Building Systems
Open Studio and Python Data Driven Project

Prepared by:
Rafael Ferrato, Dimitrios Papadopoulos , and Benjamin Traskunov

Table of context:

Sketch Up and Open Studio:

Background:

The building modeled in the sketch up program is an office building in Thessaloniki, Greece. The building has a pentagon shape with a circular vitrine on one side of the structure. There is only one floor with 3 office spaces and one common area (located in the space of the vitrine. An image of the model from sketch up is represented below in figure 1.

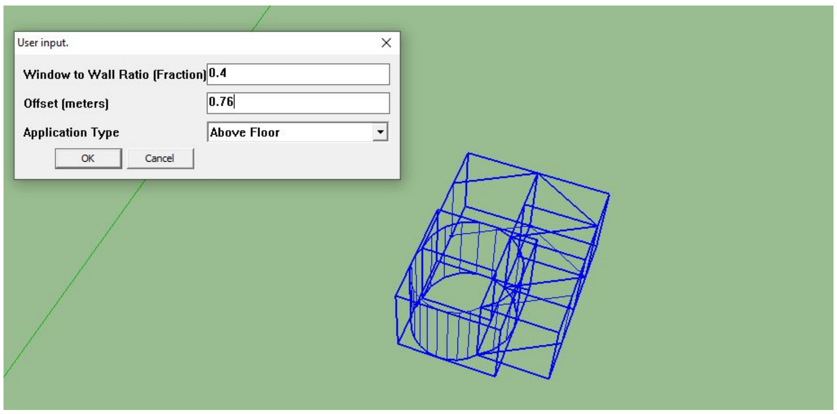

Figure 1: Base model for project analysis with windows .

There were three types of walls analyzed in the energy plus portion of this project: wood, cement, and metal. The makeup of the walls is represented in the table below.

#	Type	Material 1 (Inside)	Material 2	Material 3 (Outside 2/3)	Material 4	Material 5 (Outside)
1	Wood	1 in. Stucco	G 05 25 mm wood	F04 Wall air space resistance	G 05 25 mm wood	1 in. Stucco
2	Cement	G01a 19 mm gypsum board	F 16 Acoustic	8 in Concrete	-	-
3	Metal	½ in Gypsum	I01 25mm insulation board	F08	-	-

Table 1: Wall material makeup

Three different locations for weather related data were picked from the Energy plus website. The locations were Thessaloniki-Greece, Newark-New Jersey, and Brasilia-Brazil. After running the Energy plus software, the location dependant heating and cooling loads were compared.

The usage schedule was kept the same for all cases. The specific type of schedule was a standard schedule included in the software called medium office building occupancy. Everything else was kept at default settings.

Procedure:

The sketch up model , along with the weather and location data, were all loaded into Energy PLus software. Three tests were run for each location (one for each wall type).The final values were then generated into a report (within the software); some of the charts have been reproduced in the result section.

Results:

For each case there is a chart showing the monthly variations of heating load [BTU], cooling load [BTU], and average outdoor temperature [F]. In addition to this, there are tables showing sum of loads for the year, district heating/cooling, and electricity usage. One variable that certainly influenced the energy consumption was the variation of outside dry bulb temperature. This lead to Brasilia having the lowest energy consumption and Newark New Jersey having the greatest. Further correlation of the temperature to energy use is evident in all the location monthly energy consumption amount, which follows the peaks and valleys of the outside temperature.

CASE 1 - BRASÍLIA, BRAZIL.

The data for Cooling Load and Heating Load (all sub cases) has very small variations but they were still displayed with their values tabulated. The variations are not surprising; cement, is not a good heat transfer material and thus does not lose heat or warm up as fast as metal. Wooden walls in heating, had slightly higher heating loads but for cooling had a 50 (+) KBTU difference. The temperature variation is very small which helps in accurately predicting the needed energy consumption. From the following charts, It is recommended to use cement walls in Brazilia.

1.1. Wall type 1 in Brasilia.

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	71.8	71.5	72.7	70.7	69.2	64.1	66.2	68.0	72.6	74.2	71.2	71.5
Cooling Load (MBtu)	63.92	59.8	72.47	57.21	60.17	35.28	41.17	48.92	63.56	72.74	57.15	53.6
Heating Load (MBtu)	0.05	0.17	0.03	0.9	2.15	8.42	6.62	5.57	0.74	0.41	0.09	0.07

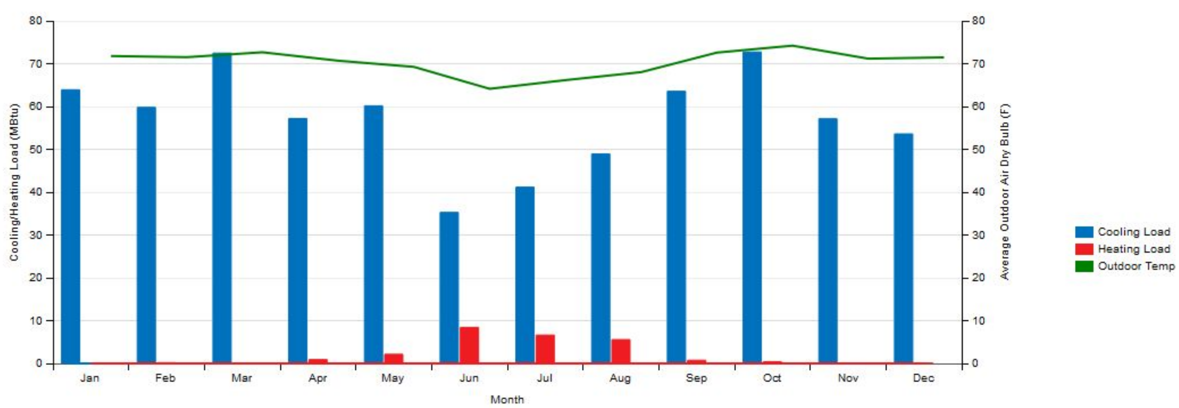

Figure 2: Loads for wall Case 1.1 in Brasilia

End Use - view table

End Use	Consumption (kBtu)
Heating	25,231
Cooling	685,992
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 3: Annual loads for wall Case 1.1 in Brasilia

