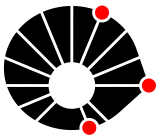

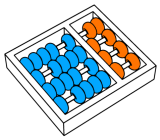

Influência de fatores climáticos no consumo de energia elétrica: um estudo de caso na Unicamp

A. S. Gonçalves *L. F. Gonzalez* *J. F. Borin*

Relatório Técnico - IC-PFG-19-52
Projeto Final de Graduação
2019 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Influência de fatores climáticos no consumo de energia elétrica: um estudo de caso na Unicamp

André de Souza Gonçalves [*] Luis Fernando Gomez Gonzalez [†]
Juliana Freitag Borin [‡]

Resumo

Este trabalho destinou-se analisar os efeitos de fatores climáticos no consumo de energia elétrica do prédio da Prefeitura da UNICAMP utilizando técnicas de aprendizado de máquina de modo a gerar um melhor conhecimento sobre o consumo energético e posterior gerenciamento mais eficiente desse recurso da universidade. O melhor modelo analisado obteve erro quadrático de 20.54 para os dados de teste comparando o valor real com o previsto.

1 Introdução

Em busca de promover a sustentabilidade e um melhor gerenciamento da infraestrutura e da mobilidade, muitas cidades estão utilizando tecnologias da Internet das Coisas para interconectar dispositivos, sistemas, ambientes e pessoas. As chamadas *Smart Cities* utilizam os dados dessas relações de modo a gerarem, entre outros benefícios, um uso mais eficiente dos recursos naturais e financeiros. Neste contexto, o consumo energético mais eficiente e sustentável tem estado em evidência.

Através do Projeto *Smart Campus*, diversos pesquisadores vêm realizando trabalhos com o conceito de Internet das Coisas de modo a facilitar a vida de todos que usufruem dos espaços da Unicamp, além de facilitar a tomada de decisões dos gestores da universidade, baseando-as em dados mais fidedignos à realidade. Um dos projetos pilotos do *Smart Campus* consiste em coletar dados de consumo energético no prédio da Prefeitura do campus. Embora esses dados estejam sendo coletados há mais de um ano, ainda não se tinham realizadas análises destes dados de modo a gerar conhecimento e apoiar a gestão deste recurso.

Vários fatores podem influenciar no aumento ou na diminuição de consumo energético. Dado que os aparelhos elétricos que têm um peso maior nesse consumo no prédio da Prefeitura são os aparelhos de ar condicionado, fatores climáticos, como temperatura elevada, umidade relativa baixa, dentre outros, estimulam as pessoas do prédio a ligarem esses aparelhos.

[*]Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.
[†]KonkerLabs
[‡]Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

Com a crescente popularização de técnicas de Aprendizado de Máquina, novas pesquisas surgem dentro desse tema, inclusive com aplicações na predição de clima. Em sua pesquisa, A. H. M. Jakaria [1] utilizou diversos modelos de Aprendizado de Máquina na tentativa de prever o clima no estado de Tenessee, nos Estados Unidos. Essa pesquisa motivou e levantou questionamentos sobre a aplicação de alguns dos modelos na predição de consumo elétrico tendo os fatores climáticos como informação de entrada.

O CEPAGRI[1] realiza a coleta e armazenamento de dados de vários fatores climáticos, que, juntamente com as medidas de corrente elétrica coletadas pelo dispositivo no prédio da Prefeitura da Unicamp e armazenadas na Plataforma Konker[2],possibilita realizar estudos sobre a correlação entre consumo elétrico e clima com o uso de técnicas de Aprendizado de Máquina.

2 Justificativa

Ainda na Unicamp, o projeto Campus Sustentável tem como uma das frentes a distribuição de medidores inteligentes para coletar dados de consumo energético em vários prédios da universidade. Esses dados brutos também precisarão ser analisados.

Acreditamos que o uso de técnicas de aprendizado de máquina pode apoiar a análise desses dados com o intuito de gerar um melhor conhecimento sobre o consumo energético e posterior gerenciamento mais eficiente dos recursos energéticos da universidade pelos seus gestores.

3 Objetivos

Esse projeto tem como objetivo auxiliar os gestores da Prefeitura da UNICAMP no entendimento do consumo elétrico do prédio e nas tomadas de decisão nas discussões que tangem o assunto de economia energética, através da predição esperada de consumo em um dado período.

4 Desenvolvimento do Trabalho

O desenvolvimento do trabalho consistiu em 6 etapas: na primeira, procurou-se entender quais dados se tinham disponíveis para estudo e como fazer análises a partir desses dados; na segunda, houve uma análise dos fatores climáticos e as suas influências sobre a corrente medida no prédio da prefeitura; na terceira, realizou-se um estudo mais detalhado sobre a relação da temperatura com a corrente medida; em seguida, na quarta etapa, realizou-se uma análise de 4 modelos de aprendizado de máquina e de seus resultados, além de utilizar técnicas de otimizações em alguns modelos; na quinta, foi feita a escolha do melhor modelo para uma análise dos resultados; e por fim, na sexta etapa, realizou-se uma projeção do consumo elétrico de novembro de 2019 utilizando a temperatura média medida no mesmo mês de 2013 a 2017.

[1] Centro de Pesquisas Meteorológicas e Climáticas Aplicadas à Agricultura.

[2] Plataforma que ajuda negócios a construir e operar uma solução de Internet das Coisas

4.1 Coleta de dados

No primeiro momento do projeto, procurou-se entender quais dados e informações estão disponíveis para análises. Com o intuito de coletar as informações de corrente, foi instalado um dispositivo na caixa elétrica do prédio da Prefeitura da UNICAMP que realiza uma medida da amperagem da corrente a cada minuto. Já para os dados climáticos, como temperatura, sensação térmica, umidade, pressão, dentre outros, a coleta é feita a partir das medidas capturadas pelo CEPAGRI a cada 10 minutos. Todos esses dados são armazenados e gerenciados na plataforma Konker, e a requisição de dados é feita pela API da plataforma, que retorna as informações em formato JSON.

Com os arquivos JSON do dispositivo "medidor", que contém os dados medidos no prédio da Prefeitura e do dispositivo "cepagri", que possui os dados climáticos coletados pelo site do CEPAGRI, foram criados um *dataframe* para cada dispositivo. Através da técnica de *resampling*, que consiste em ajustar os dados coletados do *dataframe* a uma determinada frequência, ajustou-se os dados do medidor para cada minuto, por exemplo, 10:00:00, 10:01:00 e 10:02:00. Como os dados coletados pelo CEPAGRI ocorrem a cada 10 minutos, é necessário fazer o ajuste desses dados para que tenhamos medida a cada minuto. Assumindo que a variação climática é muito baixa a cada 10 minutos, foi utilizada a técnica de *downsampling* com preenchimento por proximidade, para gerar os dados dentro do intervalo de 10 minutos baseado na medida mais próxima. Por exemplo, para os horários de 10:00:00 e 10:10:00 tendo temperaturas de 27º e 28º, respectivamente, a temperatura para 10:04:00 será 27º, e para 10:07:00 será 28º.

Com os dois *dataframes* ajustados na mesma frequência temporal, é possível realizar o *inner join*[3] dessas duas tabelas, utilizando o índice, que é o próprio *timestamp*, de modo que tenhamos informações sobre o clima na UNICAMP e a corrente medida pelo dispositivo a cada minuto. Com isso, obteve-se dados de 06/05/2019 às 15:00 até 03/11/2019 às 23:00, totalizando 261.106 medidas.

4.2 Influência de fatores climáticos no consumo elétrico

Nesta etapa do projeto, procurou-se entender como elementos climáticos influenciam no consumo elétrico do prédio da Prefeitura da UNICAMP. Então, coletou-se dados de temperatura, sensação térmica, umidade relativa, índice de pluviosidade, vento e pressão pelo dispositivo do CEPAGRI e medida de corrente pelo dispositivo do prédio.

Para analisar a influência, utilizou-se uma Matriz de Correlação, onde cada parâmetro é avaliado quanto a sua variação comparada com a variação de outro parâmetro. O resultado pode ser visto na Figura 1.

[3]Operação de unir duas tabelas em que ambas possuem uma determinada chave única.

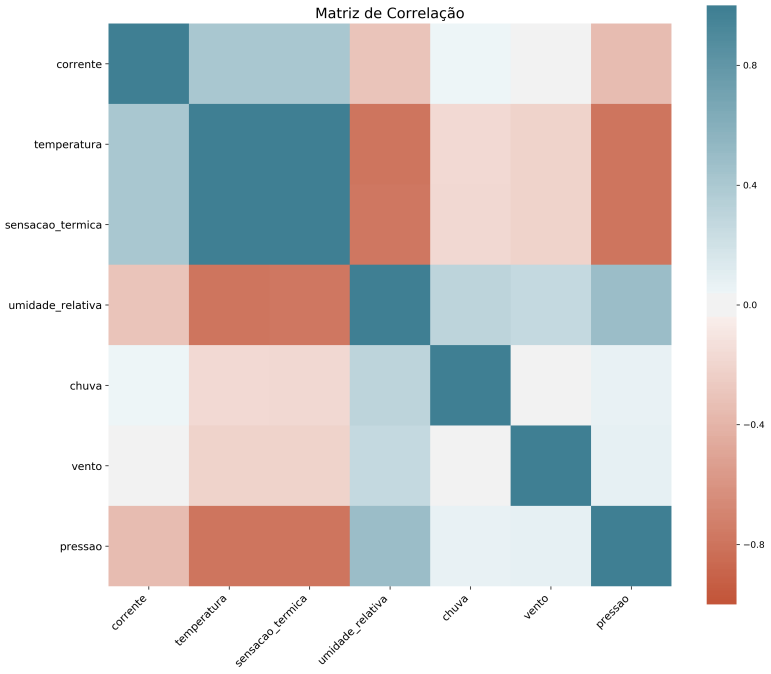

Figura 1: Matriz de Correlação dos fatores climáticos com a corrente elétrica.

