CSCI6515 Fall 2020 — Project
Evaluation of models for weather forecasting

Ridham Dabhi
B00853506
Dalhousie University
Halifax, NS, Canada
Email: ridhamdabhi@dal.ca

***Abstract*—Weather is imperative because of our dependence on weather conditions. Forecasting climate variables like temperature in a timely and accurate manner helps us to take necessary precautions to minimize weather-associated risks. Forecasts based temperature and precipitation are used by various industries like agriculture. This paper summarizes brief analysis of available approaches for forecasting temperatures using machine learning techniques. The main focus of this project is to build models and evaluate accuracy of each model. Models are evaluated based on how accurate results they predict. For model evaluation, Linear Regression model, Deep Neural Network (DNN) model, and Time Series Forecasting using Tensorflow are implemented on the same sample dataset.**

***Keywords*— Deep Neural Network, Time Series, Forecasting, Regression, Correlation, Outliers**

1. Introduction

The burgeoning research in the fields of Artificial Intelligence and machine learning has given rise to numerous weather prediction models. But the problem of accurately predicting or forecasting the weather still persists [1]. Weather forecasting is when weather conditions of a particular geographical area are predicted. Many essential or non-essential elements of life are dependent on forecasts. Traditional weather forecasting techniques are going obsolete because of unpredictable data outcomes from nature because of the rising sea-levels and global warming.

Machine learning allows us to find hidden relations and patterns in datasets to validate existing data or to predict outcomes based on the detected relations/patterns after processing certain input conditions, i.e., the generated models can be used to predict weather based on the observed historical data. The most widely used empirical approaches which are used for weather prediction, are regression, artificial neural network, fuzzy logic and group method of data handling [2].

This project involves analyzing various existing approaches for weather prediction. After researching and analyzing the effectiveness of diverse approaches, I implemented three models — Linear Regression model, Deep Neural Network (DNN) model, and a model created using Time Series forecasting.

The project definition is selected and the implementations are done to include most of the concepts that I learned in the class, and the assignments of the CSCI6515 Machine Learning with Big Data course. The methods for predicting the climate variables like temperature using different machine learning approaches requires a variety of data. I used a readily available dataset with a lot of records of daily weather variables. The selected dataset has diverse values collected for a long period, which will be described later in this report along with the dataset.

Big data is large, more complex data sets, especially from new data sources. These data sets are so voluminous that traditional data processing software just can't manage them [3]. As expected, the selected dataset may be classified as big data, as the weather conditions change drastically on most days based on different factors like temperature, precipitation, and time of the year.

In summary, the project's objective is to create and evaluate models and calculate their accuracy (effectiveness) based on the statistics and predictions for comparison.

2. Available Solutions

There are countless approaches to achieve weather forecast prediction from historical data implemented to date. I have researched diverse techniques to predict climate variables like temperature, the amount of rainfall, and the type of weather (like cloudy or sunny). Most techniques that I found were focused on training a model to determine average temperature of particular days based on historical input dataset.

Many of the implementations are different types of regression. In fact, regression is the most effective technique and it yielded high accuracies in almost all instances. Other implementations include

2.1. Using Regression

Regression is a method of modelling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables [4].

To select a good regression model, analyzing the data's different metrics like R-square, Adjusted r-square, and statistical significance of features of the dataset should be done. The data is usually divided into two parts — training part, and the validation part. The accuracy can be calculated and compared by different methods to find the optimal model.

The advantages of regression models is that it is one of the simplest model that can be used to evaluate and predict the relationship between multiple predictor features/variables and the predicted feature/variable (in the case of this project, the predicted feature is the Average daily temperature). Also, another advantage of using regression models is that they are computationally efficient, i.e., they make predictions in a very short amount of time compared to other models.

The disadvantage of using Regression models is that they make strong assumptions that the predictor variables and the predicted variables are related which might not be the case in some instances. Also, they are critically affected by outliers, biasing the model sometimes.

2.2. Using Neural Networks

In multi-layer artificial neural networks, there are neurons (nodes) placed in a similar to the human brain. Each neuron is connected to other neurons with certain coefficients [5]. During training, information is distributed to these connection points so that the network is learned. They receive numerical quantities (input signals), process them, and later transmits the processed signals to the nodes connected below the current node. The neural network models are not limited to predicting linear values, i.e., they have the potential to perform non-linear evaluations/operations.

