Conceção e otimização de modelos de *Machine Learning*

Diogo Rebelo[pg50327], Daniel Xavier[pg50310], Henrique Alvelos[pg50414], and João Cerquido[pg50469]

[1] Minho's University, Department of Informatics, 4710-057 Braga, Portugal
[2] email:{pg50327,pg50310,pg50414,pg50469}@alunos.uminho.pt

Abstract. Abordam-se dois problemas específicos, um que se relaciona com a previsão de incidentes rodoviários, numa cidade portuguesa, em 2021, e outro que se prende com previsão do número de ocupantes de uma sala. Ambos os problemas foram alvo da aplicação de técnicas de *ML* conhecidas. Para o primeiro caso, o objetivo foi desenvolver modelos capazes de minimizar o número de acidentes e vítimas em estradas. Para isso, utilizamos um conjunto de dados com informações históricas de incidentes rodoviários, incluindo registos temporais, condições climáticas e as características na via. Para o segundo caso, pretendia-se a conceção e otimização de modelos capazes de minimizar o consumo energético, baseando no número de ocupantes da respetiva sala. Para isso, analisaram-se fatores do interior da respetiva sala, desde condições ambientais a movimento. Aplicamos maioritariamente técnicas de *ML*, baseadas em árvores de decisão, e otimizamos os modelos através da validação cruzada e da seleção de hiperparâmetros. Os resultados mostraram que os modelos desenvolvidos tiveram uma *accuracy* de até 94% e um *R2 Score* de até 93.7%.

Keywords: *ML* · Incidentes Rodoviários · Ocupação Interior · Árvores de Decisão · Conceção de Modelos · Otimização de Modelos · Metodologias (CRISP-DM) · AutoML

1 Introdução

No âmbito da unidade curricular de Dados e Aprendizagem Automática (DAA), integrada no plano de estudos do 1º ano do Mestrado em Engenharia Informática (MEI), foi-nos proposta a conceção e otimização de modelos de ML, através da análise de dois *datasets*, um fornecido pela equipa docente e outro selecionado pelo grupo de trabalho. Este relatório descreve todo o enquadramento teórico que serve de base para a aplicação mais prática aquando do tratamento dos dados e elaboração dos modelos preditivos.

Em relação à estrutura e organização do projeto, começamos pela extração do dados, a qual se baseou apenas na seleção de um *dataset*, já que o outro foi disponibilizado pelos docentes. Prosseguimos para a respetiva compreensão e descrição dos dados, através de uma análise de qualidade dos dados previamente

extraídos. Tomamos as ações necessárias de transformação e limpeza e, finalmente, iniciamos a criação de vários modelos, confrontando-os com a execução de testes.

De um modo geral, em termos de trabalho, destacaram-se duas fases: uma primeira de adequação dos dados (*datasets*) ao trabalho a desenvolver, percebendo de que forma modificar certos aspetos poderia influenciar o(s) modelo(s) a desenvolver, e, uma segunda fase, de *tuning* dos vários modelos, de modo a otimizar a previsão requerida. Sendo assim, o presente relatório apresenta uma análise detalhada e justificada de todo o procedimento efetuado em cada um dos *datasets*. É, assim, possível observar quais os métodos de visualização e exploração de dados, assim como os modelos de aprendizagem aplicados e os respetivos resultados. Além disso, o presente documento faz também alusão aos testes submetidos na plataforma *Kaggle*, o "anfitrião" de uma competição desenvolvida pela equipa docente, na tentativa de auxiliar o processo de otimização.

Como objetivos principais, destaca-se a obtenção e consolidação de conhecimentos alusivos à realização de *pipelines* de dados, conceção e otimização de modelos de ML, compreensão das principais técnicas associadas à exploração, transformação e modelação de dados. É de extrema importância evidenciar ao longo do projeto a metodologia seguida e, por isso, o presente documento segue cada fase da metodologia selecionada, CRISP-DM. Este relatório é um instrumento de estudo e avaliação do grupo 39[7.1].

2 *Datasets* da Competição e de Grupo

2.1 *Business Understanding*

Background O conjunto de dados, fornecido pela equipa docente, contém informações alusivas principalmente às condições climatéricas na estradas de uma cidade portuguesa, em 2021, permitindo prever o nível de incidentes rodoviários, ou seja, pretende-se prever a classe de incidentes, a qual pode ser *None, Low, Medium, High, Very High*, tratando-se, por isso, de um problema de classificação *multiclass*. Esta previsão permite intuitivamente a identificação de um padrão no tráfego rodoviário, ao longo do tempo, contribuindo para a minimização da ocorrência deste tipo de incidentes.

Já em relação ao *dataset* de grupo, este contém informações alusivas às condições ambientais no interior de uma sala, desde a presença ou não de movimento, até à temperatura ou mesmo luminosidade, medidas por sensores. Neste caso, trata-se de um problema de regressão, onde se pretende prever o número de ocupantes de uma sala (N), desde 0 até N. Também neste caso se identificam padrões na utilização da sala, de modo a minimizar o consumo energético localizado.

Recursos O trabalho socorreu-se de recursos a nível de *Staff*, nomeadamente, o próprio grupo de trabalho e a equipa docente; a nível de *Data*, com as extrações pontuais dos dados (nos *datasets*; e a nível computacional, inerente às

máquinas utilizadas no desenvolvimento do trabalho e todas as tecnologias envolvidas (*Python, Anaconda, Vscode, Jupyter Notebook*

Requirements, assumptions and constraints Como requisitos principais destaca-se a existência de uma taxa de acerto elevada, de modelos corretos a nível técnico e científico, e a qualidade dos resultados obtidos. Para isso, assume-se a veracidade mínima dos dados e a permissão para o seu uso. Como restrições, destaca-se as restrições de prazo estabelecidas (entrega do projeto até dia 12 de janeiro de 2023), as restrições de dados a utilizar (*dataset* da equipa docente) e as restrições que limitam o número de elementos do grupo de trabalho, não existindo restrições de custo associadas.

3 *Dataset* da Competição

3.1 *Data Understanding*

Describe Data Nesta fase, realizaram-se todas as tarefas necessárias para que melhor se compreendessem os atributos que compõem as colunas do *dataset*. Cada elemento do grupo começou por visualizar as colunas e as linhas do dataset, ganhando um conhecimento geral do mesmo. Para isso, recorreu-se a funções como `info()`, `describe()`, `value_counts`, `head()`, que, para além de fornecerem uma ideia sintética da estrutura do *dataset*, contribuem para a compreensão das suas características estatísticas, como máximos, mínimos, médias, modas, percentis, e tipos de dados associados (incluindo *missing values*). Para descrever as nossas *features* e o *target*, observe-se a seguinte tabela. Todas as *features* dizem respeito ao *record_date* da respetiva linha e na cidade de Guimarães.

