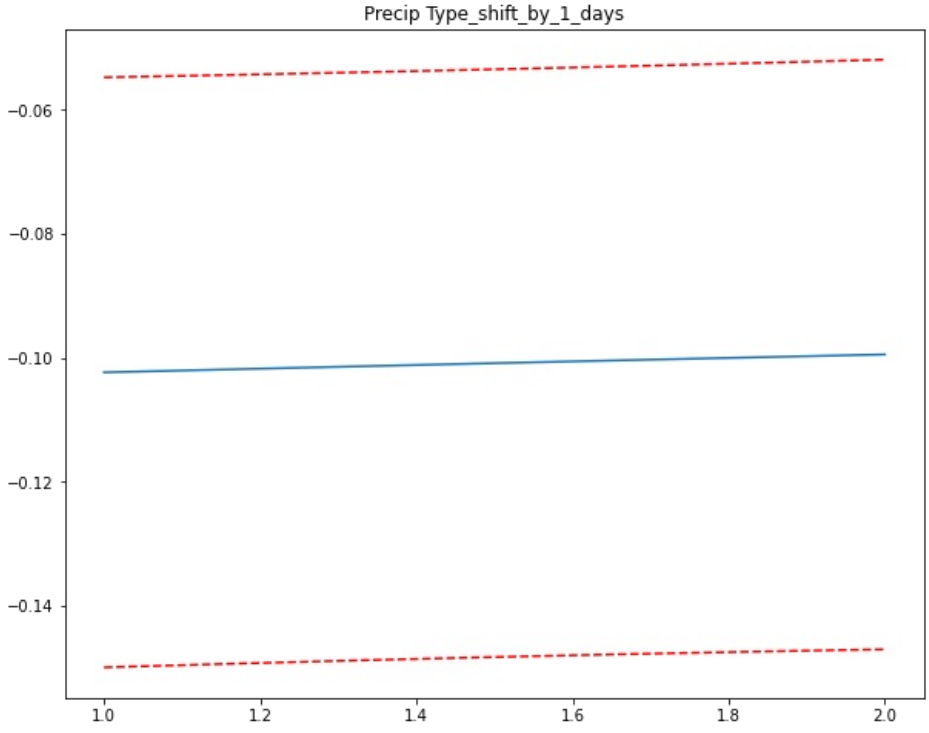

0.3
0.2
0.1
0.0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0

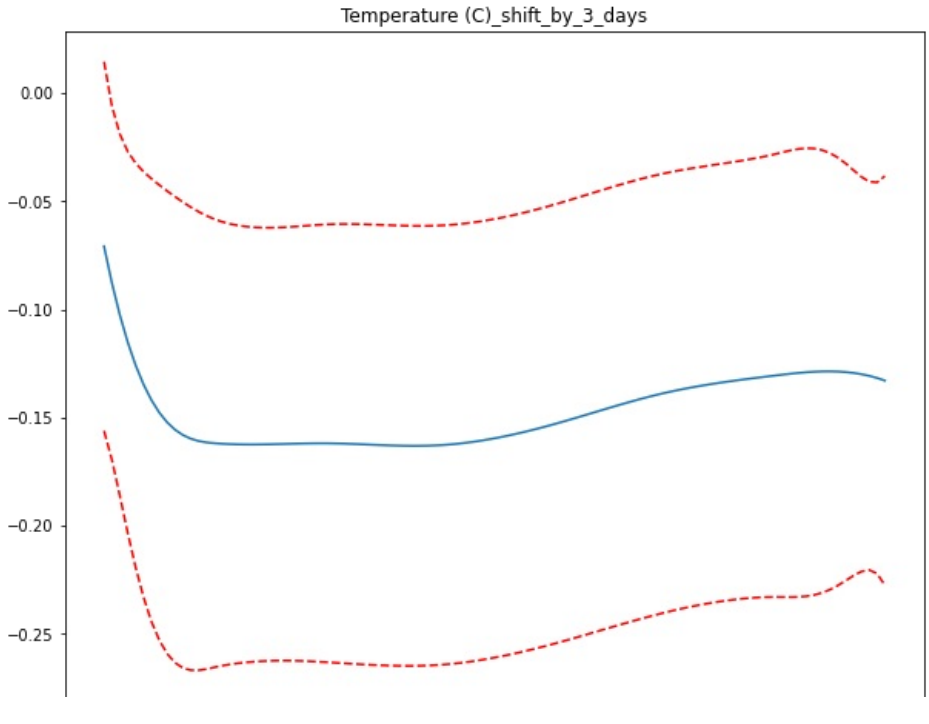

Wind Bearing (degrees)_shift_by_1_days
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

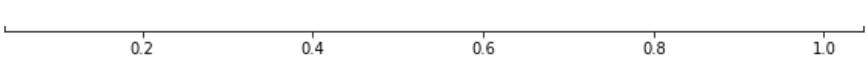

Visibility (km)_shift_by_1_days
−0.025
−0.050
−0.075
−0.100
−0.125
−0.150
−0.175
−0.200
0.0
0.2
0.4
0.6
0.8
1.0

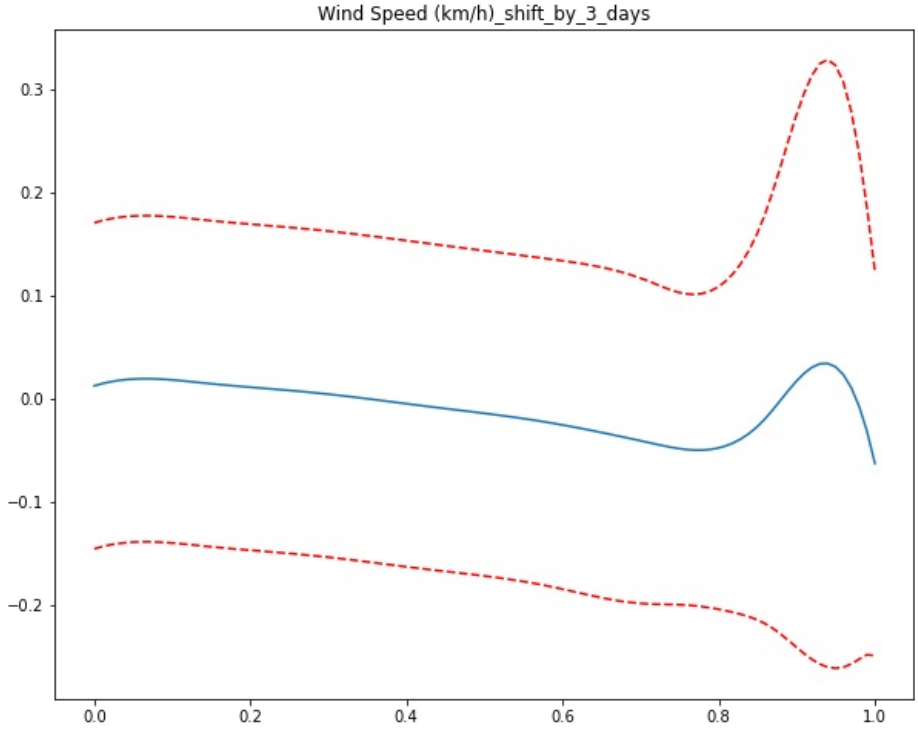

Pressure (millibars)_shift_by_1_days

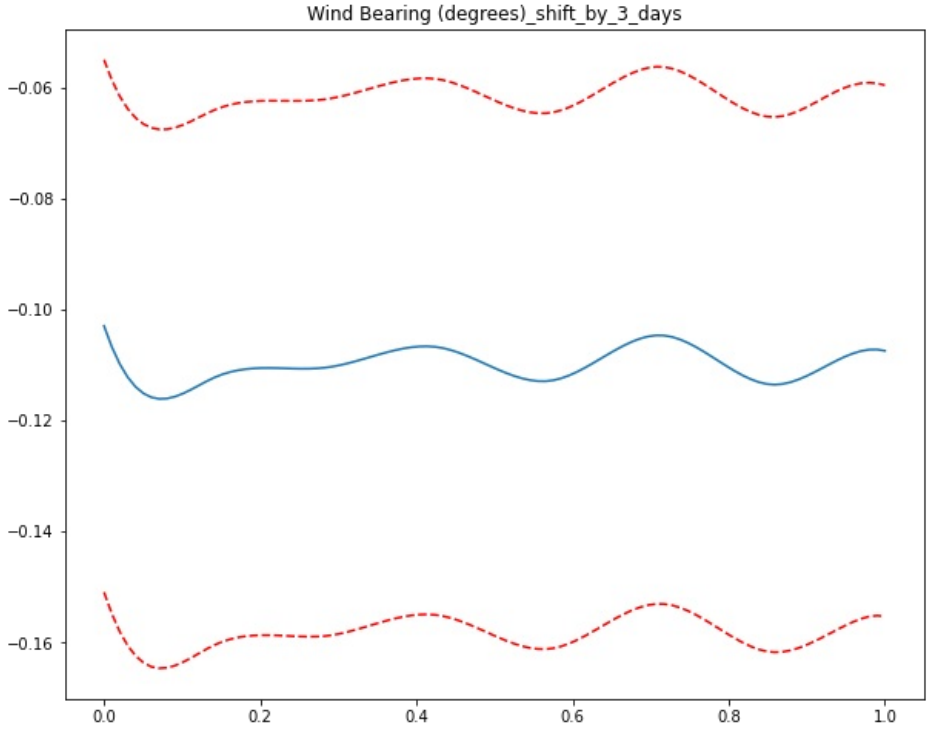

1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.0
0.2
0.4
0.6
0.8
1.0

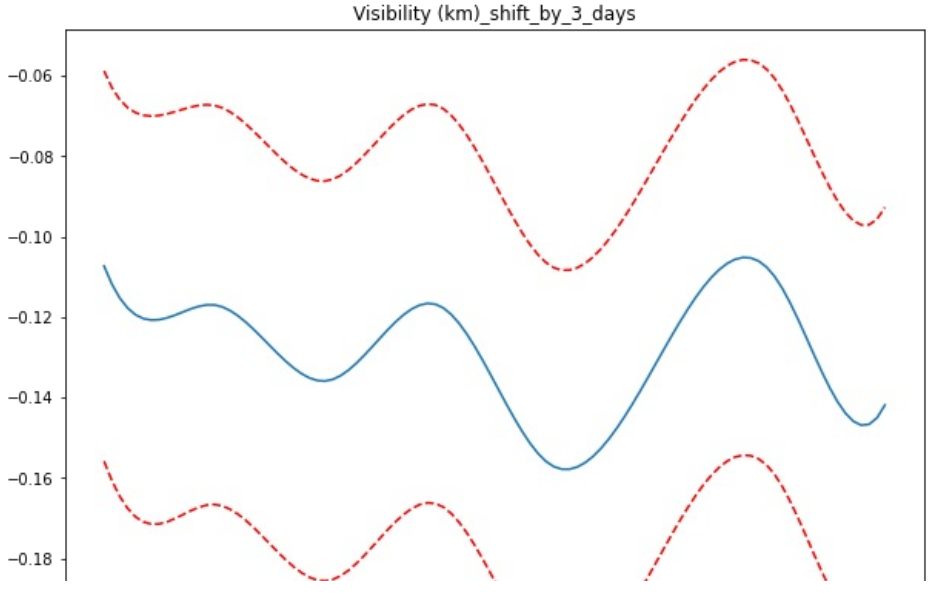

Precip Type_shift_by_1_days
−0.06
−0.08
−0.10
−0.12
−0.14
1.0
1.2
1.4
1.6
1.8
2.0

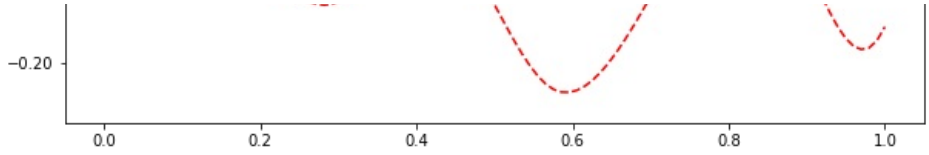

Temperature (C)_shift_by_3_days
0.00
−0.05
−0.10
−0.15
−0.20
−0.25

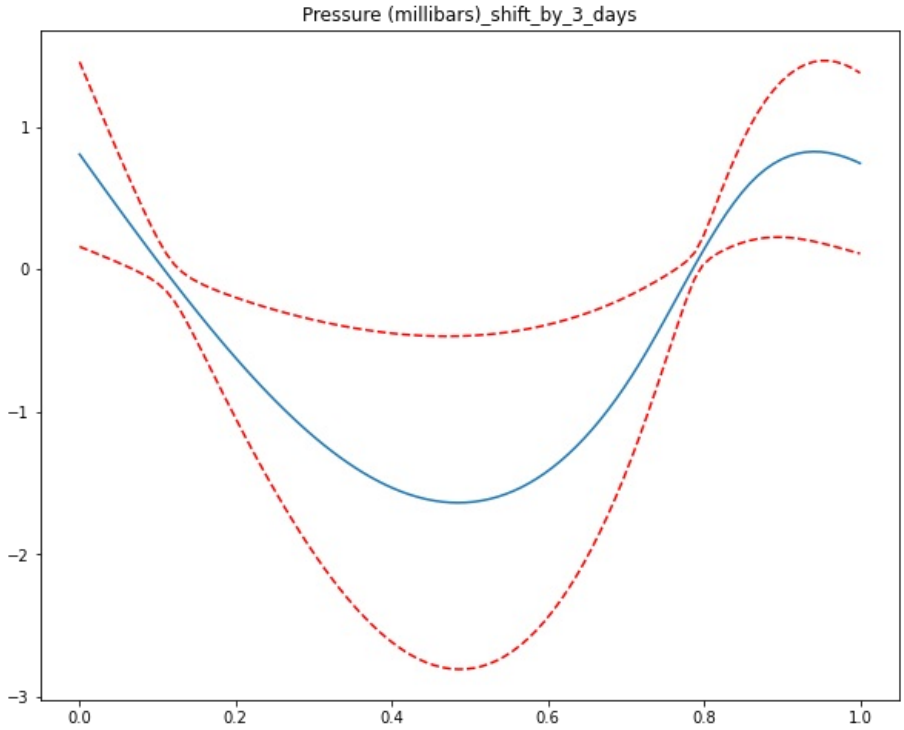

0.2
0.4
0.6
0.8
1.0

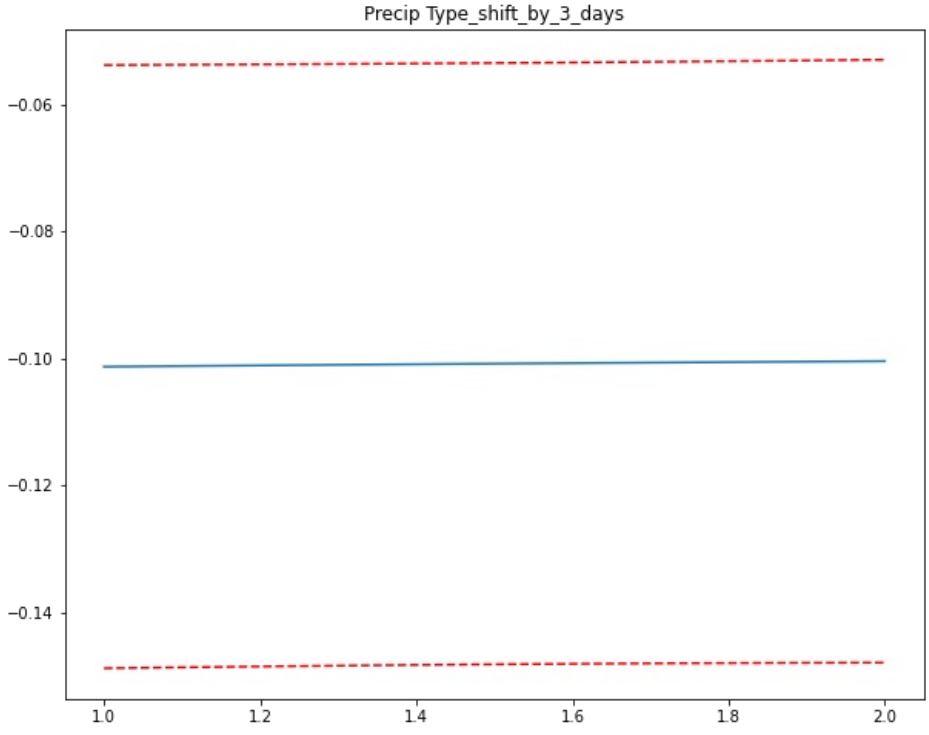

Wind Speed (km/h)_shift_by_3_days
0.3
0.2
0.1
0.0
−0.1
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

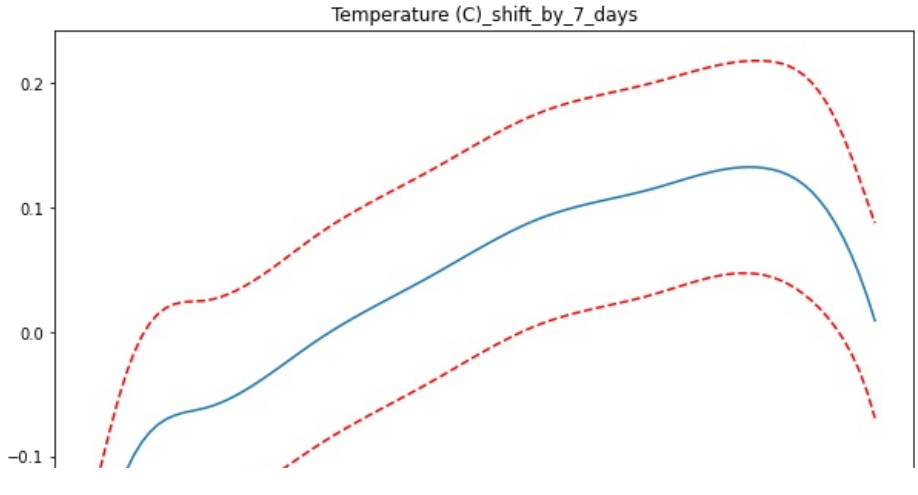

Wind Bearing (degrees)_shift_by_3_days
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

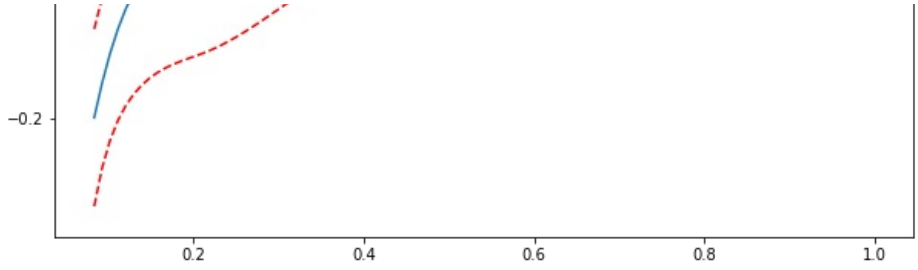

Visibility (km)_shift_by_3_days
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
−0.18

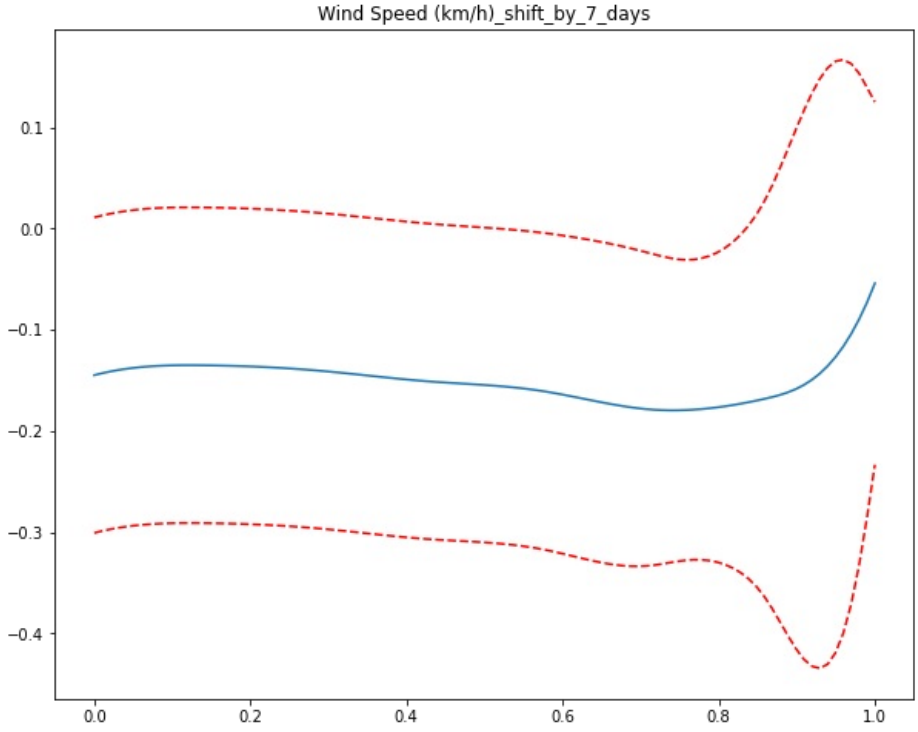

−0.20
0.0
0.2
0.4
0.6
0.8
1.0

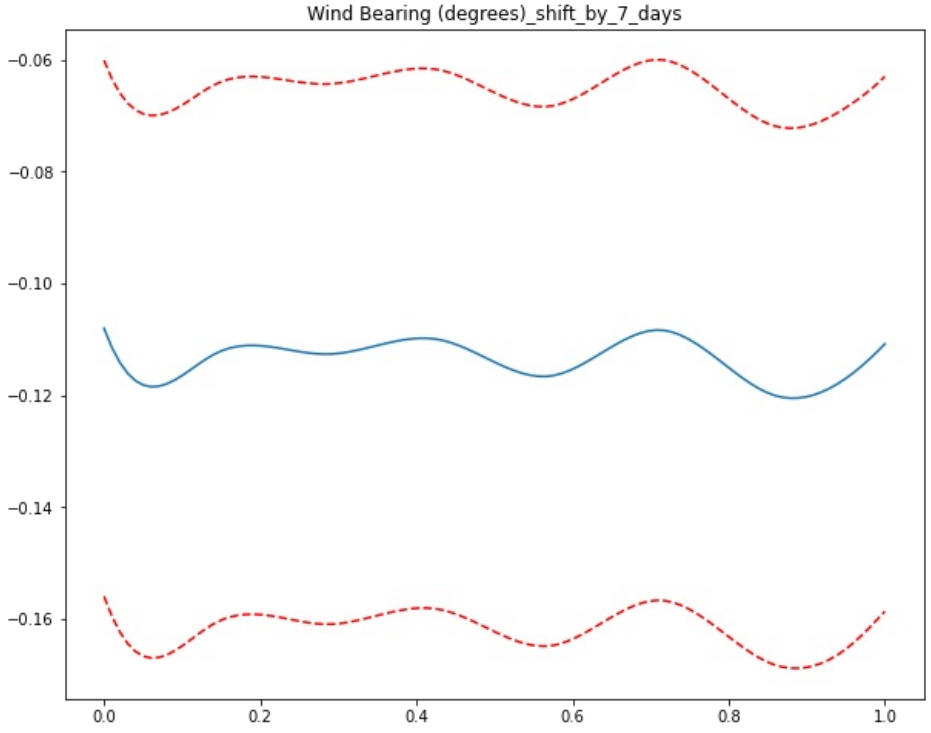

Pressure (millibars)_shift_by_3_days
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0