1.2. Wall type 2 in Brasilia.

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	71.8	71.5	72.7	70.7	69.2	64.1	66.2	68.0	72.6	74.2	71.2	71.5
Cooling Load (MBtu)	59.05	54.55	67.46	52.72	54.27	30.01	36.32	43.75	58.98	67.45	52.62	48.83
Heating Load (MBtu)	0.07	0.18	0.03	0.7	1.66	7.01	5.52	4.6	0.64	0.36	0.12	0.09

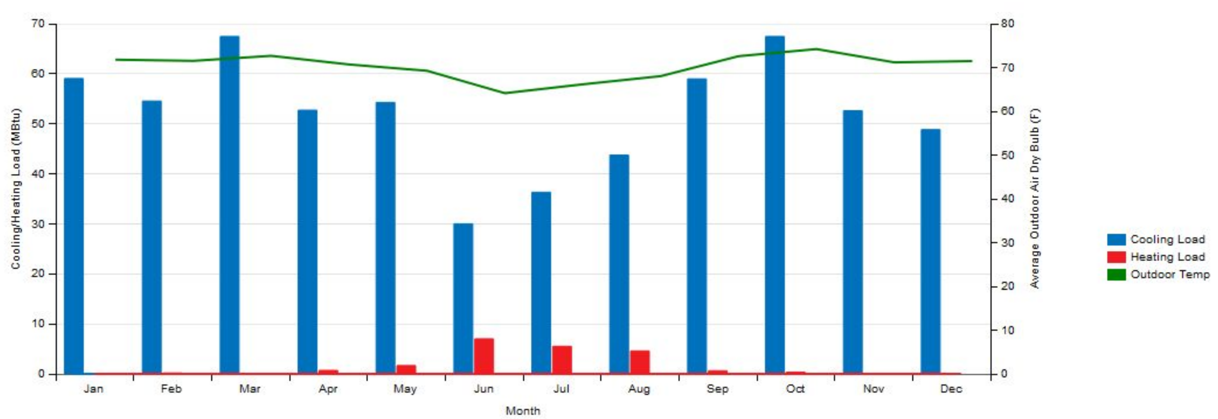

Figure 4: Loads for wall Case 1.2 in Brasilia

End Use - view table

End Use	Consumption (kBtu)
Heating	20,975
Cooling	626,024
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 5: Annual loads for wall Case 1.2 in Brasilia

1.3. Wall type 3 in Brasilia.

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	71.8	71.5	72.7	70.7	69.2	64.1	66.2	68.0	72.6	74.2	71.2	71.5
Cooling Load (MBtu)	70.06	66.19	79.3	64.61	69.17	44.31	50.4	57.99	70.61	78.86	63.16	60.12
Heating Load (MBtu)	1.28	1.76	1.59	3.58	5.56	14.94	12.84	11.47	3.5	2.09	1.16	0.89

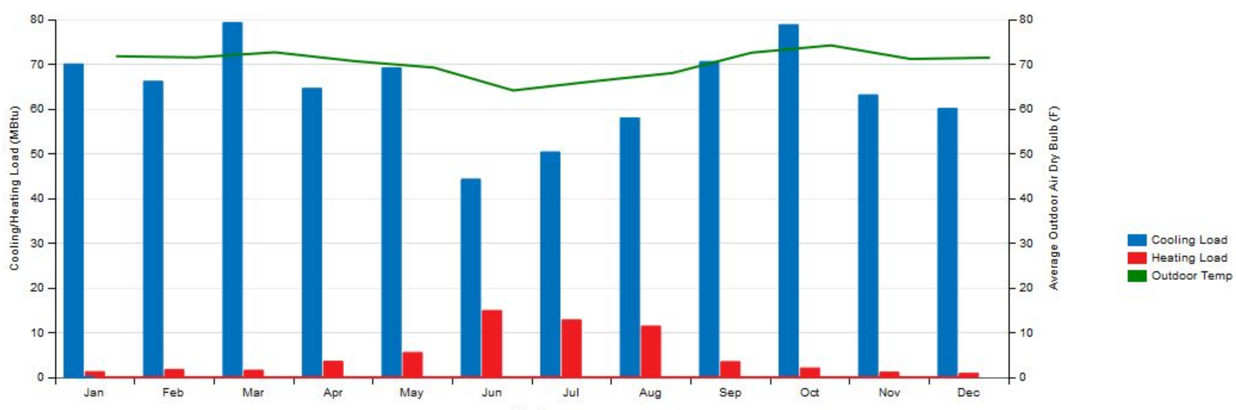

Figure 6: Loads for wall Case 1.3 in Brasilia

End Use - view table

End Use	Consumption (kBtu)
Heating	60,670
Cooling	774,793
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 7: Annual loads for wall Case 1.3 in Brasilia

CASE 2 - THESSALONIKI, GREECE.

The temperature variation in Thessaloniki is much larger than in Brasilia. Due to this, the values for heating and cooling loads are larger . The growth of the values follow the trend of temperature (cooling follows the trend while heating follows the inverse).While in Brazilia, cement walls were the best option to keep cooling and heating values low, for Thessaloniki, cement proved to be the best choice for cooling setups and wood for heating setup. The selection of which material to use will have to involve a techno-economic optimization to see what is the most affordable and reasonable option.

