PNNL-36312

Evaluating the Accuracy of Machine Learning Forecasts

September 2024

Rosa Saldivar

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-36312

Evaluating the Accuracy of Machine Learning Forecasts

September 2024

Rosa Saldivar

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Evaluating the Accuracy of Machine Learning Forecasts

Rosa Saldivar

Visiting Faculty Program
California State University, Los Angeles

Pacific Northwest National Lab
Richland, Washington

ABSTRACT

To improve the accuracy of forecasting in machine learning, we must investigate multiple machine learning models and see how accurately they can predict values after training. We used seven machine learning models to try and get more accurate predictions. The models that were used were ARIMA, SES, MLP, CART, LightGBM, and XGBoost. We used a processed dataset from a Terminal at LAX that had the number of people traveling through terminal X every hour in March from 2015-2019. We trained our models with the dates March 6 - March 19 to predict the value for March 20th and the hours 6:00 am to 6:00 pm since those are the most popular traveling hours. By using the different models, we had varying results of accuracy when estimating the amount of people traveling through terminal X on March 20th. We know that machine learning models are helpful for forecasting and by seeing how accurately these models can predict, we can see how forecasting can be helpful for other issues. Using these methods, airports can use forecasting to predict the amount of people coming in and out and can use these predictions to prepare their resource management, operational efficiency, and overall passenger experience.

I. INTRODUCTION

Time series forecasting is making a scientifical prediction based on previous time-stamped data. This can be done by using machine learning models. A machine learning model is a computer program that can recognize patterns and make predictions based on those patterns. After reviewing different machine learning models that time series forecast, it is evident that there are many models that do not produce accurate predictions. We wanted to build, train and test multiple models with airport travel data to see if we could produce an accurately predicting model. The seven models we decided to work on were the AutoRegressive Integrated Model (ARIMA), Simple Exponential Smoothing Model (SES), MultiLayer Perceptron Model (MLP), Classification and Regression Trees (CART)- Discission Tree Model, Classification and Regression Trees (CART)- Random Forest Model, Light Gradient Boosting Machine Model (LightGBM), and the eXtreme Gradient Boosting Model (XGBoost). However, I only got the chance to work with three of them: ARIMA, SES, and MLP. Our goal with the research was to find the most accurate machine learning model. In this paper we will be exploring the three models I mentioned above, including looking at the code and the graphed results of the model's predictions. We will also be looking at the steps to be building these models including the setting up of data and graphing of the data. We will then be discussing how to interpret the results. Finally, I will be concluding with my final thought and impact of this research.

II. MOTIVATION

After reviewing the airport travel data, it is clear that it would be difficult to predict the amount of people that would be coming in and out of an airport. In conducting this research, we wanted to try and find a helpful forecasting method to determine the amount of people going through the airport in the following hours of selected data. We also wanted to see how finding the most accurate model might be helpful for these terminals.

III. METHODS

We tried to set up our research in a way that seemed simplest. We started with graphing the data followed by implementing the machine learning models (which included organizing the data, building the model, training the model, and testing the model) and then by analyzing our results. I have provided more detail about the steps below.

A. Graphing the Data

We started by using a preprocessed dataset that specifically excluded data past 2019 to avoid any inconsistencies due to the COVID-19 pandemic. The data included the columns "Hour", "Week Number", "Weekday", "date2015", "Value2015", "date2016", "Value2016", etc. continuing until the year 2019. After looking at this data we decided it would be helpful to graph the data to give insight into the patterns and trends in the data and to understand the distribution of the data and see how it might affect our forecasts. All work including the machine learning models and graphs were created using the coding language Python. Figures 1-4 below show the graphs of the data.