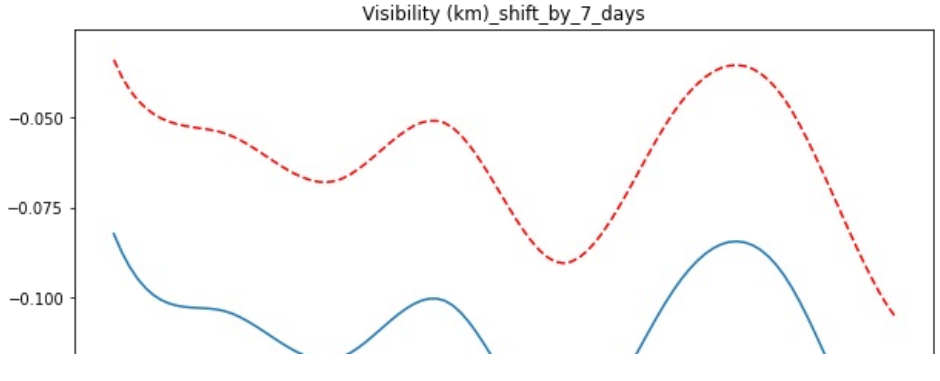

Precip Type_shift_by_3_days
−0.06
−0.08
−0.10
−0.12
−0.14
1.0
1.2
1.4
1.6
1.8
2.0

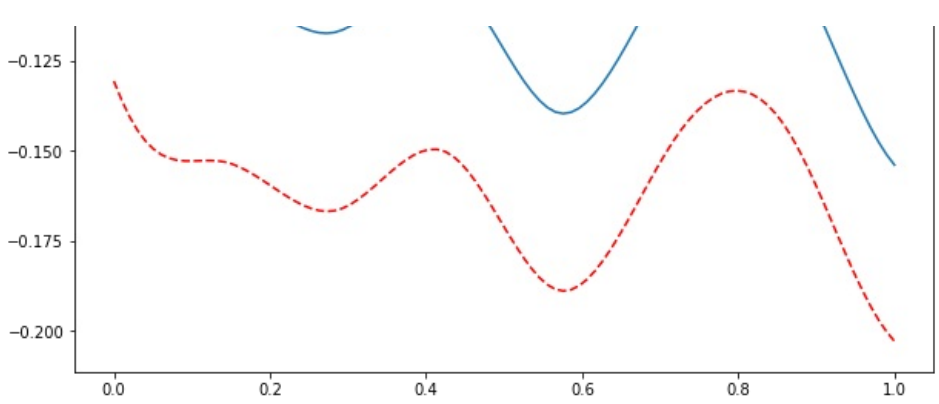

Temperature (C)_shift_by_7_days
0.2
0.1
0.0
−0.1

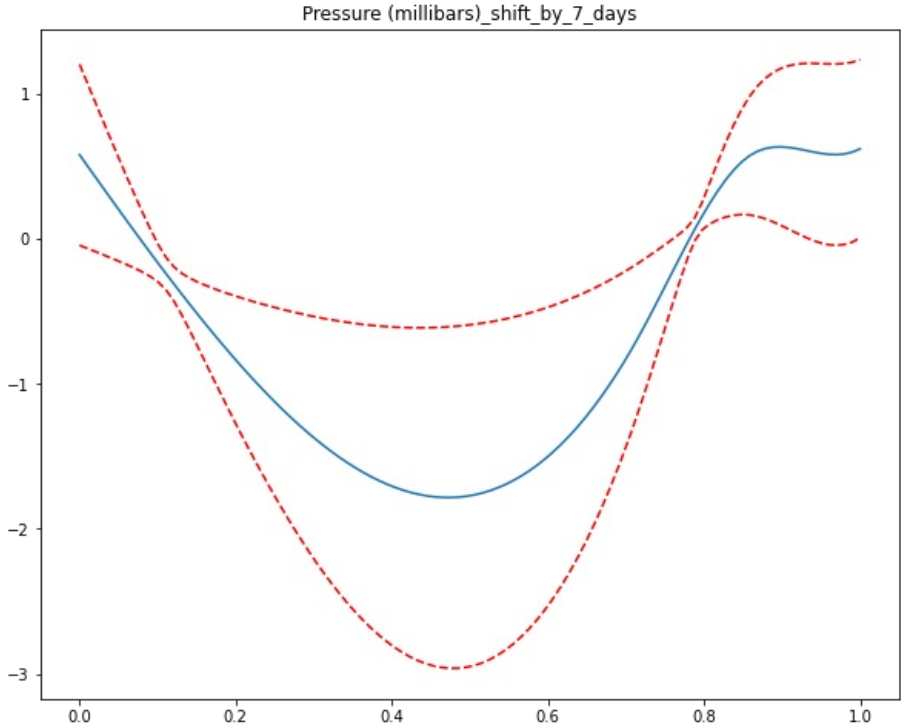

−0.2
0.2
0.4
0.6
0.8
1.0

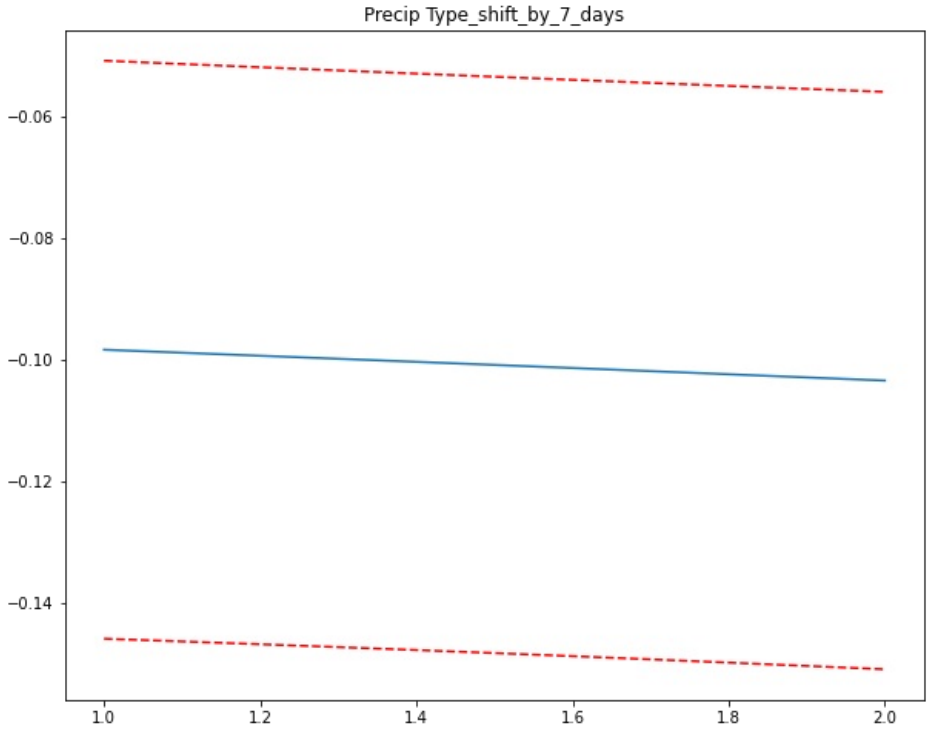

Wind Speed (km/h)_shift_by_7_days
0.1
0.0
−0.1
−0.2
−0.3
−0.4
0.0
0.2
0.4
0.6
0.8
1.0

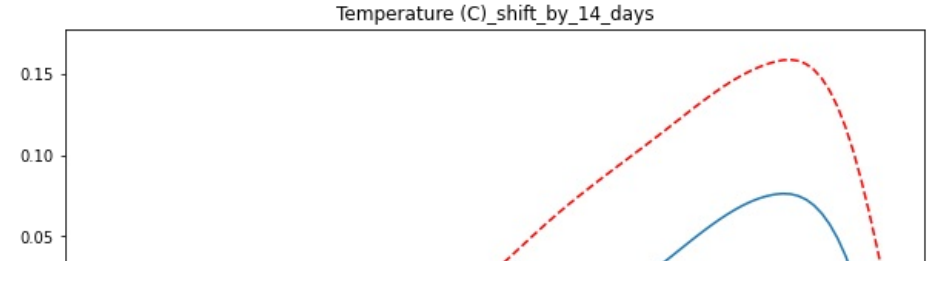

Wind Bearing (degrees)_shift_by_7_days
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

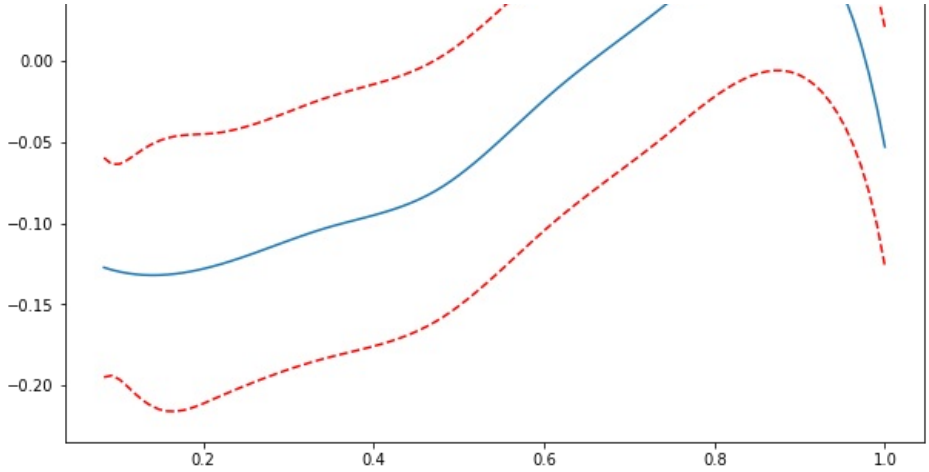

Visibility (km)_shift_by_7_days
−0.050
−0.075
−0.100

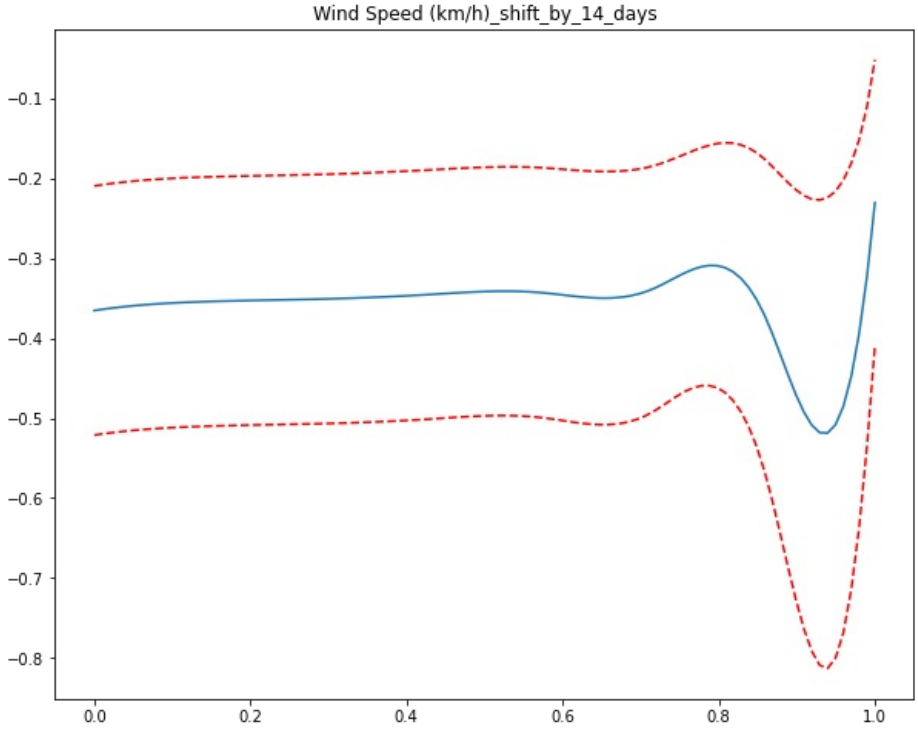

−0.125
−0.150
−0.175
−0.200
0.0
0.2
0.4
0.6
0.8
1.0

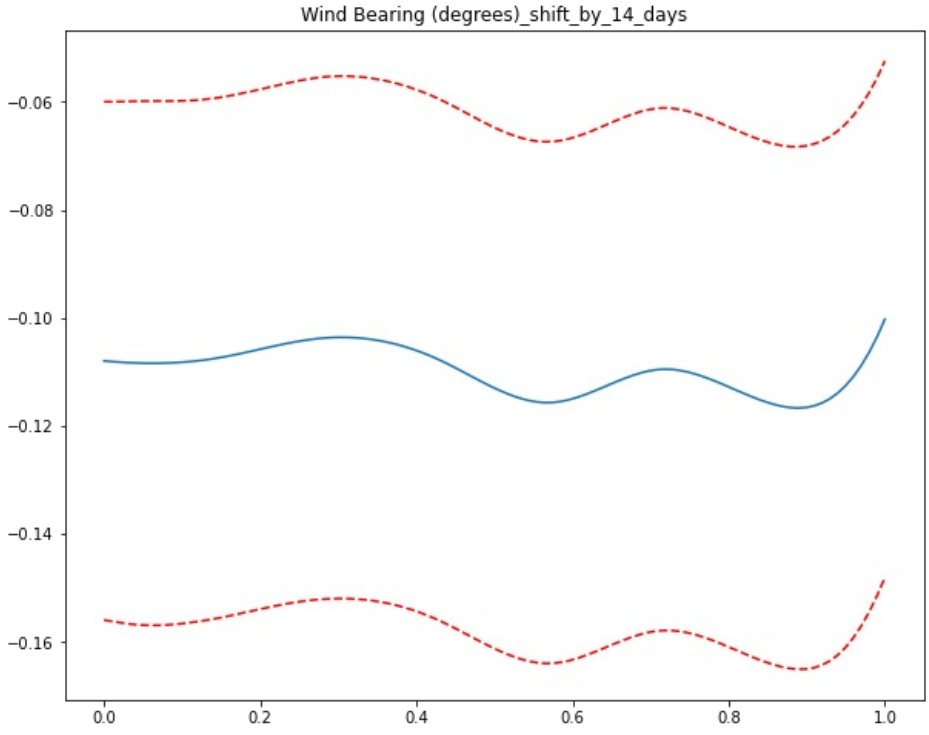

Pressure (millibars)_shift_by_7_days
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0

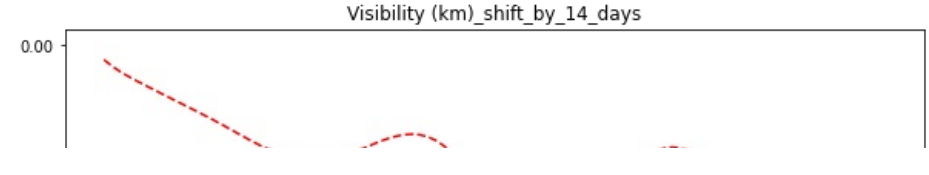

Precip Type_shift_by_7_days
−0.06
−0.08
−0.10
−0.12
−0.14
1.0
1.2
1.4
1.6
1.8
2.0

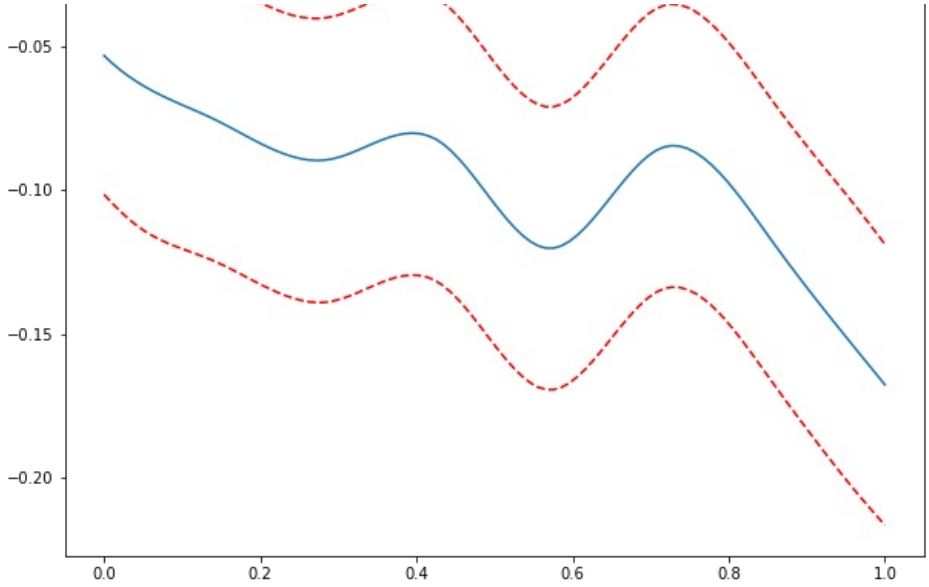

Temperature (C)_shift_by_14_days
0.15
0.10
0.05


0.00
−0.05
−0.10
−0.15
−0.20
0.2
0.4
0.6
0.8
1.0

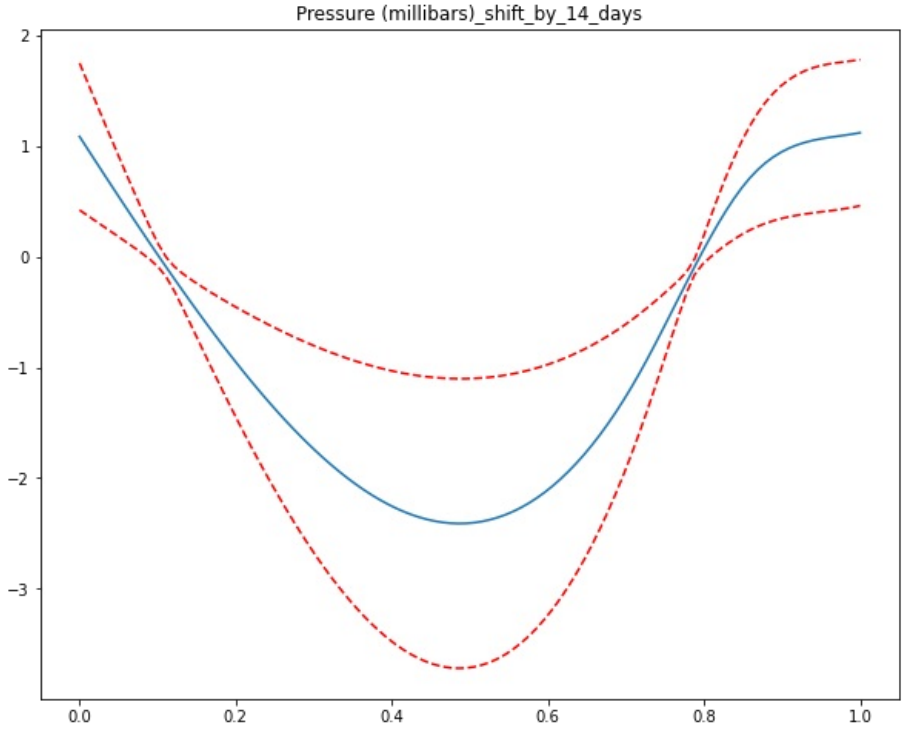

Wind Speed (km/h)_shift_by_14_days
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
−0.7
−0.8
0.0
0.2
0.4
0.6
0.8
1.0