Pela Figura 1, percebe-se que a corrente só possui uma variação positiva com os fatores de temperatura e sensação térmica, variação quase nula com fatores de chuva e vento e variação negativa com umidade relativa e pressão.

Agora, analisando e correlacionando a temperatura com os demais parâmetros, pode-se perceber que ela possui uma relação inversa mais intensa com a umidade relativa e com a pressão. Como a temperatura está fortemente inversamente relacionada com esses dois fatores e diretamente relacionada com a sensação térmica, a temperatura se mostra como o parâmetro climático mais importante, uma vez que ele rege o comportamento dos demais. Incluir esses parâmetros climáticos pode fazer com que modelos de Aprendizado de Máquina fiquem mais enviesados porque a temperatura estaria sendo reforçada nos outros parâmetros.

4.3 Efeitos da temperatura no consumo elétrico

Temos até então que a temperatura é o atributo que mais se correlaciona com a variação de corrente no prédio da Prefeitura da UNICAMP. Nessa etapa do trabalho, procurou-se entender se existem outros fatores, que não sejam climáticos, que podem também afetar o consumo elétrico.

4.3.1 A correlação entre a temperatura e a corrente

Com o objetivo de entender de maneira visual a relação entre temperatura e corrente, coletou-se os dados com a temperatura e corrente média a cada hora e gerou-se os gráficos da temperatura e corrente média ao longo do tempo para cada mês que podem ser vistos nas Figuras de 2 a 7.