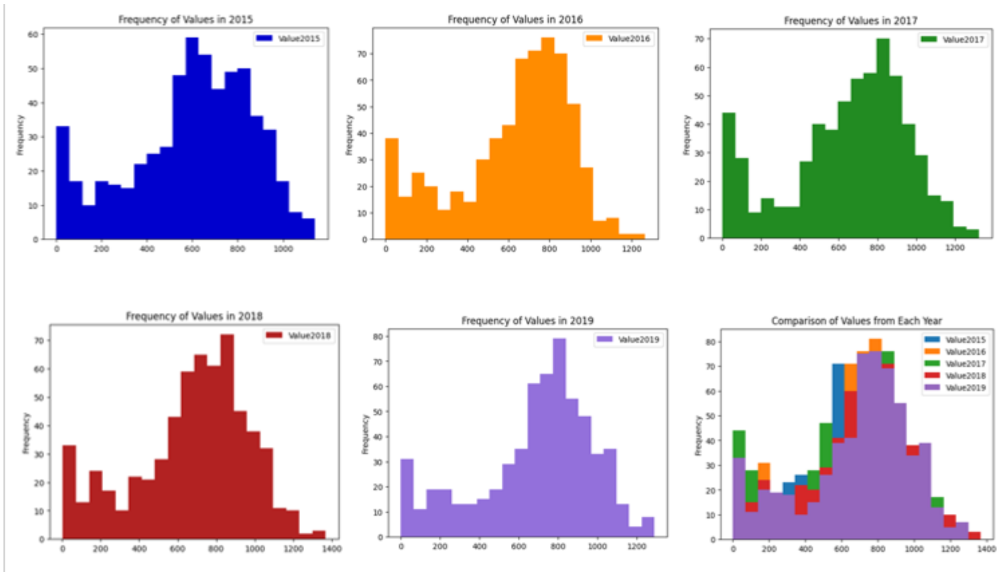


Figure 1. These graphs show the frequency of each value in their respective year. The values indicate the number of people who were at the airport at a specified hour. The final graph shows a comparison of each of these graphs together.

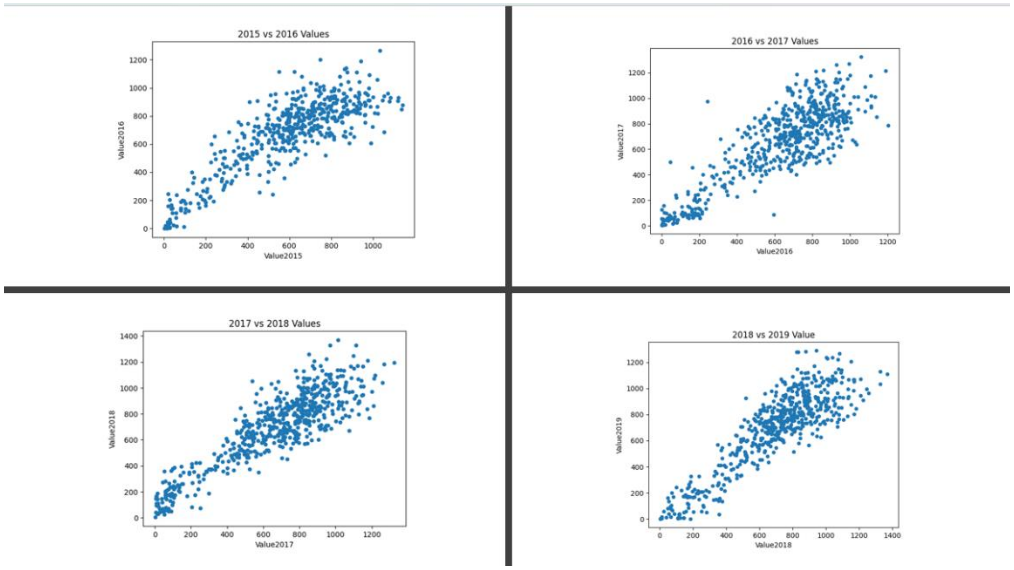


Figure 2. These graphs depict the comparison of values between each year. The values indicate the number of people who were at the airport at a specified hour. It shows how similar the values are from year to year.