The advantages of using neural network for prediction is that it can even work with incomplete knowledge. However, the performance loss is based on how vital is the missing information. It also has fault tolerance up to some extent. Moreover, they have high numerical strength and possess the ability to perform parallel processing.

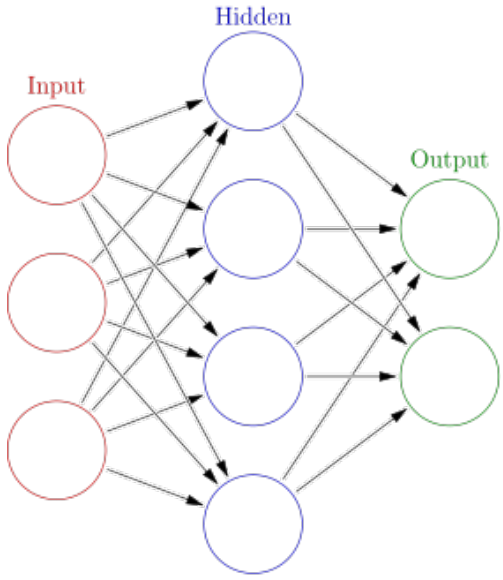

Figure 1. Graph depiction of a neural network [6]

There are a lot of advantages of using neural networks model for making predictions, but there are certain disadvantages of using it. One of the disadvantages is that they can only work with numerical information. So, before introducing the problems to neural networks, they have to be transformed/translated into numerical values. The quality of this translation will highly affect the performance of the neural network. Another disadvantage is that there is no way to determine the structure of neural networks. It is usually achieved through trial and error. Also, one of the biggest problem with the neural networks is that they give us the solutions (predictions) but doesn't show how it produces it. This diminishes trust in the neural network.

2.3. Using Time Series Forecasting

Time series is a series of data points indexed (or listed/graphed) in time order. Time series data are simply measurements or events that are tracked over time [7].

The advantages of using time series forecasting is that it can be applied anywhere, because temporal effects exist everywhere. Another advantage of using time series forecasting is that it can deal with temporal effects in time series analysis.

The main disadvantage of using time series based model is that there is always a way to get predictions with same accuracy faster using other models, rather than getting predictions using a time series model. Moreover, these models are ineffective in problems where temporal effects are weak, so using other non-temporal models for those problems may make more sense.

3. Implemented Solutions

While keeping the requirements of project in mind, I have followed a top-down approach to implement different models. At first stage I have collected and pre-processed the historical weather data. The second stage have witnessed implementation of the implementation of the Linear Regression model. The next stage contains the implementation of Deep Neural Network to predict weather (temperature). The final stage consists the methodology to implement Time Series model with the same dataset provided in above two models. As we know there is a specific accuracy associated with every model we implement, I have calculated the accuracy of every model after the implementation and shown them in the output which is then compared at the end. The highest accuracy model says that it was able to forecast weather with maximum precision and there was only a slight difference between the actual weather metrics and predicted weather metrics. The figure below shows high-level diagram process flow for this project. Further every stage of the process is discussed in detail.

Figure 2. Project approach for model evaluation

3.1. Data Pre-processing

This stage starts with the collection of datasets for the project. Initially I have tried to find historical weather data from many different sources which provided daily and monthly datasets. For this project, I needed the daily dataset because I need to find the pattern of weather data to predict it in future. Hence more daily dataset will result in more accurate results. I have researched about the type of measures that are recorded for the weather at any moment. I came across many measures such as minimum temperature, maximum temperature, dew point, precipitation, and atmospheric pressure. For the purpose to analyse all the features of weather we need to collect maximum data and need to pre-process data to know which factors are contributing to predicting temperature. I came across a dataset [8] which contains daily weather data for the Austin KATT station from 2013 to 2017. This dataset contains all the features that align with the dataset requirements.

