DATS6313 Final Project Presentation Script

My name is Erika, and my project is on using Time series techniques to predict appliance energy consumption.

Slide 1: In this presentation, I will start with introducing the project by showing an overview of the chosen data set. Then I will talk about my EDA process before moving on to building the required models. Finally, I will summarize and conclude the presentation.

Slide 2: The chosen data set that meets all the requirements for this project is called Appliance Energy Prediction from UCI Machine Learning library. It has more than 19000 data points, with a total of 27 continuous variables (show in slide 3). The forecast variable is “appliances”, which in this case is energy consumption measured in watt per hour.

Experimental data of appliances energy use in a low energy building. Energy consumption data is collected every 10 minutes to a house inside the electrical power consumption, environmental data is the use of wireless sensor network monitoring temperature and humidity, including living room, kitchen, bathroom, laundry and so on. (move to slide 3) We can see these variables, such as T\_1 showing (…), T\_2 showing…., RH\_3 showing (…) The sensor collects data every ten minutes. Weather from the nearest airport weather station (Belgium) was downloaded from a public data set merged together with the experimental data sets using the date and time column. All the outdoors variables (move slide 3 again) are from this data, such as T\_0 which shows …. And Wind speed.

Two random variables have been included in the data set for testing the regression models and to filter out non predictive attributes (parameters)

Slide 4: I will now explain the EDA process. First, crucial for time series, is stationarity checking. We see that rolling mean and variance stabilize after all samples are included. P-value for adf is 0.00, so we can reject the null hypothesis; for KPSS is 0.1 so we fail to reject the null hypothesis. All these are evidence that the data is stationary. No further transformation is needed.

Slide 5: Now for decomposition. I applied the STL method, which shows that there is no significant trend or seasonal pattern that needs to be dealt with.

Slide 6: For feature selection – First, we see that the condition number is extremely high, which supports a strong case of multicollinearity. For the correlation matrix, with 27 variables, it was too big to show here, but we see some high correlation between some temperature and humidity variables that were worth noting. VIF values are high for these variables. Both PCA and SVD results show that we can reduce the features and still explain a lot of variance in the data. PCA for example, show that ~20 features can explain almost all of the variance. In general, all metrics support dimensionality reduction.

Slide 7: The chosen method for feature elimination is backward stepwise selection, with alpha = 0.05 as the criteria. Only 19 features were kept; the 2 random variables, pressure, and some temperature and humidity variables were removed.

Slide 8: For the models, I will be talking about the base-models which are (read off slide). The holt-winters method was also used, along with multiple linear regressions, and ARMA/ARIMA model.

Slide 9: For base models, I used them to perform h-step predictions. Performance-wise, Average performed the best, with the lowest RMSE, and exponential smoothing performed the worst. The holt-winters method might not do very well either, as it is an expansion of exponential smoothing to include seasonal components (that this data does not seem to have)

Slide 10: As theorized, the model did not perform as well as Average. I used additive methods, with seasonal periods of 144, as there are 144 10-min intervals every day.

Slide 11: Moving on to regressions. I initially ran one-step predictions without cross validation, and then reran the program to include 5-fold cross-validation. R2 went from 0.17 to 0.07 after CV. RMSE went from 85 to 94. F-test is significant with p-value basically 0, which shows that at least some of the predictors significantly contribute to the model. Most features are statistically significant, except dewpoint temperature and outside humidity, which both had t-probaility larger than 0.1.

Slide 12: For regression residual analysis, the ACF/PACF plots do not resemble patterns for white noise. It does not cut off after lag 0. Doing Q-value analysis also shows that Q is larger than Q-critical. Residual is not white. This model needs to be adjusted in the future.

Slide 13: For ARMA/ARIMA preliminary order selection, I created a GPAC table for the data. It looks like the clearest pattern is k=1 and j=0. You could also argue for k=8, j=10 but personally for me, that is less clear. The ACF is tails off and PACF cuts off after lag 1. Both GPAC and ACF/PACF shows a strong argument for an AR(1) model.

Slide 14: Since the model would not have any seasonal components, or moving average components, I plugged ARIMA(1,0,0) into the ARIMA package to estimate coefficients for the model. The residuals’ ACF/PACF seem to cut off after first lag, and Q-value also supports that residuals are white. RMSE is 131.51, which means it performed worse than Average method and multiple linear regressions.

Slide 15: Some more residual analysis here. Variance of forecast errors is much, much larger than variance of errors. This suggests that the model does not generalize well. It performs worse on unseen data. After performing Zero/pole cancellation, it shows that the final coefficient matches the current one. Confidence interval suggests that all parameters are significant since it has no 0 in between.

Slide 16: As shown above, the best performing model was Average, which is one of the base models. Since it is not complete and fully bug-free, I did not include – but a LSTM was built, and shown the best performance, with a much lower RMSE than the rest. If there was more time, I would have done the following:

1. Redo feature selection
2. Test other ARIMA model orders
3. Finish a LSTM model
4. Try 2-3 other models, such as K-NN, SVM or Random Forests

There is a paper in 2020 for the Journal of Physics: Conference Series that documented their work on the same data set, and they achieved an RMSE of 2.48 for LSTM with a different set of features included. Given time, I would like to replicate what they do in the paper to see if I could achieve the same results.