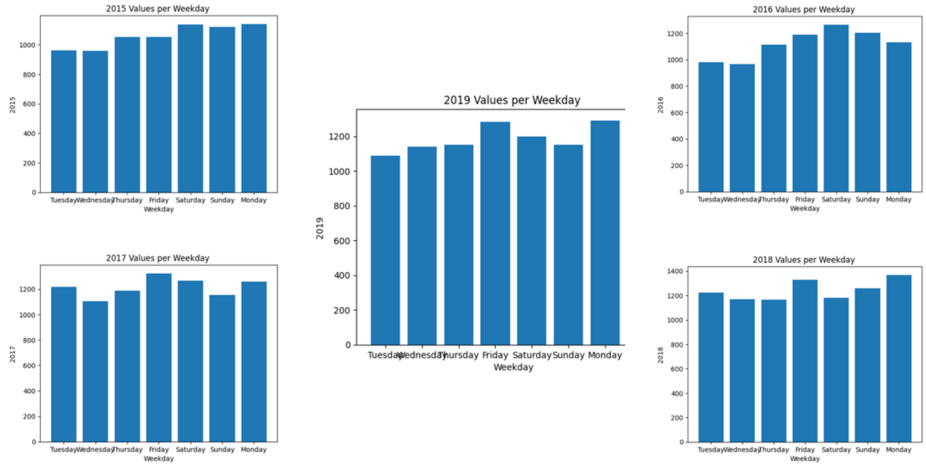


Figure 3. These graphs show the average of the value per weekday in their respective year. The values indicate the number of people who were at the airport at a specified hour. It reveals which days were more popular for traveling in this terminal each year.

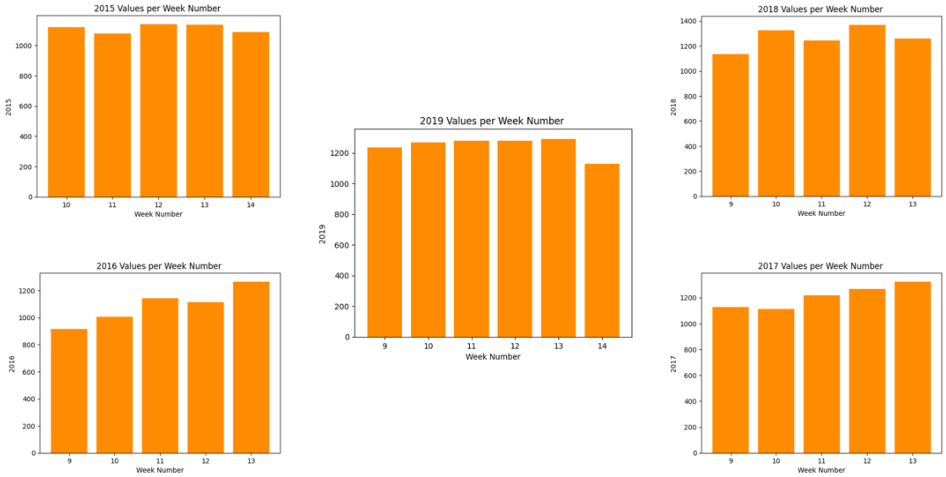
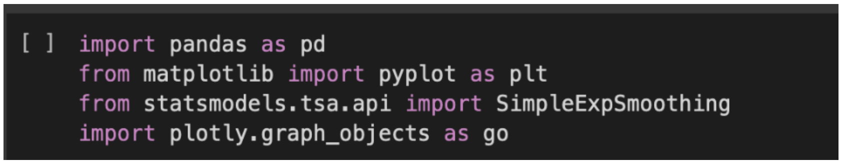


Figure 4. These graphs depict the average of the value per week number in their respective year. The values indicate the number of people who were at the airport at a specified hour. The graphs show which week numbers were popular for traveling in this terminal each year.