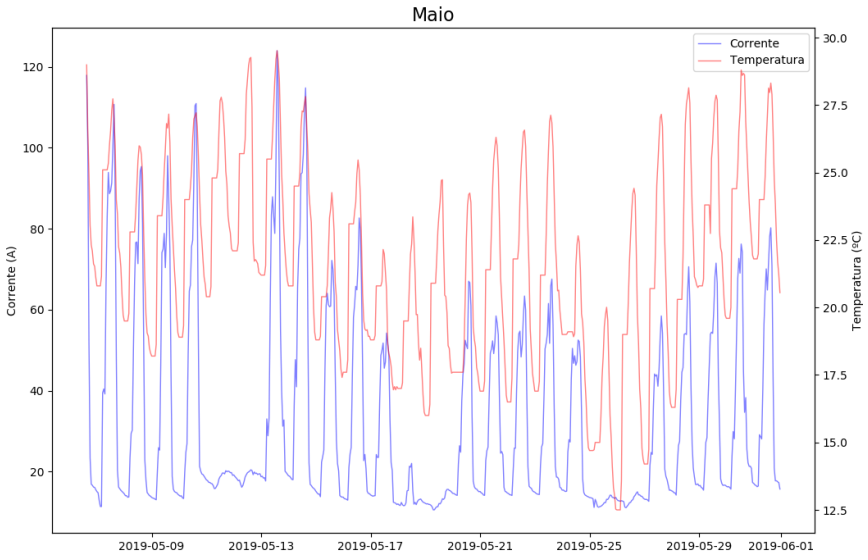

Figura 2: Distribuição de corrente para faixas de temperatura - Maio

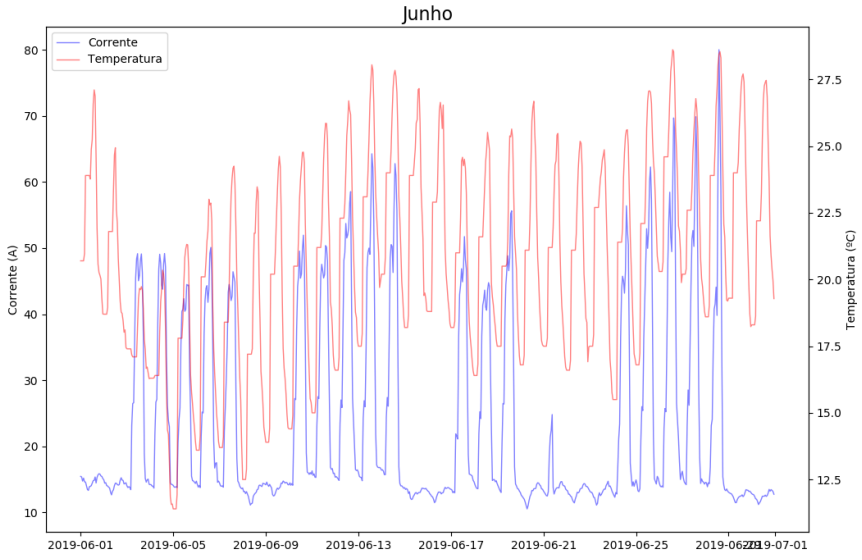

Figura 3: Distribuição de corrente para faixas de temperatura - Junho

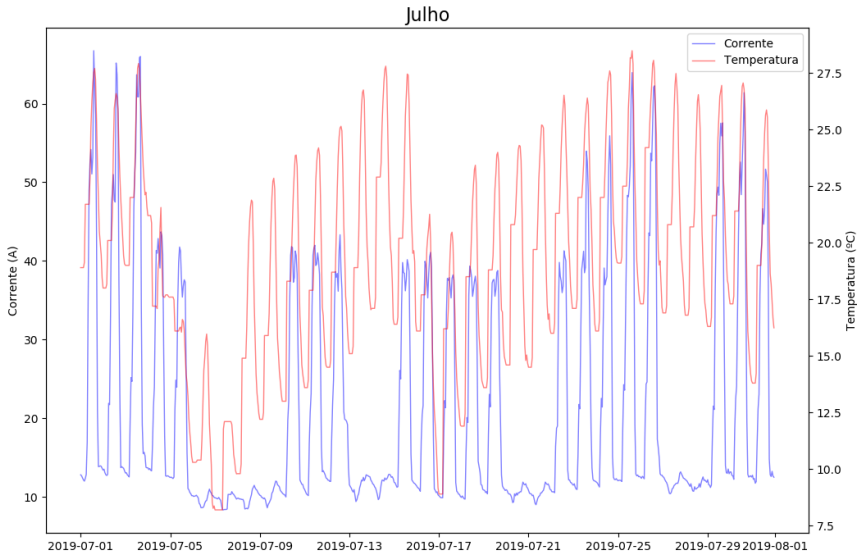

Figura 4: Distribuição de corrente para faixas de temperatura - Julho

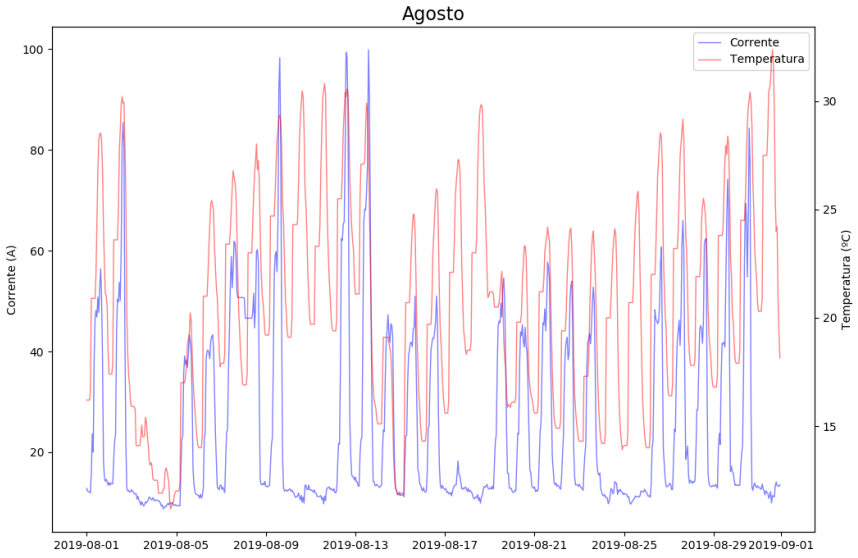

Figura 5: Distribuição de corrente para faixas de temperatura - Agosto

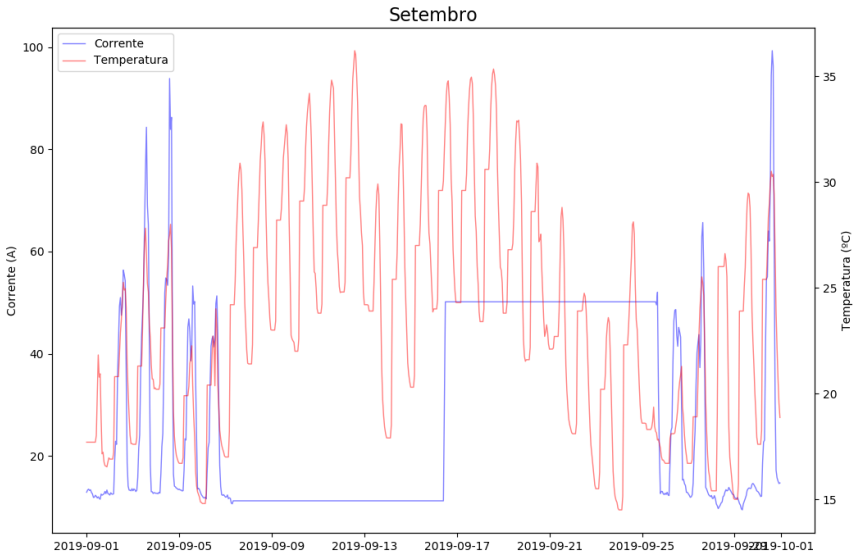

Figura 6: Distribuição de corrente para faixas de temperatura - Setembro

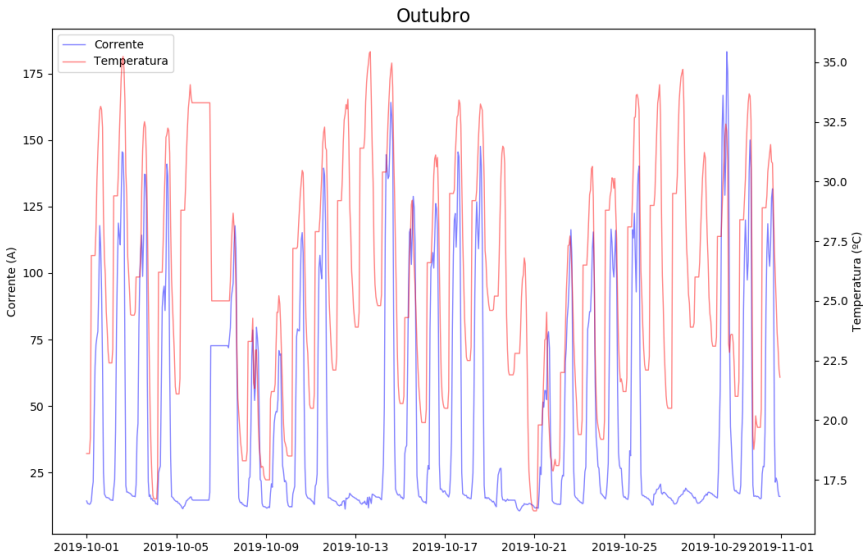

Figura 7: Distribuição de corrente para faixas de temperatura - Outubro

Pela Figura 6 que representa o mês de setembro, pode-se perceber que ouve uma falha no medidor de corrente. Como foi utilizada a técnica de *downsampling*, que utiliza o valor mais próximo encontrado das medidas faltantes, todo o período ficou com apenas duas medidas. Esses casos de erro não foram removidos das análises de modelos de Aprendizado de Máquina porque quis-se avaliar se os mesmos conseguem identificar esse caso e predizer a corrente nesse período.

Assim, com o intuito de entender o efeito da influência humana no consumo elétrico ligando objetos eletrônicos, principalmente ar condicionado, no prédio da Prefeitura, selecionou-se as medidas que ocorreram durante horário comercial, das oito horas da manhã às seis da tarde, e de segunda a sexta. Com isso, gerou-se um gráfico de correlação de temperatura de corrente, ou seja, qual foi a corrente medida para uma dada temperatura, e um boxplot de correntes medidas para faixas de 5º de temperatura para cada mês.

O gráfico boxplot ajuda a compreender a distribuição de uma amostra de dados. O boxplot divide os dados em 3 seções principais. As hastes inferior e superior representam o valor mínimo e máximo expressivos na distribuição. Quaisquer pontos que não esteja entre essas duas hastes é chamado de *outlier*, que são pontos com ocorrências tão baixas que não influenciam a distribuição, de modo a se definir um novo valor mínimo ou máximo. Os dados localizados dentro do retângulo correspondem a 50% da distribuição, e linha dentro do retângulo corresponde a mediana.

Os resultados podem ser vistos nas Figuras de 8 a 13.

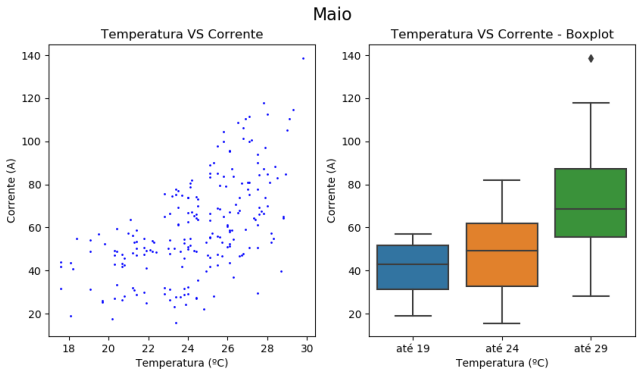

Figura 8: Distribuição de corrente para faixas de temperatura - Maio

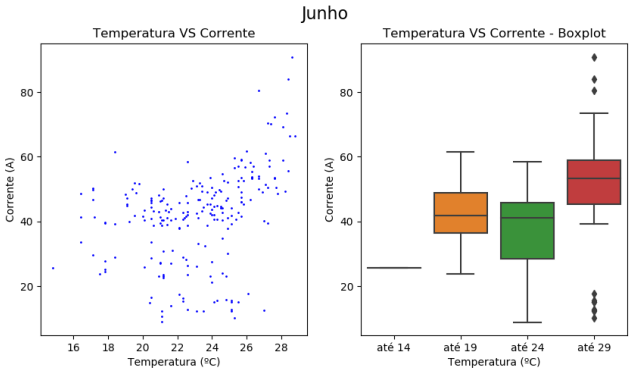

Figura 9: Distribuição de corrente para faixas de temperatura - Junho

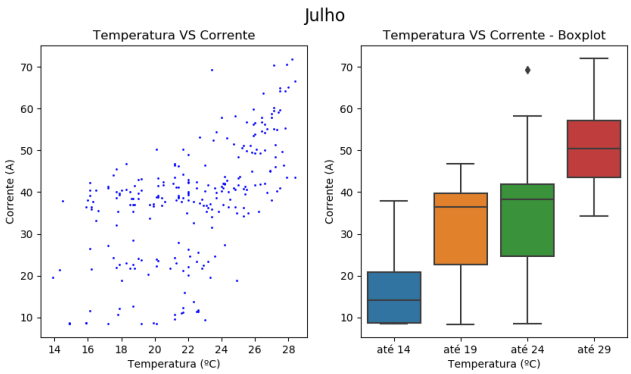

Figura 10: Distribuição de corrente para faixas de temperatura - Julho

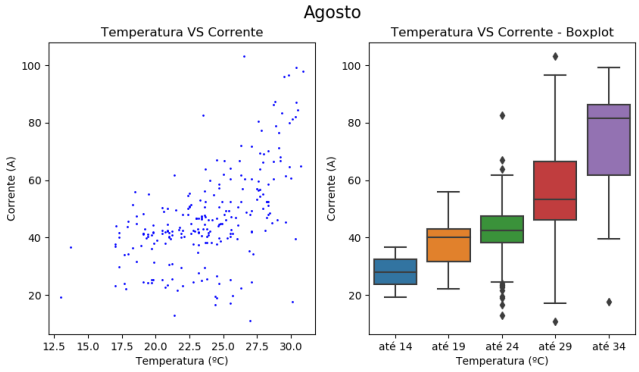

Figura 11: Distribuição de corrente para faixas de temperatura - Agosto

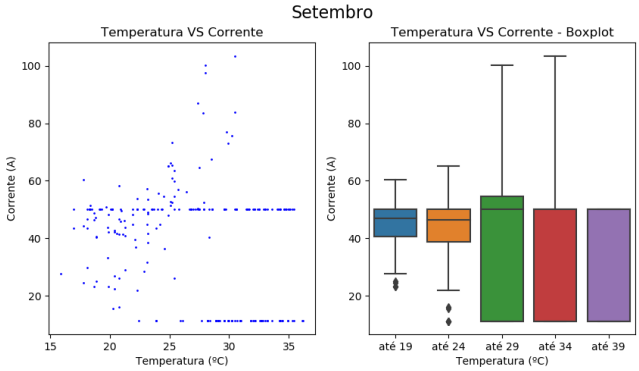

Figura 12: Distribuição de corrente para faixas de temperatura - Setembro

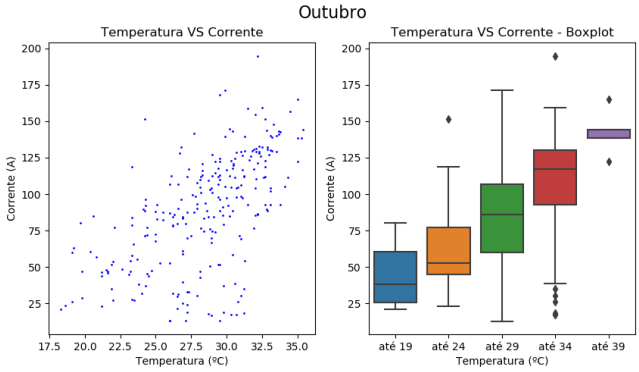

Figura 13: Distribuição de corrente para faixas de temperatura - Outubro

Dos gráficos de distribuição de temperatura e corrente média a cada hora, percebe-se que há um forte aumento da corrente ao passo que a temperatura aumenta em todos os

meses, muito provavelmente pelo efeito de ligar ar condicionados no prédio.