2.1. Wall type 1 in Thessaloniki

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	43.1	43.9	49.1	55.8	65.2	74.0	78.4	77.8	70.3	60.7	52.0	45.3
Cooling Load (MBtu)	0.0	0.03	0.27	4.48	31.58	68.05	94.78	96.82	45.9	16.09	0.91	0.0
Heating Load (MBtu)	124.96	100.33	72.91	30.36	9.24	0.26	0.0	0.0	1.16	21.24	56.35	110.56

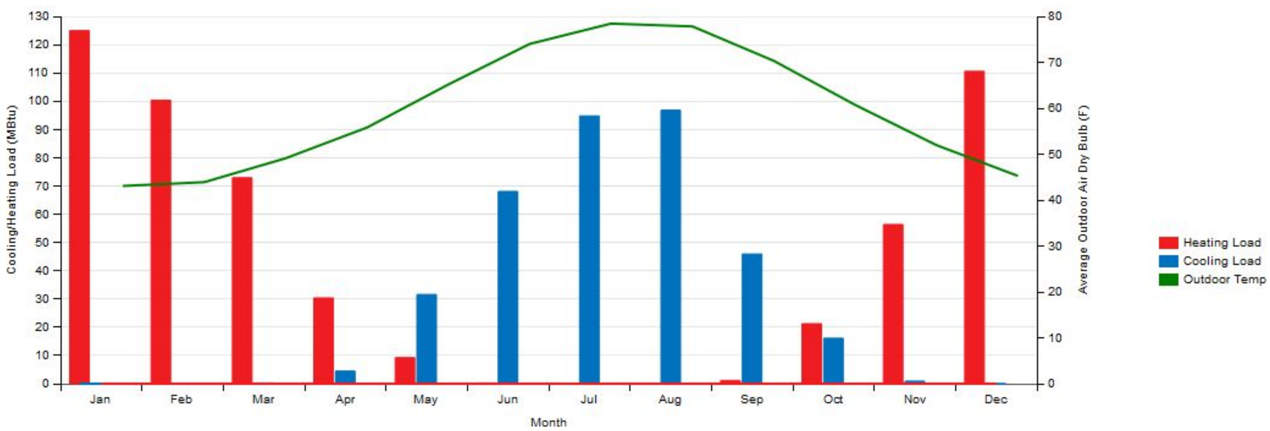

Figure 8. Loads for wall Case 2.1 in Thessaloniki.

End Use - view table

End Use	Consumption (kBtu)
Heating	527,365
Cooling	358,910
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 9. Annual loads for wall Case 2.1 in Thessaloniki.

2.2. Wall type 2 in Thessaloniki

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	43.1	43.9	49.1	55.8	65.2	74.0	78.4	77.8	70.3	60.7	52.0	45.3
Cooling Load (MBtu)	0.0	0.01	0.07	2.77	26.96	60.14	86.02	88.31	40.01	12.27	0.51	0.0
Heating Load (MBtu)	129.08	103.72	75.37	29.62	8.76	0.29	0.0	0.0	0.98	20.39	57.06	114.05

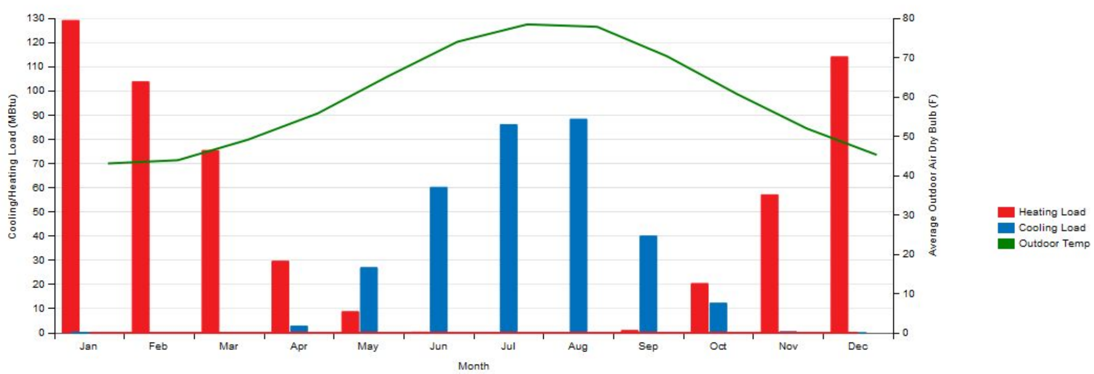

Figure 10. Loads for wall Case 2.2 in Thessaloniki.

End Use - view table

End Use	Consumption (kBtu)
Heating	539,308
Cooling	317,064
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 11. Annual loads for wall Case 2.2 in Thessaloniki.

2.3. Wall type 3 in Thessaloniki

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	43.1	43.9	49.1	55.8	65.2	74.0	78.4	77.8	70.3	60.7	52.0	45.3
Cooling Load (MBtu)	0.0	0.06	0.57	5.78	34.94	71.83	99.38	100.7	49.62	20.49	1.82	0.0
Heating Load (MBtu)	129.48	106.95	79.33	35.9	13.67	1.54	0.39	0.3	3.84	25.11	61.34	114.27