Atributo	**Descrição**	***DType***	**Tipo**	**Exemplo**
city_name	Cidade onde ocorreu a extração dos dados em causa.	*object*	Categórico Nominal	"Guimaras"
magnitude_of_delay	Magnitude do atraso, provocado pelos incidentes.	*object*	Categórico Ordinal	"UNDEFINED", "MODERATE", etc.
delay_in_seconds	Atraso, em segundos, provocado pelos incidentes.	*int64*	Numérico Discreto	"10", "162", etc.
affected_roads	Estradas afectadas pelos incidentes.	*object*	Categórico	"N101,N101"
record_date	O timestamp associado ao registo.	*object*	Categórico	"2021-03-15 23:00"
luminosity	O nível de luminosidade.	*object*	Categórico Ordinal	"DARK", "LIGHT", etc.
avg_temperature	Valor médio da temperatura.	*float64*	Numérico Discreto	"13.0", "7.0", etc.
avg_atm_pressure	Valor médio da pressão atmosférica.	*float64*	Numérico Discreto	"1024.0", "999.0", etc.
avg_humidity	Valor médio de humidade.	*float64*	Numérico Discreto	"8.0", "92.2", etc.
avg_wind_speed	Valor médio da velocidade do vento.	*float64*	Numérico Discreto	"10.0", "2.0", etc.
avg_precipitation	Valor médio de precipitação.	*float64*	Numérico Discreto	"0.0"
avg_rain	Avaliação qualitativa do nível de precipitação.	*object*	Categórico Ordinal	"Sem chuva", "chuva forte", etc.
incidents (target)	Nível de incidentes rodoviários.	*object*	Categórico Ordinal	"Low", "High", etc.

Table 1: Features & Target.

Explore Data Nesta fase, aproveitamos para explorar os dados visualizados, ou seja, através de vários tipos de gráficos, como histogramas, gráficos de densidade e de probabilidade, procuramos estabelecer relações entre as *features* e o *target* (correlações). Verificamos para cada coluna do *dataset*, a sua distribuição (normal ou não), e a sua ocorrência tendo em conta cada nível de incidentes. Então, destas duas sub-fases anteriores, observamos , de um modo geral, o seguinte:

1. Existiam colunas com o mesmo valor em todos os registos (*avg_precipitation*);
2. Apenas uma coluna apresentava valores em falta (*affected_roads*);
3. Algumas colunas apresentam dados razoavelmente distribuídos normalmente, outras têm *skewness* (*delay_in_seconds, avg_atm_pressure, avg_humidity*);
4. Detetou-se a presença de *outliers*.

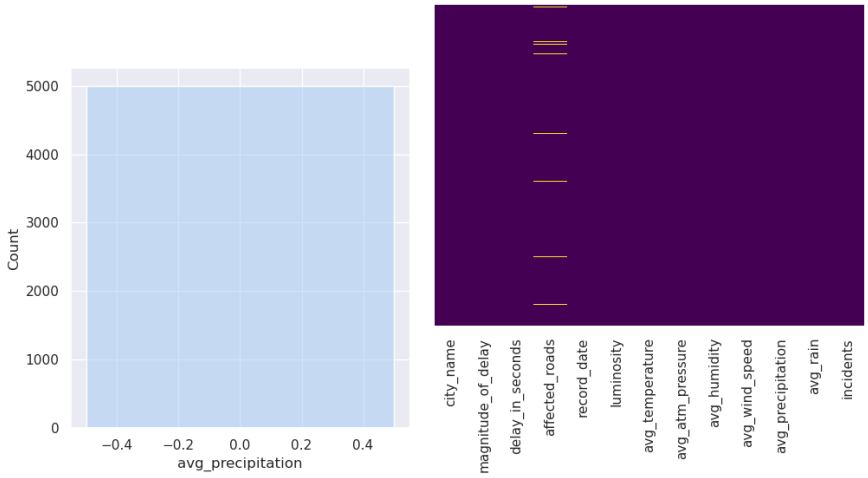


Fig. 1: pontos [1] e [2], respetivamente.

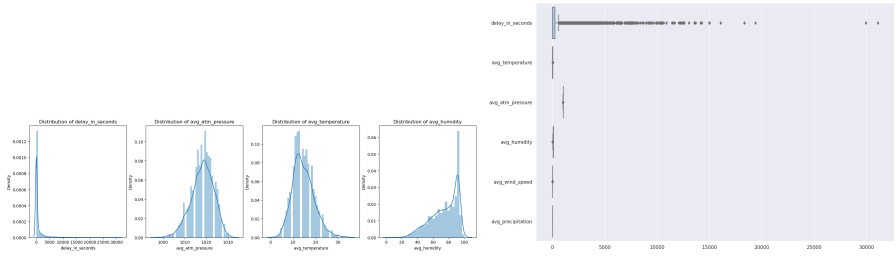


Fig. 2: pontos [3] e [4]

3.2 *Data Preparation*

Nesta fase, preparamos os dados de modo a que pudessem ser utilizados pelos vários modelos. Este processo surge detalhado nos *notebooks* disponibilizados, sendo que estes surgem numerados, para que possa ser possível compreender bem todo o processo de tratamento, quais as modificações efetuadas, de acordo com os resultados que iam sendo obtidos. O tratamento seguinte foi o ponto de origem de todos os cenários, constituindo assim o `cenário 0`. Note-se que aqui não se encontram todos os processos de tratamento, mas apenas os que foram feitos no *dataset* que funcionou como ponto de partida para todos os cenários.

***Cleaning* - Remoção de colunas desnecessárias** Para conseguirmos determinar a importância de cada atributo, optamos por analisar a natureza dos valores: para os que eram categóricos, avaliar os possíveis valores que estes podiam tomar; para os numéricos, compreender como estes variavam. Então, reparamos que a o valor da cidade na coluna *city_name* era sempre o mesmo (`Guimarães`), assim como os valores na coluna *avg_precipitation* (`0.0`). Então, estas *features* não traziam novo conhecimento ao modelo, constituindo apenas ruído ao processo de modelação e o que poderiam levar a que o processo de previsão se tornasse tendencioso. Então, foram removidas.

***Transform* - Transformação de colunas categóricas em numéricas** É sabido que alguns modelos de aprendizagem automática apresentam incompatibilidade com valores categóricos e que funcionam melhor com tipos numéricos [4]. É por esta razão que as *features avg_rain, magnitude_of_delay, luminosity* e o *target incidents* foram convertidas em números naturais. Todos estes atributos representam níveis, pelo que têm uma ordem intrínseca associada. Então, esta transformação tem em conta essa característica.