Wind Bearing (degrees)_shift_by_14_days
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

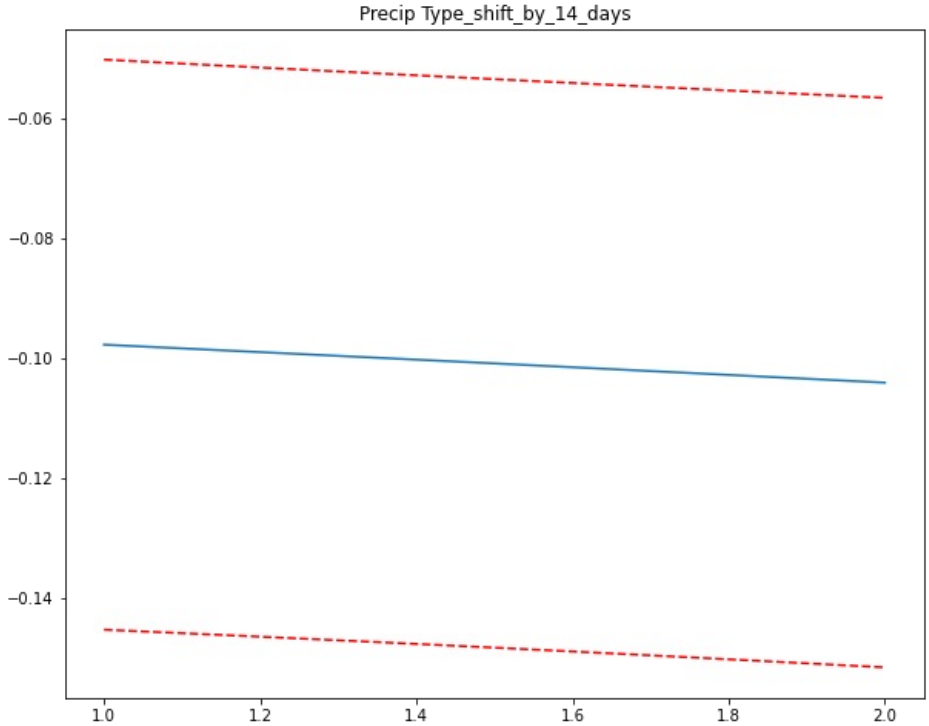

Visibility (km)_shift_by_14_days
0.00



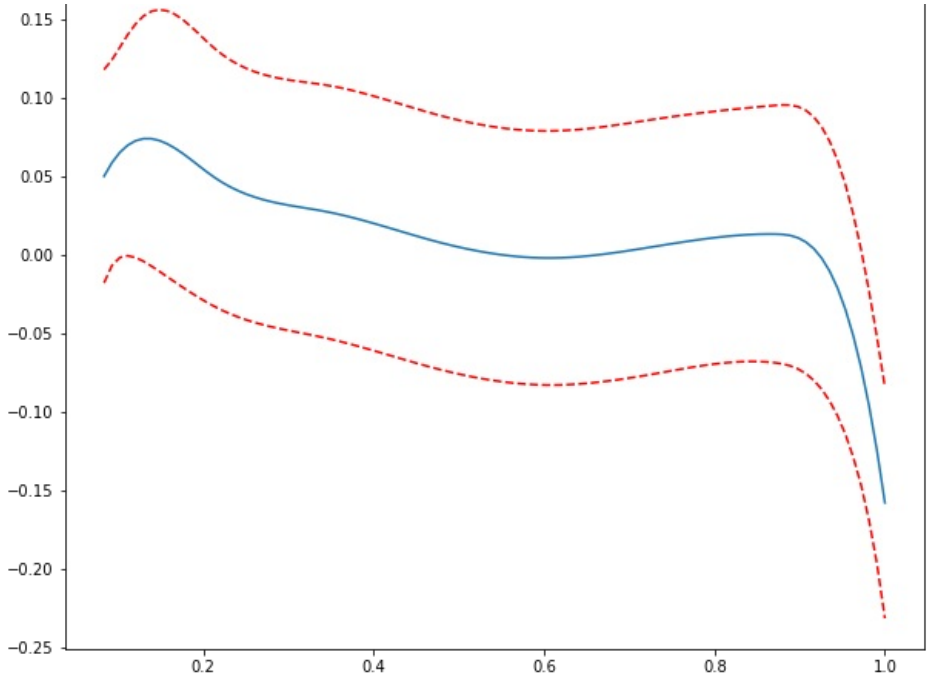
Pressure (millibars)_shift_by_14_days

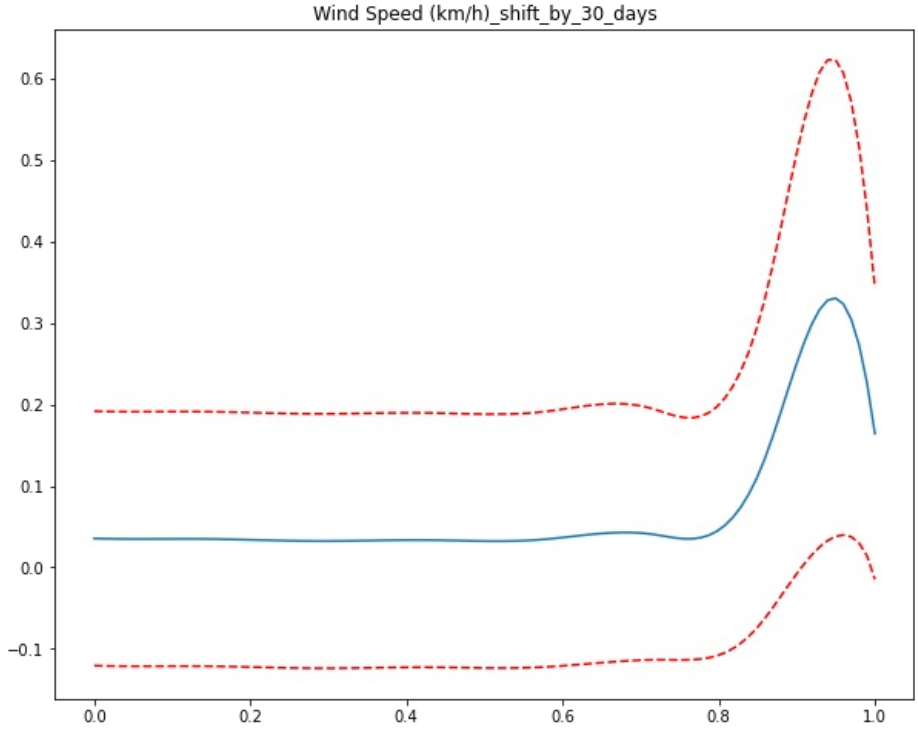


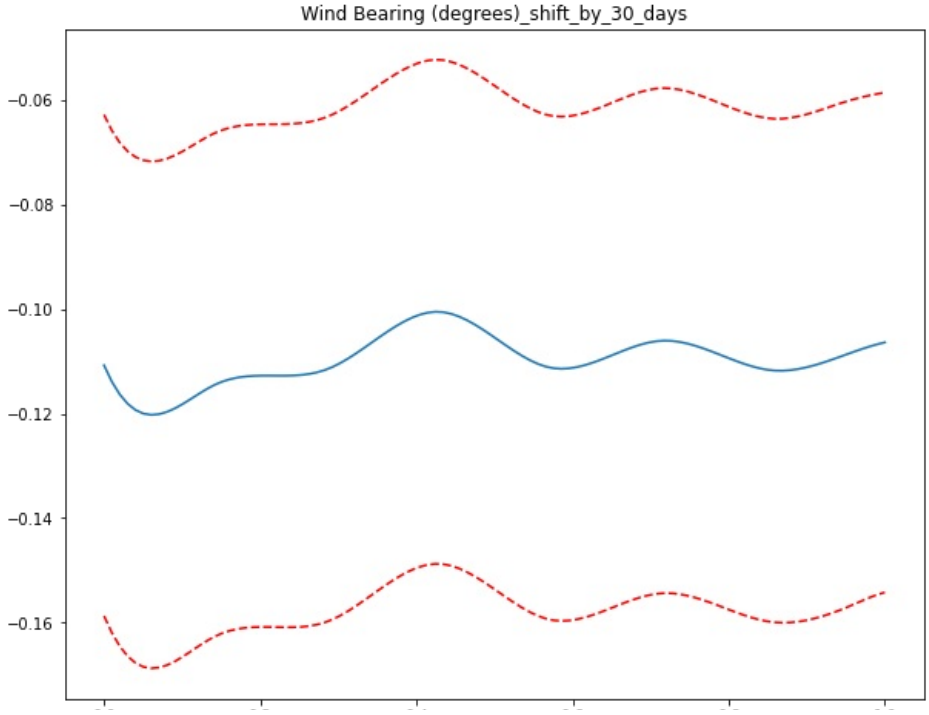
Precip Type_shift_by_14_days



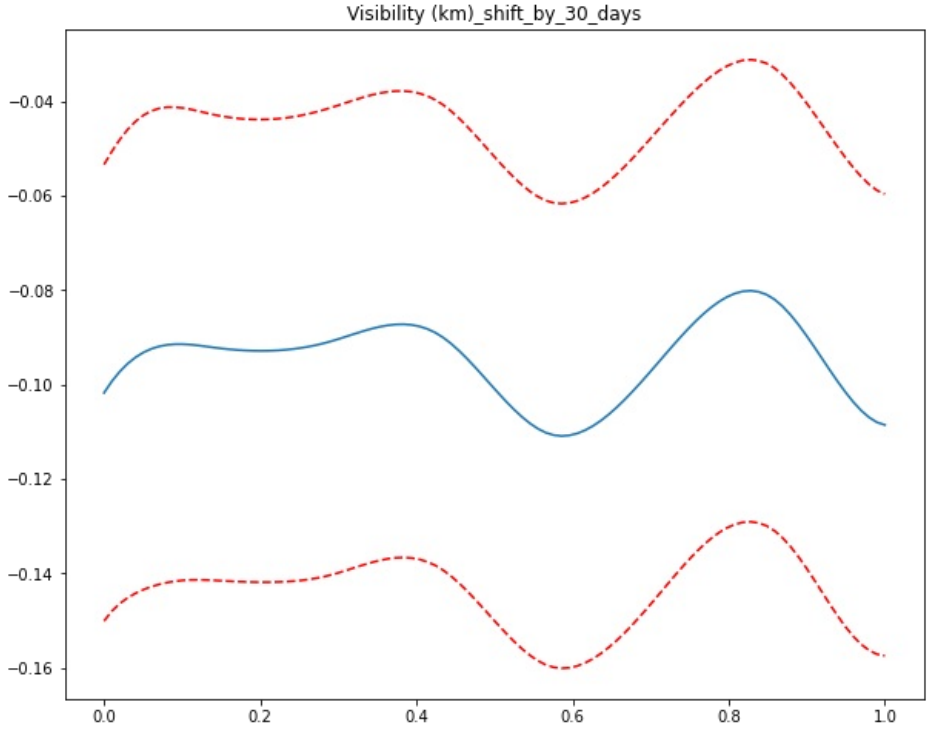
Temperature (C)_shift_by_30_days

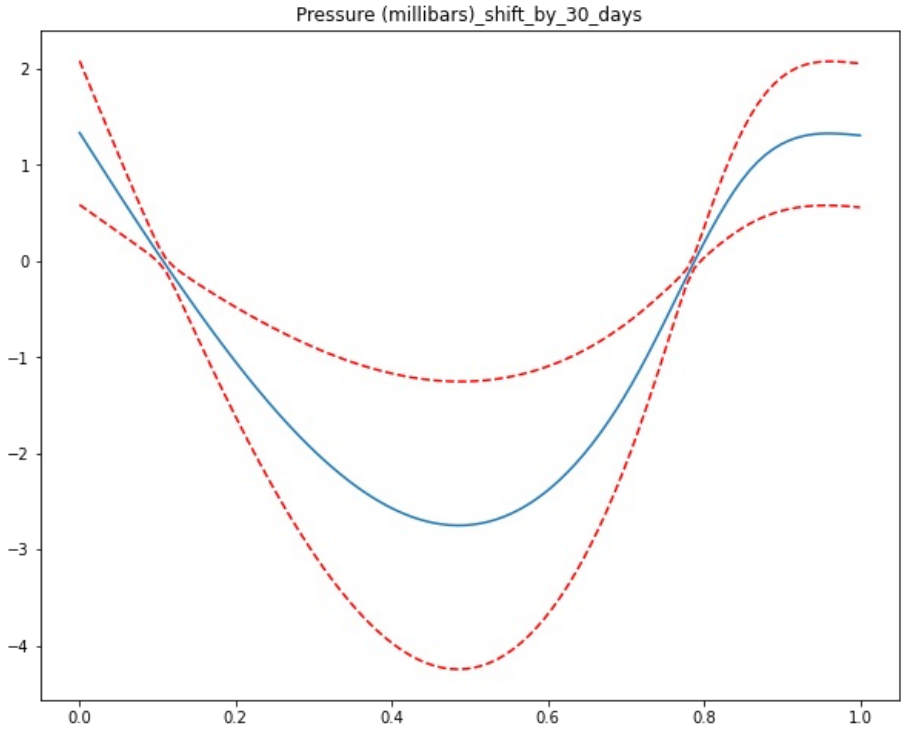

0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
0.2
0.4
0.6
0.8
1.0

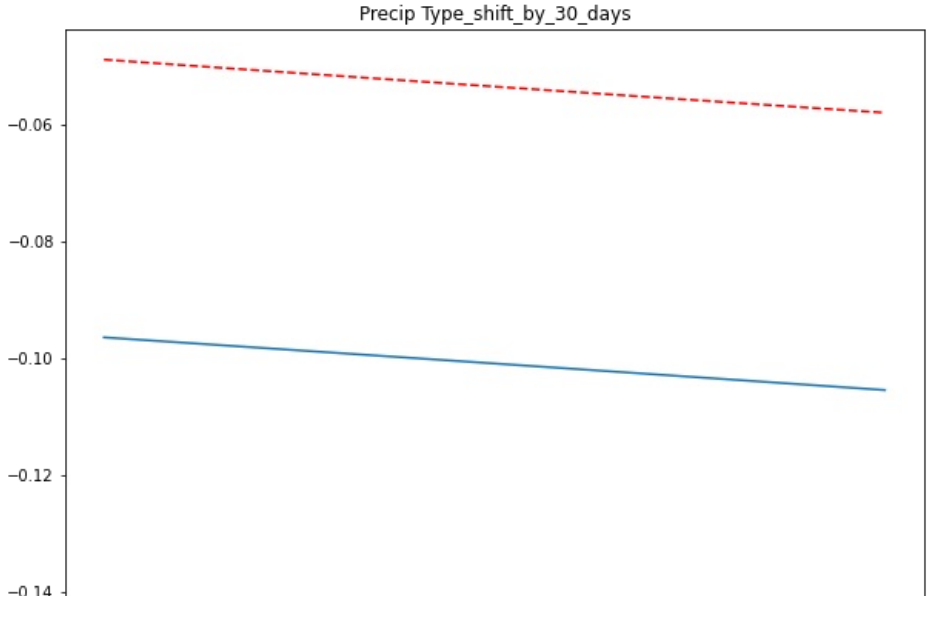

Wind Speed (km/h)_shift_by_30_days
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0

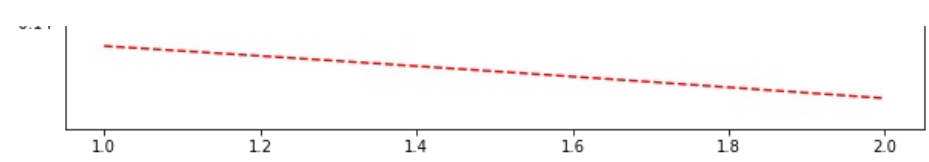

Wind Bearing (degrees)_shift_by_30_days
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16

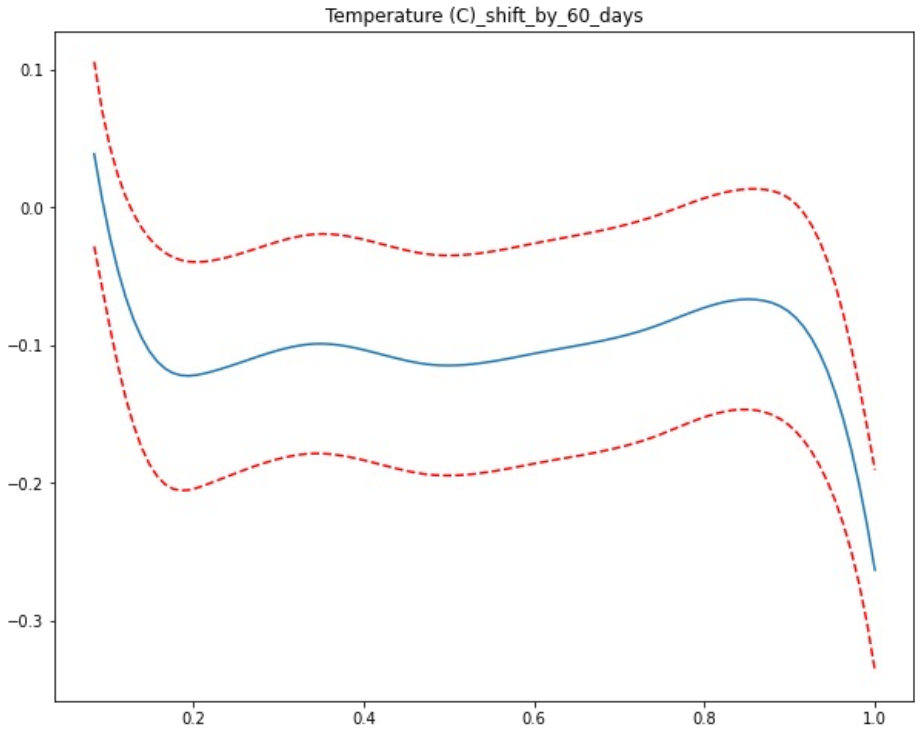
0.0
0.2
0.4
0.6
0.8
1.0

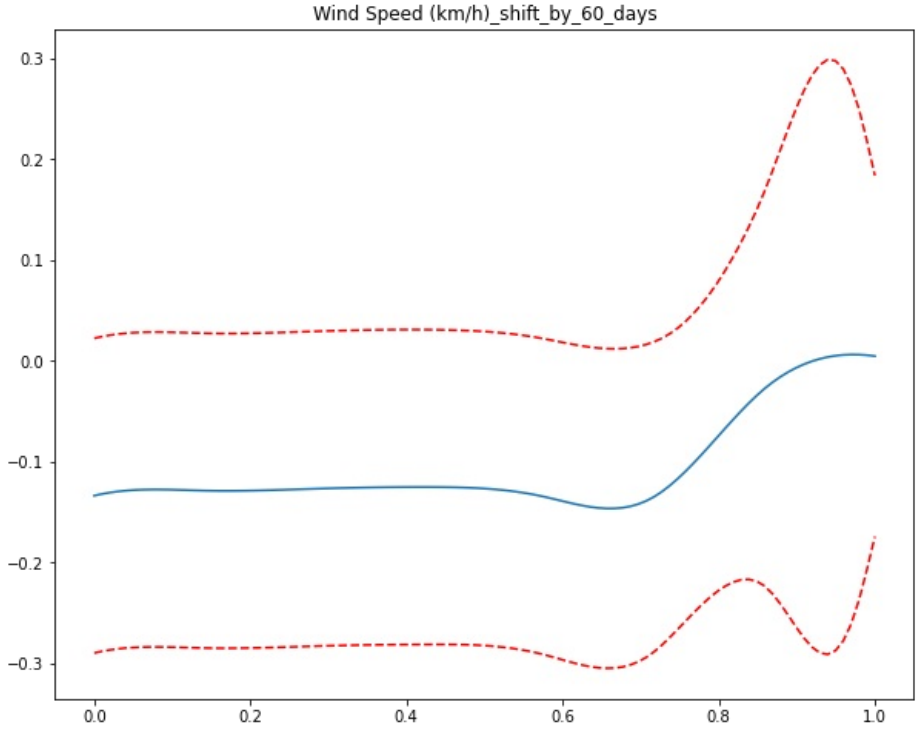

Visibility (km)_shift_by_30_days
−0.04
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