Analisando os gráficos boxplot, percebe-se que para as temperaturas mais quentes, acima de 25º, as hastes de mínimo e máximo são mais distantes entre si e os retângulos possuem intervalos de valores maiores do que nas faixas de temperaturas mais frias. Isso mostra que não há um padrão humano quanto ao consumo elétrico para as temperaturas mais quentes, pois 50% das medidas de corrente estão dentro de um intervalo maior de corrente.

4.3.2 O consumo elétrico ao longo dos meses

Percebendo a ausência de um padrão de consumo para as temperaturas mais quentes, fez-se uma análise para entender o padrão de consumo elétrico ao longo do dia e da semana. Para isso, os dados de temperatura e corrente foram coletados a cada minuto, e com eles, gerou-se gráficos de mapa de calor (*heatmap*) com a corrente e temperatura média para cada hora e dia da semana. O resultado pode ser visto nas Figuras 14 a 19, onde os números no eixo X de 0 a 6 representam os dias da semana iniciando na segunda-feira e no eixo Y são as horas do dia.

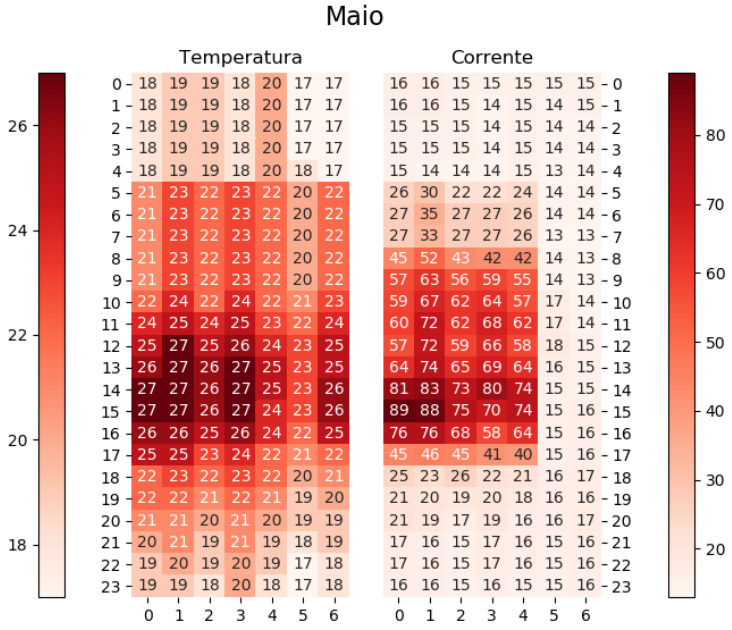

Figura 14: Heatmap de Temperatura e Corrente - Maio

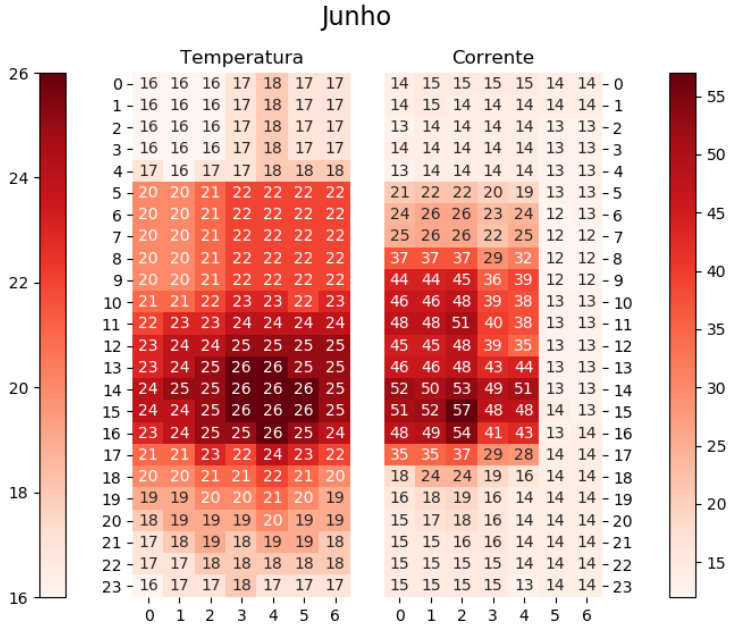

Figura 15: Heatmap de Temperatura e Corrente - Junho

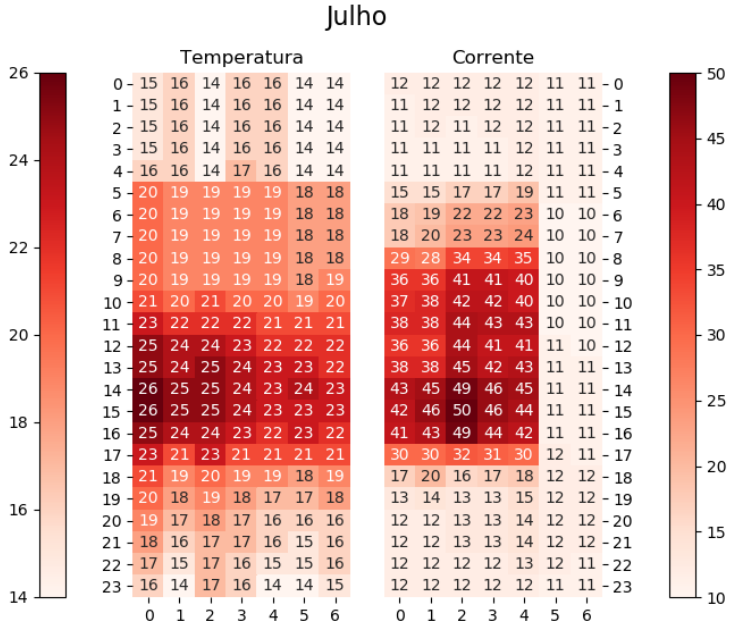

Figura 16: Heatmap de Temperatura e Corrente - Julho

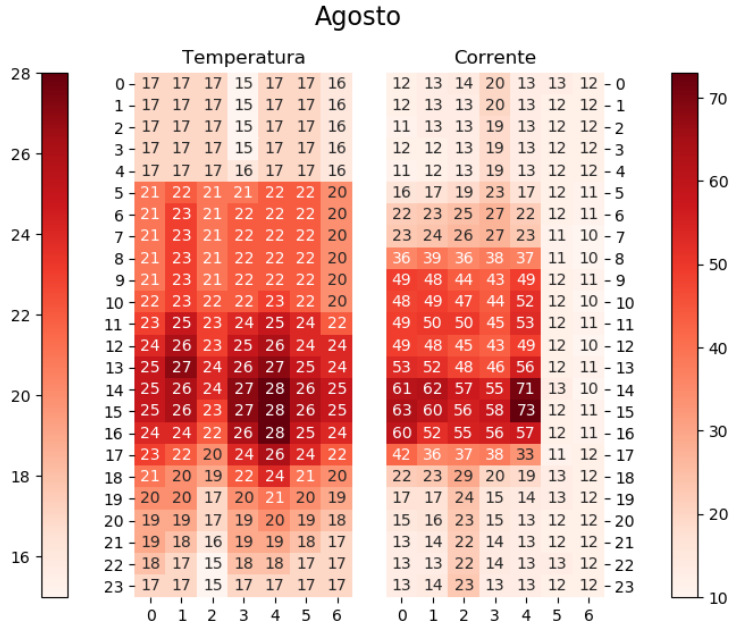

Figura 17: Heatmap de Temperatura e Corrente - Agosto

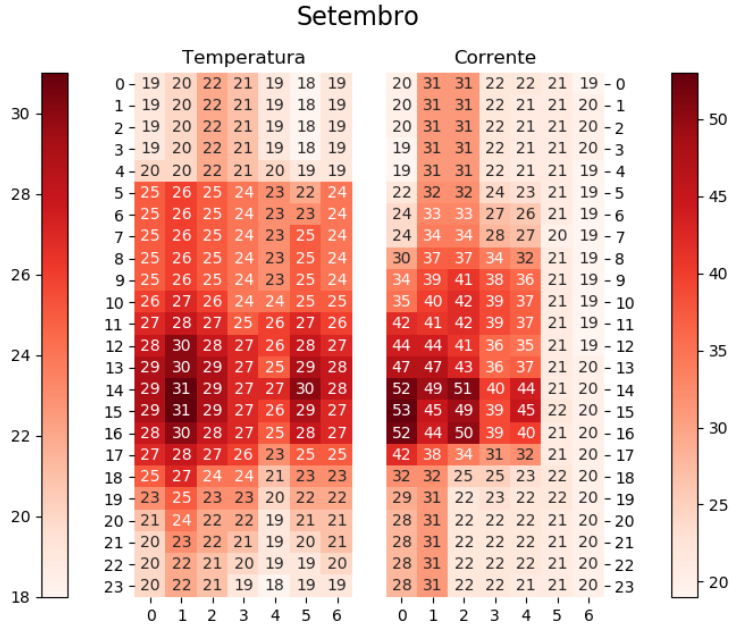

Figura 18: Heatmap de Temperatura e Corrente - Setembro

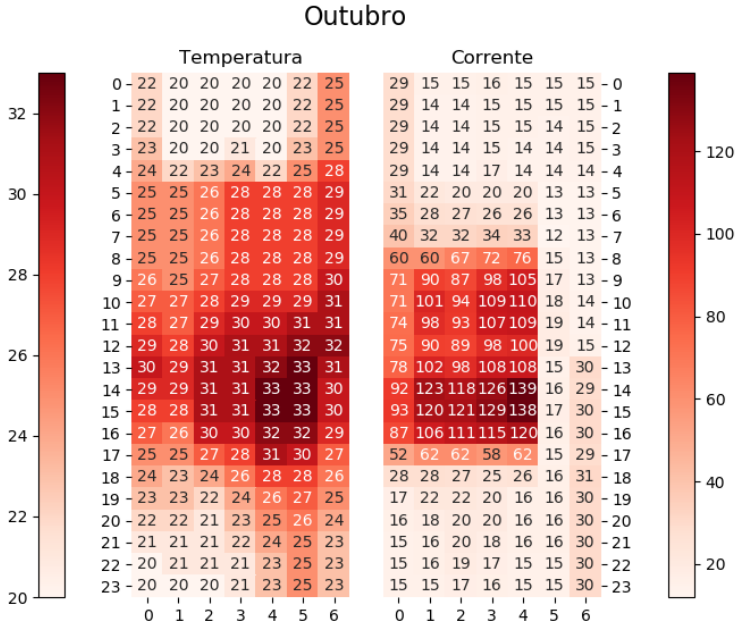

Figura 19: Heatmap de Temperatura e Corrente - Outubro

Com base nos gráficos de *heatmap*, além de enaltecer a correlação de aumento de corrente elétrica com o aumento da temperatura, a hora do dia também possui influência no consumo, dado que é entre as 14:00 e 16:00 que ocorre um maior consumo em todos os meses. O dia da semana também possui uma influência quando analisamos o consumo, principalmente os finais de semana quando o consumo é mínimo, visto que não há atividades no prédio nesses dias. Os demais dias comerciais se destacam diferentemente em cada um dos meses, assim sendo, não se pode assumir que nas segundas-feiras o consumo tende a ser maior que nas terças-feiras, por exemplo.

Outro fator temporal que podemos perceber como importante é que a cada mês a temperatura média muda dado a época do ano, pois os meses de junho a setembro tendem a ser mais frios. Outubro, por exemplo, por ser o mês mais quente dentre os meses analisados, teve as medidas mais altas de corrente média.

4.4 Análise de modelos de Aprendizado de Máquina

Das etapas de estudo dos dados e correlação entre eles, temos que a temperatura, hora do dia e dia da semana são os parâmetros selecionados para serem usados nos modelos de Aprendizado de Máquina nesta etapa do trabalho.

Para tratar esses efeitos de sazonalidade climática e tentar acompanhar o padrão de consumo energético das pessoas no prédio da Prefeitura, decidiu-se incluir o número da semana no ano como um parâmetro para prever o consumo elétrico.

Para que fosse possível testar a qualidade e desempenho dos modelos de Aprendizado

de Máquina, dividiu-se as 261.106 medidas, de maneira aleatória, em grupos de treino, de validação e de teste, cada um contendo, respectivamente, 70%, 20% e 10% desses dados. Os dados de treino serão utilizados para que o computador crie uma regressão ótima, baseada na lógica de cada modelo. Os dados de validação, por sua vez, são usados apenas para avaliar a qualidade das regressões criadas com dados que os modelos não utilizaram para treinar. Por fim, os dados de teste só podem ser utilizados uma única vez para avaliar a real qualidade da melhor regressão criada.

Para se testar a qualidade dos modelos, utilizou-se a técnica de Erro Médio Quadrático, que consiste em calcular a soma do quadrados da diferença entre os valores previstos e esperados ou reais conforme a equação abaixo.

$$erro = \sum (yprev - yreal)^2$$

Os modelos foram implementados utilizando a biblioteca Scikit-learn que é um software para a linguagem de programação Python e que contém os algoritmos utilizados neste projeto e diversos outros algoritmos disponíveis de maneira gratuita.

4.4.1 *Support Vector Regression*

O primeiro modelo utilizado foi o *Support Vector Regression* (SVR) [2] que consiste em encontrar vetores de suporte que dividem os dados em seções para agrupar os dados em grupos para posterior classificação ou regressão de novos dados. O parâmetro que foi variado nesse modelo foi o de penalidade, que consiste na tolerância de erros que os agrupamentos podem ter. Quanto maior a penalidade, maior o esforço de encontrar um vetor baseado nas características dos dados que de fato separem um grupo de dados de outro.