***Construct, Format* - Valores em falta e construção de conhecimento** Nesta fase, apenas uma coluna, *affected_roads* apresentava *missing values*. Analisando bem os dados deste atributo, optamos por diferentes métodos para transformar esta coluna em conhecimento benéfico para os modelos. Primeiramente, observamos que os diferentes valores que a coluna poderia tomar eram: ‘‘`,`’’ ou ‘‘`N101, ...`’’. O primeiro caso parece ser uma anomalia, o autor provavelmente quereria expressar a inexistência de estradas afetadas, pelo que, assumimos isso como um valor nulo. Então, construimos uma função que observa os valores desta coluna e cria uma outra com a ocorrência de estradas diferentes afetadas: começa por substituir todos os casos ‘‘`,`’’ por 0. Para os restantes casos, conta cada ocorrência `[N,R]xxx` diferente. Tem-se uma nova coluna *affected_roads_number*, eliminando-se a *affected_roads*. Num cenário mais tardio, percebemos que o tipo de estrada não estava a ser tido em conta e que alterar o método de tratamento poderia vir a melhorar os modelos. Neste contexto, criamos uma nova função para tratar estes valores, a qual, criava duas novas colunas `Regional` e `National`, com o número de estradas encontradas para cada tipo.

***Construct* - Tratamento de registos de datas** O formato da *feature record-date* não era muito ideal para ser utilizado pelo modelo, então, numa tentativa de melhorar isso, optamos por, pela realização de *feature engeneering*, obtendo, para cada data, os respetivos dia, mês, ano, hora e minutos. Também obtivemos o dia do ano e o dia da semana. Todos estes dados constituem uma representação numérica. Mais tarde, veio-se a reparar que as colunas *year* e *minute* era sempre iguais, pelo que foram também removidas.

***Construct, Select, Transform* - Tratamento de outliers** Ao longo do projeto, como se mostra nos cenários na fase de modelação, foram detetados e tratados *outliers* de diferentes formas [1]: em relação à deteção, foi realizada através da distância interquartil (IQR) e através do método de *Mean Absolute Deviation* (MAD); em relação ao seu tratamento, optou-se por substituição e remoção. Por questão de simplicidade, optou-se por uma análise univariada. O processo de deteção foi realizado de acordo com a distribuição de cada *feature*: os *outliers* das que estavam distribuídas normalmente foram detetados através de MAD, já que dos três métodos (distância IQR, MAD e Z-Score), é o que é mais robusto para distribuições normais de dados (o Z-Score é também assume dados normais, mas é menos robusto) [3]; os *outliers* das que não estavam distribuídas normalmente foram detetados através da distância IQR, já que é um método mais simples e facilmente extensível para dados com *skewness*. Quanto ao tratamento, experimentou-se remoção e substituição pela moda, média e mediana, selecionando-se o melhor método, tendo em conta os resultados obtidos. Numa fase mais tardia, também se experimentou realizar a substituição de acordo com a percentagem de *outliers*. Durante o processo de tratamento (substituição) tivemos em conta que a mediana seria uma medida melhor de tendência central do que a média, e o intervalo interquartil seria uma medida melhor de dispersão do que o desvio padrão, já que a média e o desvio padrão são sensíveis a todos os pontos do conjunto de dados, incluindo os valores discrepantes. Mas a mediana e a distância IQR podem ignorar esses valores discrepantes, fornecendo medições mais precisas dos dados.

Select* - *Feature engeneering Para além das *features* de registo temporal mencionadas, também extraímos mais algumas, como a averiguação sobre se o dia em questão era uma feriado, ou fim de semana, e a sua estação dando origem às colunas `is_weekend`, `is_holiday`, `is_season`.

***Transform* - Normalização** Observamos que algumas colunas apresentavam diferentes escalas, ainda que esta diferença não fosse muito significativa. Existem alguns modelos que necessitam de normalização [9] e, apesar dos que foram utilizados não necessitarem, esta foi objeto de análise já que poderia vir a melhorar os resultados. Existem diferentes formas de normalização [7], contudo, a utilizada consistiu em transformar os valores dos atributos entre 0 e 1, sendo 0 o valor mais baixo do tributo e 1 o mais alto, dando uma mesma importância a

todos os atributos do *dataset*. Como os atributos categóricos presentes eram todos ordinais, não houve necessidade de utilizar *encoding* para dados sem ordem intrínseca. O único caso que suscitou dúvida foi o *encoding* dos dias da semana. Contudo, para evitar a criação exagerada de colunas, optou-se por não utilizar *one-hot-enconding* ou outro semelhante.

3.3 *Modeling and Evaluation*

Estas duas fases estiveram sempre relacionadas, já que cada cenário introduzia uma mudança nos dados que se traduzia num resultado diferente e que, então, requeria uma nova avaliação. Abaixo segue-se o conjunto de cenários que o grupo enfrentou ao longo da modelação. É sabido que diferentes modelos necessitam diferentes processamentos, que se adequem mais ao tipo de modelo utilizado, então, apesar de se ter realizado um pré-processamento comum, após se apurarem os dois melhores modelos, foi realizado e experimentado um tratamento de dados diferentes, que viesse melhorar o desempenho desse modelo específico.

Select Modeling Technique Como forma de também experimentar algumas ferramentas transversais, o grupo recorreu a técnicas de *AutoML* como primeiro passo para perceber quais os melhores modelos para os nossos dados, bem como as melhores métricas. Utilizando-se o módulo `Pycaret` [5] de alto nível, construi-se um experimento: `experiment = setup(incidentes, target='incidents')`, com parâmetros *default*, obtendo-se a seguinte tabela com informações dos processos que os dados experimentaram (tabela da esquerda). De seguida, é possível imprimir uma comparação de todos os modelos possíveis, onde os valores a amarelo são as melhores métricas (tabela da direita).