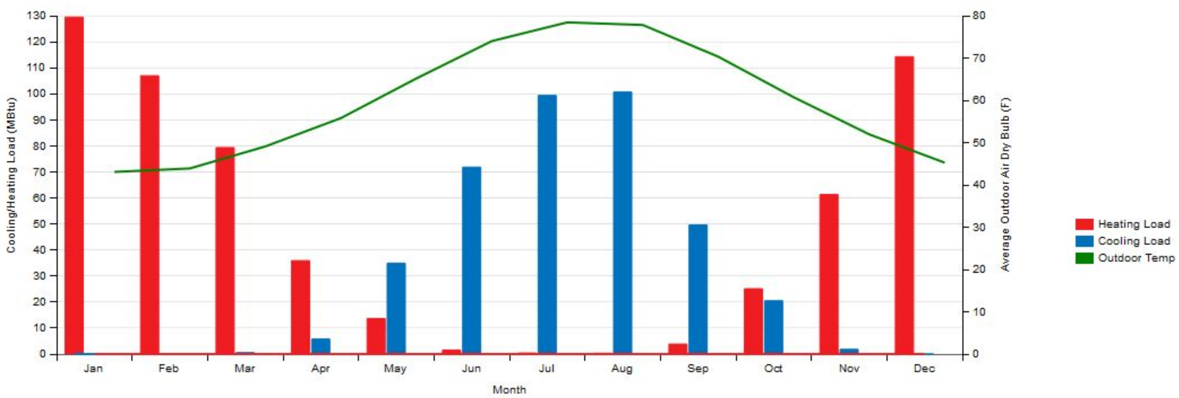

Figure 12. Loads for wall Case 2.3 in Thessaloniki.

End Use - view table

End Use	Consumption (kBtu)
Heating	572,121
Cooling	385,193
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 13. Annual loads for wall Case 2.3 in Thessaloniki.

CASE 3 - NEWARK, UNITED STATES.

Newark New Jersey, being the city most distant from the equator, had the worst heating loads of but the best cooling load of all the cities. A material for such climate, will definitely lean towards something that resists to change in temperature and can insulate a space. Both heating and cooling loads varied by 50 KBTU among all wall types, for Newark, cement proved to be the best option for reducing all the loads.

3.1. Wall type 1 in Newark

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	29.4	33.2	41.2	49.7	62.8	70.8	75.5	75.8	67.6	58.0	45.4	35.5
Cooling Load (MBtu)	0.0	0.0	0.15	2.55	23.59	51.36	87.41	89.03	33.66	7.74	0.01	0.0
Heating Load (MBtu)	291.85	223.86	174.16	86.21	15.7	1.35	0.24	0.09	4.09	33.64	120.6	235.83

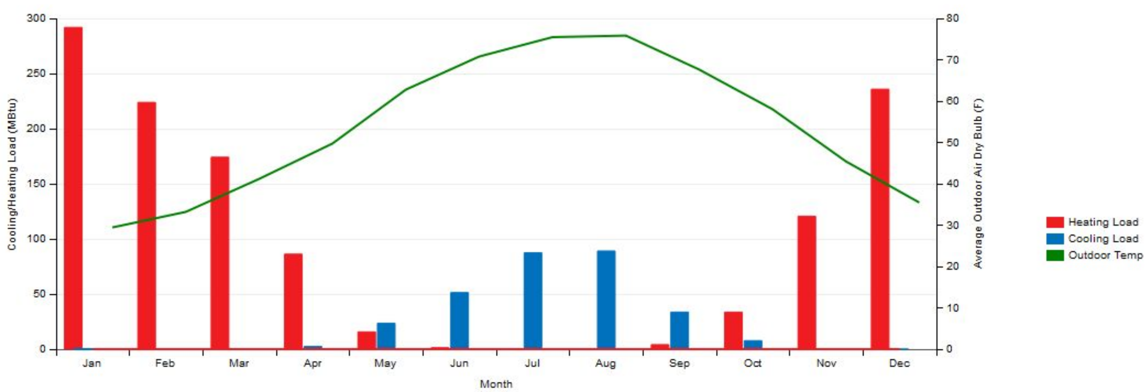

Figure 14. Loads for wall Case 3.1 in Newark.

End Use - view table

End Use	Consumption (kBtu)
Heating	1,187,605
Cooling	295,520
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 15. Annual loads for wall Case 3.1 in Newark.

3.2. Wall type 2 in Newark

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	29.4	33.2	41.2	49.7	62.8	70.8	75.5	75.8	67.6	58.0	45.4	35.5
Cooling Load (MBtu)	0.0	0.0	0.11	1.93	20.7	43.7	74.28	75.11	29.26	6.73	0.01	0.0
Heating Load (MBtu)	281.23	214.71	167.51	82.66	14.69	1.21	0.23	0.08	3.38	30.46	113.91	225.01

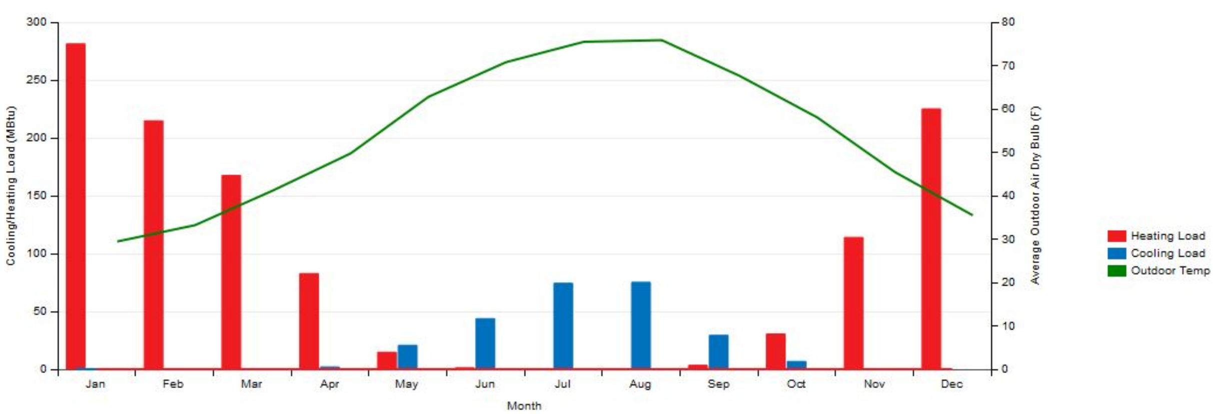

Figure 16. Loads for wall Case 3.2 in Newark.