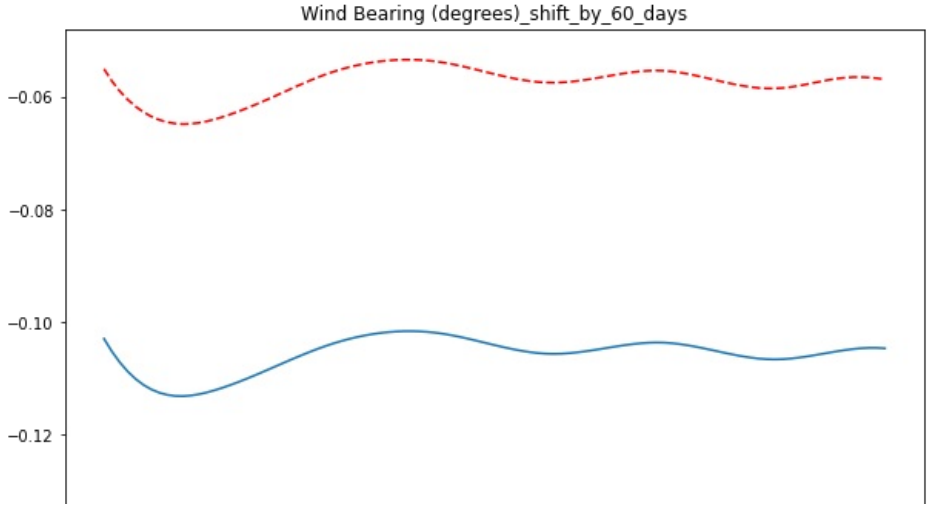

Pressure (millibars)_shift_by_30_days
2
1
0
−1
−2
−3
−4
0.0
0.2
0.4
0.6
0.8
1.0

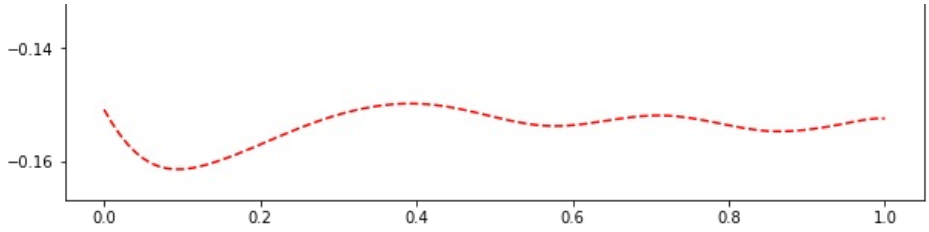

Precip Type_shift_by_30_days
−0.06
−0.08
−0.10
−0.12
−0.14

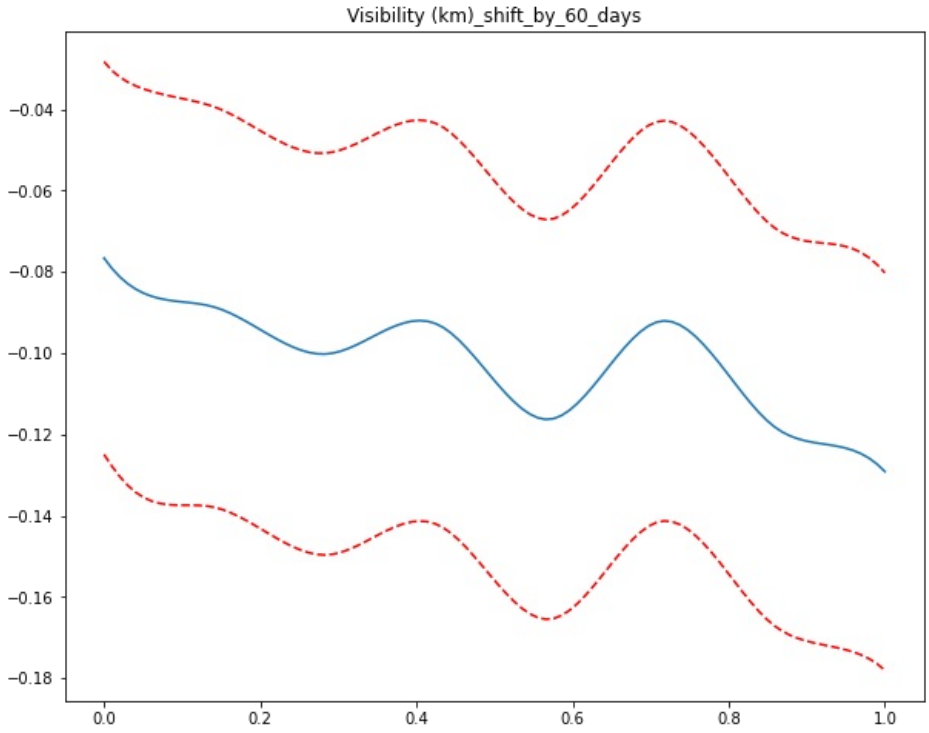

1.0
1.2
1.4
1.6
1.8
2.0


Temperature (C)_shift_by_60_days
0.1
0.0
−0.1
−0.2
−0.3
0.2
0.4
0.6
0.8
1.0


Wind Speed (km/h)_shift_by_60_days
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.2
0.4
0.6
0.8
1.0


Wind Bearing (degrees)_shift_by_60_days
−0.06
−0.08
−0.10
−0.12





Run-2

The Train starts now from 10% of Dataset to 60 %
The Test Starts now from 60 to 70%

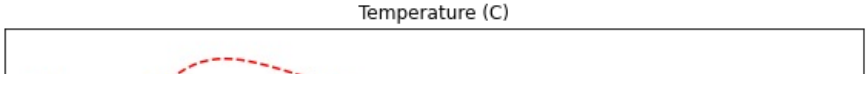
In [139…

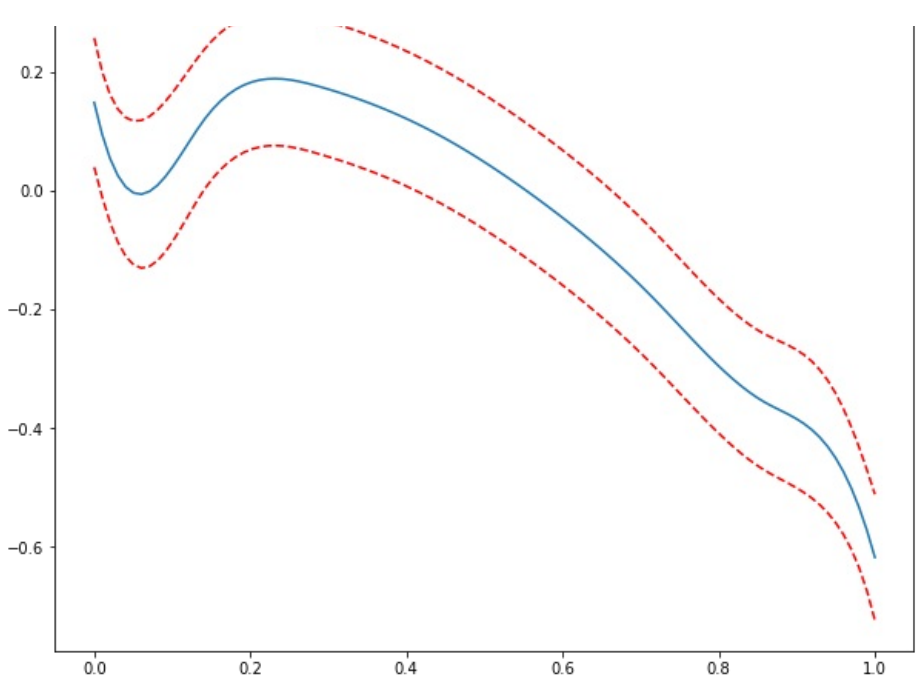
```
data_train = data[int(0.1*(len(data))):int(0.6*(len(data)))]
data_test = data[int(0.6*(len(data))):int(0.7*(len(data)))]
lams = np.random.rand(400, 42)
lams = lams * 42 - 3
lams = np.exp(lams)
print(lams.shape)
gam = LinearGAM(n_splines=10).gridsearch(data_train[[col for col in data_train if col!='Humidity']].values,data_t
from sklearn.metrics import mean_squared_error
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
print(mean_squared_error(gam.predict(data_test[[col for col in data_train if col!='Humidity']]),data_test['Humidi
titles = data_train.columns[0:41]

for i in range(0,41):
    plt.figure(figsize=(10,8))
    XX = gam.generate_X_grid(term=i)
    plt.plot(XX[:, i], gam.partial_dependence(term=i, X=XX))
    plt.plot(XX[:, i], gam.partial_dependence(term=i, X=XX,   width=.95)[1], c='r', ls='--')

    plt.title(titles[i])
    plt.show()
```

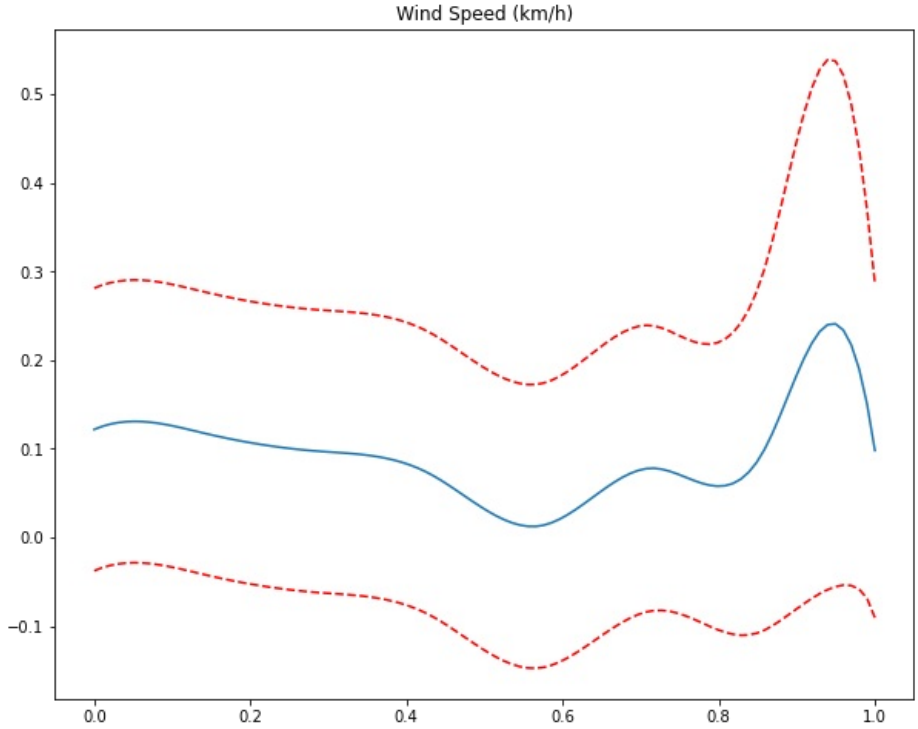
```
N/A% (0 of 11) |                        | Elapsed Time: 0:00:00 ETA:  --:--:--
```

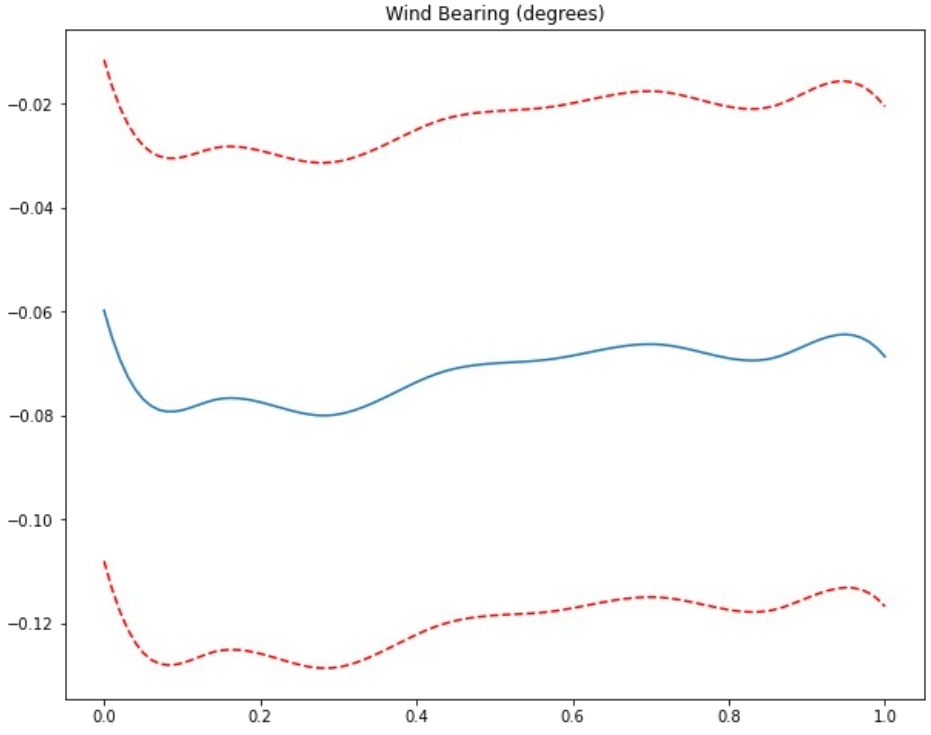
```
(400, 42)
```

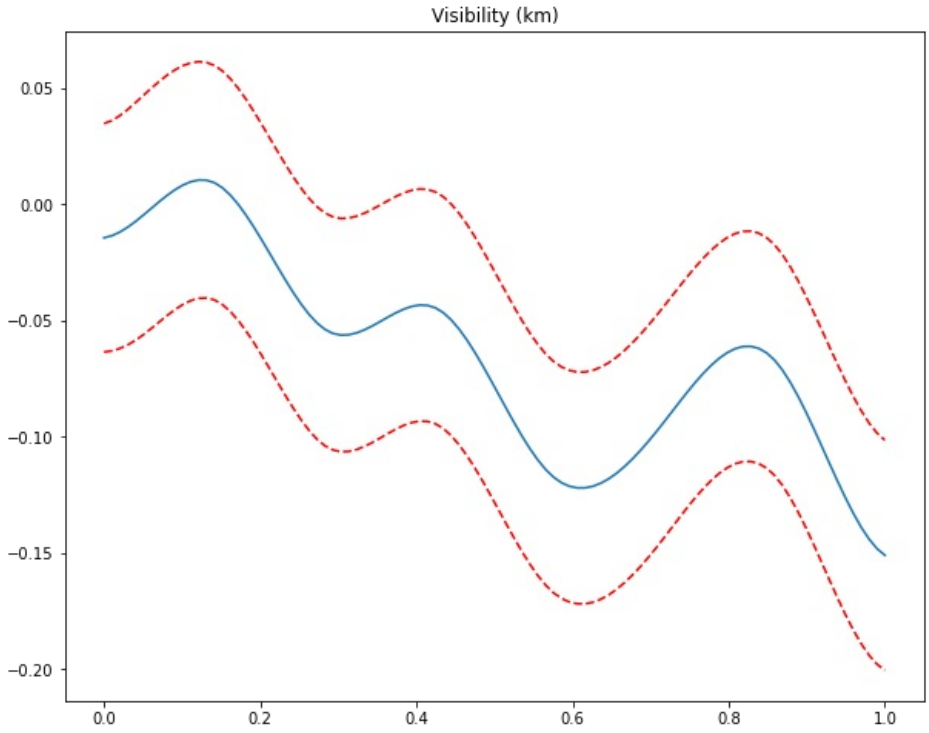
```
100% (11 of 11) |########################| Elapsed Time: 0:01:41 Time:  0:01:41
0.010900262674145173
0.10440432306253018
```

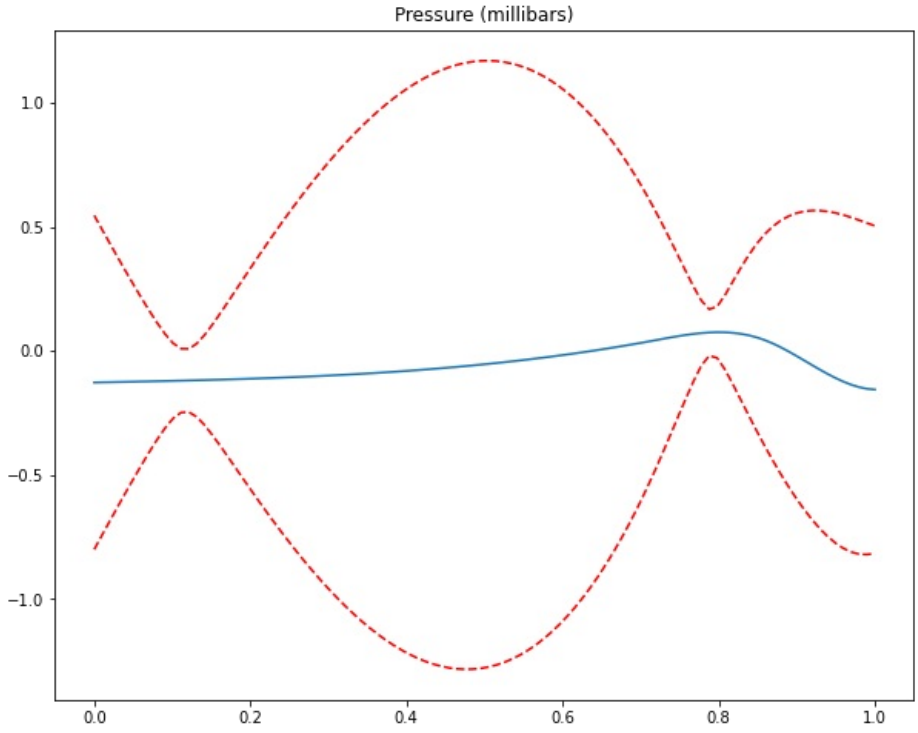


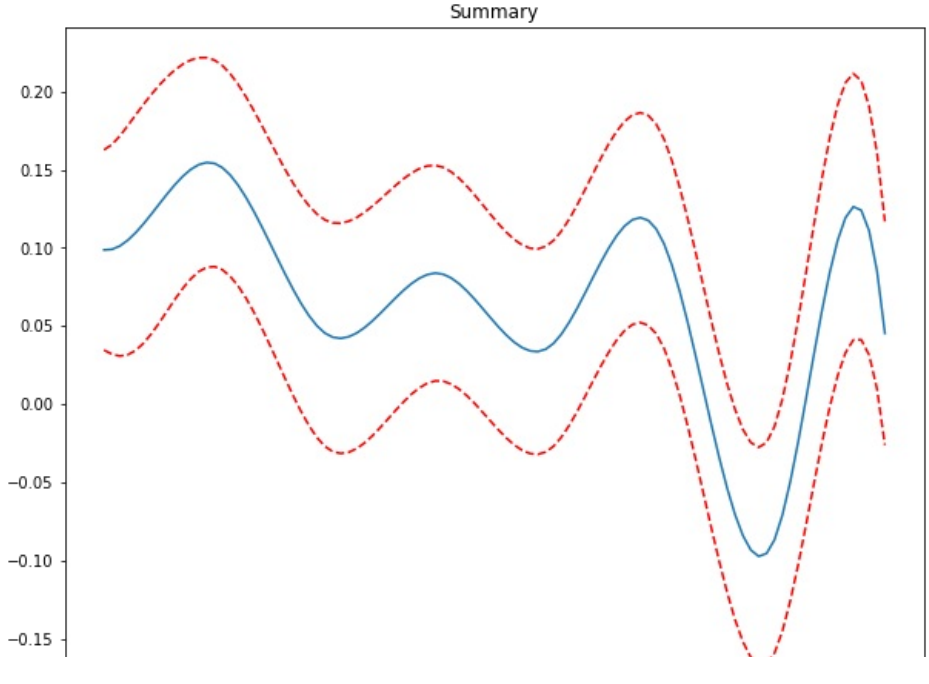

0.2
0.0
−0.2
−0.4
−0.6
0.0
0.2
0.4
0.6
0.8
1.0


Wind Speed (km/h)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0

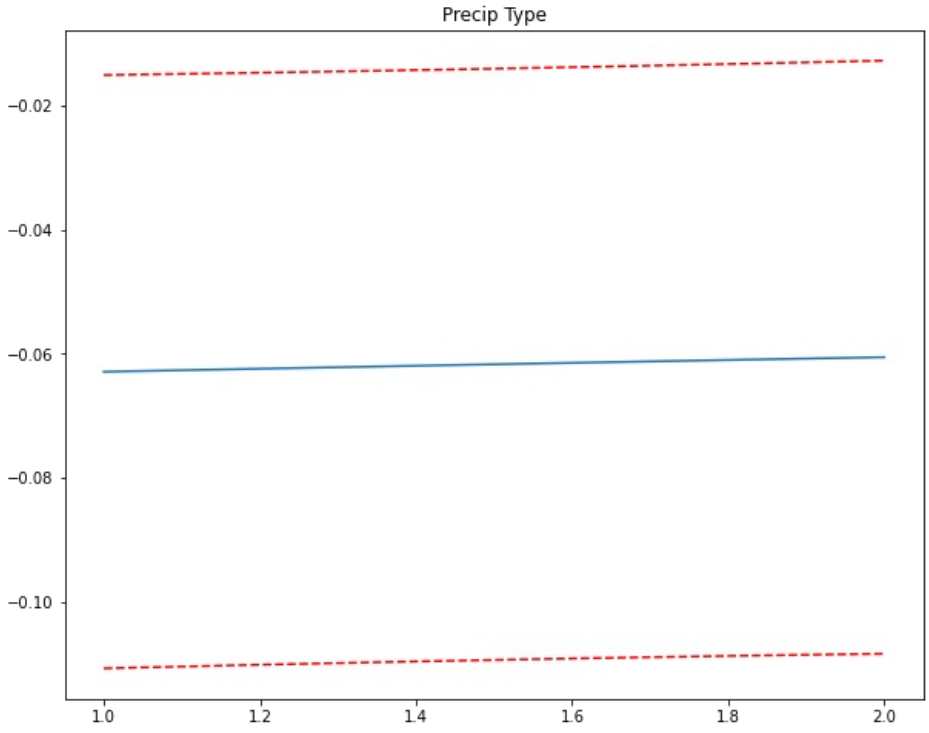

Wind Bearing (degrees)
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0

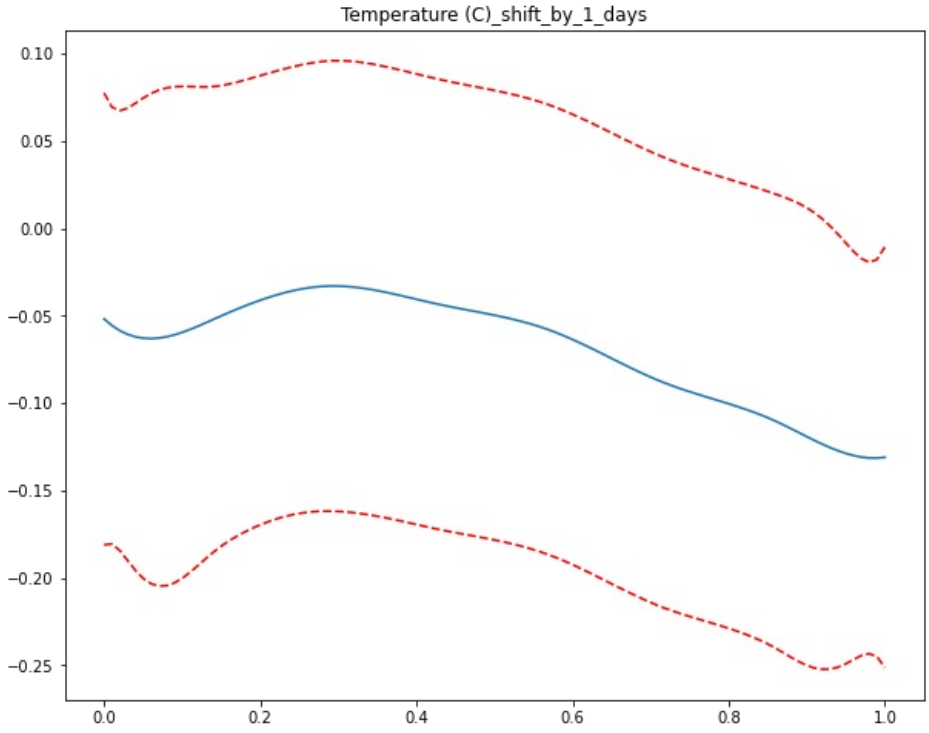

Visibility (km)
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0.0
0.2
0.4
0.6
0.8
1.0