	Description	Value
0	Session id	8135
1	Target	incidents
2	Target type	Multiclass
3	Original data shape	(5000, 15)
4	Transformed data shape	(5000, 15)
5	Transformed train set shape	(3499, 15)
6	Transformed test set shape	(1501, 15)
7	Numeric features	14
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	constant
12	Low variance threshold	0
13	Fold Generator	StratifiedKFold
14	Fold Number	10
15	CPU Jobs	-1
16	Use GPU	False
17	Log Experiment	False
18	Experiment Name	clf-default-name
19	USI	0606

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.9248	0.9927	0.9248	0.9262	0.9247	0.8985	0.8988	7.3880
lightgbm	Light Gradient Boosting Machine	0.9246	0.9922	0.9246	0.9260	0.9243	0.8981	0.8985	0.2140
rf	Random Forest Classifier	0.9225	0.9921	0.9225	0.9240	0.9223	0.8954	0.8958	0.1160
gbc	Gradient Boosting Classifier	0.9085	0.9882	0.9085	0.9093	0.9081	0.8763	0.8767	0.7350
dt	Decision Tree Classifier	0.9043	0.9409	0.9043	0.9065	0.9046	0.8709	0.8712	0.0170
et	Extra Trees Classifier	0.8863	0.9863	0.8863	0.8859	0.8849	0.8456	0.8461	0.1550
knn	K Neighbors Classifier	0.8380	0.9657	0.8380	0.8401	0.8377	0.7810	0.7815	0.0260
qda	Quadratic Discriminant Analysis	0.6608	0.9026	0.6608	0.6488	0.6337	0.5180	0.5324	0.0180
lda	Linear Discriminant Analysis	0.6462	0.8482	0.6462	0.6402	0.6107	0.4915	0.5076	0.0180
lr	Logistic Regression	0.6127	0.8503	0.6127	0.5873	0.5732	0.4426	0.4576	0.6610
ridge	Ridge Classifier	0.5919	0.0000	0.5919	0.6179	0.5045	0.3881	0.4299	0.0140
ada	Ada Boost Classifier	0.5762	0.8096	0.5762	0.5887	0.5638	0.4128	0.4216	0.0590
nb	Naive Bayes	0.5399	0.8146	0.5399	0.4809	0.4711	0.3223	0.3525	0.0170
svm	SVM - Linear Kernel	0.4395	0.0000	0.4395	0.3934	0.3662	0.2361	0.2839	0.0260
dummy	Dummy Classifier	0.4055	0.5000	0.4055	0.1645	0.2340	0.0000	0.0000	0.0190

Fig. 3: Processamento e Modelação Automáticos.

A amarelo encontram-se as métricas dos melhores modelos. Então, começamos por efetuar manualmente cada tipo de modelo (os top-6), imprimindo para cada um a sua matriz de confusão e o seu `accuracy report`. Apesar disso, foi efetudo

um processamento individual para os top-2 modelos *RandomForesteClassifier e Light Gradient Boosting Machine*, de modo a alcançar as melhores métricas, tendo também maior variedade de modelos diferentes analisados. A aboradagem utilizadada foi, então, selecionar os 2 melhores modelos, tentando provocar uma melhoria nos mesmos, surgindo, assim, o contexto da otimização.

Build Model Nesta fase, segue-se o conjunto de cenários abordados, bem como o conjunto de decisões tomadas e respetivas justificações.

Cenário 0 Este cenário consistiu no *dataset* inicial pré-processado, sem mais alterações, servindo apenas de base para análise. A este conjunto de dados foram aplicados diretamente os top-6 modelos. A métrica utilizada para avaliação dos modelos foi a `accuracy` e, em alguns casos, a validação por `cross-validation`.

Cenário 1 - *Feature Engeneering* Método 1 de tratamento da coluna *affected_roads*; Remoção de colunas redundantes como *minute* e *year*; Extração das `is_weekend, is_holidays` e `season`. A maioria dos modelos experimentou melhoria.

Cenário 2 - *Feature Engeneering* com método diferente Método 2 de tratamento da coluna *affected_roads*; Remoção de colunas redundantes como *minute* e *year*; Extração das `is_weekend, is_holidays` e `season`. A maioria dos modelos experimentou melhoria, prosseguimos com o segundo método de tratamento das estradas afetadas.

Cenário 3 - Tratamento de *outliers* Deteção de *outliers* para colunas numéricas por distância IQR e por MAD, com substituição (média, moda, mediana) ou remoção. O método que mostrou melhorar a `accuracy` foi o de remoção, todavia, através de submissão no *Kaggle*, compreendemos que, com mais dados, o desempenho era muito pior, o que suscitou a existência de *overfitting*, já que com remoção se perdia informação importante e o modelo ficava mais tendencioso, considerando também o elevado número de *outliers*.

Cenário 4 - Tratamento de *outliers* com transformações logarítmicas Deteção de *outliers* para colunas numéricas por distância IQR e por MAD, com substituição (média, moda, mediana) ou remoção, para colunas que seguem uma distribuição normal e aplicação de transformação logarítmica para as que têm *skewness*. O método que mostrou melhorar a `accuracy` foi na mesma o de remoção, todavia, para o caso do LGBM, este segundo método com transformações mostrou-se melhor. A transformação log. têm a característica de normalizar os dados e reduzir o impacto dos *outliers*.

De seguida, efetuou-se um processamento individual para os top-2 modelos, tendo em conta os resultados obtidos nos cenários anteriores. Seguem-se um resumo do processamento efetuado.

Processamento individual - *Random Forest* Para este modelo, o tratamento de *outliers*, não veio melhorar o modelo, então, experimentou-se apenas `normalização`, mas não teve grande impacto no modelo, tendo vindo piorar o modelo. É compreensível tendo em conta que o próprio algoritmo já possui alguma normalização. Efetuou-se `cross-validation` e `feature importance` com diferentes *thresholds*, concluindo que as *features* inicialmente adicionadas, não tinham uma contribuição muito significativa para o modelo e que o aumento do desempenho era proveniente essencialmente da utilização do método 2 de tratamento da coluna *affected_roads*.

Processamento individual - *Ligth Gradient Boosting Machine* Para este modelo, utilizou-se o método 2 de tratamento de *outliers*, substituindo-os pela média, no caso de atributos sem *skewness*, e aplicando a transformação log. nos restantes. Experimentou-se também `normalização`, mas não teve grande impacto no modelo, tendo vindo piorar o modelo. Também se avaliou o modelo através de `cross-validation`.

3.4 Avaliação e Análise Crítica de Resultados

Referimos quais as métricas que foram utilizadas para avaliar o desempenho dos modelos e a respetiva justificação. Seguem-se os resultados da aplicação de cada cenário. Foi realizado um *RoadMap* temporal com as várias submissões: todo o tratamento e modelos utilizados [página 33].