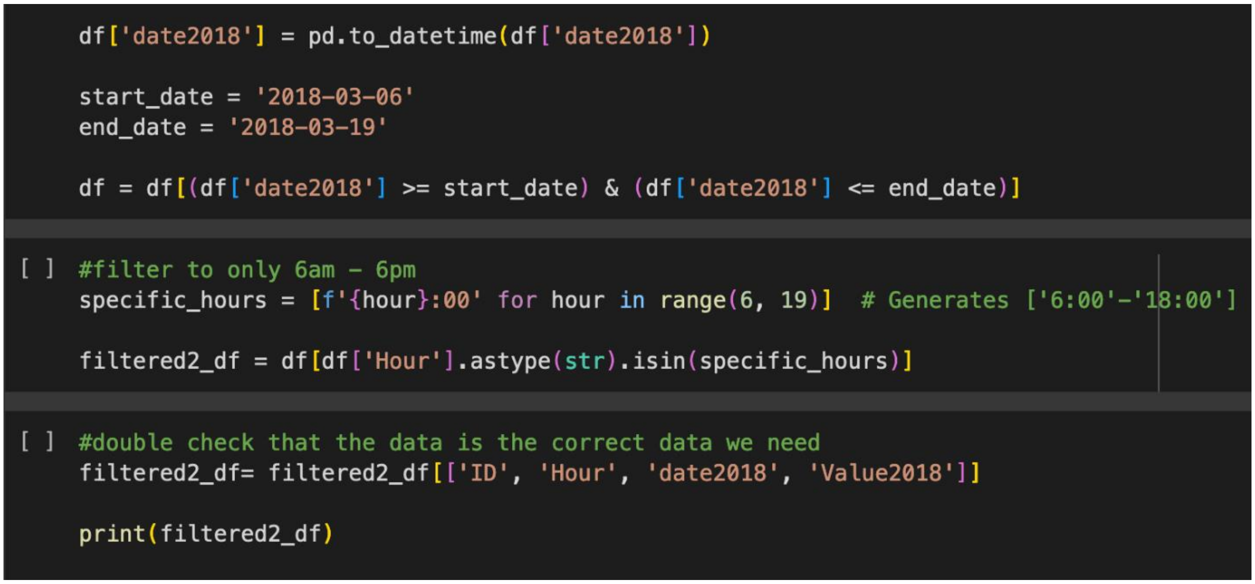
B. Preparing for the Models

Although the data was already preprocessed before graphing, we had specific requirement for the models we were going to be implementing. We wanted the models to train with the data that was specified to March 6th – March 19th of the year 2018 and limit the hours of each day to 6am- 6pm. We used code in Python to do this rather than manually having to remove all other data from the dataset. We also needed to set up the code with the appropriate imports for each model in order to start working on the training and testing. The final thing needed to prepare for the models was creating two separate subsets of data for training and testing. The training subset would be the March 6th – 19th hours and the testing subset would the March 20th hours. The training subset is used for training the model to predict the forecast, while the testing subset is used for testing if the forecast is accurate. Figure 5 shows the imports and Figure 6 shows the data filtering.



```
[ ]  import pandas as pd
     from matplotlib import pyplot as plt
     from statsmodels.tsa.api import SimpleExpSmoothing
     import plotly.graph_objects as go
```

Figure 5. This code shows all the imports needed for the Simple Exponential Smoothing Model. It is an example of how the other models are set up with the necessary imports.



```
     df['date2018'] = pd.to_datetime(df['date2018'])

     start_date = '2018-03-06'
     end_date = '2018-03-19'

     df = df[(df['date2018'] >= start_date) & (df['date2018'] <= end_date)]
```

```
[ ]  #filter to only 6am - 6pm
     specific_hours = [f'{hour}:00' for hour in range(6, 19)]  # Generates ['6:00'-'18:00']

     filtered2_df = df[df['Hour'].astype(str).isin(specific_hours)]
```

```
[ ]  #double check that the data is the correct data we need
     filtered2_df= filtered2_df[['ID', 'Hour', 'date2018', 'Value2018']]

     print(filtered2_df)
```

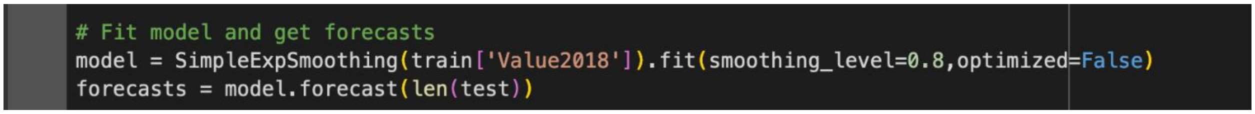
Figure 6. The code shows the filtering of data to ensure that only the 6th -19th of 2018 are selected at the specified time.

C. Building, Training, and Testing the Models

After filtering through the dataset to specify the data we wanted, we finally were ready to start building the models. Each of the models were different when it came to building them. Some required extra steps before fitting the models, however the training and testing were the same because you fit the model with the training subset and then you use the model to predict what the testing subset will be.

1. Simple Exponential Smoothing Model

The Simple Exponential Smoothing model or SES model is a very straightforward forecasting method that only needed one main parameter which is the smoothing factor. The smoothing factor can only be set to values between 0 and 1. A higher smoothing factor is recommended if you have rapid changes in data and need a more responsive forecast and a lower smoothing factor is recommended if you have more stable data and need to smooth out certain fluctuations. I chose higher smoothing factor for our data. Below in Figure 7 is the implementation of the SES model.