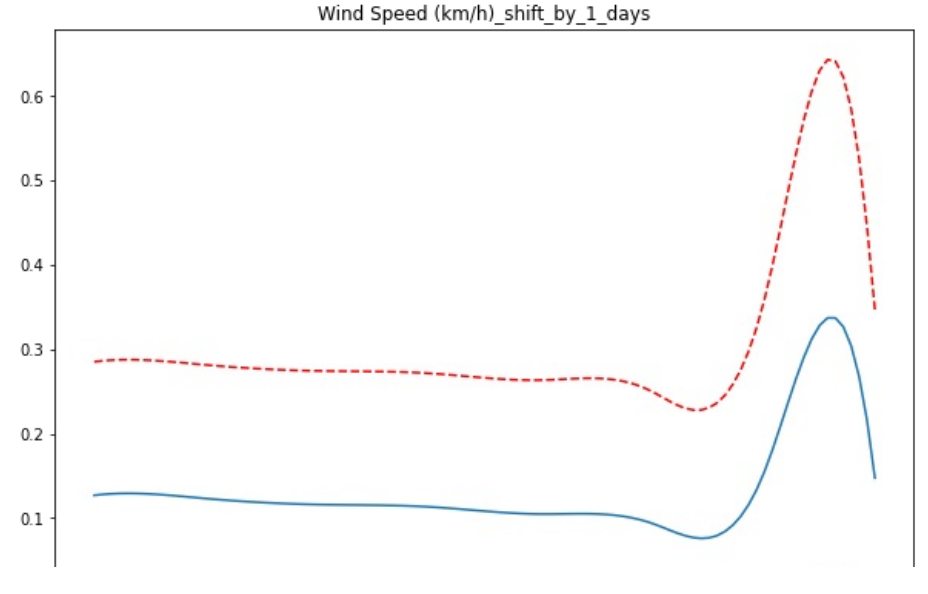

Pressure (millibars)
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

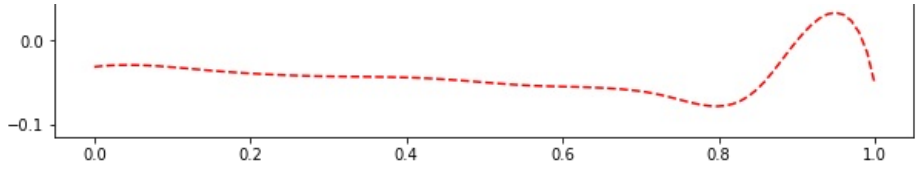

Summary
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15

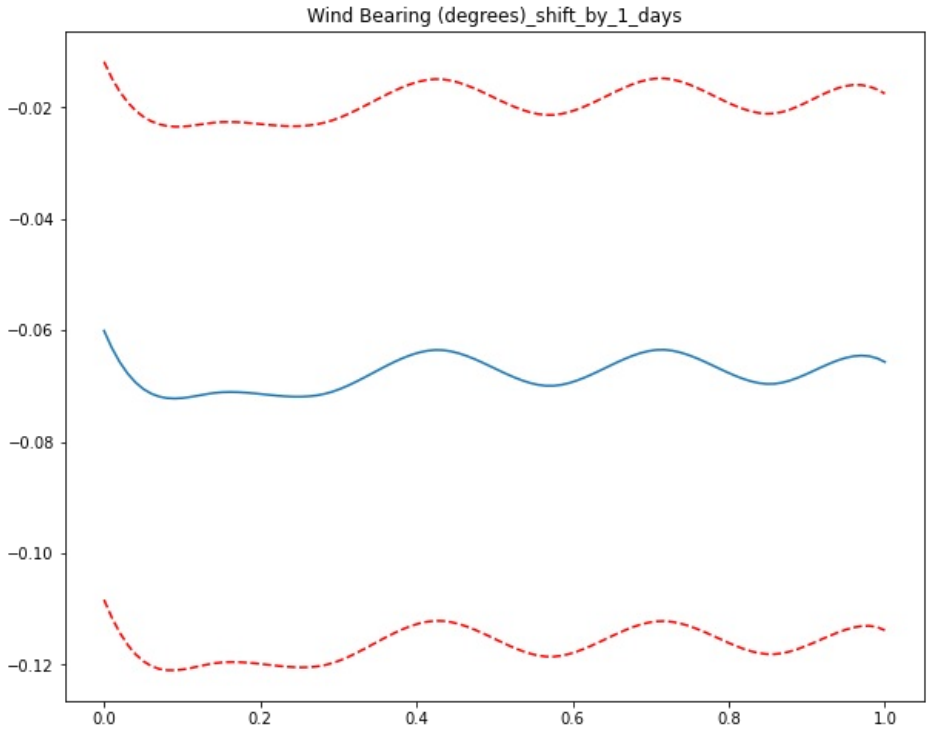

0
5
10
15
20
25

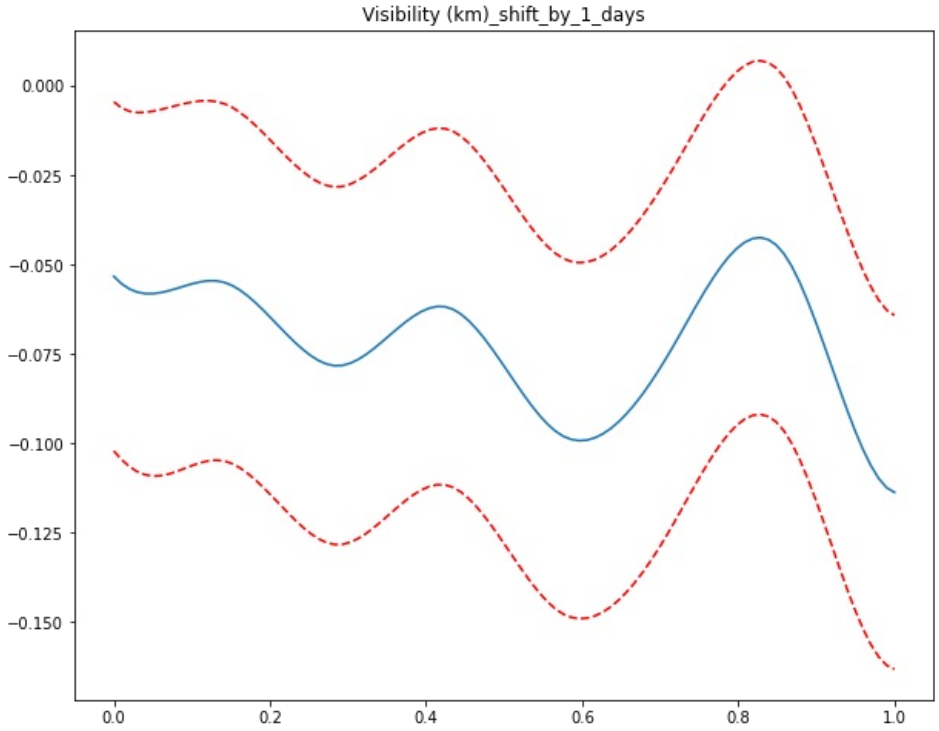

Precip Type
−0.02
−0.04
−0.06
−0.08
−0.10
1.0
1.2
1.4
1.6
1.8
2.0

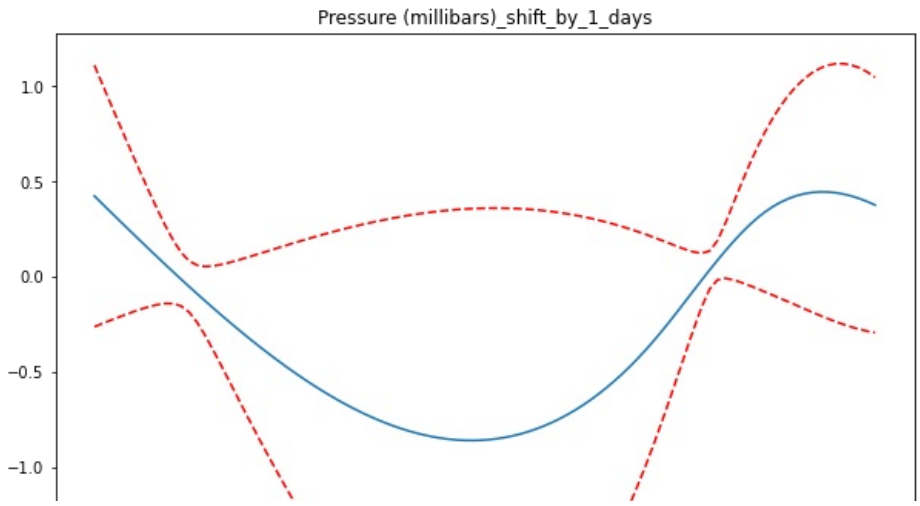

Temperature (C)_shift_by_1_days
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
0.0
0.2
0.4
0.6
0.8
1.0

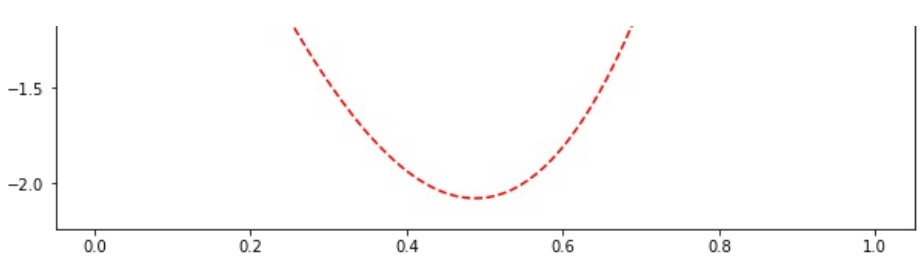

Wind Speed (km/h)_shift_by_1_days
0.6
0.5
0.4
0.3
0.2
0.1

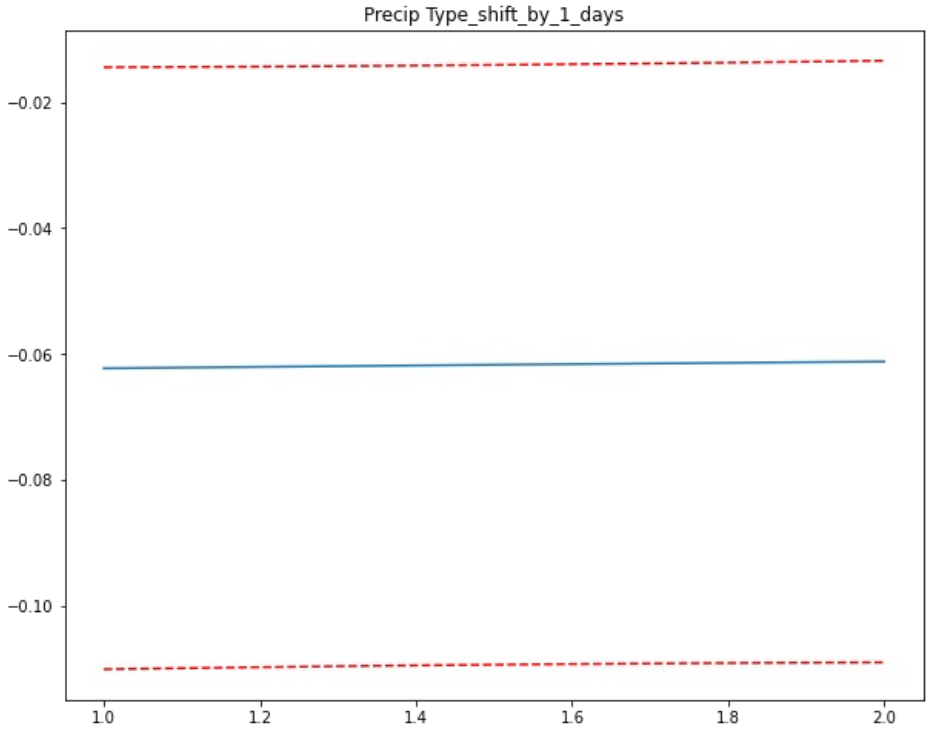

0.0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0

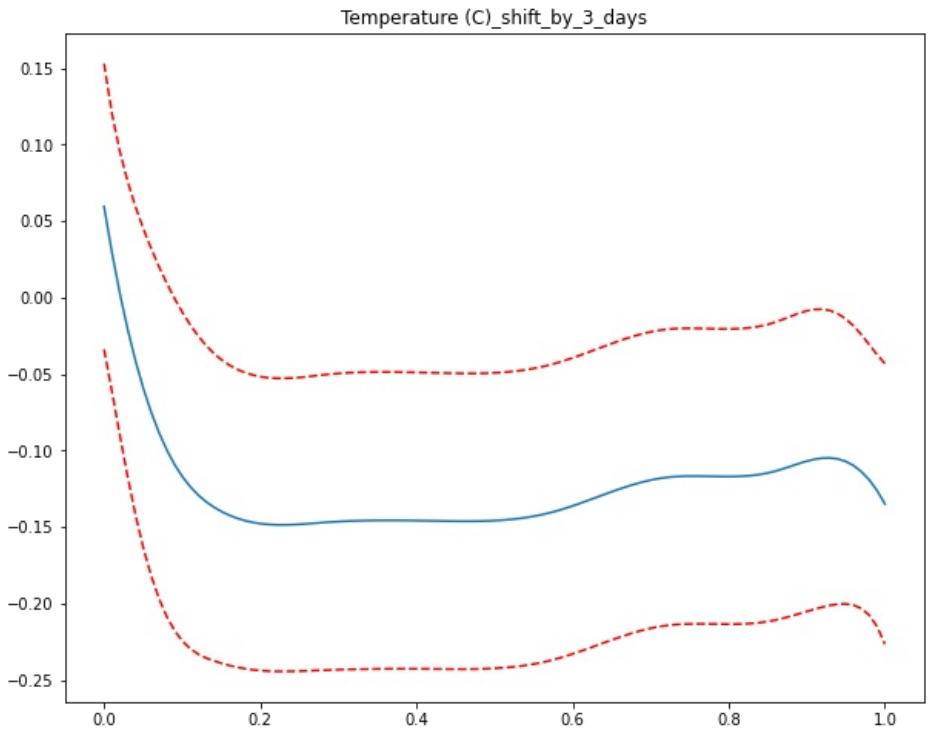

Wind Bearing (degrees)_shift_by_1_days
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0

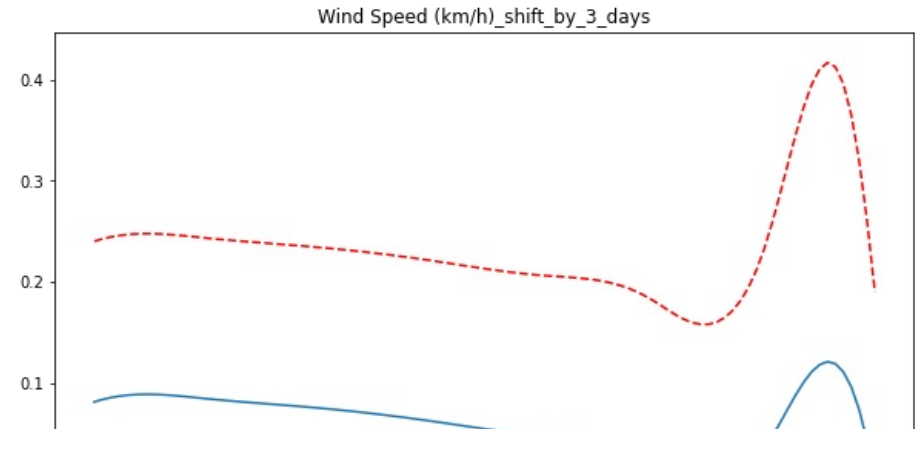

Visibility (km)_shift_by_1_days
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−0.150
0.0
0.2
0.4
0.6
0.8
1.0

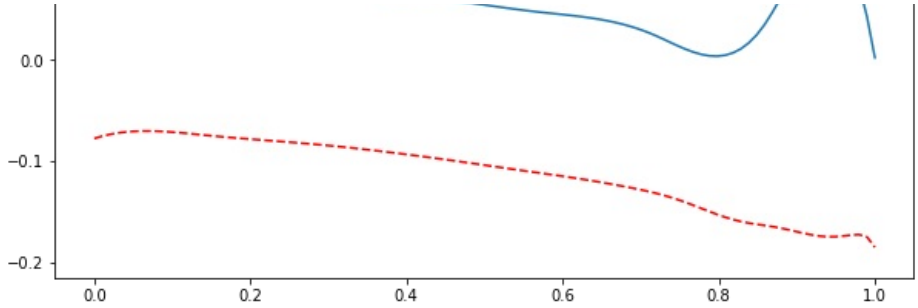

Pressure (millibars)_shift_by_1_days
1.0
0.5
0.0
−0.5
−1.0

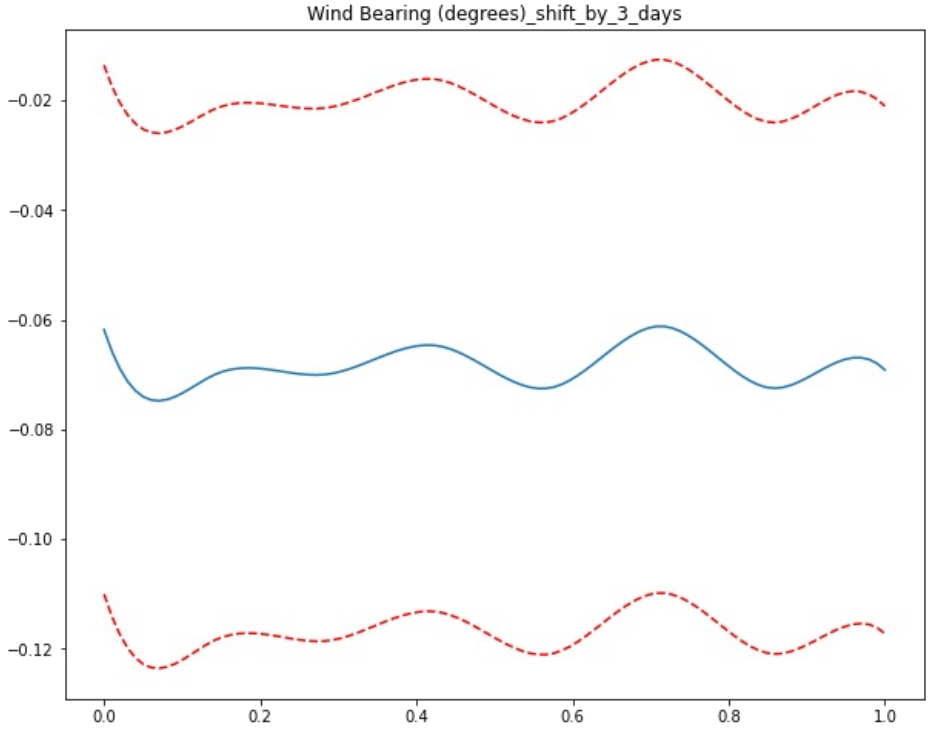

−1.5
−2.0
0.0
0.2
0.4
0.6
0.8
1.0

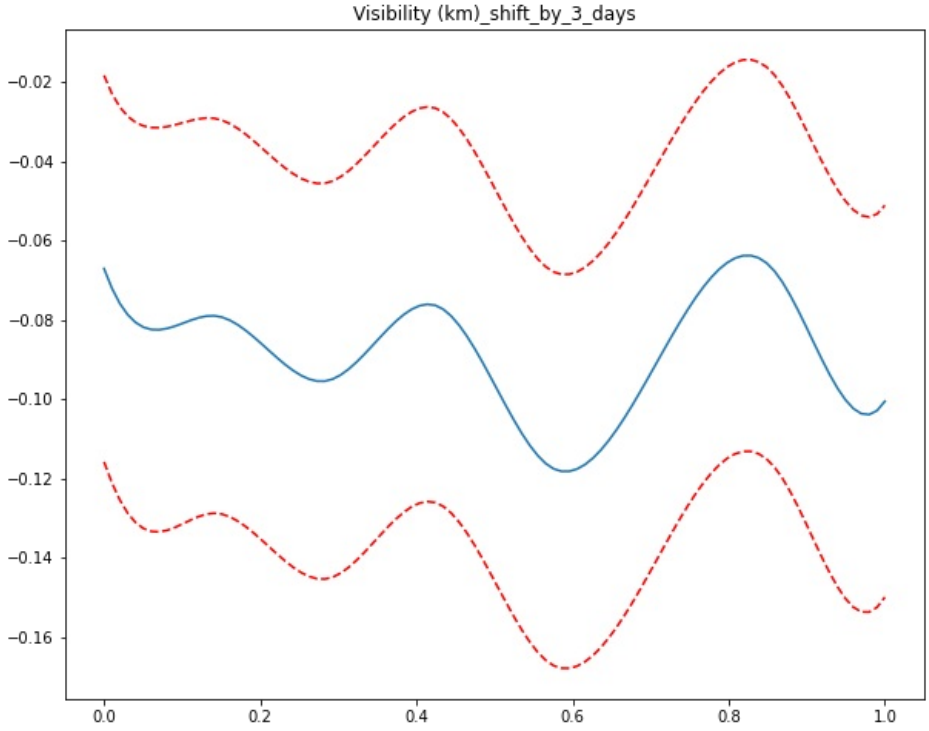

Precip Type_shift_by_1_days
−0.02
−0.04
−0.06
−0.08
−0.10
1.0
1.2
1.4
1.6
1.8
2.0