Métricas de Avaliação Utilizadas Para avaliar o desempenho do modelo, o grupo optou por recorrer às seguintes métricas de classificação:

- **Accuracy:** Percentagem de previsões corretas feitas pelo modelo. É uma métrica comum, mas pode ser enganosa em conjuntos de dados desbalanceados. Como o nosso conjunto de dados não se encontrava muito desbalanceado, consideramos uma boa métrica a utilizar;
- **Matriz de Confusão:** uma tabela que mostra o número de vezes em que o modelo classificou corretamente ou incorretamente cada classe. A diagonal principal da matriz mostra o número de vezes em que o modelo classificou corretamente cada classe, enquanto os elementos fora da diagonal mostram o número de vezes em que o modelo classificou incorretamente cada classe;
- **Curva ROC (*Receiver Operating Characteristic)*:** um gráfico que mostra a taxa de verdadeiros positivos em relação à taxa de falsos positivos para diferentes pontos de corte do modelo. A área sob a curva (AUC) é uma medida de quão bem o modelo distingue entre duas classes. Esta área foi utilizada como forma de servir de despiste ao *overfitting*.
- **F1 Score:** a média harmónica da precisão e da revocação. É uma medida útil quando queremos equilibrar a precisão e a revocação. Foi sendo observada, ao longo da avaliação, para verificar que classes estavam a ser bem previstas.

Análise de Resultados Nesta secção, surge detalhada a análise dos resultados que foram sendo obtidos, de acordo com as métricas referidas anteriormente. Urge também referir a forma como foi obtido melhor desempenho através das várias submissões efetuadas.

Modelo	Antes	Depois	Obs
K Neighbors Classifier	0.8312	0.83120	**Manteve**
Extra Trees Classifier	0.8968	0.89760	**Melhorou**
Gradient Boosting Classifier	0.8984	0.90000	**Melhorou**
Decision Tree Classifier	0.9080	0.90640	Piorou
Random Forest Classifier	0.9168	0.92480	**Melhorou**
Light Gradient Boosting Machine	0.9232	0.92240	Piorou

Modelo	Antes	Depois	Obs	Depois (M2)
K Neighbors Classifier	0.8312	0.83120	**Manteve**	0.83360
Extra Trees Classifier	0.8968	0.89760	**Melhorou**	0.90640
Gradient Boosting Classifier	0.8984	0.90000	**Melhorou**	0.90960
Decision Tree Classifier	0.9080	0.90640	Piorou	0.92240
Random Forest Classifier	0.9168	0.92480	**Melhorou**	0.92960
Light Gradient Boosting Machine	0.9232	0.92240	Piorou	0.93760

Fig. 4: Resultados cenário 1 (cima) e 2 (baixo).

Comparação entre os cenários 0 e 1 O que diferencia ambos os cenários é a existência de mais *features* extraídas no cenário 1. Na primeira tabela acima, observamos como variou a accuracy antes e depois da extração. De um modo geral, houve uma melhoria no desempenho, mesmo não tendo sido significativa, sendo que apenas os modelos *Decision Tree Classifier* e LGBM experimentaram um pior desempenho.

Comparação entre os cenários 0 e 2 O que diferencia ambos os cenários é a forma de tratamento da coluna *affected_roads*, a qual, no caso do cenário 2, foi tratada criando-se duas novas colunas, para cada tipo de estrada, com a sua ocorrência. Como se observa na segunda tabela acima, esta mudança veio melhorar imenso o desempenho dos vários modelos (confrontando a segunda e última colunas).

Então, mantivemos o tratamento do cenário 2.

Modelo	Antes	Depois (moda)	Depois (mediana)	Depois (media)	Depois (removido)
K Neighbors Classifier	0.83360	0.76400	0.82320	0.76480	0.89124
Extra Trees Classifier	0.90640	0.90400	0.91040	0.89600	0.94864
Gradient Boosting Classifier	0.90960	0.89760	0.90400	0.90000	0.95670
Decision Tree Classifier	0.92240	0.89680	0.91280	0.89760	***0.96274***
Random Forest Classifier	0.92960	0.91440	0.91920	0.91520	***0.96777***
Light Gradient Boosting Machine	0.93760	0.91920	0.92320	0.92080	***0.96979***

Modelo	Antes	Depois (moda)	Depois (moda) M2	Depois (mediana)	Depois (mediana) M2	Depois (media)	Depois (media)M2	Depois (removido)	Depois (removido)M2
K Neighbors Classifier	0.83360	0.76400	0.83360	0.82320	0.83360	0.76480	0.83360	0.89124	0.84194
Extra Trees Classifier	0.90640	0.90400	*0.90960*	0.91040	0.90960	0.89600	0.91040	0.94864	0.89274
Gradient Boosting Classifier	0.90960	0.89760	*0.91200*	0.90400	0.90800	0.90000	0.90800	0.95670	0.92661
Decision Tree Classifier	0.92240	0.89680	*0.92240*	0.91280	0.92240	0.89760	0.92240	***0.96274***	0.93710
Random Forest Classifier	0.92960	0.91440	0.92720	0.91920	0.92880	0.91520	0.92320	***0.96777***	0.93871
Light Gradient Boosting Machine	0.93760	0.91920	0.93760	0.92320	0.93840	0.92080	**0.94000**	***0.96979***	0.94677

Fig. 5: Resultados cenário 3 (cima) e 4 (baixo).

Comparação entre os cenários 2 e 3 O que diferencia ambos os cenários, é a existência de tratamento de *outliers*, no cenário 3. Conseguimos observar, para cada modelo, qual o melhor método de substituição. Como se observou, houve um aumento elevado da accuracy de todos os modelos, quando se realizou a remoção dos *outliers*. Apesar disso, optamos por não envergar pelo método de remoção, dada a percentagem de *outliers* razoavelmente elevada para algumas *features*. Esta remoção levou claramente a *overfitting*, como se comprovou também através da submissão no *Kaggle*, onde, com 70% dos dados, a accuracy diminuiu drasticamente. Isto significaria que a elevada accuracy se devia à boa previsão da classe maioritária.

Comparação entre os cenários 2 e 4 O que diferencia ambos os cenários é a existência de um tratamento com transformação logarítmica para as *features*

com *skewness*. Começamos a focar-nos nos top-2 modelos, o *Random Forest* e LGBM, sendo que, excetuando o método de remoção, pelos motivos anteriores, verificamos qual o melhor método seguinte: para o *Random Forest*, qualquer tratamento não veio melhorar. Para o LGBM, a susbtituição pela média melhorou um pouco o modelo, decidimos envergar por aí. Então, efetuamos um processamento individual para ambos os modelos, mantendo para o primeiro o cenário 0 e, para o segundo, o cenário 4.

4 *Dataset* de Grupo

4.1 *Data Understanding*

Describe Data O processo *getting in touch* para com o *dataset* ocorreu da mesma forma como para o anterior. Para descrever as nossas *features* e o *target*, observe-se a seguinte tabela.