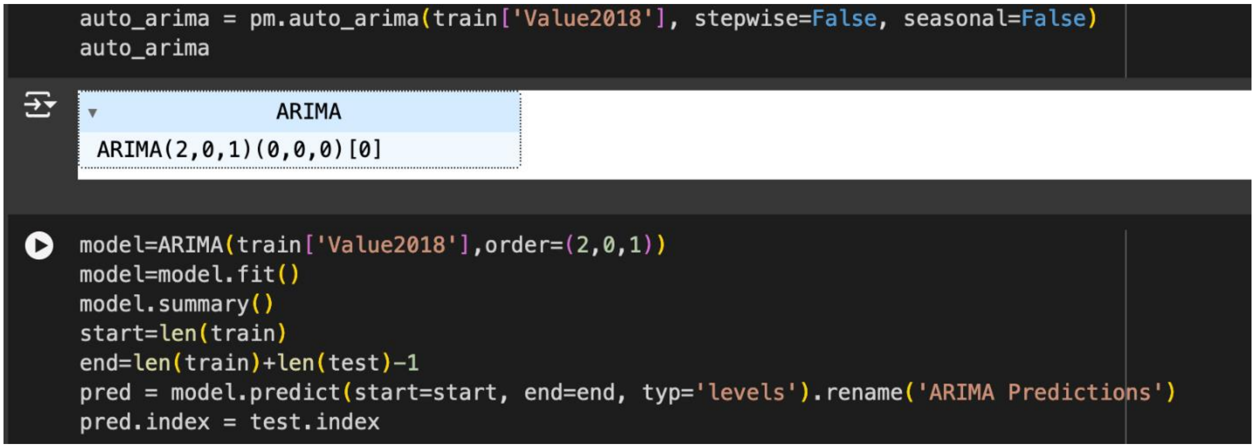


```
# Fit model and get forecasts
model = SimpleExpSmoothing(train['Value2018']).fit(smoothing_level=0.8,optimized=False)
forecasts = model.forecast(len(test))
```

Figure 7. This code shows the fitting of the SES model.

2. AutoRegressive Integrated Moving Average Model

The AutoRegressive Integrated Moving Average model or ARIMA model is a more advanced model than the SES model. This model combines autoregression, integration, and movie average to predict. The parameters needed for this model are *p* (number of lag observations), *d* (number of times data needs to be differenced), and *q* (size of moving average window). It is quite tedious and takes numerous steps and tries to pick what the best values are for these variables, however there is tool called "auto_arima" that will calculate these values for you. Figure 8 shows the implementation of the ARIMA model.



```
auto_arima = pm.auto_arima(train['Value2018'], stepwise=False, seasonal=False)
auto_arima
```

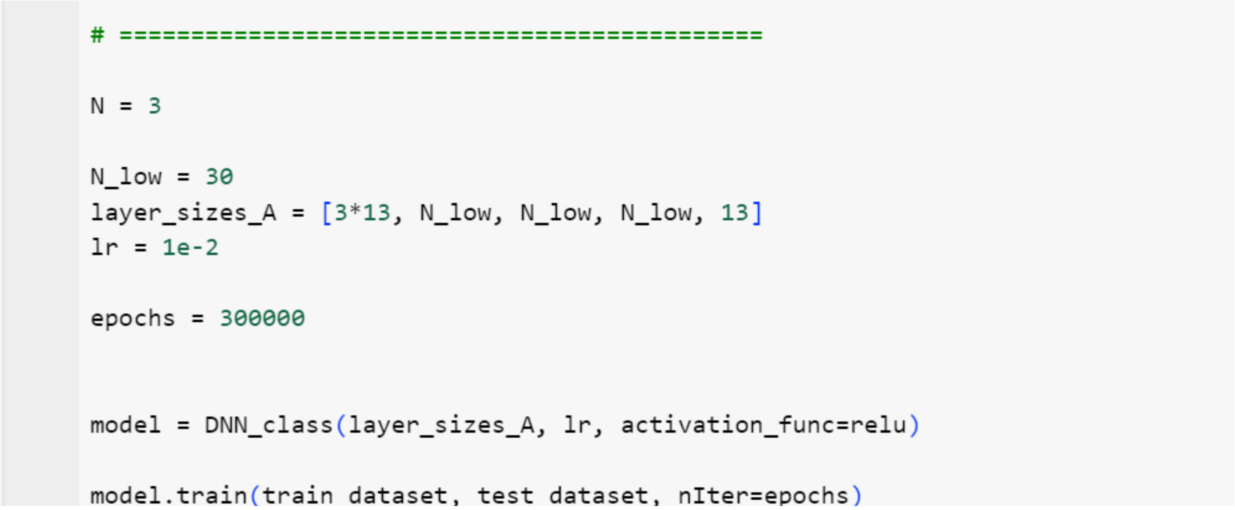
```
ARIMA
ARIMA(2,0,1)(0,0,0)[0]
```

```
model=ARIMA(train['Value2018'],order=(2,0,1))
model=model.fit()
model.summary()
start=len(train)
end=len(train)+len(test)-1
pred = model.predict(start=start, end=end, typ='levels').rename('ARIMA Predictions')
pred.index = test.index
```