A velocidade de gerar uma projeção para os casos de treino para diversos valores de penalidade estava muito lenta, ou simplesmente não finalizava. Com isso, ajustou-se os dados para que os mesmos estivessem com o valor médio a cada hora ao invés de a cada minuto. Assim, o modelo conseguiu operar corretamente. Os resultados do modelo SVR para valores diversos de penalidade podem ser vistos na tabela 1. Uma boa prática para esse modelo é fazer testes de penalidades utilizando potências de dez.

	0.1	1	10	100	1000
Treino	402.53	595.86	108.42	9.52	0.17
Validação	400.45	600.04	111.72	12.82	4.15

Tabela 1: Erros de *Support Vector Regression* para vários valores de penalidade

Pode-se perceber pela tabela que a melhor regressão gerada teve um erro de 0.17 para os casos de treino e de 4.15 para os casos de validação. Entretanto, como queremos prever o consumo utilizando os diversos valores possíveis de temperatura e corrente a cada hora e não os seus valores médios, o modelo SVR não é o modelo ideal para o nosso projeto de predição de consumo elétrico, apesar do baixo valor de erro para os casos de treino e de validação.

4.4.2 *Multi-layer Perceptron - Adam*

O próximo modelo testado foi o *Multi-layer Perceptron* (MLP) [3] com a otimização Adam, um modelo de aprendizado supervisionado baseado em camadas contendo "neurônios artificiais": componentes fundamentais de processamento de informação usando transformações algébricas no formato de funções de ativação. Este modelo se assemelha ao modo como uma rede neural biológica funciona, especialmente quando utilizadas funções de ativação não-lineares. A função de ativação nesse modelo funciona como um filtro, selecionando quais informações serão passadas para a próxima camada. Um *Multi-layer Perceptron* possui pelo menos 3 camadas: a camada de entrada, que contém os dados de entrada; a camada escondida, que pode ter mais de uma camada dentro da camada escondida, cada qual contendo números determinados de perceptrons; e a camada de saída, que contém o resultado das operações das outras camadas que é utilizado para se comparar com os dados esperados. Ele é a evolução natural da primeira rede neural construída, o *Perceptron*, que possuía apenas uma camada.

Existem diversas funções de ativação que podem ser utilizadas em cada camada de uma rede neural. Para cada camada, cada um de seus neurônios recebe o resultado de uma operação da camada anterior e utiliza uma função de ativação sobre esse resultado, modificando o valor que será passado para a próxima camada. As funções de ativação mais famosas são:

- Identidade: retorna o valor sem operação nenhuma, baseado na função identidade f(x) = x;
- Logística: retorna o valor de 0 ou 1 usando a função f(x) = 1 / (1 + exp(-x));
- Tangente Hiperbólica: retorna o resultado da operação f(x) = tanh(x);
- ReLU: retorna 0 se $x \leq 0$ ou f(x) = x para $x > 0$.

Um procedimento que é utilizado dando a base para otimizações de redes neurais é o de *Stochastic Gratient Descent* que consiste em atualizar os pesos contidos em cada ligação entre neurônios de duas camadas adjascentes utilizando o gradiente daquele peso com o objetivo de diminuir o erro que aquele peso influencia no resultado final. Dependendo do número de camadas e de neurônios por camadas, o modelo MLP utilizando *Stochastic Gratient Descent* pode ser bastante ineficiente e ineficaz. Uma maneira de contornar isso é utilizando técnicas de otimização como a Adam (*adaptative moment estimation*) [4]. Essa otimização consiste em atualizar o *learning rate* (i.e. velocidade com que cada peso é atualizado a cada etapa de treino dos modelos) de acordo com o mais recente gradiente encontrado para um determinado peso, fazendo com que o modelo consiga chegar em um resultado ótimo mais rápido.

Para o experimento do projeto, foi utilizado MLP com otimização Adam e ativação com ReLU para diversos valores de camadas escondidas. Os resultados podem ser vistos na Tabela 2.

	10	20	30	40	50	60	70	80	90	100
Treino	373.5	260.9	249.0	263.0	218.0	206.4	203.0	220.1	209.8	193.7
Validação	372.8	261.5	249.9	262.2	218.5	206.8	202.3	220.5	211.6	195.8

Tabela 2: Erros de MLP-Adam para vários valores de camada escondida

Percebe-se pela Tabela 2 que o erro está bastante alto apesar da constante diminuição do mesmo até 80 camadas escondidas. Isso pode ter se dado pela característica dos próprios dados, onde tem-se padrão de consumos bem diferentes olhando horário comercial e finais de semana, por exemplo. Além disso, o modelo realiza computações utilizando o valor bruto dos dados. O dia da semana, por exemplo, tem consumo mínimo para valores 5 e 6, sábado e domingo, respectivamente. Se a corrente crescesse ao longo dos dias da semana e ao longo das horas do dia, o modelo MLP poderia ter tido resultados melhores.