End Use - view table

End Use	Consumption (kBtu)
Heating	1,135,087
Cooling	251,835
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 17. Annual loads for wall Case 3.2 in Newark.

3.3. Wall type 3 in Newark

HVAC Load Profiles

Monthly Load Profiles - view table

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Average Outdoor Air Dry Bulb (F)	29.4	33.2	41.2	49.7	62.8	70.8	75.5	75.8	67.6	58.0	45.4	35.5
Cooling Load (MBtu)	0.0	0.0	0.54	3.72	26.64	55.06	86.33	87.3	38.43	10.43	0.17	0.0
Heating Load (MBtu)	276.7	213.75	167.76	85.06	18.26	2.21	0.64	0.37	6.39	36.42	114.88	223.8

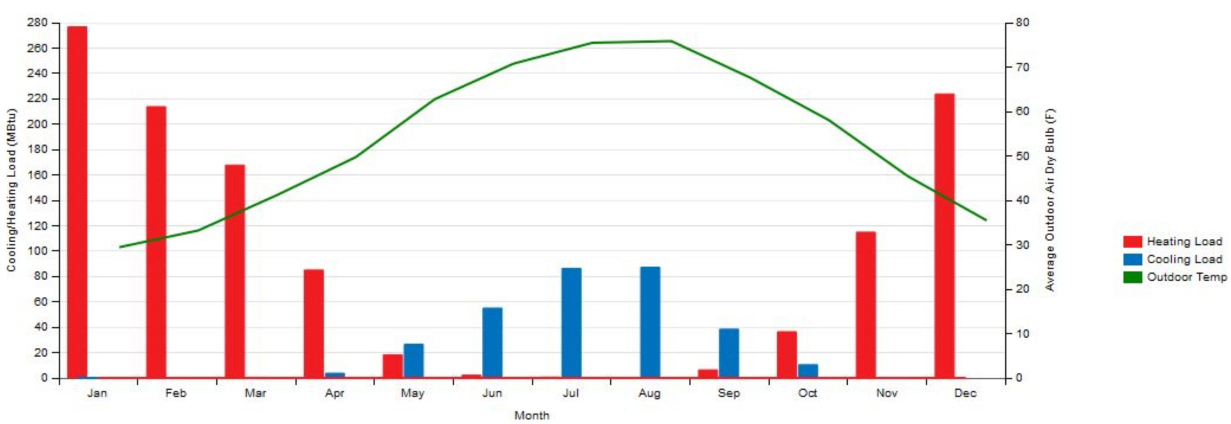

Figure 18. Loads for wall Case 3.3 in Newark.

End Use - view table

End Use	Consumption (kBtu)
Heating	1,146,224
Cooling	308,609
Interior Lighting	143,585
Exterior Lighting	0
Interior Equipment	438,470

Figure 19. Annual loads for wall Case 3.3 in Newark.

Final Remarks:

The results of this project are by no means the final solution for what walls should go in all three locations, but are a great indicator for the initial steps of construction and design planning. Cement proved to be a solution that be utilized world wide to help keep loads down (and costs) while metal proved to be the worst choice. In order to have a better exact decision on wall construction for the office building, more information needs to be provided to the software.

Machine Learning:

Background:

In this portion of the project, the group was tasked with selecting a database and with python, create a machine learning code that could predict occupancy in a room with data such as temperature, CO2 concentration, humidity, light, and etc. The group selected an already finished project from UC Irvine University Machine Learning repository. The raw data was from a 201 research project that aimed to predict office room occupancy using sensor collected data and output 0 or 1 (un occupied or occupied).

Procedure:

Data source and libraries used:

The library used in this project were: pandas, numpy, matplotlib, seaborn and os. Each library helped make the processing and prediction processes. Pandas, an obvious choice for processing data sets in scientific fields combine with numpy's embedded calculation methods helped the code crunch numbers quickly and smoothly. Matplotlib helped the team presents its results in clear and versatile graphics. Seaborn (library based on Matplotlib) was used to help visualise statistical data. Keras is a library written in python that helps users design and experiment with deep neural networks.

The steps in the code were as follows: Import modules, import datasets, creating timestamps, classification of the Neural Network, Training and Testing, Checkpoint, Training, and Results. Each section is explain in detail below.

Importing modules:

In this section, all the previous libraries are imported. Then the path for the folder where everything will be saved and the raw data (CSV file) is written into the code. A check is made to make sure such files exists before proceeding. Then the code moves to the importing datasets section.

Importing datasets:

This section is very straight forward in the code. For future references the data set is given a reference name and is printed so that the coder can see the first few and last 5 rows of the data.

Creating Timesteps:

The data sets time and date information is not in a format that is easily read by python, therefore it must be transformed into an easily read time stamp format. This is done by taking the old dataframe and with a function [pd.to_datetime()] transform it into the necessary format. Another print is done afterwards to make sure that the data is converted properly.

Classification - Neural Network:

This section had a few sub steps included in it. The first was to import some features from the keras library. The features from Keras that were imported were model (collection of data from the neural network), layers (building blocks of Keras models) and Model checkpoint (Snapshot of a run , also serves as a safety net incase the code crashes). The model used in this code was a sequential model. This model allows us to create the neural network layer by layer.

Secondly, the data needed to be classified. The humidity ratio is also removed in this step since only the Humidity value will be used.This categorized data was then reorganized so that the neural network does not get used to the order (not always a given when predicting). The next step is to normalize the values in the Temperature,Humidity,Light, and CO2 columns. The values were normalized using the formula: $(data_i - data_{mean}) / data_{std}$. The new values were entered into the data frame. Finally , two lists were made; one had the normalized data in it and the other had the binary occupancy value in it. With the data cleaned and prepared, the code moves onto the training portion.