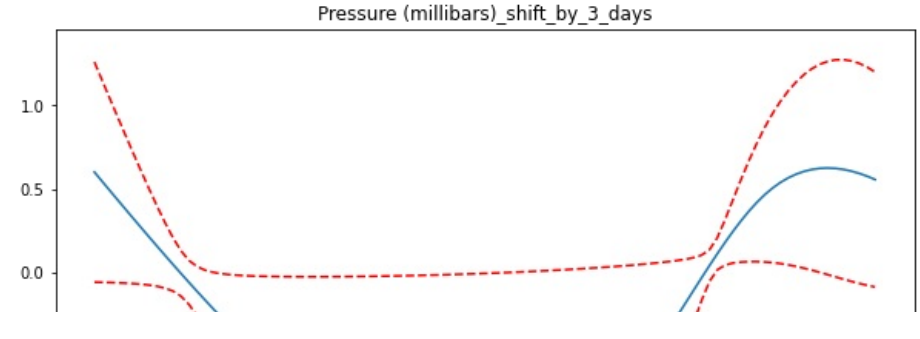

Temperature (C)_shift_by_3_days
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
0.0
0.2
0.4
0.6
0.8
1.0

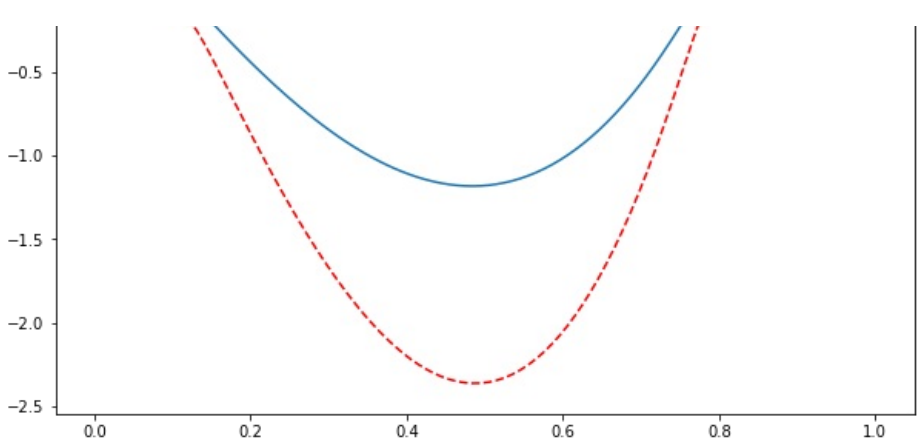

Wind Speed (km/h)_shift_by_3_days
0.4
0.3
0.2
0.1

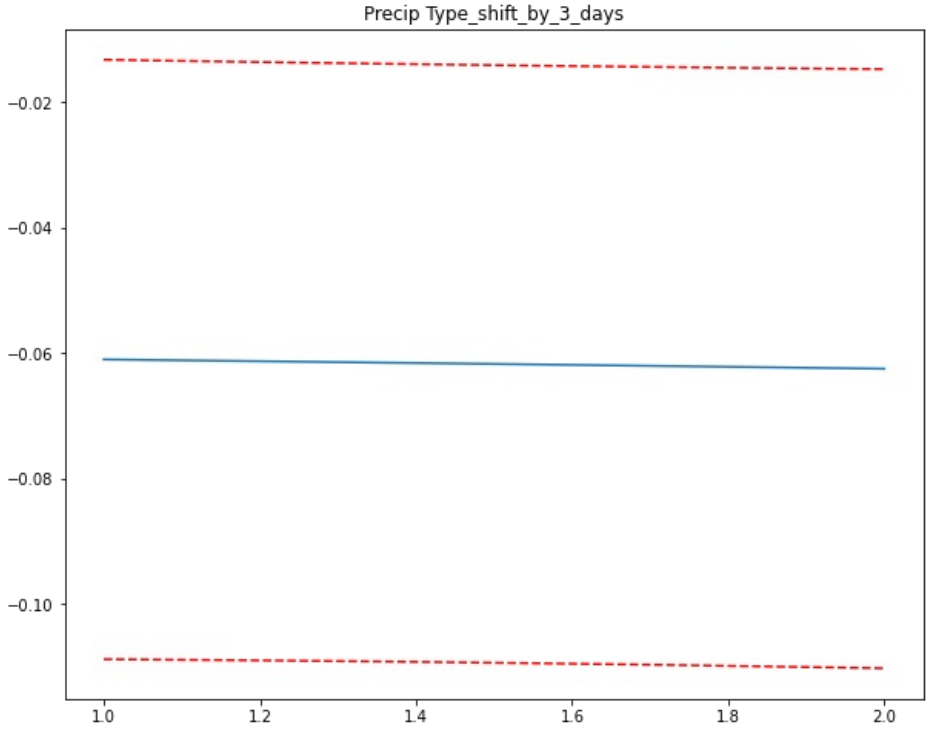

0.0
−0.1
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

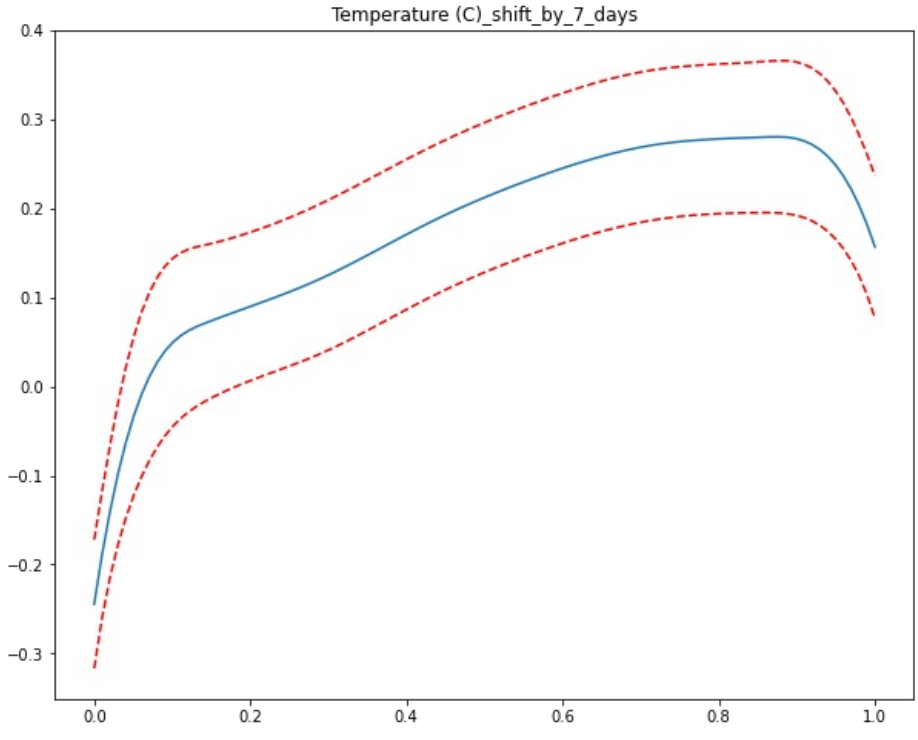

Wind Bearing (degrees)_shift_by_3_days
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0

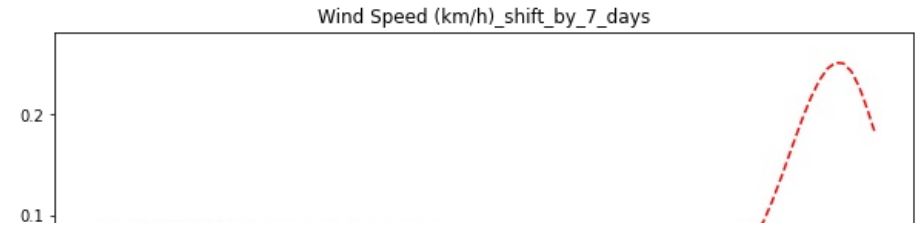

Visibility (km)_shift_by_3_days
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0

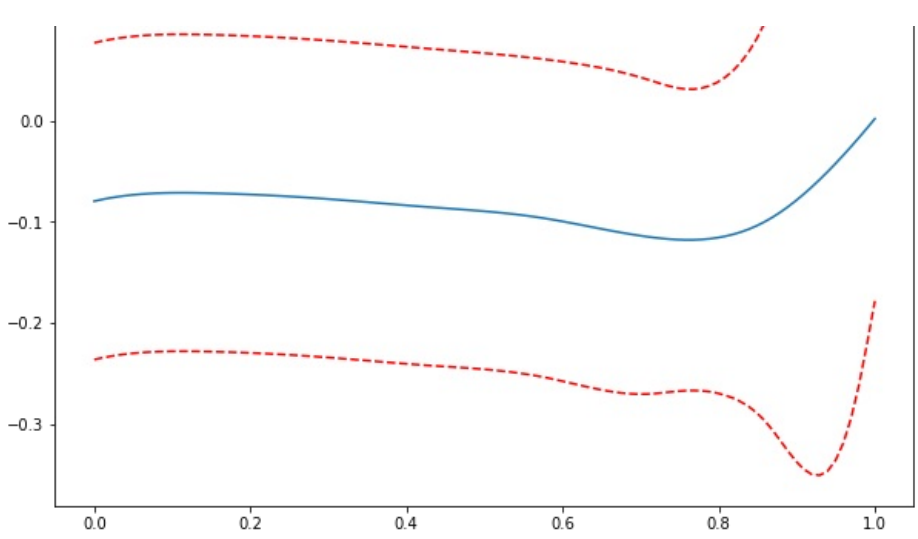

Pressure (millibars)_shift_by_3_days
1.0
0.5
0.0

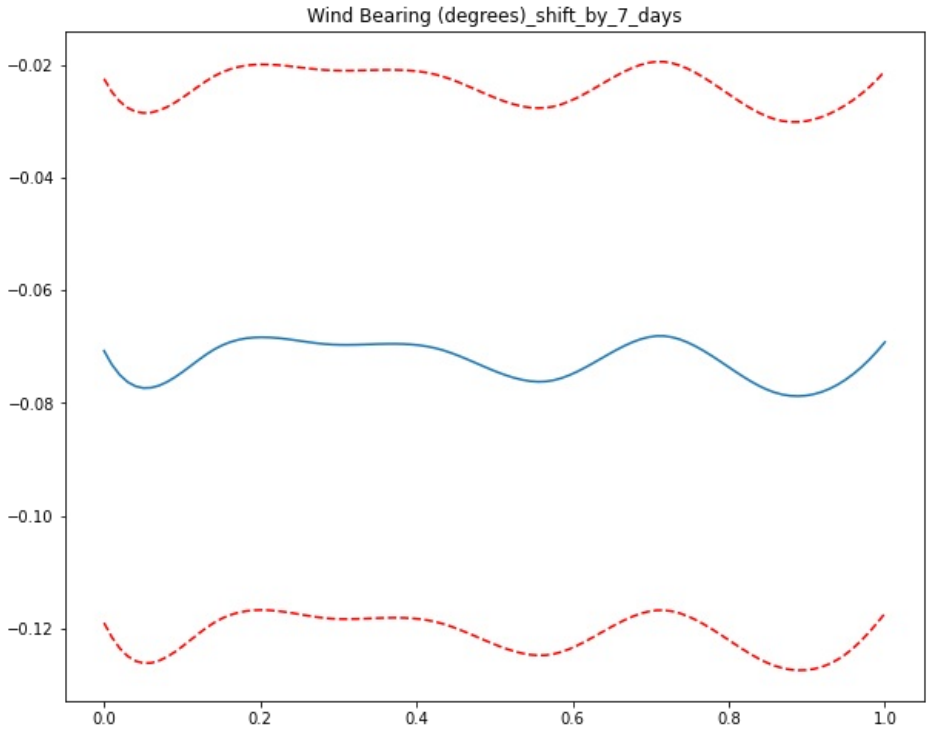

−0.5
−1.0
−1.5
−2.0
−2.5
0.0
0.2
0.4
0.6
0.8
1.0

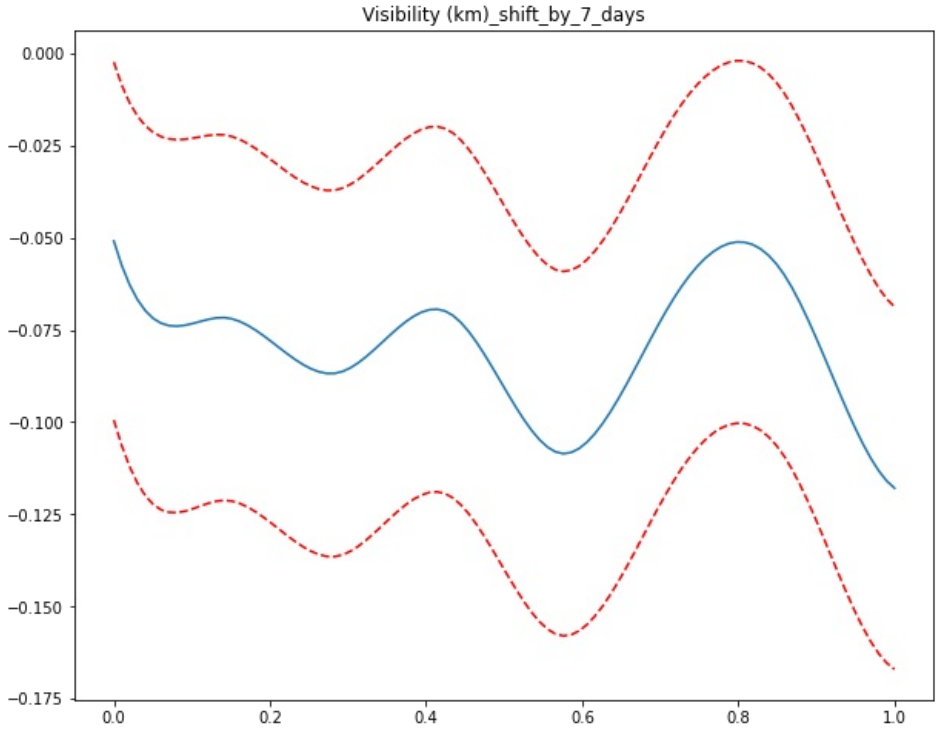

Precip Type_shift_by_3_days
−0.02
−0.04
−0.06
−0.08
−0.10
1.0
1.2
1.4
1.6
1.8
2.0

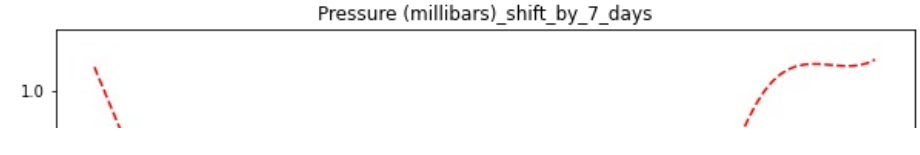

Temperature (C)_shift_by_7_days
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.0
0.2
0.4
0.6
0.8
1.0

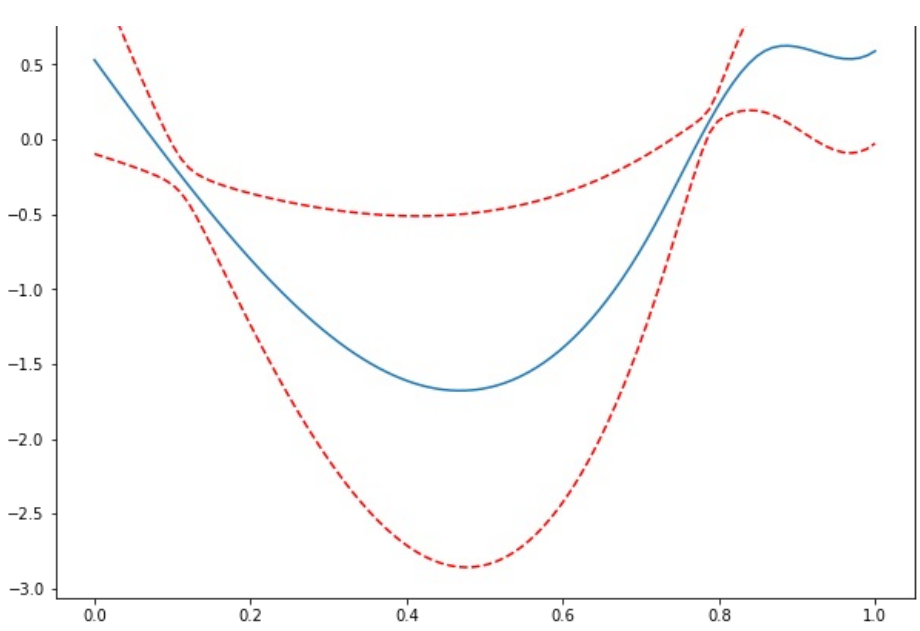

Wind Speed (km/h)_shift_by_7_days
0.2
0.1


0.0
−0.1
−0.2
−0.3
0.0
0.2
0.4
0.6
0.8
1.0

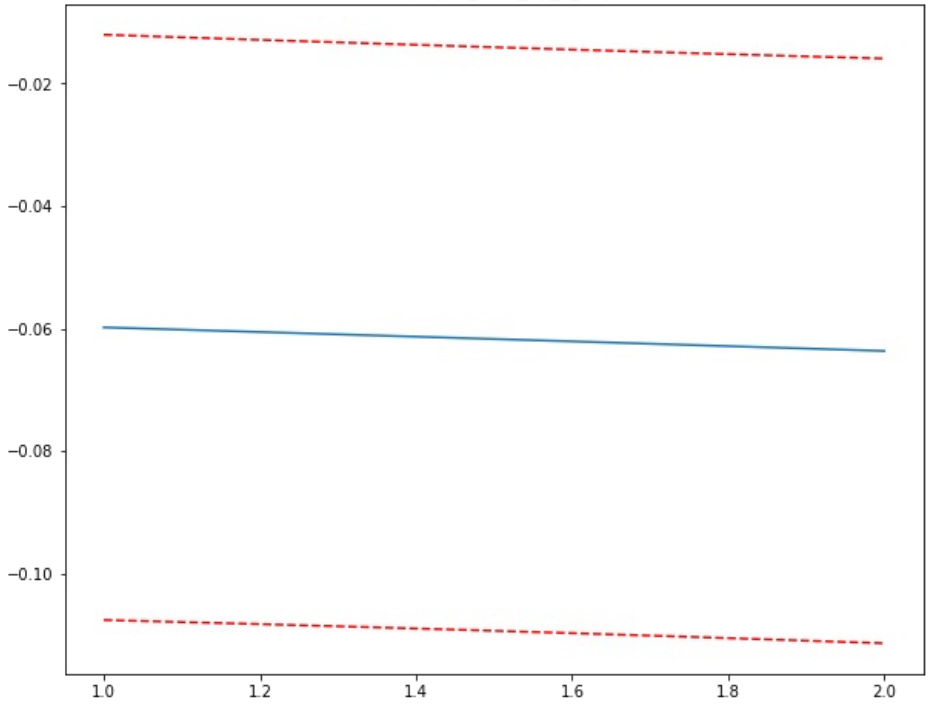

Wind Bearing (degrees)_shift_by_7_days
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0


Visibility (km)_shift_by_7_days
0.000
−0.025
−0.050
−0.075
−0.100
−0.125
−0.150
−0.175
0.0
0.2
0.4
0.6
0.8
1.0

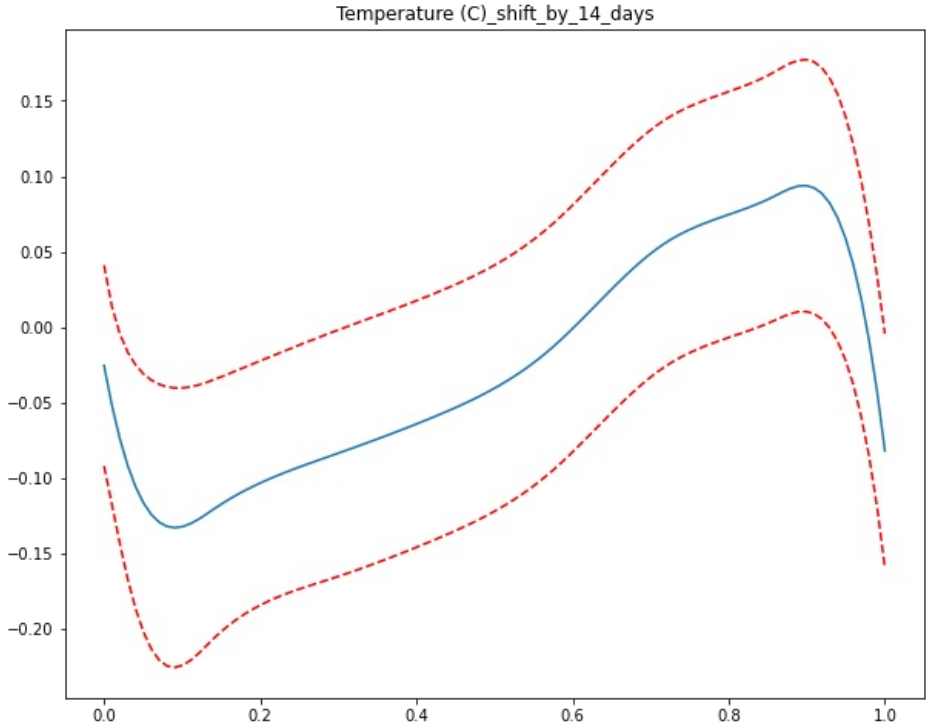

Pressure (millibars)_shift_by_7_days
1.0



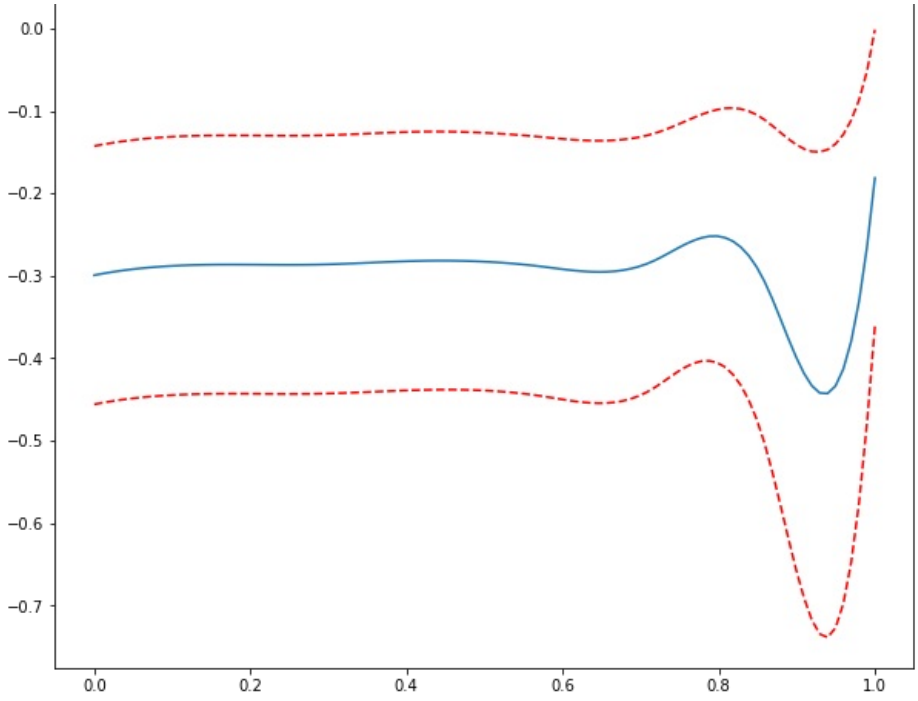
Precip Type_shift_by_7_days



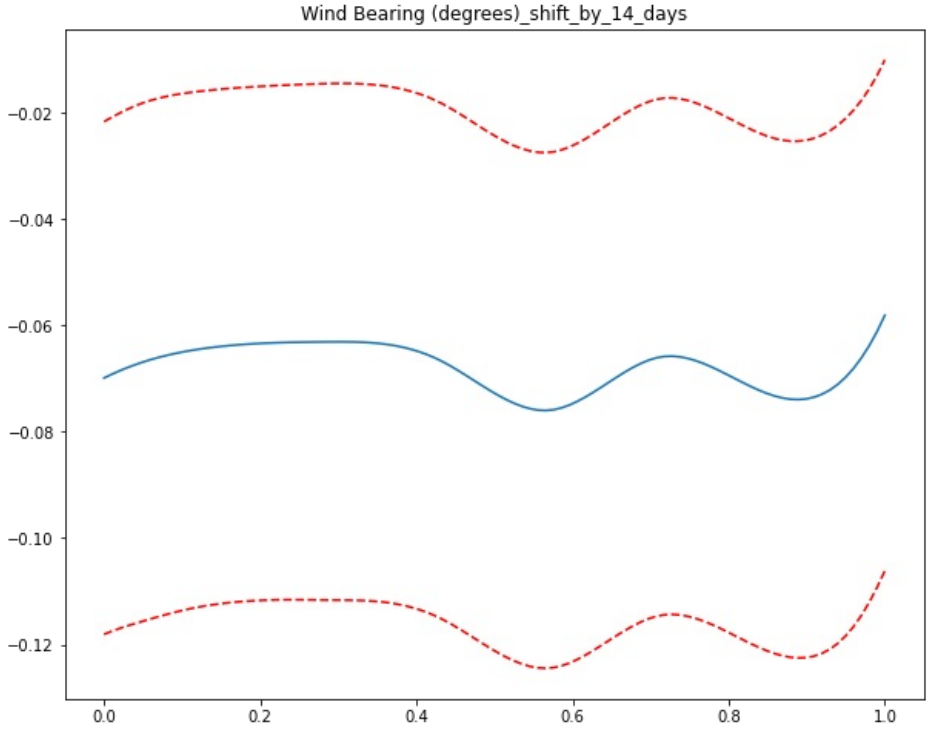
Temperature (C)_shift_by_14_days



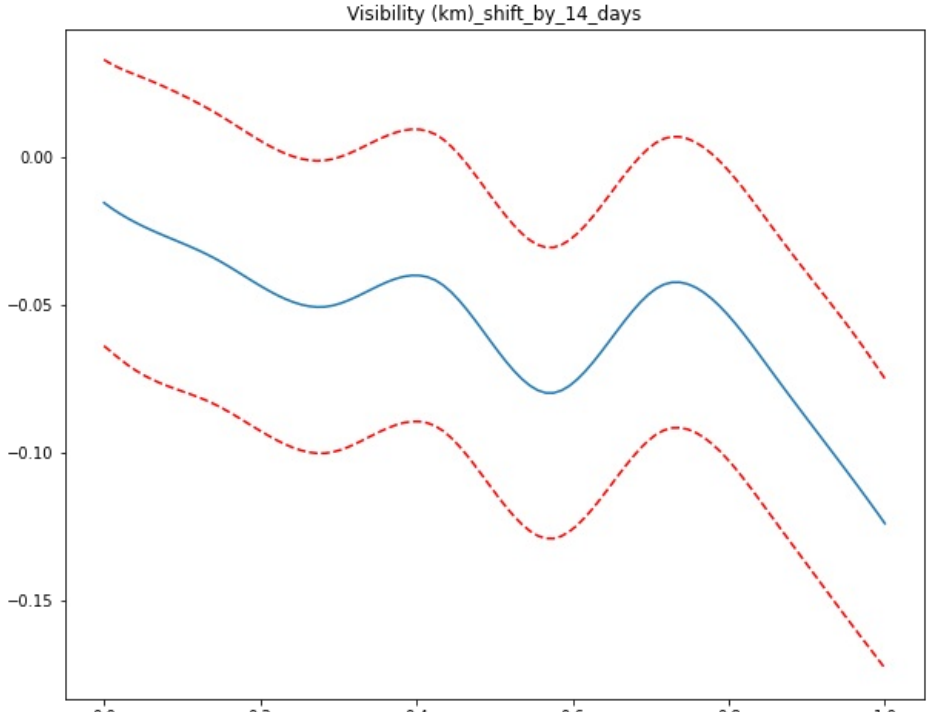
Wind Speed (km/h)_shift_by_14_days

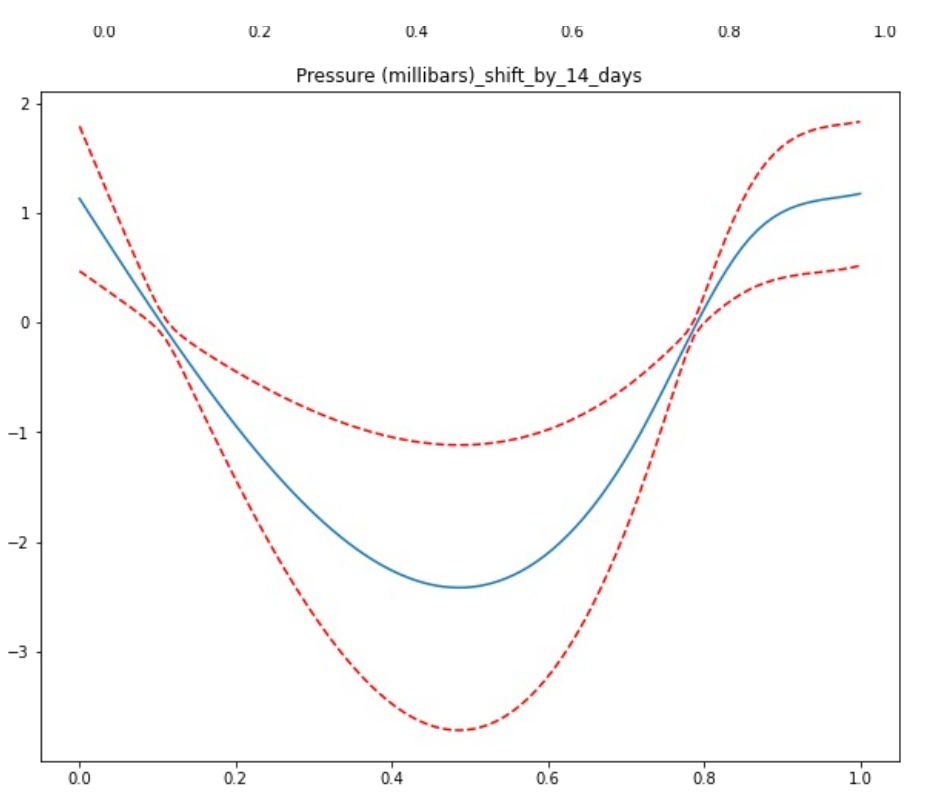


Wind Bearing (degrees)_shift_by_14_days

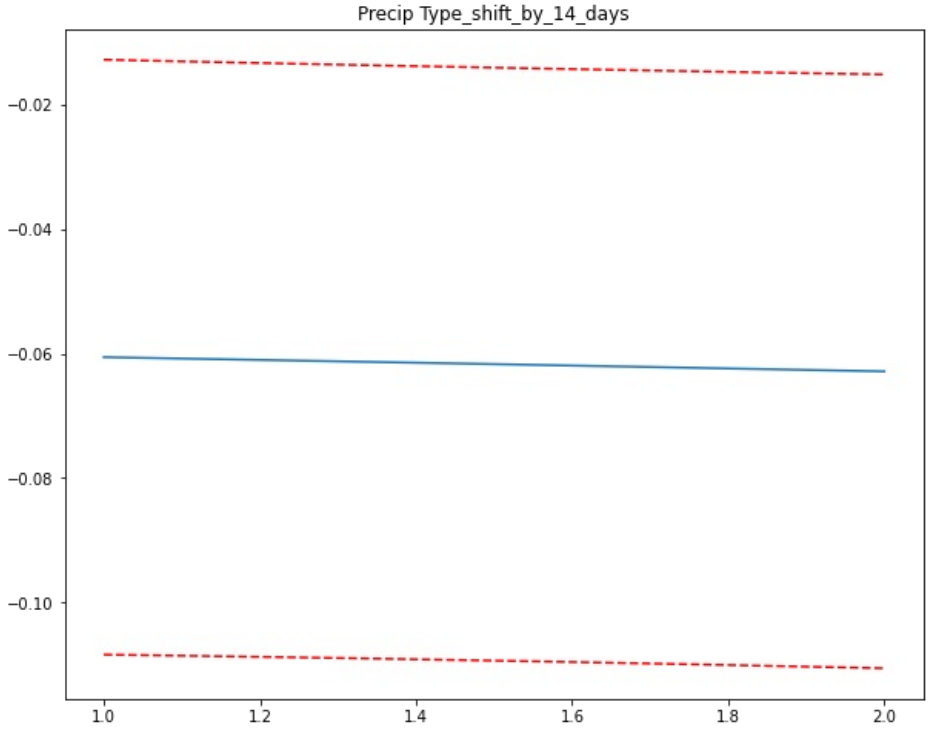


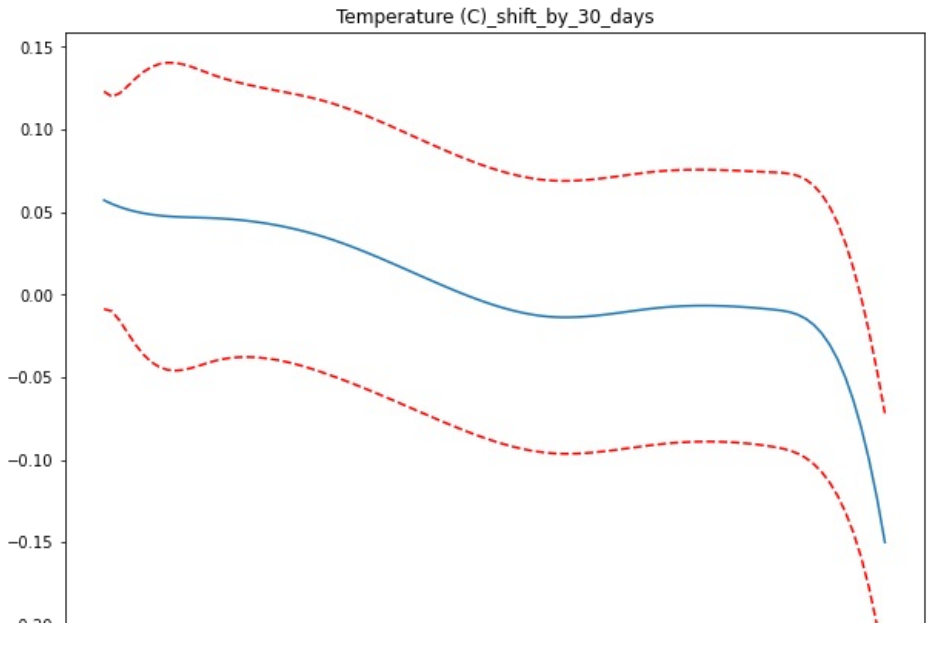
Visibility (km)_shift_by_14_days

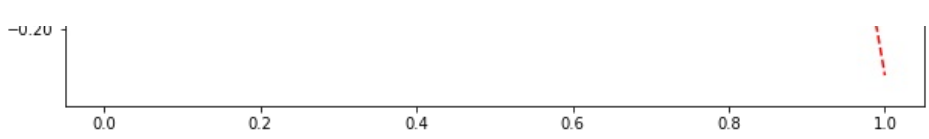


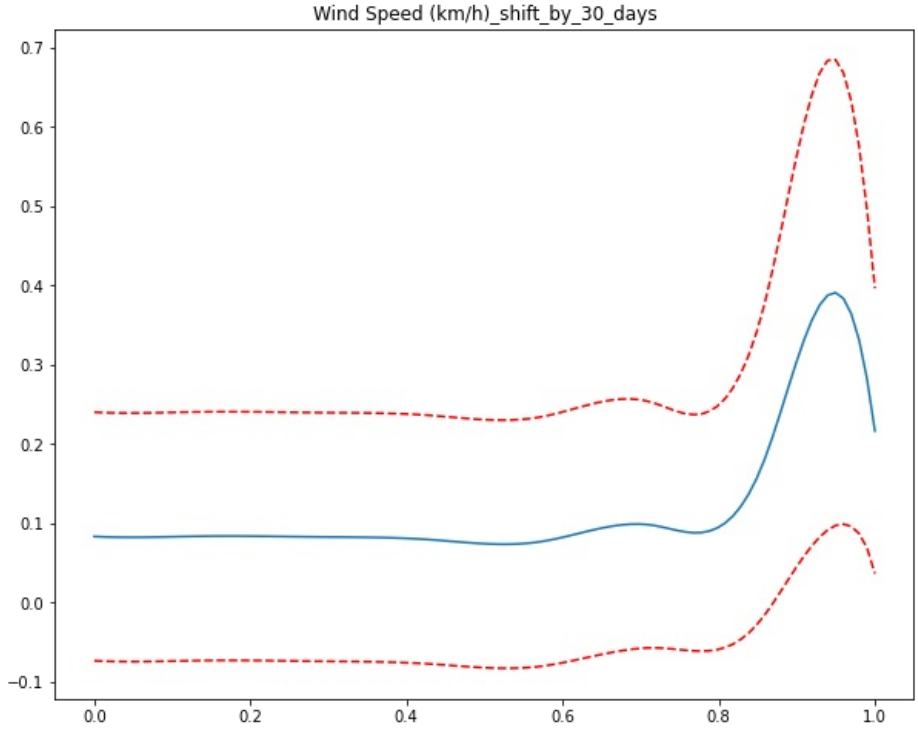

0.0
0.2
0.4
0.6
0.8
1.0

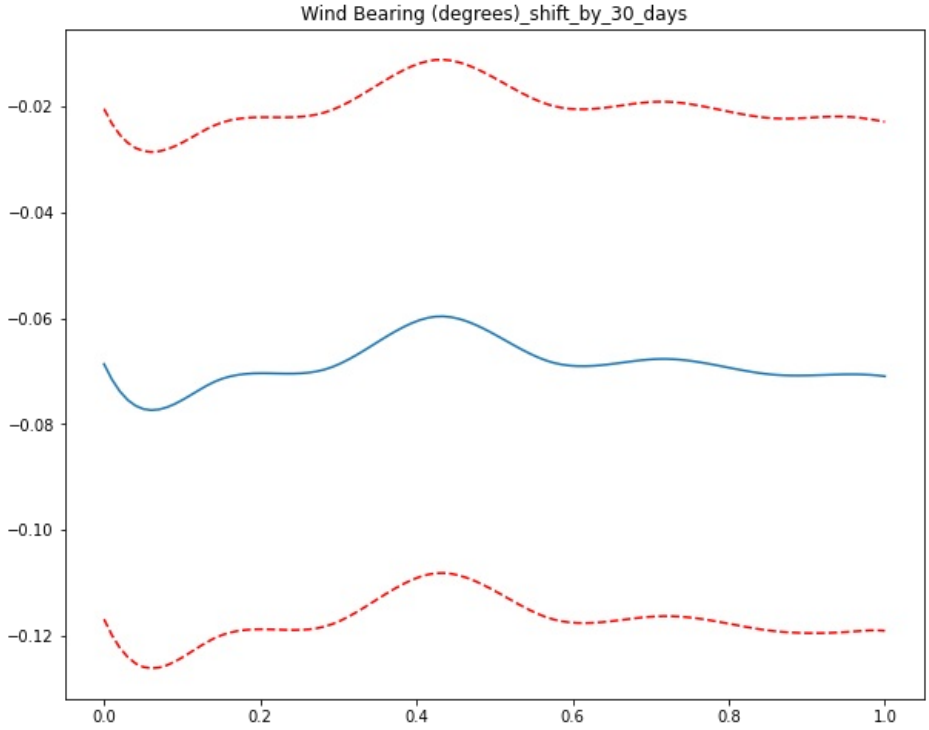
Pressure (millibars)_shift_by_14_days
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0