After I collected all the necessary data, I needed to conduct research about the trends which are seen in weather data. After reading several research papers I came across a the idea that the previous days measurement values are taken into consideration for any specific day which could be helpful in anyways. For the considering the previous days information, I needed to change the dataset in a way that includes new columns. I have scripted a code which generates specified columns which contains weather data for previous days based on the specified number of days (I have used number of days=2 in this case). In the case when data is not available for previous days, I have considered that value as null. After generating the new dataset with increase columns, I have dropped the rows will null values so that it will not affect performance of any model.

I have now more data then collected initially, hence I need to perform some data cleaning to ensure that there is no redundant and unnecessary data. For predicting temperature, I only need to keep columns which can affect the temperature measure. Hence, I removed every other column which shown very less interrelation with temperature and were unnecessary. After data cleaning I am left with the columns such as minimum temperature, maximum temperature, mean temperature and the new defined columns. Another task in data cleaning is to ensure that every data in the dataset is numeric because the temperature is measured in digits. Hence if any rows have non-numeric data, it can cause a wrong signal and can affect the accuracy of our models. For this problem I have used pandas to search for any non-numeric numbers and replace it with NaN. Now we have all the necessary cleaned data, we are ready to apply different models and predict the average temperature for future years.

3.2. Linear Regression model

Linear regression is the technique which finds the relation between the output and the dataset considering the dependent and independent variables. This is a simple approach which is used to linearly predict the data without considering any other fundamental factors which can change the decision. The very first thing with our dataset is to know which factors are related to temperature, this could be any direct relation or indirect relation. Hence to calculate this we need to find the co-relation between temperature and every other column in the dataset. Correlation value is calculated in the range of -1 and 1. If the value of correlation is near to 1 then it is known to be highly related. If the value is near to 0 or negative, then we can say that the relation is rarely observed.

Also, I calculated and found outliers in the dataset to see if there is any potential for introducing spurious data artifacts, because they can significantly impact or bias the models.

To find the correlation between average temperature and other columns I have used the function “corr” which is pre-defined in pandas. This function automatically calculates correlation and provides us with the correlation score. I have selected all the columns which has correlation score more than 0.6 which I believe that has direct relation with

	count	mean	std	min	25%	50%	75%	max	outliers
VisibilityHighMiles	1307.0	9.991584	0.163489	5.00	10.00	10.0	10.00	10.00	True
VisibilityAvgMiles	1307.0	9.162204	1.458883	2.00	9.00	10.0	10.00	10.00	True
WindGustMPH	1315.0	21.373384	5.875657	9.00	17.00	21.0	25.00	57.00	True
SeaLevelPressureAvgInches_1	1315.0	30.022943	0.172205	29.55	29.91	30.0	30.10	30.74	True
SeaLevelPressureAvgInches_2	1314.0	30.023029	0.172242	29.55	29.91	30.0	30.10	30.74	True
HumidityHighPercent_1	1316.0	87.890578	11.023178	37.00	85.00	90.0	94.00	100.00	True
HumidityHighPercent_2	1315.0	87.904943	11.015042	37.00	85.00	90.0	94.00	100.00	True
PrecipitationSumInches_1	1194.0	0.126374	0.448362	0.00	0.00	0.0	0.01	5.20	True
PrecipitationSumInches_2	1193.0	0.126479	0.448535	0.00	0.00	0.0	0.01	5.20	True

Figure 3. Outliers table (Transposed)

changing temperature. Hence the columns with score more than 0.6 were average temperature, minimum temperature, maximum temperature, dew point average, dew point high and dew point low. To further visualise the correlation I have used the library matplotlib which constructs the graphs between average temperature and correlated feature. Some of the scatter graphs that were plotted to identify trends are shown below.