Training and Testing:

The length of the data that is used to train the program correlates to the programs prediction accuracy. It was decided that 80% of the data will be used for training ; the remaining 20% will be used for testing the model.

Checkpoint:

The neural network will be able to save checkpoints with every run. After the run, the neural network will compare the results accuracy and save the best run.

Training:

When training , It is necessary to make sure that there is no over use the data or under use it. The the results very printed to see how accurate each run (Epoch) was.

Plotting:

In order to track how accurate the model is with every run, a dictionary with the accuracy values was created to record every Epoch test and the final values were plotted. Another line that was plotted was a validation accuracy array that the neural network has never seen before (testing part of the model processes). The graph with both lines (training blue and testing green) is represented below in figure 20.

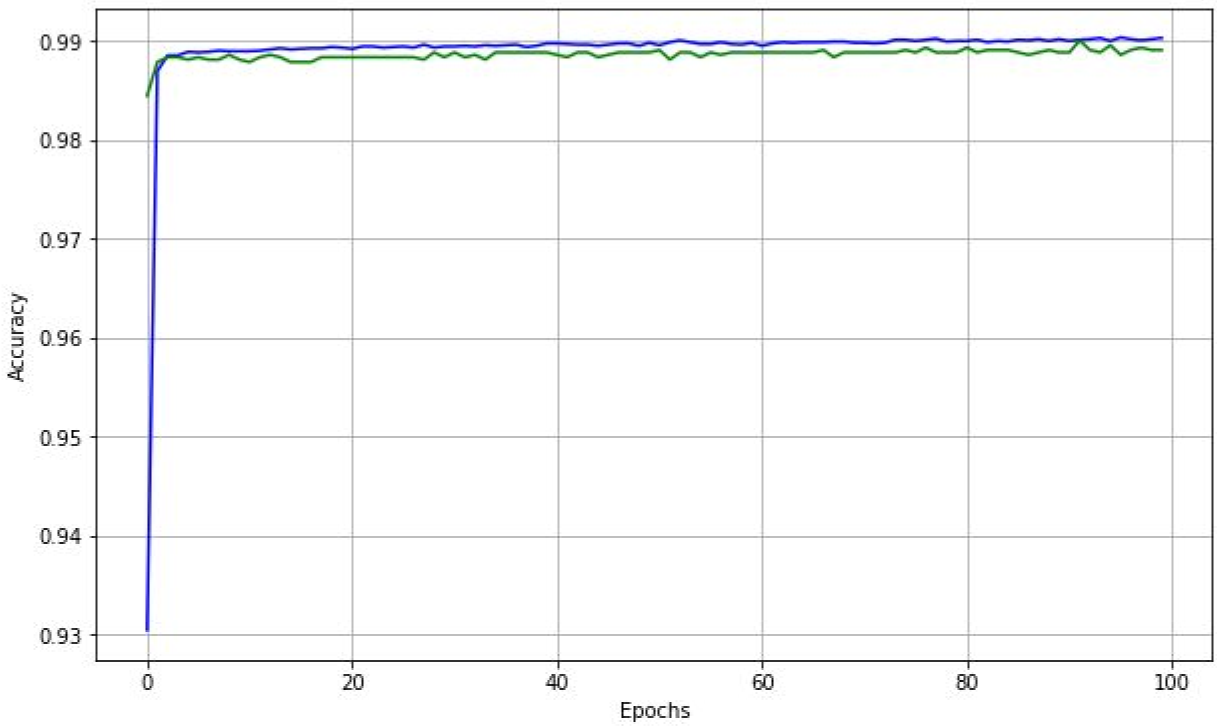

Figure 20: Comparison between training accuracy (blue) and testing accuracy (green)

Results:

The results of the project were broken down into multiple sections. The sections were: Occupancy Classification Validation, and Occupancy 2 Week Predictions. The 2 week prediction portion had three models: Linear Regression, Cross Validation, and Random Forests. Each section involved a different method to test the neural networks accuracy and in the end all were compared to see which one is the best.

Occupancy Classification Validation:

In this section, two functions were created, one was to get the index in the data frame and the other was to send it to the neural network and generate a prediction for the selected index (date and time). This allowed the user to pick what time they want to predict and have the program guess the results based on its training . The functions are represented below in figure 21.

```
[ ]  def get_index_by_timestamp(df, time_stamp):
         index = DATA_DF[DATA_DF["Time_Stamp"] == time_stamp].index[0]
         return index
```

```
[ ]  def predict_occupancy(df, index):

         test_x = np.array([[ df["Temperature"][index], df["Humidity"][index], df["Light"][index], df["CO2"][index]
         prediction = model.predict_classes(test_x)[0][0]
         if prediction == 1 :
           print("Neural Network Prediction: Room is occupied")
         else :
           print("Neural Network Prediction: Room is empty")

         real_occupancy = df["Occupancy"][index]
         if real_occupancy == 1:
           print ("Real occupancy: The room is occupied")
         else:
           print ("Real occupancy: room is empty")
```

Figure 21: Selected time occupancy prediction functions

Occupancy 2 Week Predictions:

The three models used in this portion Linear Regression - Split, Cross Validation, and Random Forest, were all part of the sklearn model. Sklearn library (Scikit-learn), is an open source Machine learning library that allows comparison, validation and parameter selection.

Linear Regression- Split:

To use the linear regression model, it was necessary to dates and times to float values. This was achieved by importing date time library from datetime and converting all the time stamps from timestamp format to floats.

Following this, the train_test_split model was imported from the sklearn library and define what the train and test data was (80% train 20% test). The next step was to import the linear regression model from the sklearn library and using it , output the predicted occupancy for each time step. The results could then be plotted in a scatter plot graph (figure 22) with the x-axis being time and y axis being the predicted occupancy.