Atributo	Descrição	DType	Tipo	Exemplo
Date	Data da recolha dos valores captados pelos sensores.	object	Categórico Ordinal	"2018/01/11"
Time	Momento temporal de captação.	object		"09:00:09"
S[1,2,3,4]_Temp	Temperatura captada pelo sensor [1,2,3,4] em graus Celsius.	int64	Numérico Contínuo	"24.94"
S[1,2,3,4]_Light	Luminosidade captada pelo sensor [1,2,3,4] em LUX (unidade SI de fluxo luminoso por unidade de área, ou seja da densidade de intensidade luminosa conhecida por iluminância).			"121"
S[1,2,3,4]_Sound	Som captado pelo sensor 1 em Volts (o que determina a diferença entre sinais de áudio é a sua voltagem).			"0.08", "3.16"
S5_CO2	Concentração de CO2 captada pelo sensor 5 em PPM.			"355"
S5_CO2_Slope	Inclinação de CO2 captada pelo sensor 5.			"4.873077"
S[6,7]_PIR	Sensor [6,7] PIR (digital passive infrared) para deteção de movimento (através de infravermelhos).			"0" ou "1"
Room_Occupancy_Count (target)	Número de pessoas na sala (determinado manualmente por uma pessoa).		Numérico Discreto	"0", "4"

Table 2: Features Target.

Explore Data Metodologia seguida exatamente como no dataset anterior. Destas duas sub-fases anteriores, observamos , de um modo geral, o seguinte:

1. Existência de desbalanceamento ao nível do *target*: a classe 0 é maioritária;
2. Exploração efetuada da ocupação ao longo das horas de cada dia, para cada *feature*: confirmação da menor ocupação em alturas de férias/feriados, por exemplo;
3. Existência de dados que não seguem uma distribuição normal;
4. Existência de muitos *outliers* na generalidade dos atributos;

4.2 *Data Preparation*

Nesta fase, preparamos os dados de modo a que pudessem ser utilizados pelos vários modelos. Este processo surge detalhado nos *notebooks* disponibilizados,

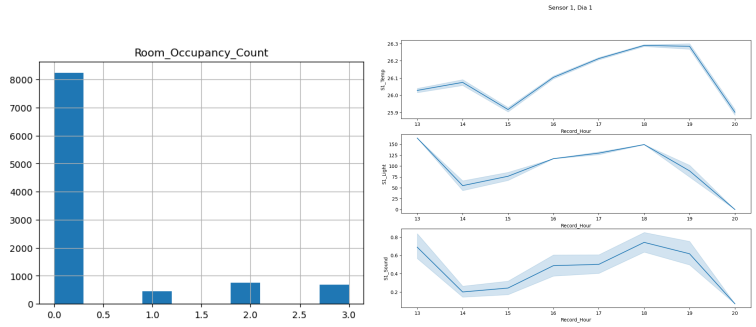


Fig. 6: pontos [1] e [2], respetivamente.

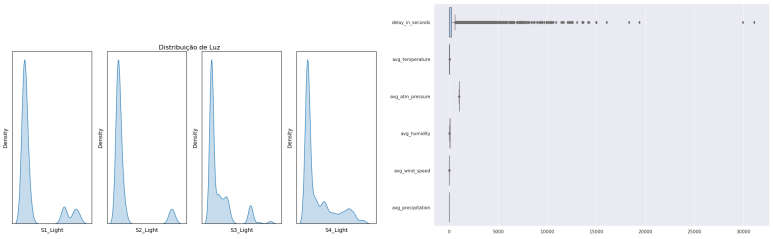


Fig. 7: pontos [3] e [4]

sendo que estes surgem numerados, para que possa ser possível compreender bem todo o processo de tratamento, quais as modificações efetuadas, de acordo com os resultados que iam sendo obtidos. O tratamento seguinte foi o ponto de origem de todos os cenários, constituindo assim o `cenário 0`. Note-se que aqui não se encontram todos os processos de tratamento, mas apenas os que foram feitos no *dataset* que funcionou como ponto de partida para todos os cenários.

***Construct, Format* - Valores em falta e Duplicados** Verificamos que não existiam valores em falta no *dataset*. Neste caso, tratando-se cada registo de uma captação única de cada sensor, seria impossível e incoerente aceitar que existissem valores duplicados nos dados, então, verificamos a existência de duplicados e removêmo-los.

***Transform* - Desbalanceamento** O número de registos para uma ocupação igual a 0 era muito superior, quando comparada com os restantes valores (1,2,3 e 4). É sabido que o desbalanceamento causa problemas no treino dos modelos, todavia, apenas em casos muito específicos. Sendo assim, o grupo experimentou uma forma de balanceamento ainda não muito explorada e também não muito adotada na insdústria, chamada *Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise* (SMOGN) [8]. Este algoritmo cria registos aleatórios das classes minoritárias, de modo a balancear os dados. Esta

abordagem acabou por tomar complicações, pela qual foi abandonada. O grupo optou por não envergar para a utilização de algoritmos de *data augmentation* como GANs e VANs, já que teríamos de os implementar nós próprios e seria algo muito complexo. Assim sendo, para contornar o problema do *overfitting*, o grupo utilizou métricas específicas e formas de validação adequadas, que reduzem o seu impacto.

Select - Feature Selection Verificamos a existência de colunas que mediam uma mesma coisa, ou seja, existiam diferentes sensores que mediam uma mesma coisa e que estariam correlacionados. A existência de multicolinearidade [10], ou seja, de *features* correlacionadas entre si, constitui um problema para o treino dos modelos [6], pelo que, especialmente neste caso, deveria ser corrigido. Então, tendo em conta a distribuição dos dados, que não era normal, o método mais adequado para calcular a correlação é o de *Spearman* [2]. A técnica utilizada para seleção de *features* foi o teste de hipóteses [**?**], pelo que, duas colunas estariam correlacionadas se o valor de correlação fosse significante, tendo em conta o nível de significância. Para cada par de *features*, calculamos a sua correlação, definimos a hipótese nula (*features* correlacionadas) e a alternativa (*features* não correlacionadas), calculamos o *p-value* e comparámo-lo com o nível de significância definido. Se o *p-value* fosse inferior à significância, não poderíamos rejeitar a hipótese nula, pelo que aceitaríamos a alternativa. Sabendo que *features* estavam relacionadas, bastava saber quais delas estavam mais relacionadas com o *target*. Assim, eliminamos as *features S[2,3,4]_Temp, S[2,3,4]_Light, S[2,3,4]_Sound, S5_CO2_Slope, S7_PIR*.

Select - Feature engeneering Efetuou-se a transformação das datas e das horas, exatamente como no *dataset* da competição. Como forma de experimentação, também recorremos a *Wrapper methods* para comprovar que a nossa seleção estava correta, utilzando *Forward selection, Backward elimination e Bidirectional elimination (Stepwise Selection)*. Observamos que atributos apareciam mais vezes e realizamos modelos para verificar se as métricas melhoravam ou não.