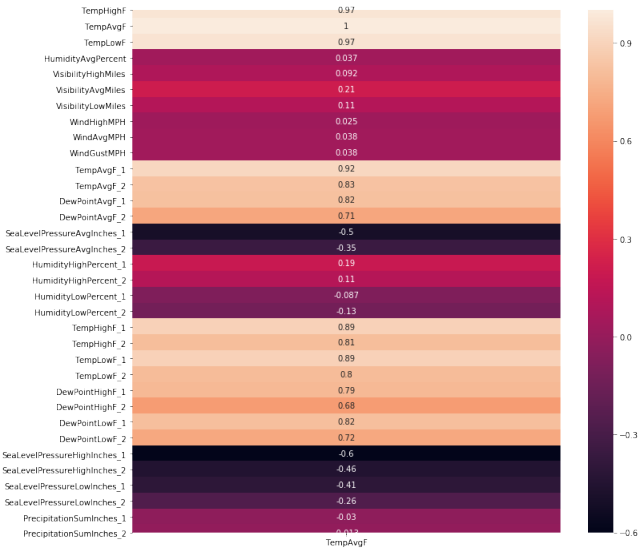

Figure 4. Correlation matrix heatmap

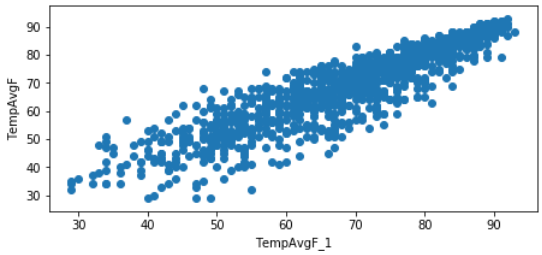

Figure 5. Scatter graph trend for a day's average temperature vs previous day's average temperature

The next step in this process is to apply OLS model to our selected features and generate the summary by selecting

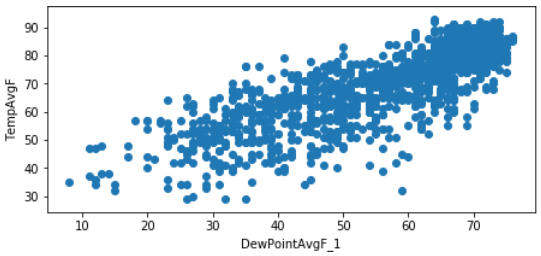

Figure 6. Scatter graph trend for a day's dew temperature vs previous day's dew temperature

the value of alpha. Here I have selected the value of alpha as 0.05. I have provided the dataset to fit the model and then have generated the summary to analyse the value of co-efficient p-value. If value of co-efficient is more than our alpha value for any columns, then we discard that data and provide our new data frame to fit the OLS model. In the first stage value of DewPointHighF_1 is 0.553 which is higher than our alpha value 0.05 hence we will discard it. In the next stage the value of DewPointAvgF_2 is 0.234 hence it will also be removed. Applying this new data frame once to OLS produces summary which does not have value more than alpha which means we found necessary features to be applied to our Linear Regression model and are able to predict the temperature with high accuracy.

OLS Regression Results

Dep. Variable:	TempAvgF	**R-squared:**	0.866
Model:	OLS	**Adj. R-squared:**	0.865
Method:	Least Squares	**F-statistic:**	835.2
Date:	Mon, 21 Dec 2020	**Prob (F-statistic):**	0.00
Time:	21:00:51	**Log-Likelihood:**	-3962.6
No. Observations:	1299	**AIC:**	7947.
Df Residuals:	1288	**BIC:**	8004.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	10.3149	1.010	10.210	0.000	8.333	12.297
TempAvgF_1	-1.1920	0.550	-2.168	0.030	-2.271	-0.113
TempAvgF_2	-1.3249	0.551	-2.405	0.016	-2.406	-0.244
TempLowF_1	0.9666	0.279	3.465	0.001	0.419	1.514
TempLowF_2	0.8569	0.278	3.081	0.002	0.311	1.402
DewPointAvgF_1	-0.2923	0.045	-6.516	0.000	-0.380	-0.204
DewPointHighF_2	-0.1305	0.028	-4.593	0.000	-0.186	-0.075
DewPointLowF_1	0.4186	0.037	11.448	0.000	0.347	0.490
DewPointLowF_2	-0.1542	0.029	-5.239	0.000	-0.212	-0.096
TempHighF_1	1.0592	0.276	3.835	0.000	0.517	1.601
TempHighF_2	0.6722	0.278	2.422	0.016	0.128	1.217

Omnibus:	119.448	**Durbin-Watson:**	1.960
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	203.962
Skew:	-0.638	**Prob(JB):**	5.13e-45
Kurtosis:	4.463	**Cond. No.**	1.53e+03

Figure 7. Final table with p-values under alpha (0.05)

At the last step I have applied our data frame to the

regressor and have calculated the average temperature, explained variance, mean absolute error and median absolute error. This measure are very useful in visualising accuracy of our dataset and our Linear Regression model.

3.3. Deep Neural Network (DNN) model

The previously implemented linear regression model makes a ridged assumption of a linear relationship between the dependent and independent variables. Neural networks tackles this problem by utilizing both — linear, as well as non-linear — operations based learning techniques. As explained above, neural networks are inspired by biological neurons in brain where they receive input signals, process it, and transmit the processes signal downstream agents in the network [9].

In this implementation, I implemented supervised learning. This means that the model is trained using known target outcomes. Moreover, the predictions are numerical values. So, we have to use regressor prediction algorithms.

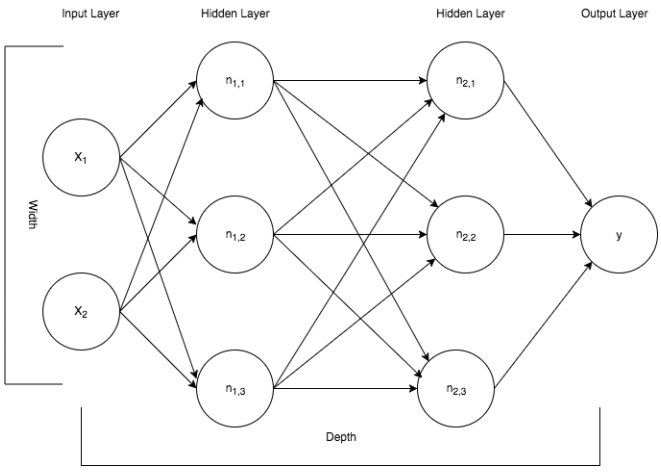

Figure 8. Graph depiction of a neural network [9]

The model works the following way: The neural network's input layer (say, x1 and x2) are feeding the neural network . These features are processed and transmitted into two layers (hidden layers). In the figure depiction, the each hidden layer contains three nodes (neurons). Then, the signal exits the neural network. It gets aggregated at the output layer as a predicted numerical value.

Each arrow in the figure represents the mathematical transformation of the passed numerical value. At the tip of the arrow, the passed numerical value is multiplied by the corresponding weight of that path. This is done for each node inside the neural network. The converging nodes' values are the aggregate of the multiplied weight and sum of the products that indicate a linear operation in a neural

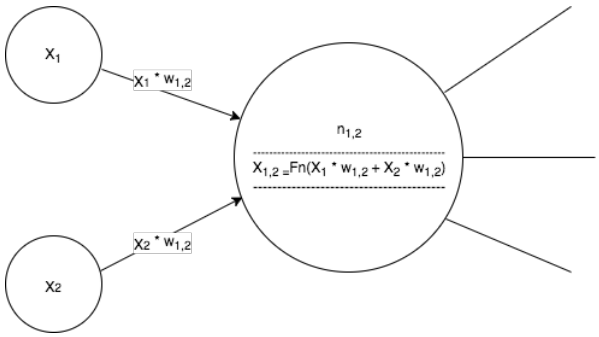

Figure 9. Depiction of activation function in a neural network [9]

network. After summing, a non-linear function is applied to the calculated value. This function introduces an activation function. This function is the reason behind the non-linear characteristics of neural networks.

Implementation of this model is done by creating a training data X, and the expected prediction data y. In supervised learning, while training, the actual values are also passed to compare with the predicted values and evolve the model as required. Model optimization is done by feeding example data into the neural networks and then evaluating and adjusting the weights in the neural network as required.

Using Tensorflow's DNNRegressor [10], I created a model and plotted a training steps versus Loss (Sum of Squared Errors — SSE) for the created model.

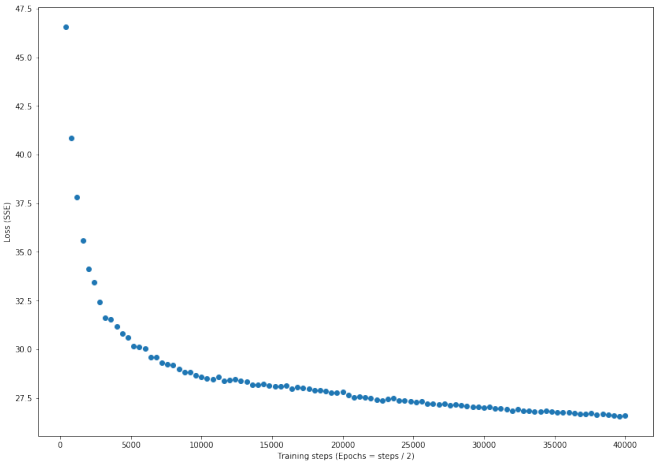

Figure 10. Training steps vs Loss (SSE) Graph for built model

I specified the loop iterations to 100 and the batch size to 400 for the building the model. The evaluate method is run and captured in an array during each loop iteration. The train method selects a random batch of records from the input and passes it to the neural network, until an output is achieved (prediction). Then, the weights in the neural network are adjusted. The loss values are calculated in each iteration.

Eventually, adjusting weights after each iteration causes the loss value to drop. However, over training the model may result in addition of noise which may affect the accuracy of the model. The evaluations array stores the average loss, the step, the current loss of the iteration, and some other diagnostic values.

```
{'average_loss': 46.558548,
 'label/mean': 68.44615,
 'loss': 46.558548,
 'prediction/mean': 68.912,
 'global_step': 400}
```

Figure 11. Sample value of an element in evaluation array

3.4. Time Series model

Implementing Time Series Sequential model is comparatively simple than the Linear Regression and Deep Neural Network (DNN) model. My research said that this model is not suitable where data might vary with time, i.e., the data has weak temporal effects. However, I went ahead and implemented it anyway to see how it works/fails to make predictions with high accuracy.

The implementation is started by creating a DataFrame with just the date and the feature to be predicted (Average temperature). Then I plotted the Time versus Average Temperature graph to view the trends of the data.

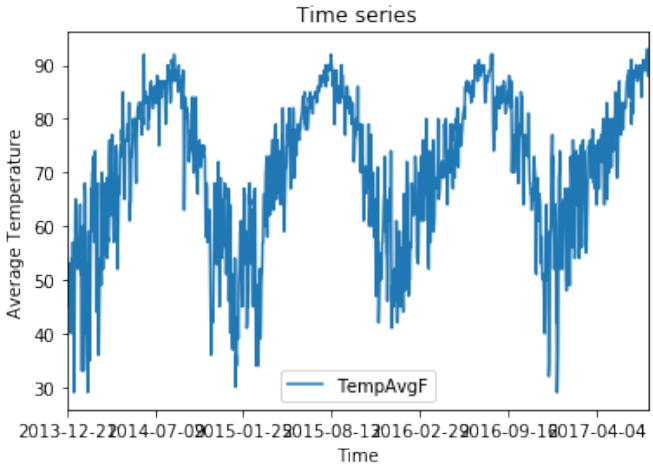

Figure 12. Time vs Average Temperature (TempAvgF) Graph

This is followed by transforming the features by scaling each feature using Sklearn's MinMaxScaler [11]. Then, create data array sequences and prediction array sequences based in number of iterations specified (history size) (in this case, 20). This is followed by scaling the training data and reshaping it for input. Then based on these, a model is built, using which predictions can be made.

4. Model Comparison

The below table elaborates three metrics according to three model uses to predict weather. For explained variance, Linear Regression, DNN Regressor and Time Series has values 0.87, 0.87, and 0.58 respectively. These values changes to 3.73, 3.61, and 6.17 in case of Means Absolute error associated with the three models. Moreover, when a Median Absolute error is calculated, it gives a value of 2.37, 2.59, and 4.87 for Linear Regression, DNN Regressor, and Time Series respectively. In case of explained variance, Linear Regression and DNN Regressor, both have equal and high variance. For mean absolute error, DNN regressor has a good result. Meanwhile, median absolute error gives an excellent value in Linear Regression model.

Accuracy Measure	Linear Regression	DNN Regressor	Time Series
Explained Variance	0.87	0.87	0.58
Mean Absolute Error	3.73	3.61	6.17
Median Absolute Error	2.37	2.59	4.87

Hence, it can be concluded that either DNN Regressor predictions or linear regression predictions are accurate up to some extent and can be used for accurate prediction of weather variables like temperature in datasets similar to the sample dataset.

5. Conclusion and Future Work

This project helped understand how to leverage diverse machine learning techniques to achieve accurate weather predictions. To summarize, this project consisted of researching and analyzing different approaches/techniques for predicting weather variables. The suggested solution consisted of preprocessing data, feature selection, training different models (Linear regression, DNN, and Time Series), plotting different visualizations to better understand trends and make comparisons.

Out of the implemented models, predictions made using Linear Regression and Deep Neural Network (DNN) were found to be most accurate out of the implemented models. In my opinion, I believe that using linear regression for temperature prediction will be more efficient because it produces prediction with accuracy similar to that of DNN model, but in significantly lesser time. On the other hand, if the data does not have a lot of correlation between predictor variable and the predicting variable, then using the DNN model will be more beneficial, rather than using linear regression.

For future work, we can go even further and process fine-grained data. In other words, in this project, daily weather records were used to predict daily temperatures, but an hourly (or even minutes) based data records can enable us

to predict hourly weather variables very accurately. In this case, the neural network might able to find some predictor connections that would not have been known otherwise. Moreover, the rise in temperatures (global warming) is causing irregularities in climate, making it even more to make forecasts using previous historical data. For example, climate change (rising temperatures, melting icebergs, or rising sea levels) was not a great issue a few years ago. So, it would be difficult to make predictions using that data, but not impossible if some new breakthrough comes along the way in the future.

References

[1] Sanyam Gupta, Indumathy K, and Govind Singhal, "Weather Prediction Using Normal Equation Method and Linear regression Techniques",) International Journal of Computer Science and Information Technologies, Vol. 7 (3) , 2016, 1490-1493.

[2] E. Sreehari and P. G. S. Ghantasala, "Climate Changes Prediction Using Simple Linear Regression," Journal of Computational and Theoretical Nanoscience, vol. 16, no. 2, pp. 655–658, 2019.

[3] Oracle, "What Is Big Data?," What Is Big Data? — Oracle Canada, 2020. [Online]. Available: https://www.oracle.com/ca-en/big-data/what-is-big-data.html. [Accessed: 20-Dec-2020].

[4] R. Gandhi, "Introduction to Machine Learning Algorithms: Linear Regression," Medium, 28-May-2018. [Online]. Available: https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a. [Accessed: 20-Dec-2020].

[5] M. M. Mijwel, "Artificial Neural Networks Advantages and Disadvantages," LinkedIn, Jan-2018. [Online]. Available: https://www.linkedin.com/pulse/artificial-neural-networks-advantages-disadvantages-maad-m-mijwel. [Accessed: 20-Dec-2020].

[6] Wikipedia, "Artificial neural network," Wikipedia, 17-Dec-2020. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed: 20-Dec-2020].

[7] K. Lai, "Time Series Analysis and Weather Forecast in Python," Medium, 23-Mar-2020. [Online]. Available: https://medium.com/@llmkhoa511/time-series-analysis-and-weather-forecast-in-python-e80b664c7f71. [Accessed: 22-Dec-2020].

[8] GrubenM, "Austin Weather," Kaggle, 15-Aug-2017. [Online]. Available: https://www.kaggle.com/grubenm/austin-weather. [Accessed: 15-Dec-2020].

[9] A. McQuistan, "Using Machine Learning to Predict the Weather: Part 3," Stack Abuse. [Online]. Available: https://stackabuse.com/using-machine-learning-to-predict-the-weather-part-3/. [Accessed: 22-Dec-2020].

[10] "tf.estimator.DNNRegressor : TensorFlow Core v2.4.0," TensorFlow. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/estimator/DNNRegressor. [Accessed: 15-Dec-2020].

[11] Scikit Learn, "sklearn.preprocessing.MinMaxScaler¶," scikit, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. [Accessed: 22-Dec-2020].