Figure 8. This code shows the fitting and auto arima of the Arima model.

3. Multilayer Perceptron Model

The Multilayer Perceptron mode or MLP model is a type of neural network that is helpful for forecasting. The components of this model are the input layer (receives the data), hidden layer (performs computations on data), and the output layer (produces output). In this model, that was created by Dr. Amanda Howard. It has a specification where it takes the previous three days of data to predict the following single day. It also gets the structure and functionality from an already prepared file called “DNN_class”, which was also prepared by Dr. Howard.

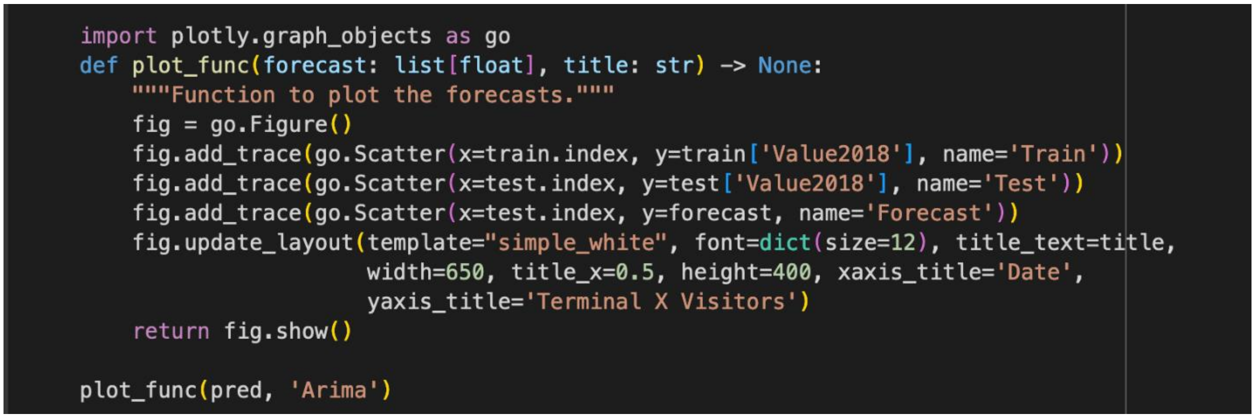


```
# ==============================================

N = 3

N_low = 30
layer_sizes_A = [3*13, N_low, N_low, N_low, 13]
lr = 1e-2

epochs = 300000


model = DNN_class(layer_sizes_A, lr, activation_func=relu)

model.train(train dataset, test dataset, nIter=epochs)
```

Figure 9. This code (prepared by Dr. Amanda Howard) shows the fitting of the MLP model.

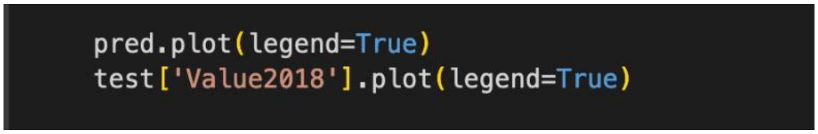
D. Graphing the Models

After creating all the models, the graphing of the models is all the same. I created two graphs in order to have a better look at the predictions. Figures 10 and 11 show the code for creating the graphs.



```
import plotly.graph_objects as go
def plot_func(forecast: list[float], title: str) -> None:
    """Function to plot the forecasts."""
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=train.index, y=train['Value2018'], name='Train'))
    fig.add_trace(go.Scatter(x=test.index, y=test['Value2018'], name='Test'))
    fig.add_trace(go.Scatter(x=test.index, y=forecast, name='Forecast'))
    fig.update_layout(template="simple_white", font=dict(size=12), title_text=title,
                      width=650, title_x=0.5, height=400, xaxis_title='Date',
                      yaxis_title='Terminal X Visitors')
    return fig.show()

plot_func(pred, 'Arima')
```