***Construct, Select, Transform* - Tratamento de outliers** O problema dos *outliers* era, neste caso, muito relevante, porque, dada a existência de muitos registos para a ocupação igual a 0, poderia ser utilizado para diminuir o ruído e assim tornar os dados mais balanceados [1]. Após termos as *features* mais relevantes, detetamos os valores discrepantes de cada uma, através da distância IQR, porque, como mencionado anteriormente, é mais robusta a dados não normalizados. O método utilzado para tratamento já se aproximou de uma análise bivariada, porque recorremos à análise do *target*. Seguimos um método de tratamento específico, elucidado no fluxograma seguinte:

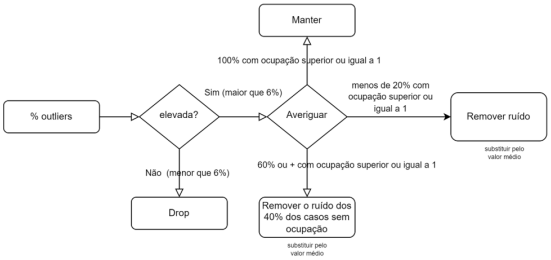


Fig. 8: Método de tratamento baseado na percentagem e no tipo de outliers.

***Transform* - Normalização** Da mesma forma como para o *dataset* anterior, realizamos uma normalização dos dados [7]. Mostrou-se relevante dada a diferença de escala entre os valores de alguns atributos. O método de normalização utilizado foi o *MinMaxScaler* que efetua a transformação dos valores de cada *feature* para valores entre os *inputs* fornecidos. Utilizou-se normalização entre 0 e 1. Esta normalização não alterou o modelo final selecionado, o *Random Forest*, contudo, para modelos que se regem por cálculos de coeficientes de regressão, a normalização mostrou-se essencial.

4.3 *Modeling and Evaluation*

Select Modeling Technique Também para este *dataset* utilzamos *AutoML* para averiguar sobre quais os modelos que poderíamos utilizar.

	Description	Value
0	Session id	946
1	Target	Room_Occupancy_Count
2	Target type	Regression
3	Data shape	(10129, 12)
4	Train data shape	(7090, 12)
5	Test data shape	(3039, 12)
6	Numeric features	11
7	Preprocess	True
8	Imputation type	simple
9	Numeric imputation	mean
10	Categorical imputation	constant
11	Low variance threshold	0
12	Fold Generator	KFold
13	Fold Number	10
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	reg-default-name
18	USI	07af

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	0.0122	0.0040	0.0590	0.9950	0.0246	0.0243	1.8530
et	Extra Trees Regressor	0.0055	0.0047	0.0611	0.9940	0.0245	0.0118	0.0870
rf	Random Forest Regressor	0.0068	0.0050	0.0636	0.9938	0.0280	0.0133	0.1510
xgboost	Extreme Gradient Boosting	0.0066	0.0063	0.0657	0.9921	0.0246	0.0145	0.1300
lightgbm	Light Gradient Boosting Machine	0.0222	0.0099	0.0961	0.9877	0.0499	0.0362	0.0400
dt	Decision Tree Regressor	0.0055	0.0097	0.0832	0.9874	0.0336	0.0117	0.0160
gbr	Gradient Boosting Regressor	0.0277	0.0144	0.1166	0.9823	0.0448	0.0525	0.1230
knn	K Neighbors Regressor	0.0155	0.0174	0.1280	0.9785	0.0608	0.0298	0.0180
ada	AdaBoost Regressor	0.0589	0.0193	0.1374	0.9760	0.0656	0.1224	0.0650
ridge	Ridge Regression	0.2237	0.1668	0.4075	0.7915	0.1989	0.3024	0.0120
lr	Linear Regression	0.2238	0.1668	0.4075	0.7915	0.1989	0.3025	0.4780
br	Bayesian Ridge	0.2234	0.1668	0.4075	0.7915	0.1989	0.3021	0.0170
en	Elastic Net	0.1873	0.1946	0.4399	0.7564	0.2107	0.3040	0.0180
par	Passive Aggressive Regressor	0.2154	0.1978	0.4436	0.7526	0.2153	0.2947	0.0130
lasso	Lasso Regression	0.1864	0.1988	0.4447	0.7512	0.2146	0.2993	0.0130
huber	Huber Regressor	0.2210	0.2064	0.4528	0.7417	0.2211	0.3189	0.0770
omp	Orthogonal Matching Pursuit	0.1981	0.2267	0.4750	0.7164	0.2116	0.4092	0.0180
lar	Least Angle Regression	0.3179	0.2529	0.4950	0.6860	0.2677	0.3399	0.0140
llar	Lasso Least Angle Regression	0.6505	0.8032	0.8954	-0.0022	0.4659	0.7772	0.0160
dummy	Dummy Regressor	0.6505	0.8032	0.8954	-0.0022	0.4659	0.7772	0.0130

Fig. 9: Processamento e Modelação Automáticos.

A amarelo encontram-se as métricas dos melhores modelos. Então, começamos por efetuar manualmente cada tipo de modelo (os top-3). Para este *dataset*,

usamos *Cross Validation* (CV), que oferece muitas formas de dividir um conjunto de dados. No entanto, estando a lidar com um problema de regressão com um grande desequilíbrio na distribuição do *target*, recorreu-se a *StratifiedK-Fold* (STK).Construiram-se os modelos de *Decision Tree, Random Forest e Extra Trees*. Para não existir problemas de compatibilidade com a técnica de CV, optamos por explorar estes algoritmos da biblioteca *sklearn*. Foi, mais tarde, efetuado um processamento individual para o melhor modelo *Random Forest Regressor*.

Build Model Nesta fase, segue-se o conjunto de cenários abordados, bem como o conjunto de decisões tomadas e respetivas justificações.

Cenário 0 Este cenário consistiu no *dataset* inicial pré-processado, sem mais alterações, servindo apenas de base para análise. A este conjunto de dados foram aplicados diretamente os top-3 modelos, determinado com `AutoML`. Em termos de processamento, efetuamos `feature selection`, para reduzir a multicorrelação, e tratamos as datas, por `feature engeneering`, extraíndo os campos relevantes. A métrica utilizada para avaliação dos modelos foram as métricas conhecidas de regressão com validação por `cross-validation`.

Cenário 1 - *Feature Selection* com *Wrappers* Explorou-se `feature selection` com `wrapper methods`, de modo a comprovar que a seleção anterior era coerente. Aplicação dos top-3 modelos. A métrica que utilizámos, teve em conta a existência de *outliers* e o desbalanceamento dos dados: RMSLE conjugado com R2_Score. Neste caso, demos mais peso ao RMSLE, devido à distribuição anormal dos dados, que ainda não estava corrigida. De um modo geral, houve diminuição de desempenho, comprovando que as *features* selecionadas pelos métodos não eram as melhores, e, como não eram as que foram selecionadas anteriormente no cenário 0, permitiram comprovar uma escolha acertada da utilização do teste de hipóteses para selecionar os atributos relevantes.