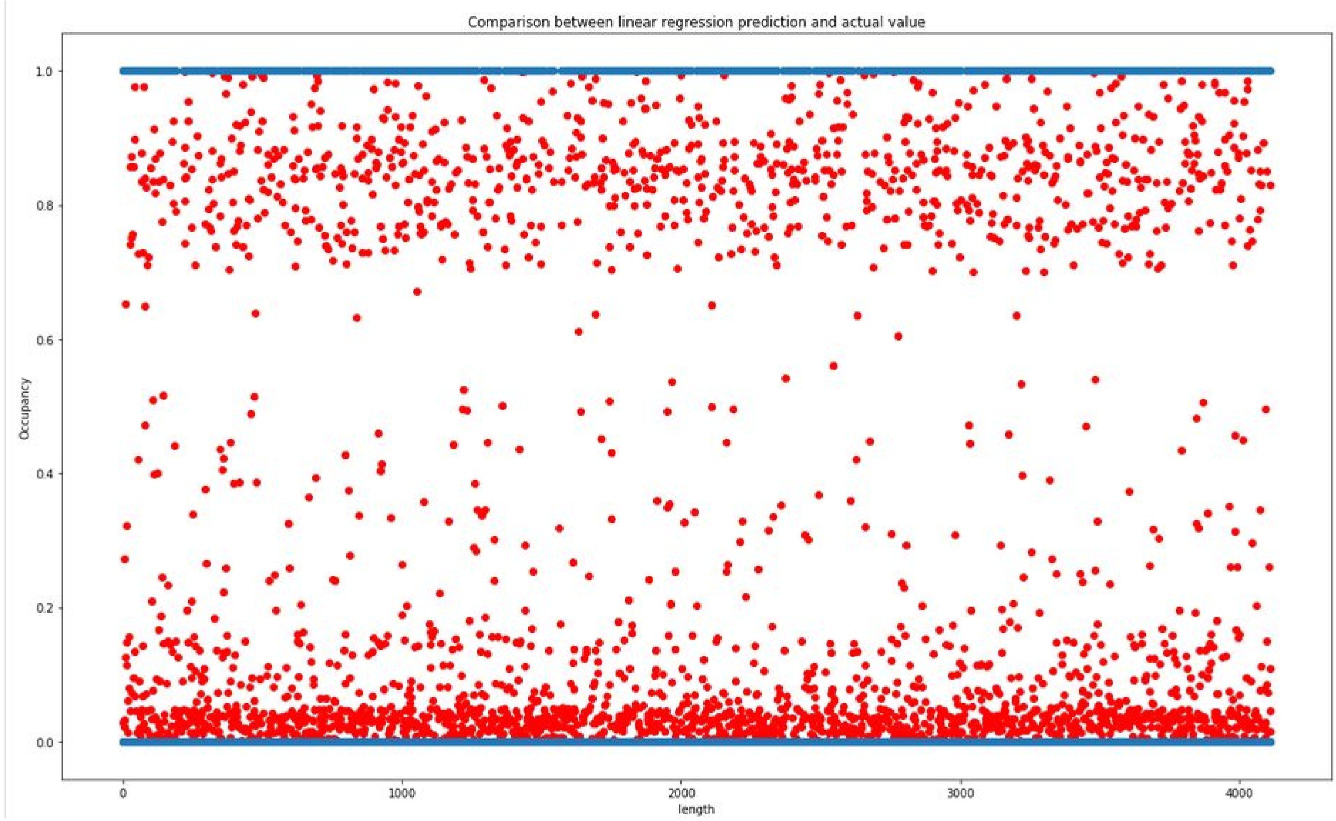

Figure 22: Linear regression occupancy prediction

In addition to the scatter plots, each method used for the 2 week predictions was accompanied by mean absolute error, mean squared error and the r2 score. The calculations for these values we derived using the specific functions from the sklearn.metrics library. The resulting values are reported in table 2 below.

Table 2	
Linear Regression Statistical Values	
Mean absolute error	0.09229028283841711
Mean squared error	0.02255208067527324
R2 score	0.8521777907509003

Cross validation:

The cross validation model is also part of the sklear library. Importing the cross_val_predict model and adding an additional column in the data frame called “Predicted Occupancy” the model was run and the results tabulated. For these predictions, there was no need to modify the timestamps to floats. If the predicted value was less than 1, the code set the prediction as 0, otherwise it would be set as 1 (0 meaning unoccupied 1 meaning occupied). The resulting prediction values (not converted to 0 or 1) are reproduced in figure 23 below. The X and Y axis for this scatter plot are the same as in the Linear regression model plot (figure 22).