Figure 10. This code shows the creation of a graph that shows the entire testing and training data along with the prediction.



```
pred.plot(legend=True)
test['Value2018'].plot(legend=True)
```

Figure 11. This code shows the creation of a graph that shows a close-up of the prediction vs. the testing data.

IV. RESULTS AND DISCUSSION

The forecast results of the each of the models are below.

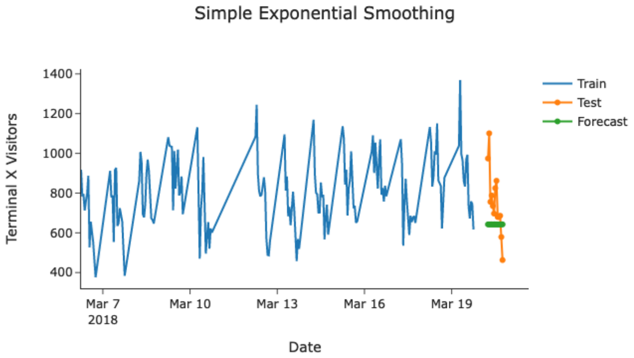


Figure 12. This graph shows the training data, testing data, and the forecast of the SES model.

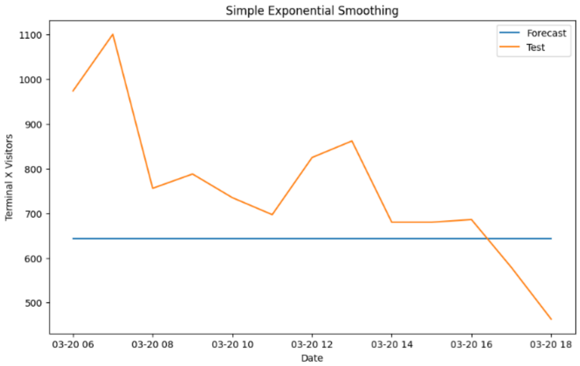


Figure 13. This graph shows a closer look at the testing data vs. the forecast of the SES model.

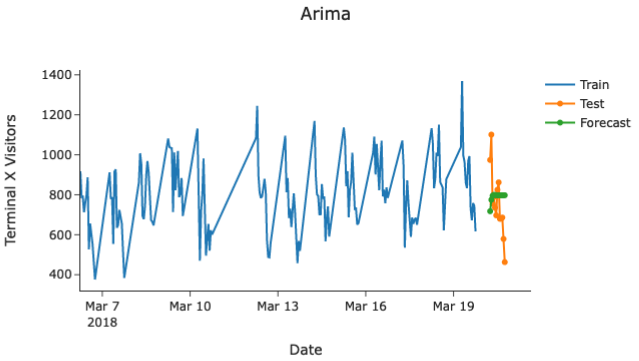


Figure 14. This graph shows the training data, testing data, and the forecast of the ARIMA model.

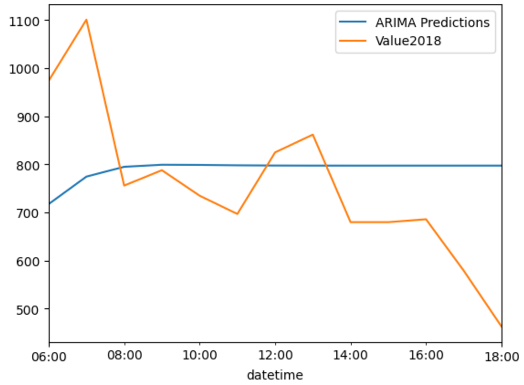


Figure 15. This graph shows a closer look at the testing data vs. the forecast of the ARIMA model.