4.4.3 *Decision Tree*

Dada a característica dos dados que possui padrões de consumo diferentes por períodos diferentes ao longo do dia e da semana, decidiu-se realizar o experimento usando o modelo de *Decision Tree* [5]. Esse é um dos modelos de Aprendizado de Máquina mais famosos na literatura principalmente por ser um dos modelos mais simples de aprender, comparando com outros modelos, bem como pela sua possibilidade de entender a influência de cada característica dos seus dados de entrada no resultado da predição realizada.

Esse modelo consiste em segmentar seus dados em diversas subdivisões de maneira condicional e de modo a representar características típicas de cada amostra de dado na sua subdivisão. De maneira ilustrativa, nas Figuras 20 e 21, temos a representação, respectivamente, do que seria uma *Decision Tree* com profundidade de até 3 e 4 subníveis para o nosso caso de estudo.

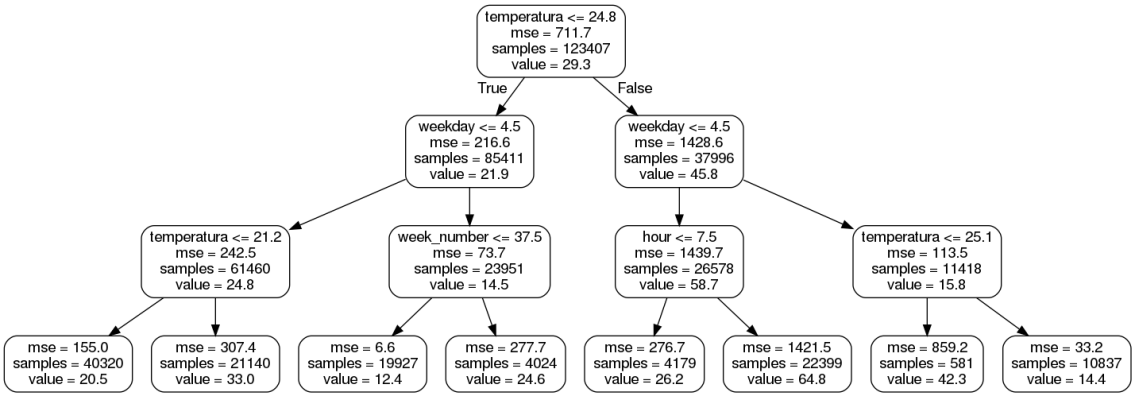

Figura 20: *Decision Tree* com profundidade de 3 níveis

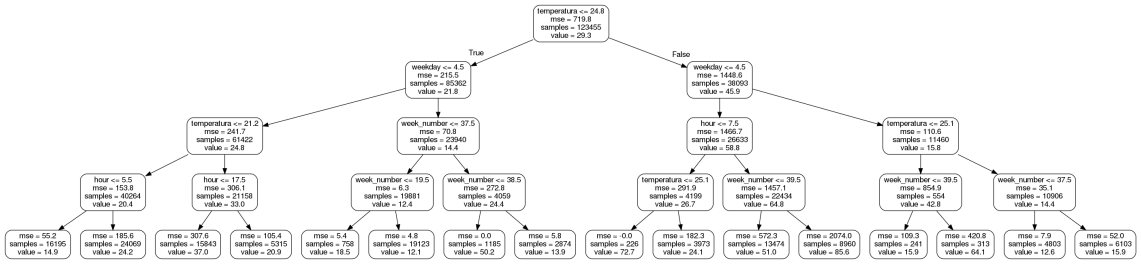

Figura 21: *Decision Tree* com profundidade de 4 níveis

Pela Figura 20, a primeira divisão ocorre na condição de a temperatura ser menor ou igual a 24.8 e em seguida verifica se se o dia da semana é menor que 4.5. Após isso, a árvore verifica um parâmetro diferente dependendo da resposta do caso anterior. Ao final, nas folhas da árvore, temos um número que corresponde ao resultado esperado dada uma informação de entrada.

Para testar esse modelo, utilizou-se a classe DecisionTreeRegressor e testou-se diferentes valores para o parâmetro max_depth, que é a profundidade máxima que a árvore pode ter no processo de encontrar uma distribuição ótima dos dados. Os resultados podem ser vistos na Tabela 3.

	5	10	15	20	25	30	35	40
Treino	187.21	65.69	21.50	15.35	15.02	15.01	15.01	15.01
Validação	187.63	65.79	22.87	17.41	17.31	17.31	17.31	17.31

Tabela 3: Erros de Decision Trees para vários valores de profundidade máxima

Pela Tabela 3, percebe-se que a partir de profundidade máxima de 25, os modelos começam a ter seus erros estabilizados tanto para os dados de treino quanto para os de validação. Vale notar que o número de folhas no final da árvore corresponde ao número dois elevado ao número máximo de folhas. Então, para o melhor modelo utilizando Decision Tree com profundidade de 25 teríamos 33.554.432 valores diferentes de saída, entretanto, como temos 182.774 dados para treino, provavelmente a árvore criou uma folha para cada dado.

Assim, escolheu-se o modelo com profundidade 15 como o ideal, pois a árvore teria no máximo 32.768 folhas, sendo em média 5 medidas de dados por folha além de ter um erro em torno de 6 unidades maior que o melhor modelo com profundidade 25. Para esse caso, não foi possível gerar uma representação visual da árvores para entender no detalhe todas as possibilidades da árvore. Porém, uma forma de mitigar isso é analisando a importância que cada variável teve durante a construção da árvore. A representação dessa importância para a árvore de profundidade máxima de 15 pode ser vista na Tabela 4.

Variável	Importância
Hora	28%
Temperatura	26%
Dia da semana	24%
Número da semana	22%

Tabela 4: Importância de cada variável para árvore de profundidade máxima de 15

Percebe-se pela tabela que a contribuição de cada variável é bem equilibrada, tendo todas mais de 20% de importância. Destaca-se a variável de hora do dia com 28% de importância na predição da corrente prevista, uma vez que ela gera mudanças mais bruscas na corrente, e em seguida a váriável de temperatura com 26%, ao se analisar as figuras de heatmap anteriores novamente.

4.4.4 *Random Forest*

O modelo de Aprendizado de Máquina *Random Forest* [6] é um modelo na categoria de *ensemble learning*, que são modelos baseados na agregação e/ou comparação de diversos outros. No caso, o modelo *Random Forest* consiste em calcular a média aritmética de um número escolhido de diferentes *Decision Trees* geradas diferentemente umas das outras.

Utilizando a classe RandomForestRegressor da biblioteca scikit-learn com geração de 10 árvores para o cálculo da média, testou-se o resultado com árvores de diversos tamanhos máximos de profundidade. Os resultados podem ser vistos na Tabela 5.

	5	10	15	20	25	30	35	40
Treino	176.33	60.28	19.48	15.27	15.13	15.13	15.13	15.13
Validação	176.38	60.77	20.85	17.39	17.46	17.38	17.34	17.39

Tabela 5: Erros de Random Forests para vários valores de profundidade máxima

Pela tabela, percebe-se um padrão semelhante ao que ocorreu utilizando apenas uma *Decision Tree*, tendo o erro estabilizado para treino a partir de profundidade 25, mas para validação teve diminuição até a melhor profundidade de 35, após isso houve um aumento. Porém, para uma profundidade máxima de 15, este modelo conseguiu erro para os dados de validação de 20.85, melhor que o erro de 22.87 no modelo anterior.

4.4.5 *Adaboosting e otimização de hiperparâmetros*

Tem-se até então que o modelo *Random Forest* com profundidade máxima de 15 foi o melhor que conseguiu representar bem os conjuntos de dados com as suas características com erro quadrático de 20.85. Entretanto, esse mesmo modelo e o modelo *Decision Tree* conseguiram valores como 17.34 e 17.31 para profundidade máximas maiores, uma diferença de erro de pelo menos 3.5.

Então, decidiu-se utilizar técnicas de otimizações de modelos de Aprendizado de Máquina

tanto no modelo *Decision Tree* quanto *Random Forest*, ambos com profundiade máxima de 15, com o foco em diminuir essa diferença encontrada comparando com os mesmos modelos mas com profundidades de árvore maiores. As duas ténicas utilizadas foram a de Adaboosting e a de Otimização de Hiperparâmetros.

O AdaBoosting [7] tem como objetivo principal criar um modelo mais forte baseado em modelos mais fracos, como *Decision Trees*. Esse novo modelo é construído utilizando pesos nas instâncias dos dados de treino, que são inicializados igualmente. O modelo é treinado utilizando um método mais fraco, as instâncias em que o modelo possui resultados mais coerentes têm seus pesos diminuídos e as instâncias de resultados incoerentes possuem seus pesos aumentados. Um próximo modelo é construído sendo ponderado com o balanço dos pesos atribuídos no passo anterior. Ou seja, exemplos com maiores erros terão maior peso para ao treino no próximo modelo, objetivando elaborar um modelo que corrija os resultados incoerentes. Esses modelos são criados sequencialmente visando reduzir o erro total. E a predição é feita utilizando a média do resultado elaborada por todos os modelos.