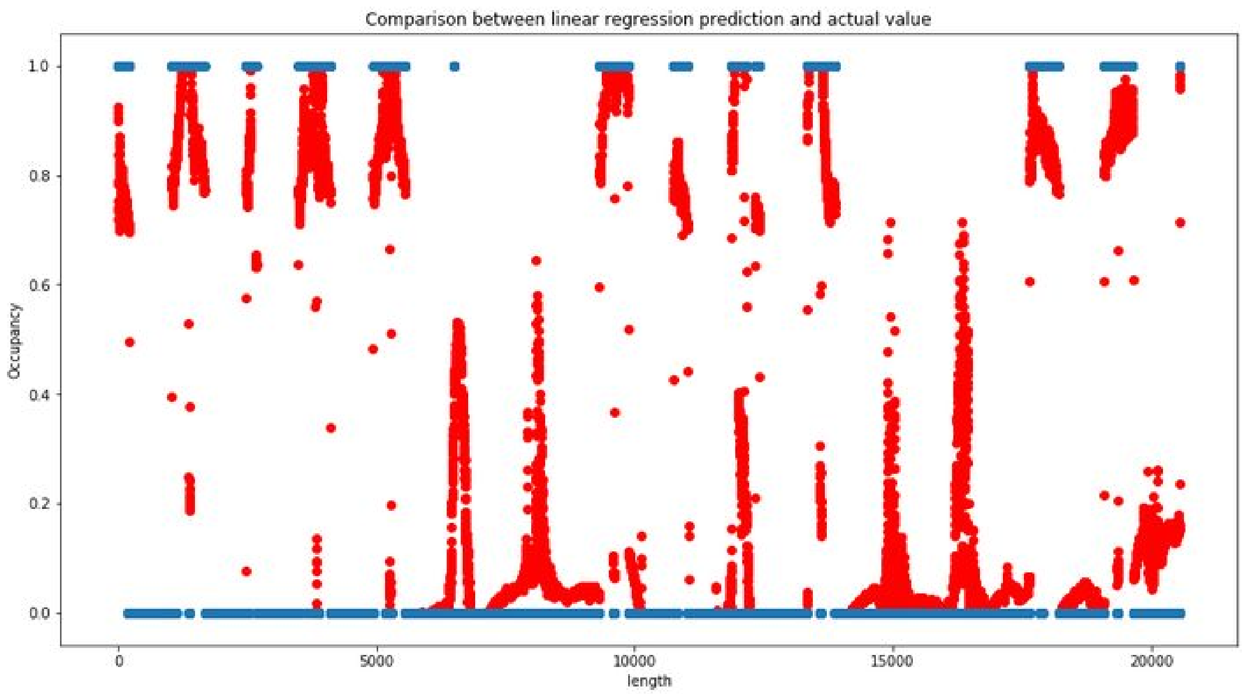

Figure 23: Cross validation occupancy prediction

Just like for the Linear regression model, statistical values were calculated to quickly understand the prediction models accuracy. The results were reproduced in table 3.

Table 3	
Cross Validation Statistical Values	
Mean absolute error	0.09269922134182103
Mean squared error	0.02442146780202939
R2 score	0.8409531208396086

Random Forest:

Random Forest is an estimator that fits a decision tree on samples of the dataset. It uses the average to improve the predictive accuracy and to control over fitting. Just like in all the other models, the new prediction values are added in a column called Predicted Occupancy. Ranging from 0 to 1. The results are displayed in figure 24.

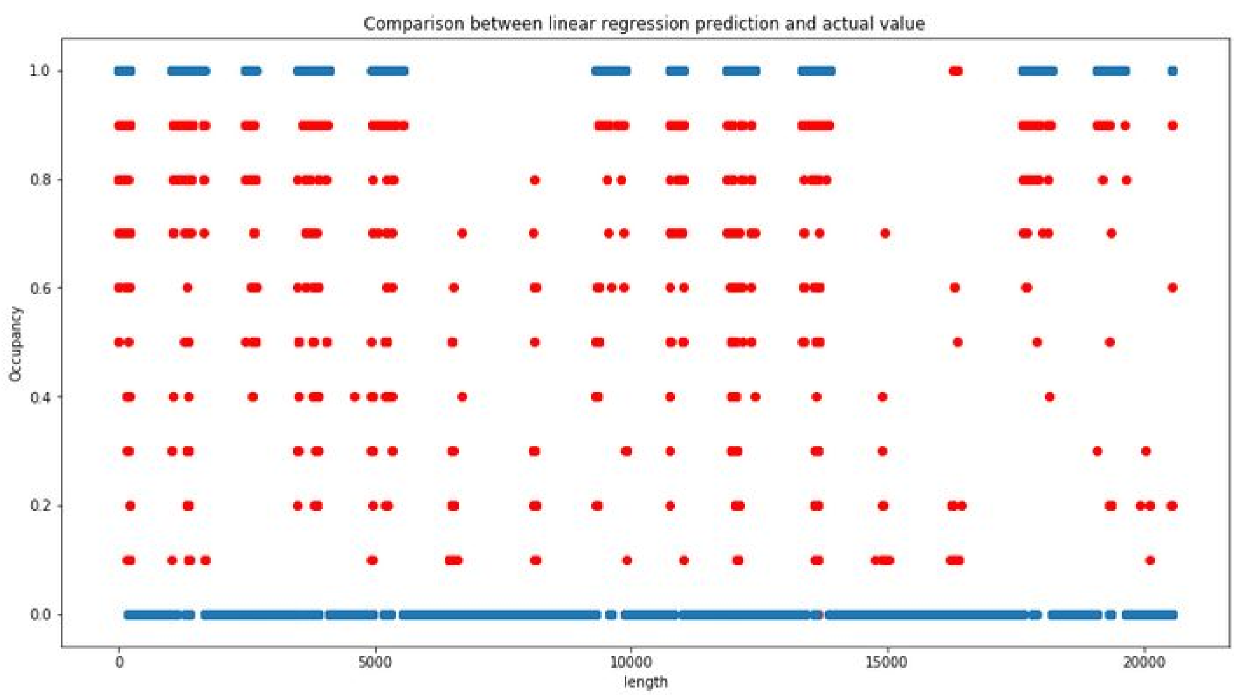

Figure 24: Random Forest occupancy prediction

The resulting statistical data for the Random forest model as found in table 4.

Table 4	
Random Forest Statistical Values	
Mean absolute error	0.02446498054474708
Mean squared error	0.013797665369649806
R2 score	0.9165850880268429

Final Remarks:

The resulting r2 scores for each model were compared side by side and the clear winner was the Random Forest model. The other models were not far behind; showing that the neural network was built well and that all that was necessary were minor tweaks to increase the accuracy. Further development can be done in terms of increased accuracy, and even prediction of the number of occupants in a room based on the same data.