**Portfolio Project Option # 1 – AI Use - Case Problem With Solution - Paper**

Daniel Diaz

For my portfolio project I created a weather app that predicts weather one day in advanced. Using the model, we could predict for as many days as we’d like. I decided to keep it to just one day to avoid getting results that we’re too unreasonable and far from the actual data. Weather prediction gets very difficult the further into the future we attempt to predict it. There are many complex systems that do not act in a linear fashion, such as atmospheric pressures, the temperature (avg, highs and lows), precipitation, amount of sunshine, etc. For this reason, alone, the weather is a model that acts very chaotically, and its systems are inherently non-linear and dependent on one another making the endeavor of predicting any characteristic of the climate very difficult. There are many scientific models for predicting the weather such as, the ECMWF and GFS (Sillin, 2019). My attempt to predict the weather in my AI program does not make use of either of these models, however, my model would certainly benefit from doing so.

In my prediction application, I’ve attempted to predict the Average Temperature (tavg), Minimum Temperature (tmin), Maximum Temperature (tmax), the daily precipitation (prcp), weather it will snow or not (snow), the average daily Average Wind Direction (given in vectors Wdirx, Wdiry by multiplying by their magnitude and sin and cosine components), the Average Wind Speed (Wdirx, Wdiry), the Daily Peak Wind Gust (wpgt), the average sea-level air pressure (pres), and the daily sunshine given in total in minutes (tsun). The models that I’ve trained to predict all these aforementioned features except the snow depth, is a Recurrent Neural Network known as the Long Short-Term Memory method. This method is known to predict time series data very well (Zhao et al, 2018). Time series needs to be in order and LSTM makes use of previous output for input in the next layer. Our model gave predictions that are quite reasonable as can be seen in the Denver, Los Angeles, and New York screenshots. The final loss from the 2 epochs I ran, gave me an average of .7574 and a mean absolute error of .4856.

The other type of model that I made use of was a classification prediction model known as the Random Forest Classifier. This is an ensemble method that makes use of decision trees to predict something. In my case, I utilized the ensemble classifier to predict whether there would be snow fall or not. My data gave snowfall in depth recorded as mm but I encoded the data to sum any amount of mm of snowfall and count that as ‘yes’ or ‘no’ (0 mm) snowfall for the day. The input to the model were the 8 other variables mentioned in the previous paragraph. Predicting snowfall is also a very complicated endeavor. To help me visualize (and this tends to help before embarking on creating any AI/ML model) the correlations and relationships between variables, I created a correlation heatmap in which we can see which values effect snowfall more than others. The chart is shown below.

Graphical user interface, application, Teams

Description automatically generated

The snowfall is correlated most with temperature average, minimum and maximum and the precipitation. Then I trained my model to create 30 decision trees. The final precision resulted in an average of .825 and an accuracy of .82. The max depth any one tree could reach was 12.

My app that I wrote can take in a city by city name and look up its latitude and longitudinal coordinates using a library known as geopy. The country of the city can be included but doesn’t have to be. It can help specify the city in question. Also, the program can take input as a zip code, as well. The program produces output in a form that humans can understand. The output is all given for the next day. The output consists of whether or not it will snow, the mean, high and low temperatures for the day in Celsius and Fahrenheit, and the prediction in the form of a dictionary for all the other features.

Gathering data was a bit of a challenge as I wanted to gather data that was robust, diverse, representative, and historically large. I was able to find an API called Meteostat. This API gathers data on all aforementioned features yearly, monthly, daily, and hourly from the National Oceanic and Atmospheric Association (NOAA) and all its different stations. From this API, I wrote a script that took in a data frame representing all the zip codes in the United States and fed their longitudinal and latitudinal coordinates to get their historical climate data. Unfortunately, this was too much data that the API could hand over in a single request, so I decided to utilize the shuffle method in Sklearn to shuffle all the zip codes in every state and territory in the U.S. and select 50 zip codes in each state at random. This was the data I used to train my models. The time period I used was from January 1st, 1980 – December 31st, 2021. The data was given for every day (daily). This generated just shy of 2 GB’s worth of data (about 22 million rows). It should be noted that the data is specific to the U.S., meaning that the models may have a biased prediction towards the United States weather data. My program can predict the climate data for any valid region of the world, but the model’s accuracy and precision can vary more so than they would for regions and states in North America.

**References**

US Department of Commerce, N. O. A. A. (2013, March 11). The challenges and complexities of weather forecasting, by Mark Bloomer. National Weather Service. Retrieved December 30, 2021, from https://www.weather.gov/car/weatherforecasting

Sharda, R., Delen, D., & Turban, E. (2021). Analytics, Data Science, & Artificial Intelligence: Systems for Decision Support. Pearson.

Sillin, J. (2019, December 18). Model Mania: What Are The ECMWF and GFS Models, and Why Are They Different? Blog.Weather.Us. Retrieved February 13, 2022, from https://blog.weather.us/model-mania-what-are-the-ecmwf-and-gfs-models-and-why-are-they-different/

Brownlee, J. (2020, August 28). Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. Machine Learning Mastery. Retrieved February 13, 2022, from https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

Li, Y., Zhu, Z., Kong, D., Han, H., & Zhao, Y. (2019). EA-LSTM: Evolutionary attention-based LSTM for time series prediction. Knowledge-Based Systems, 181, 104785. https://doi.org/10.1016/j.knosys.2019.05.028

Jarrahi, M. H. (2018). Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making. Business Horizons, 61(4), 577–586. https://doi.org/10.1016/j.bushor.2018.03.007

The AI Index Report – Artificial Intelligence Index. (2021, January 1). Stanford University Human-Centered Artificial Intelligence. Retrieved December 23, 2021, from https://aiindex.stanford.edu/report/

Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Ilavarasan, P. V., Janssen, M., Jones, P., Kar, A. K., Kizgin, H., Kronemann, B., Lal, B., Lucini, B., … Williams, M. D. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. International Journal of Information Management, 57. https://doi.org/10.1016/j.ijinfomgt.2019.08.002

Kelly, K. (2012, December 24). Better Than Human: Why Robots Will — And Must — Take Our Jobs. Wired. Retrieved December 23, 2021, from https://www.wired.com/2012/12/ff-robots-will-take-our-jobs/

The state of AI in 2020. (2020, November 17). McKinsey & Company. Retrieved December 23, 2021, from https://www.mckinsey.com/Business-Functions/McKinsey-Analytics/Our-Insights/Global-survey-The-state-of-AI-in-2020

Subhajini, A. C. (2018). APPLICATION OF NEURAL NETWORKS IN WEATHER FORECASTING. Eajournals, 4(1), 8–18.

Valasek, J., & Bokadia, S. (2001). Severe weather avoidance using informed heuristic search. AIAA Guidance, Navigation, and Control Conference and Exhibit. https://doi.org/10.2514/6.2001-4232

R. Donida Labati, A. Genovese, V. Piuri, F. Scotti and G. Sforza, "A Decision Support System for Wind Power Production," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 1, pp. 290-304, Jan. 2020, doi: 10.1109/TSMC.2017.2783681.

Brownlee, J. (2020, October 31). Random Forest for Time Series forecasting. Machine Learning Mastery. Retrieved January 30, 2022, from https://machinelearningmastery.com/random-forest-for-time-series-forecasting/

Van Assche, A., Vens, C., Blockeel, H., &amp; Džeroski, S. (2006). First Order Random Forests: Learning Relational classifiers with complex aggregates. Machine Learning, 64(1-3), 149–182. https://doi.org/10.1007/s10994-006-8713-9