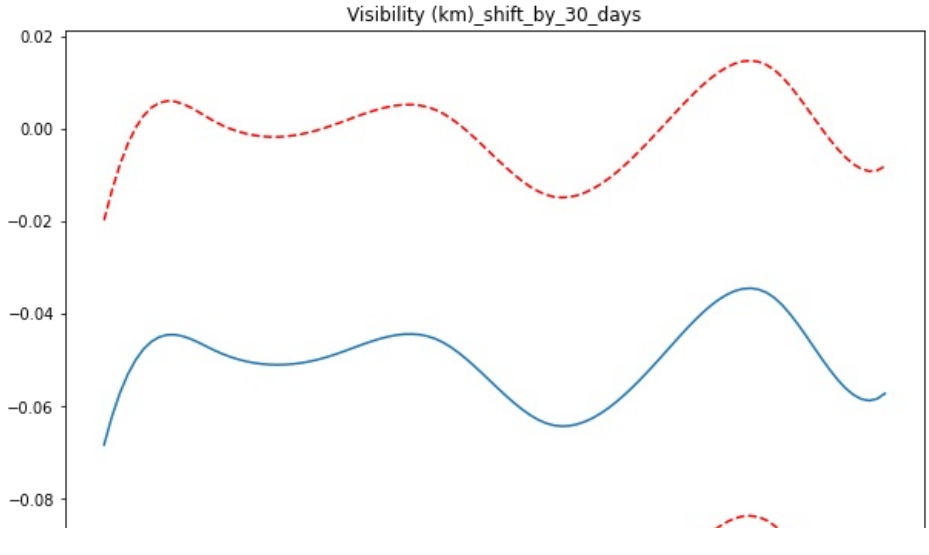

Precip Type_shift_by_14_days
−0.02
−0.04
−0.06
−0.08
−0.10
1.0
1.2
1.4
1.6
1.8
2.0

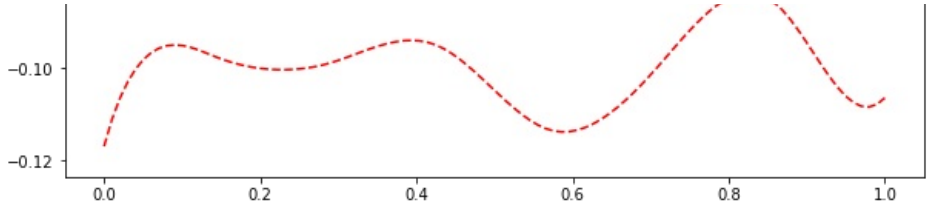

Temperature (C)_shift_by_30_days
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15

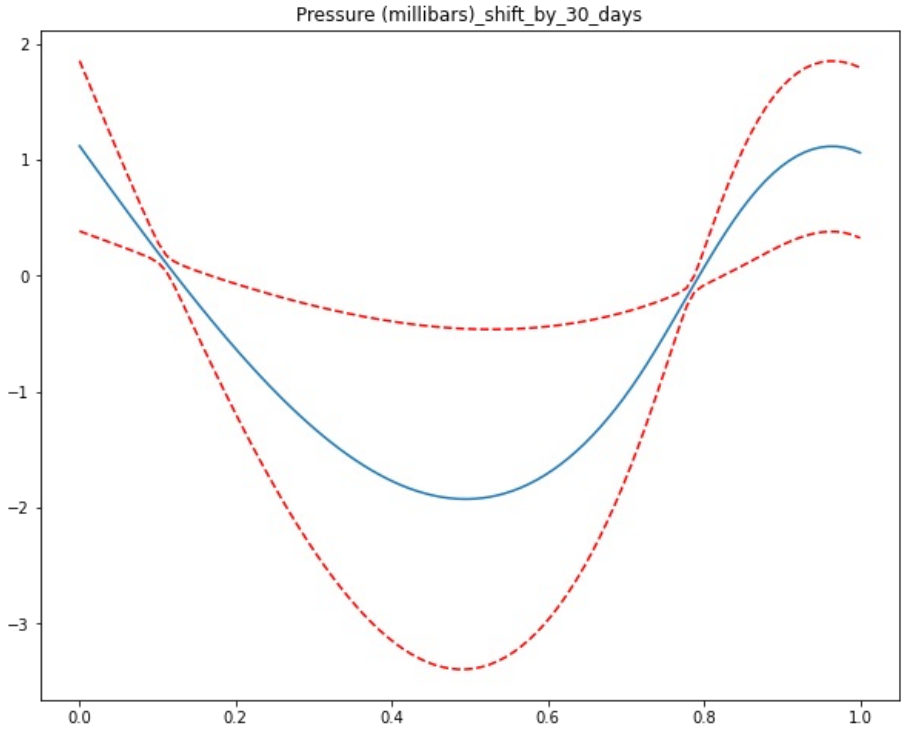

0.0
0.2
0.4
0.6
0.8
1.0

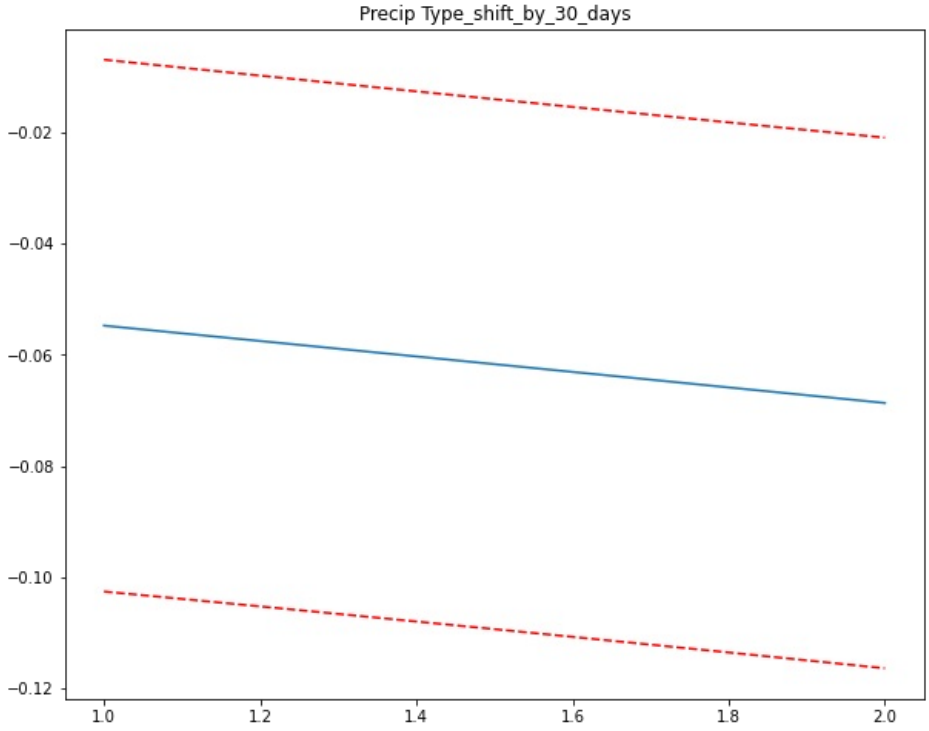

Wind Speed (km/h)_shift_by_30_days
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
0.0
0.2
0.4
0.6
0.8
1.0

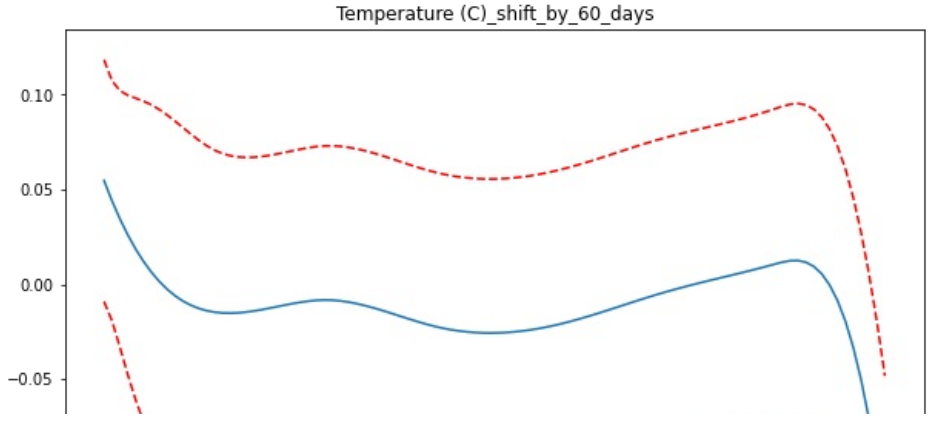

Wind Bearing (degrees)_shift_by_30_days
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0

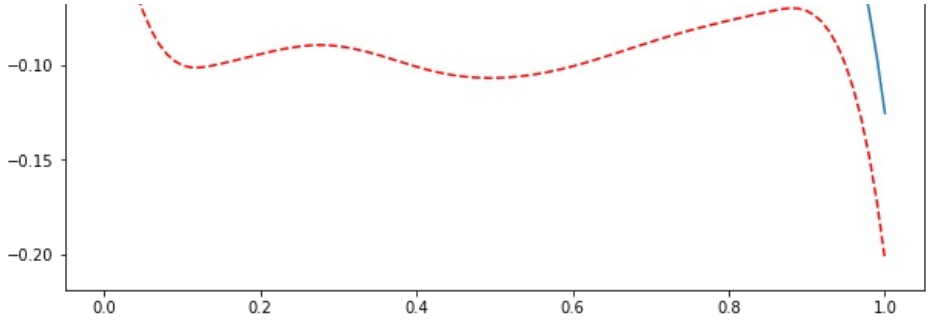

Visibility (km)_shift_by_30_days
0.02
0.00
−0.02
−0.04
−0.06
−0.08

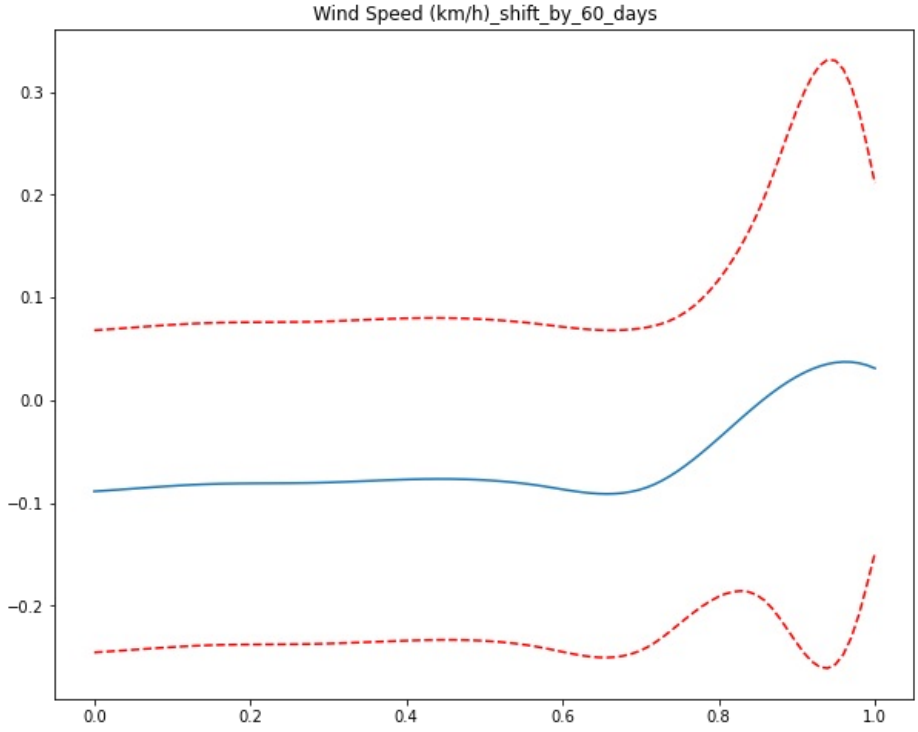

−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0

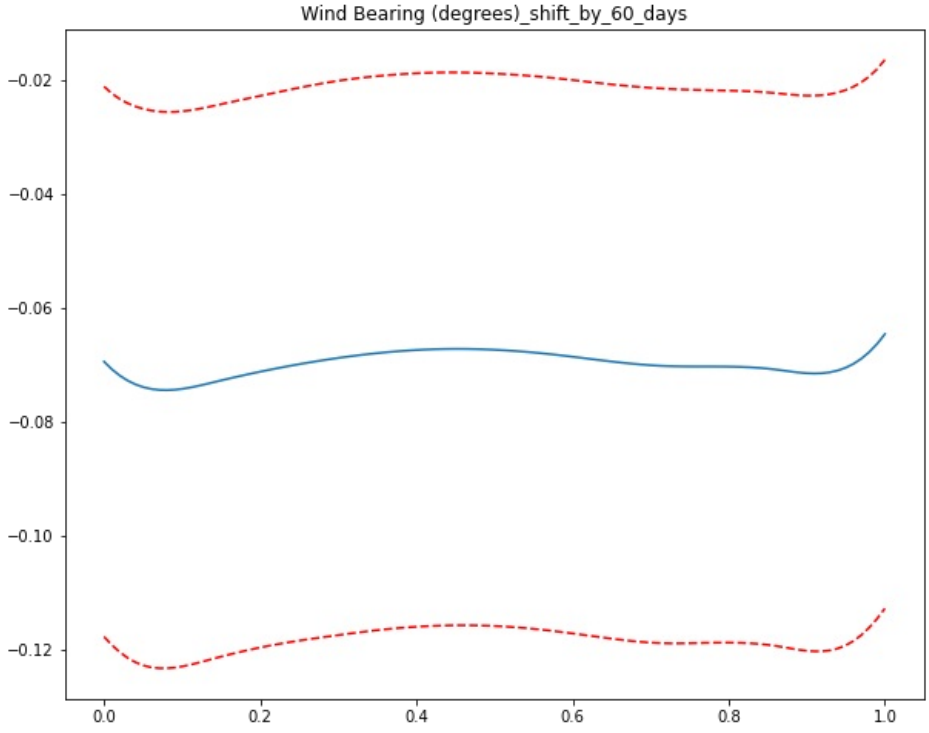

Pressure (millibars)_shift_by_30_days
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0

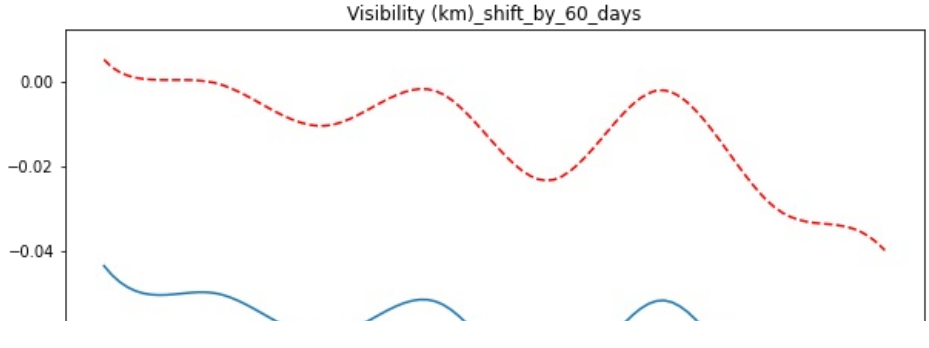

Precip Type_shift_by_30_days
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
1.0
1.2
1.4
1.6
1.8
2.0

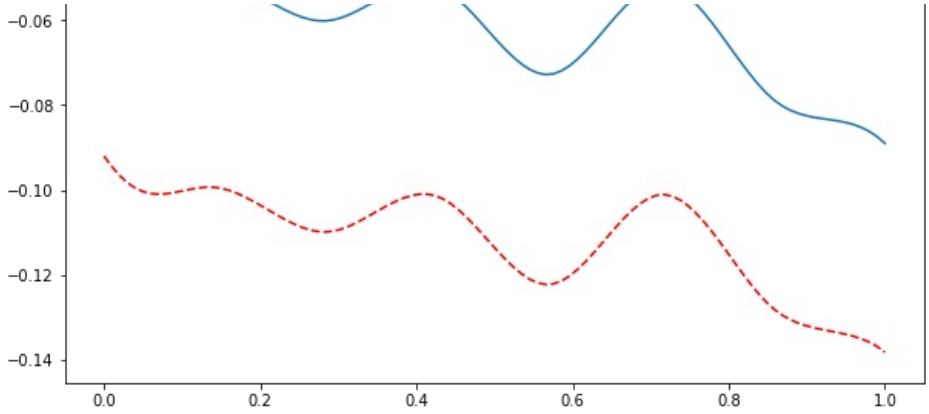

Temperature (C)_shift_by_60_days
0.10
0.05
0.00
−0.05


−0.10
−0.15
−0.20
0.0
0.2
0.4
0.6
0.8
1.0


Wind Speed (km/h)_shift_by_60_days
0.3
0.2
0.1
0.0
−0.1
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


Wind Bearing (degrees)_shift_by_60_days
−0.02
−0.04
−0.06
−0.08
−0.10
−0.12
0.0
0.2
0.4
0.6
0.8
1.0

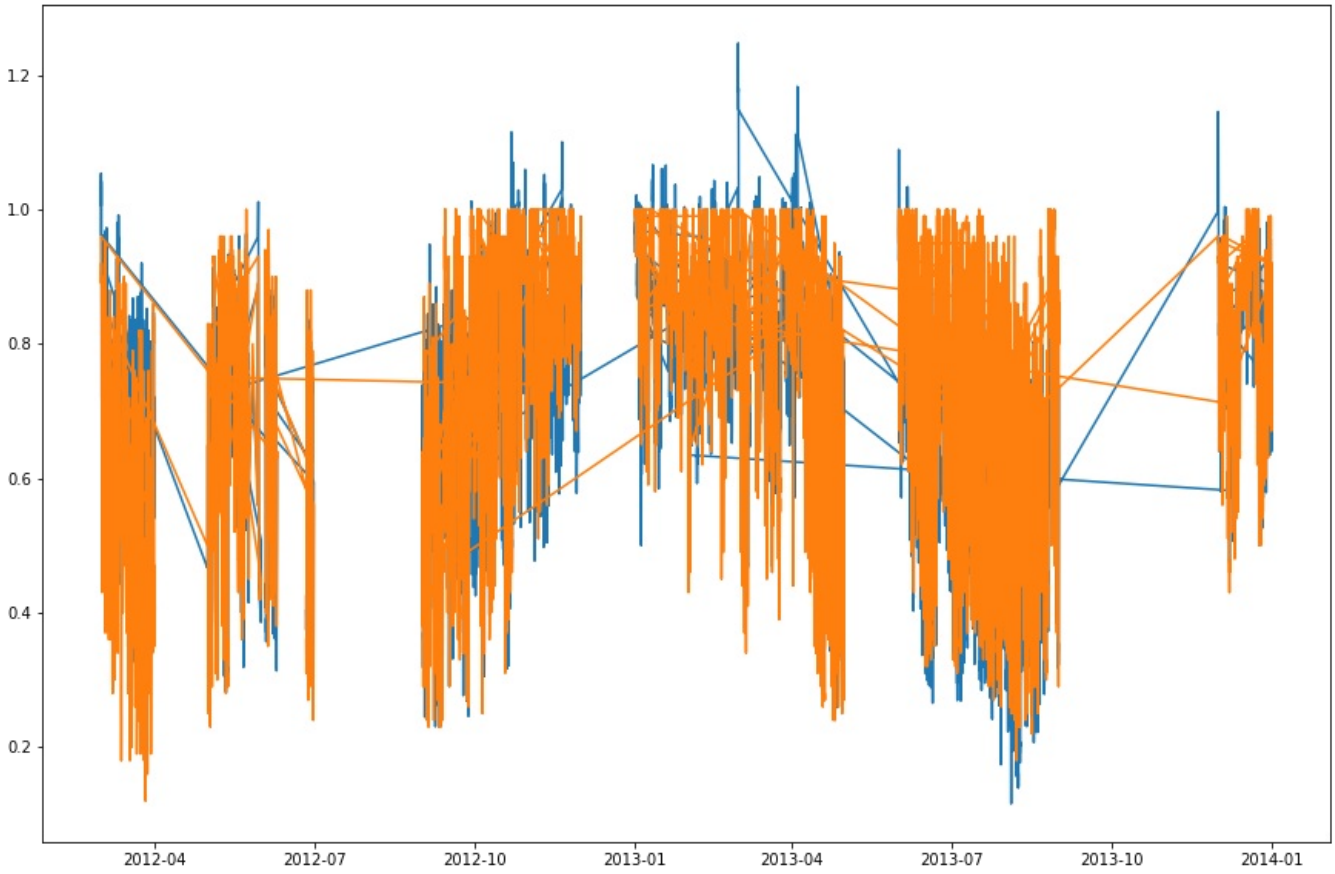

Visibility (km)_shift_by_60_days
0.00
−0.02
−0.04



In [146…
```
dataset = pd.read_csv('weatherHistory.csv')
```

In [153…
```
plt.figure(figsize=(15,10))
plt.plot( pd.to_datetime(dataset['Formatted Date'].iloc[data_test.index],utc=True), gam.predict(data_test[[col fo
plt.plot(pd.to_datetime(dataset['Formatted Date'].iloc[data_test.index],utc=True),data_test['Humidity'] )
```

Out[153… `[<matplotlib.lines.Line2D at 0x1f98664b730>]`



Conclusion,

We Got Best Result of -

- RMSE score: 0.09669071699053888

- MSE Score: 0.009349094752144484

  On top of that based on Results of Above Graph we were able to Capture even the Trends in Humidity for most of the entries in Test Sets. Unfortunately we cannot create confusion matrix as it is a Regression PRoblem.