A Otimização de Hiperparâmetros (*Hyperparameter Tuning*) [9], por sua vez, consiste em cruzar e testar diferentes valores de hiperparâmetros procurando a melhor combinação para o modelo a ser treinado. Hiperparâmetros são parâmetros utilizados nos modelos de Aprendizado de Máquina de modo a modificar a regressão que será gerada. No caso das *Decision Trees* e das *Random Forests* um exemplo de hiperparâmetro é a profundidade máxima (max_depth), já para o modelo SVR, um exemplo seria a penalidade (penalty). Assim, existem diversos hiperparâmetros que podem ser testados de acordo com o modelo utilizado. Para avaliar a qualidade de cada um dos testes gerados, utilizou-se o erro quadrático médio, no processo de encontrar os melhores hiperparâmetros.

Assim, utilizando a classe AdaBoostRegressor da biblioteca do Scikit-Learn, gerou-se a regressão otimizada dos modelos *Decision Tree* e *Random Forest*, ambos com profundidade máxima de 15. Com a classe GridSearchCV da mesma biblioteca, relizou-se o teste para os hiperparâmetros n_estimators e learning_rate do modelo AdaBoost que são, respectivamente, o número de modelos máximos a serem testados em busca do modelo ótimo e a velocidade que o modelo diminui a contribuição, ou pesos, dos modelos novos gerados. O resultado dos melhores hiperparâmetros encontrados, bem como o valor do erro quadrático para os dados de treino e de validação para o AdaBoost Decision Tree e para o AdaBoost Random Forest podem ser vistos na Tabela 6.

Modelo	n_estimators	learning_rate	Treino	Validação
AdaBoost Decision Tree	50	0.1	18.26	19.73
AdaBoost Random Forest	50	0.1	18.25	19.70

Tabela 6: Resultado dos modelos AdaBoost Decision Tree e AdaBoost Random Forest com Otimização de Hiperparâmetros

Percebe-se pela Tabela 6 que ambos modelos otimizados tiveram os mesmos valores para hiperparâmetros e resultados menores de erros quadráticos tanto para os dados de treino quanto os de validação comparando com o modelo *Random Forest* sem otimização. Dentre esses dois modelos otimizados, o AdaBoost Random Forest saiu-se melhor, reduzindo

diferença de erro de 3.5 para 2.4 comparando os resultados de dados de validação usando *Decision Tree* com profundidade máxima de 25.

4.5 Escolha do melhor modelo para predição de consumo

Após a análise de desempenho de quatro modelos de Aprendizado de Máquina e alguns desses modelos com técnicas de otimização, selecionou-se as regressões com melhores resultados de cada um deles. Um compilado desses desempenhos pode ser visto na Tabela 7.

Modelo	Treino	Validação
SVR	0.17	4.15
MLP-Adam	193.7	195.8
Decision Tree (max_depth = 25)	15.02	17.31
Random Forest (max_depth = 35)	15.13	17.34
Decision Tree (max_depth = 15)	21.50	22.87
Random Forest (max_depth = 15)	19.48	20.85
AdaBoost Decision Tree com Otim. de Hiperparâmetros.	18.26	19.73
AdaBoost Random Forest com Otim. de Hiperparâmetros.	18.25	19.70

Tabela 7: Erros dos modelos de Aprendizado de Máquina testados

Pela Tabela 7, o modelo SVR foi o que teve o melhor desempenho para os dados de treino e validação, porém foi excluído da decisão porque foi-se utilizado os valores de corrente e temperatura média a cada hora para ser treinado ao invés de todo conjunto de dados.

Assim, temos que o modelo Adaboost Random Forest com Otimização de Hiperparâmetros foi o modelo que teve o menor erro quadrático para os dados de validação, sendo, portanto, o modelo escolhido para fazer predição com os dados de teste. Com isso, utilizou-se a regressão desse modelo para analisar o erro utilizando os dados de teste, obtendo-se um resultado de erro quadrático de 20.54, 0.84 maior que o erro para os dados de validação para o mesmo modelo. A visualização do comparativo da predição de corrente comparada com a corrente nos casos de teste mês a mês pode ser vista na Figura 22.

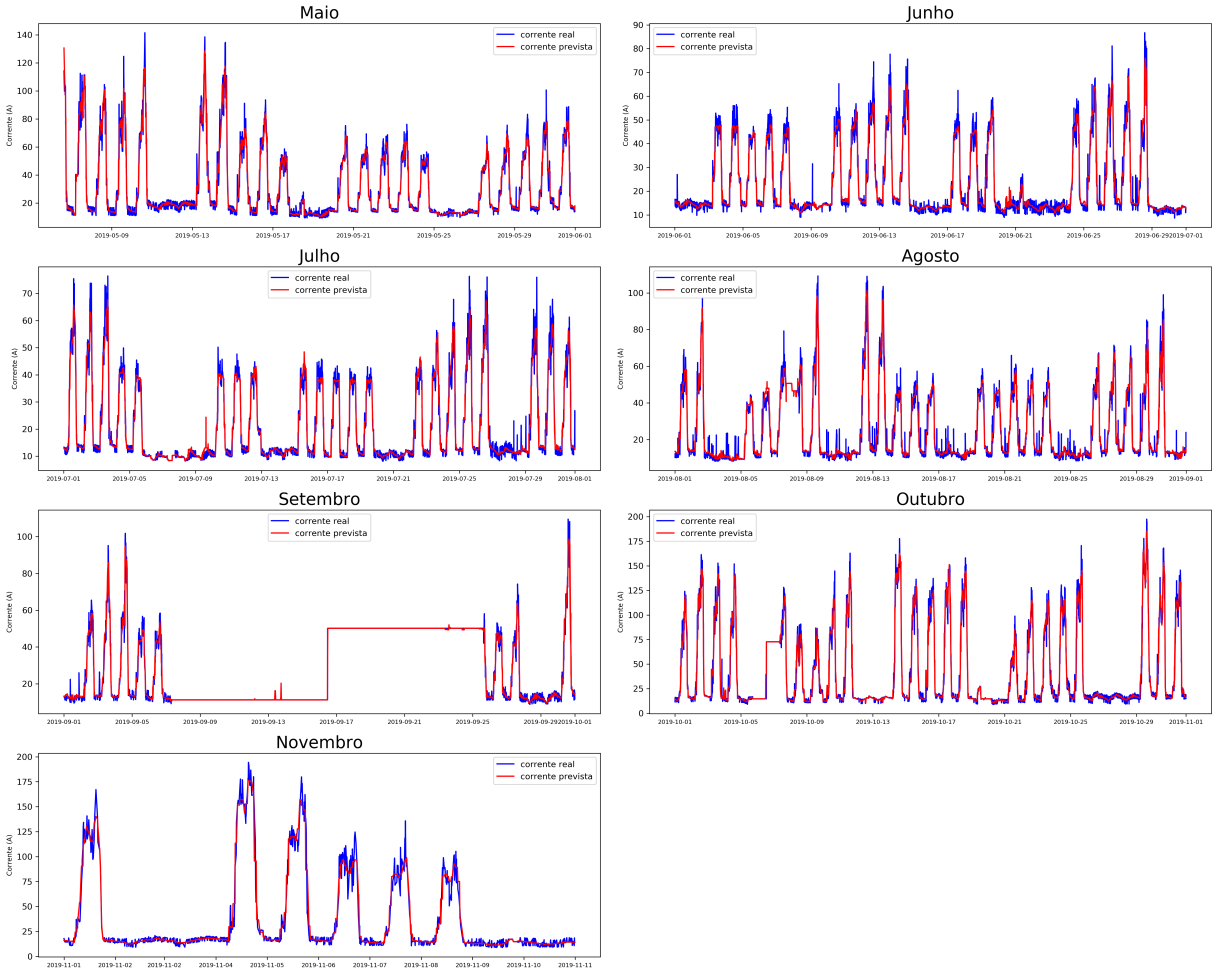

Figura 22: Comparativo da corrente real e prevista nos dados de teste por mês

4.6 Predição do consumo elétrico no mês de novembro

A última etapa do projeto consistiu em utilizar o melhor modelo encontrado para tentar predizer o consumo elétrico do mês de novembro. Para isso, conseguiu-se através da equipe do CEPAGRI uma planilha com as informações climáticas a cada hora em Campinas desde 1997 até 2016. Com isso, calculou-se e utilizou-se a temperatura média entre 2013 e 2016 para predizer a corrente que cada um desses dias teriam de hora em hora com a regressão gerada usando *Decision Tree* a partir do dia 16/11/2019, que foi o último dia de testes de modelos.

O resultado foi salvo em um arquivo CSV para que se possa comparar o resultado da regressão com as medidas que de fato ocorreram em novembro quando o mesmo finalizasse. Então com isso, gerou-se o gráfico da Figura 23 com o comparativo da corrente real medida pelo dispositivo na Prefeitura com a corrente prevista pelo modelo escolhido.

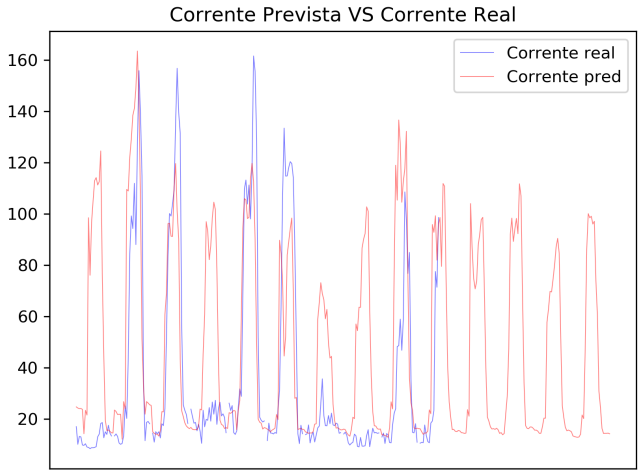

Figura 23: Comparativo corrente real e prevista em novembro

Percebe-se pela Figura 23 que o modelo conseguiu simular bem alguns padrões de subida e descida, mas não em todos os casos. Para entender onde o erro está sendo maior, gerou-se um gráfico de *heatmap* com a soma dos erros para cada dia e hora de novembro como mostrado na Figura 24.