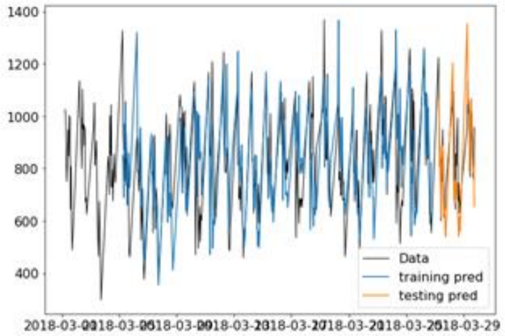


Figure 16. This graph shows the training data, testing data, and the forecasts of the MLP model.

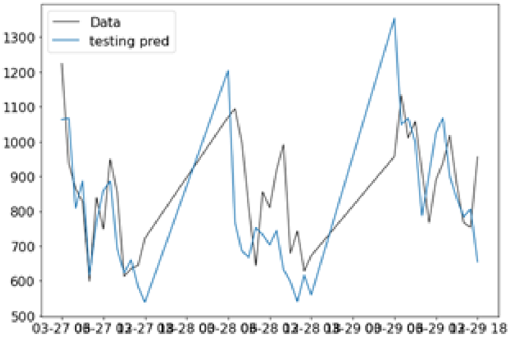


Figure 17. This graph shows a closer look at the testing data vs. the forecast of the MLP model.

Starting with the results of the Simple Exponential Smoothing Model, the prediction which is colored green in Figure 12 and in blue in Figure 13 is producing a singular value for every hour of date of March 20th. It is clear by looking at the testing data, that it should not have produced only one value.

Moving on to the AutoRegression Integrated Moving Average Model. The prediction is also not producing accurate forecasts. However, this prediction differs from the SES model because it starts at a value and then increases to a single value for the remainder of the day.

Finally, when observing the MultiLayer Perceptron Model, we can see that the prediction is more accurate to the testing data as compared to the first two models. However, this model is not measuring the hours in the day of March 20th, it is measuring the following days starting at the 20th and ending on the 29th of March.

Overall, it should be noted that the final model, MultiLayer Perceptron Model, was built and fitted differently than the previous two models. The first difference was that this model was trained with more data than the others. This model was trained with the entire month of March rather than starting at the 6th and ending at the 19th. The second difference was that there was no specified time frame for each day as there was in the first two models. Instead of the 6am to 6pm time frame, it was trained with the regular 24-hour time frame. The third difference is that in the previous models it did not specify how the prediction should be made, however in the MLP there were specific instruction to use three days of data in order to predict one day. The final difference is based off the first difference, because the model was trained with more data, the model was able to predict more days. These differences made a large difference in the prediction that these models produced.

V. CONCLUSION

After reviewing the data and the results, it is clear that there is no decision of which model is more accurate. However, if I were to revisit this research again, I have a few changes I would make. I would change the time frame for the data (6am - 6pm) because by reducing the hours to only the busy hours, it provided gaps on the time series forecasts and led to inaccurate predicting. I would also train the models with more data because it is clear based on the MLP model, that the more data these models have to train with, the more accurate the predictions seem to be. The final change I would make is to implement the other models that we had planned to use in order to have more results to compare. The results did not lead to a decision on which machine learning model was more accurate, but it led to more information on why a machine learning model might not be as accurate.

VI. IMPACT

After these changes to the models, we can then decide which model is the best at accurately predicting the values. When this is determined, these findings can allow the airports to use these predictions to prepare their resource management, operational efficiency, and overall passenger experience. Besides airports the model can also help other businesses and organizations make better decisions due to better prediction accuracy. Determining more accurate forecasting models can be helpful for everyone because it can provide better understanding of future trends and challenges.

VII. ACKNOWLEDGEMENTS

I would like to thank Dr. Amanda Howard who acted as another mentor for me and for supporting me through weekly meetings and for providing the MLP code. I would also like to thank my mentor Dr. Panos Stinis and my VFP project manager Nicole Castilleja Bentley, who both helped me get situated into this internship. Finally, I would like to thank my professor Dr. Jiang Guo, for selecting me to join him for this research internship.

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Visiting Faculty Program (VFP).

VIII. REFERENCES

1 *What Is Simple Exponential Smoothing? - Time Series Forecasting in Python* (2023).

2 S. Jaiswal, DataCamp (2024).

3 J. Frost, Statistics By Jim (2021).

4 A. Howard, PNNL (2023).

5 *ARIMA Model In Python| Time Series Forecasting #6*| (2021).

6 M. Auhl, Medium (2021).

Pacific Northwest
National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov