Methodologies and Experiments of Predicting and Forecasting Household Temperature

Name: Qinyun (Peter) Yu, 501137007

Date: July 12th, 2022

Supervised by: Dr. Alan Fung

Methodologies

Background

Local weather station data is lagged and used in the form of exogenous features to train various forecasting models. The plan is to select three local weather stations within the Greater Toronto Area and have the fourth weather station be within closest proximity of the selected three stations, as a proxy for a house. Then, by using the available weather station data from the four stations, aim to predict the fourth weather station temperatures. This will be done through various models and will be expanded upon.

Data

The data used in this report and for the experiments was obtained from the Environment Canada homepage, which allows the public to access historical weather data from local stations (airports, government headquarters, etc.,). This data can be manually downloaded day-by-day online, or more realistically, can be downloaded via the Environment and Climate Change Canada FTP site, which allows for bulk downloads and is how the current dataset was generated. Additionally, the solar radiation data can be downloaded from RETScreen with an academic license. The temporal resolution for the data is hourly, for a total of ~8760 data points per year.

For this project, data from Toronto International Airport, Buttonville, Toronto City Centre, and Toronto City was obtained, where Toronto City would act as our proxy household.

As there are several exogenous features, and data from Environment Canada is not necessarily clean, several pre-processing steps are required. One, under some of the weather stations, most of the weather labels, wind speed, and wind directions are marked as NaN. For this reason, those columns are dropped. Additionally, there are time periods where no data is recorded from the station. To compensate for these rare occasions, our dataframe is forward filled, and then backward filled for NaNs, where the previous value will be used for the NaN, and if there are any more NaNs, the next value will be used for the previous value. Two, the wind direction data is converted into dummy columns, so they can be used in our models. Three, for forecasting future values, we cannot use present data, so all the columns are lagged by an hour, while the actual (ground truth) temperature is retained. Four, outliers that fall outside the Q1 - 1.5 * IQR and Q3 + 1.5 IQR boundary are dropped. This is because the weather instruments can be faulty and these errors can be attributed towards sensor failure. Five, the weather labels can be quite descriptive, and creating dummy variables for each label would be nonsensical and increase training time significantly. For this reason, the labels were subjectively simplified, whereby labels like “snow showers”, or “heavy snow” would be both assigned to the same label of “snow”, or multi-labels like “Thunderstorms, heavy rain showers, fog” would be assigned to the label of “rain”. Finally, for a side project, the data was then split into day vs. night, and summer vs. winter, so that more specific and more potentially accurate models could be generated (since weather data varies based on the spatial location of the earth and where it sits relative to the moon and sun, which varies seasonally).

Table 1 lists the Environment Canada data used in the analysis. Additional data that were not included in the table include dates, weather labels, and wind speed direction.

Table 1: Weather Station Data

Data	**Field Name**	**Units**
Environment Canada Temperature Reading lagged	Temp_tia_1, temp_tcc_1, temp_b_1	^{o}C
Toronto City Temperature	Actual_temp	^{o}C
Weather station wind speed lagged	Wind_spd_tia_1, wind_spd_tcc_1,wind_spd_b_1	Km/h
Weather station dew point lagged	Dew_point_tia_1, dew_point_tcc_1, dew_point_b_1	^{o}C
Solar radiation lagged	Solar_rad_1	Wh/m^2
Humidex value	Hmdx_tia_1	Dimensionless
Windchill value	Wind_chill_tia_1	Dimensionless/ ^{o}C

Prediction Models

Python statistical libraries were used to predict local weather station data, specifically the Statsmodels, lightgbm, pmdarima, and Tensorflow libraries. Currently the experiments are conducted using an 80/20 train-test split, where training uses 42,288 rows of data from 2016-01-01 to 2021-01-13, and our testing uses 10,573 rows of data from 2021-01-13 to 2022-04-30. In the future, a rolling window train-test split will be introduced to better represent Timeseries data.

LightGBM

The main advantages of using LightGBM are its fast-training time and its hyperparameters that greatly influence the algorithm's ability to generalize interpretation of the data. For fast prototyping, no hyperparameters were tuned as of the current experiments. Instead, the exogenous features found in Table 1 were used for training the model under the default LightGBM settings.

Prophet

Under a MacOS environment, our data was also used to train a Prophet time series forecasting model. No settings were changed, and the results were then plotted using the Plotly interactive library.

Arimax

Using the pmdarima library, auto_arima was imported and used to perform stepwise search to minimize AIC on the aforementioned data train-test split, which resulted in a p, d, q of 4,1,5 respectively.

Performance Metrics

The fitting accuracy of the previously mentioned algorithms were evaluated and validated using several metrics including mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE), and Pearson's coefficient (R^2).

Experimentation

As per the MRP guidelines, to meet the objective of methodology and experiments, chapter 19 of "Introduction to Machine Learning" by Ethem Alpaydin must be met. The following will be the my experimentation setup.

A. Aim of the Study

The aim of the study is to assess the expected error of different algorithms (tree-based, and neural networks) to assess which type of algorithm has lower generalization error on their ability to predict and forecast time series data. More specifically, temperature from local weather stations which aim to replicate household temperatures. This study will be conducted over a number of different datasets which as of now, include day vs. night, and summer vs. winter temperature forecasting. In addition to assessing algorithm performance, the goal will also be to evaluate the importance of various features in their ability to aid in forecasting, as well as evaluating not only model accuracy, but also computation speed, ease of implementation, and interpretability.

B. Selection of the Response Variable

As highlighted in the methodologies section, a number of different metrics are used in order to capture the various model performance across different scenarios. These metrics include MAE, MAPE, sMAPE, R^2, and RMSE so far.

C. Choice of Factors and Levels

The main choice of factors will be determined through hyperparameter tuning in the tree-based learners (namely LightGBM) as well as learning rate parameters in the neural networks.

Additional potential factors include implementation of a rolling window cross-validation system to better evaluate our data in various time-based scenarios

D. Choice of Experimental Design

Currently the data is real raw data collected from Environment Canada as well as NASA for their solar radiation data. As of now, this data is collected from Toronto International Airport, Buttonville, Toronto City Centre, and Toronto City, where Toronto City temperatures represent the ground truth temperature, and the other temperatures are lagged and represented as exogenous features. For replication-sake, the rolling window cross validation comes to mind, however as of now, it's up in the air due to its difficulty in implementation and lack of availability in libraries like Sci-kit Learn. That being said, similar data can be obtained and evaluated across other environmental stations.

E. Performing the Experiment

Currently, the algorithms are all taken from various libraries, and are currently being naively run (i.e., under default parameters).

F. Statistical Analysis of Data

Through use of various error metrics, as well as different datasets, we can compare the various model performance to statistically analyze our models.

G. Conclusions and Recommendations

As of current, no conclusions or recommendations can be generated however, the plan is to evaluate the model performance based on interpretability, ease of implementation, and model accuracy.