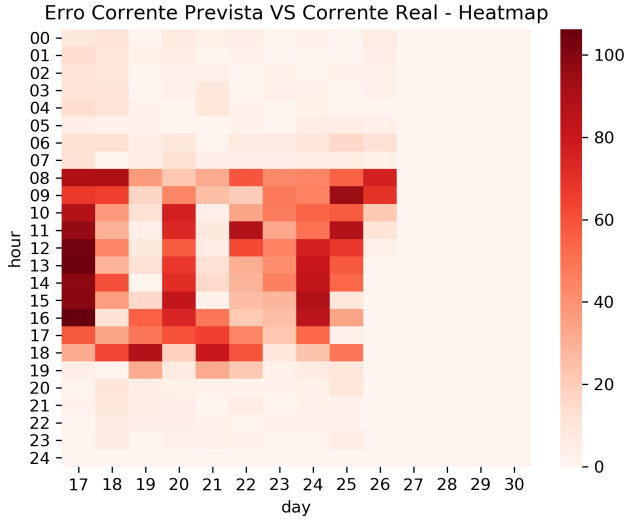

Figura 24: Erro acumulado para cada hora e dia de novembro

Também gerou-se um *heatmap* com a temperatura média para cada mês juliano de novembro que apareciam na planilha do CEPAGRI representada pela Figura 25.

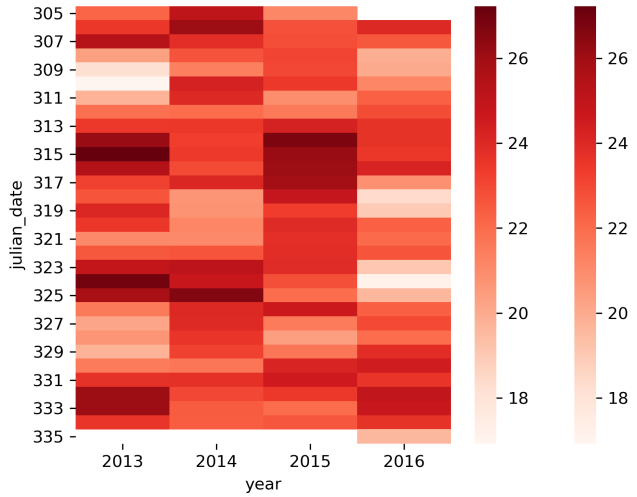

Figura 25: Temperatura para cada dia Juliano em novembro

Os erros encontram-se altos, principalmente no dia 17, que corresponde ao dia juliano 321. Isso ocorreu porque não há um padrão muito evidente de temperatura a cada ano para cada dia juliano, sendo a variável de temperatura a segunda mais importante encontrada no modelo. Além disso, o modelo utiliza o número da semana no ano como um parâmetro para acompanhar alguma mudança no padrão de consumo ao longo de uma semana, e não há informações sobre futuras semanas no mês de novembro para que o modelo treine, aumentando assim os erros.

5 Discussões dos resultados

Os modelos de Aprendizado de Máquina testados nesse projeto tiveram um desempenho bem próximos ao se analisar os erros quadráticos para os dados selecionados para conjunto de treino e para de validação. Pela própria característica dos dados, os modelos baseados em árvore de decisão se destacaram, pois o consumo elétrico varia bastante ao longo das horas do dia e nos dias da semana, tendo melhor desempenho com o modelo de *Decision Tree* com erros de 15.02 e de 17.31 para os dados de treino e validação, respectivamente. Entretanto, a árvore gerada por esse modelo pode estar gerando uma folha para cada amostra de dado, não sendo o ideal num cenário de previsão futura, onde novos dados podem não se adequarem à estrutura da árvore gerada, causando mais erros. Assim, foi escolhido o modelo AdaBoost Random Forest com profundidade máxima de 15 e com Otimização de Hiperparâmetros como o ideal, tendo resultados de erros quadráticos 18.25, 19.70 e 20.54 para treino, validação e teste, respectivamente.

Durante os experimentos, tentamos prever o consumo de energia utilizando as predições do melhor modelo da metade do mês de novembro até o fim do mesmo utilizando a temperatura média no mesmo período de 2013 a 2016 através de uma planilha fornecida pela

equipe do CEPAGRI, porém o erro quadrático comparando a corrente elétrica prevista e a real foi bem alto, chegando a mais de 1.000.

Apesar de o modelo escolhido ter conseguido representar bem o consumo de dados passados, como pode ser visto na Figura 22, ele não tem bom desempenho com dados futuros porque utiliza o número da semana do ano como um critério para as regressões. Além disso, não há uma correlação direta entre temperatura e os outros fatores climáticos com a corrente elétrica no prédio. Há nesse caso um forte fator humano, que tentou ser representado incluindo a hora do dia e o dia da semana nos experimentos, mas uma simples adição ou remoção de um ar condicionado no prédio por um determinado período para afetar o consumo elétrico, prejudicando as projeções.

6 Conclusões

Para um primeiro projeto de estudo dos dados climáticos e energéticos que eram até então apenas coletados mas não analisados para geração de novas discussões de consumo energético no prédio da Prefeitura, esse projeto conseguiu entender bem o padrão de consumo no prédio, mas ainda precisa evoluir na questão de previsão de consumo futuro. À medida que novos dados são coletados a todo momento, é possível que nos próximos anos, o volume de dados seja tão grande que consiga superar os efeitos humanos no consumo adicionando novos parâmetros, como mês e ano. Outra opção seria de conseguir identificar quantos aparelhos de ar condicionado estão ligados no momento da medição através de outros sinais.

Além disso, visto que a Unicamp paga um valor mensal que dá uma certa cota de consumo total somando todos os institutos e espaços do campus, só prever o consumo de energia do prédio da Prefeitura não é suficiente para sugerir uma negociação nesse valor. Outros prédios possuem equipamentos eletrônicos em laboratórios para pesquisas que podem consumir mais energia do que os aparelhos de ar condicionado. Assim sendo, caso novos medidores sejam instalados em mais prédios, novos modelos de Aprendizado de Máquina podem ser testados para predizer o consumo de energia para cada entidade, e assim propor uma negociação no valor pago em energia elétrica, gerando mais economia para a universidade.

As áreas de engenharia civil e arquitetura também podem ser beneficiadas no cenário de ter mais medidores de corrente elétrica em mais prédios da universidade. Através da visualização do consumo elétrico e dos fatores climáticos ao longo do tempo, é possível realizar o estudo e validação da influência da arquitetura de prédios, do uso de determinados janelas com tipos de vidros diferentes ou do uso de determinados modelos de ar condicionados para harmonização climática.

Por fim, pesquisas nesses temas podem gerar novas soluções e novas discussões sobre o uso consciente de recursos energéticos da universidade, direcionando a Unicamp para um caminho cada vez mais sustentável, reafirmando a importância de projetos como Smart Campus e Campus Sustentável e de mais estudos no conceito de Internet das Coisas no processo de geração de inovação para a universidade e sociedade como um todo.

7 Agradecimentos

Agradeço a minha orientadora, professora Juliana Freitag, por aceitar me orientar em um tema do meu interesse profissional e por todo suporte, flexibilidade e direcionamento de novas análises ao longo do semestre, o coorientador Luis Gonzalez pela ajuda técnica e teórica de modelos de Aprendizado de Máquina e pela participação nas discussões, o Bruno Kabke do CEPAGRI por disponibilizar dados metereológicos passados de Campinas e o Rafael Pereira de Sousa da Prefeitura do Campus por apresentar as necessidades principais para análise e validação de certos padrões de consumos no prédio. Também agradeço a professora Sandra Avila, por ministrar a disciplina MC886: Aprendizado de Máquina no mesmo semestre deste projeto, auxiliando tanto no entendimento dos modelos apresentados quanto discutindo resultados encontrados, e o grupo do projeto final da mesma disciplina, em especial o aluno Eduardo Yuji pelo excelente trabalho de otimização de modelos, contribuindo para o resultado final deste projeto.

Referências

[1] A.H.M. Jakaria, M.M. Hossain, M.A. Rahman, *Smart Weather Forecasting Using Machine Learning: A Case Study in Tennessee* (2018)

[2] C. Cortes, V. Vapnik, *Support-Vector Networks* (1995).

[3] F. Rosenblatt, *The Perceptron: A perceiving and recognizing automaton* (1957)

[4] D. Kingma, *Adam: A Method for Stochastic Optimization* (2015)

[5] J.R Quilan, *Induction of Decision Trees* (1986)

[6] L. Breiman, *Random Forests* (2001)

[7] R.E. Schapire, *Explaining Adaboost* (2013) (March 1996).

[8] G. Leshem, *Improvement of Adaboost Algorithm by using Random Forests as Weak Learner* (2004)

[9] M. Claesen, B. D. Moor, *Hyperparameter Search in Machine Learning* (2015)