Cenário 2- Tratamento de *outliers* Neste cenário, efetuou-se o tratamento dos *outliers*, seguindo o método anteriormente especificado na preparação dos dados. Após o tratamento, decidimos que poderíamos utilizar apenas métricas menos sensíveis a *outliers* como o R2_Score. Tendo em conta a distribuição dos dados, continuaríamos a utiliar o RMSLE, como prioridade. Observou-se, melhoria de ambos os parâmetros em todos os modelos, pelo que o tratamento foi correto e melhorou o desempenho. Os melhores resultados foram com o *Random Forest*, logo, realizaremos um processamento individual para esse modelo, o top-1.

De seguida, efetuou-se um processamento individual para os top-1 modelo, tendo em conta os resultados obtidos nos cenários anteriores. Segue-se um resumo do processamento efetuado.

Processamento individual - *Random Forest* Para este modelo, os *outliers* já estavam tratados, assim como se encontrava já reunido o conjunto de atributos mais relevantes. Então, experimentou-se apenas `normalização`, mas não teve grande impacto no modelo, as métricas mantiveram-se. É compreensível tendo em conta que o próprio algoritmo já possui alguma normalização. Efetuou-se `cross-validation` e `feature importance`, concluindo-se que as *features* inicialmente selecionadas, no teste de hipóteses do cenário 0, tinham uma contribuição significativa para o modelo, donde o aumento do desempenho era proveniente.

4.4 Avaliação e Análise Crítica de Resultados

Métricas de Avaliação Utilizadas

- **MAE *(Mean Absolute Error)*:** a média do erro absoluto entre as previsões do modelo e os valores reais. É calculada como a soma dos erros absolutos dividida pelo número de observações. É uma métrica que não é sensível a *outliers*;
- **MSE *(Mean Squared Error)*:**] a média dos erros ao quadrado entre as previsões do modelo e os valores reais. É calculada como a soma dos erros ao quadrado dividida pelo número de observações. É uma métrica sensível a *outliers*;
- **RMSE *(Root Mean Squared Error)*:** a raiz quadrada da média dos erros ao quadrado entre as previsões do modelo e os valores reais. É uma medida do desvio médio das previsões do modelo em relação aos valores reais. Sensível a *outliers*;
- **r2_Score:** um coeficiente de determinação que mede o quão bem o modelo se ajusta aos dados. Ele varia de 0 a 1, sendo 1 o ajuste perfeito e 0 o ajuste pior possível. Não é sensível a *outliers*.
- **RMSLE *(Root Mean Squared Logarithmic Error)*:** a raiz quadrada do logaritmo do erro médio ao quadrado entre as previsões do modelo e os valores reais. É uma medida do desvio médio das previsões do modelo em relação aos valores reais, mas é mais robusta em relação a *outliers* do que o RMSE. Utilzamos mais esta métrica porque ser uma medida mais estável do desempenho do modelo, que tem em conta algum desbalanceamento que possa estar presente. Após o tratamento de *outliers*, já se poderia recorrer ao r2_score apenas.

Análise de Resultados Seguem-se os resultados obtidos em cada cenário, com as respetivas formas de comparação e justificações para a seleção do que melhor se adequava aos modelos.

Comparação entre os cenários 0 e 1 Como neste caso os outliers ainda não se encontravam tratados, a métrica RMSLE é mais robusta, pelo que vamos dar-lhe um maior peso. Estes cenários dizem respeito aos dados sem e com seleção de atributos, utilzando *wrappers*.

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.008152	0.002607	0.060198	0.061134	0.245313	0.923407	0.10839

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.009863	0.002966	0.073253	0.057951	0.238802	0.927402	0.108763

Fig. 10: Resultados entre o Cenário 0 e 1, para o modelo *Decision Trees*.

Utilizando as *features* selecionadas pelos *wrappers*, verificamos que há um aumento do RMSLE, o que é pior, e um aumento do R2 Score, o que é bom. Como o RMSLE tem maior peso, podemos afirmar que, de um modo geral, o modelo é pior.

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.436026	0.009631	0.062394	0.056071	0.235008	0.929755	0.103187

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.468816	0.009551	0.070982	0.052714	0.227873	0.933965	0.104607

Fig. 11: Resultados entre o Cenário 0 e 1, para o modelo *Random Forest*.

Utilizando as *features* selecionadas pelos *wrappers*, verificamos que há um aumento do RMSLE, o que é pior, e um aumento do R2 Score, o que é bom. Como o RMSLE tem maior peso, podemos afirmar que, de um modo geral, o modelo é pior.

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.189558	0.009823	0.100988	0.087753	0.295433	0.890095	0.127621

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.177921	0.008762	0.088962	0.059161	0.241024	0.925897	0.094383

Fig. 12: Resultados entre o Cenário 0 e 1, para o modelo *Extra Trees*.

Utilizando as *features* selecionadas pelos *wrappers*, verificamos que há uma diminuição do RMSLE, o que é melhor, e um aumento do R2 Score, o que é bom, podemos afirmar que, de um modo geral, o modelo é melhor.

Ora, as *features* selecionadas pelos métodos foram `S3_Temp, S1_Light, S5_CO2_Slope e S7_PIR` e, que, por acaso, contrárias às que foram obtidas pelo teste de hipóstes. Como a *performance* piorou, podemos afirmar que o teste de hipóteses continua a ser o melhor cenário e que essa técnica de *feature selection* foi bem conseguida. Mantemos o cenário 0.

Comparação entre os cenários 0 e 2 Houve grande melhoria na generalidade das métricas. Estes cenários dizem respeito aos dados com e sem *outliers*.

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.009863	0.002966	0.073253	0.057951	0.238802	0.927402	0.108763

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.007298	0.002828	0.049592	0.046388	0.21421	0.934622	0.095693

Fig. 13: Resultados entre o Cenário 0 e 2, para o modelo *Decision Trees*.

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.436026	0.009631	0.062394	0.056071	0.235008	0.929755	0.103187

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.360998	0.009638	0.050714	0.044551	0.209968	0.9372	0.089815

Fig. 14: Resultados entre o Cenário 0 e 2, para o modelo *Random Forest*.

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.189558	0.009823	0.100988	0.087753	0.295433	0.890095	0.127621

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.175293	0.009337	0.088795	0.073196	0.268775	0.896795	0.117914

Fig. 15: Resultados entre o Cenário 0 e 2, para o modelo *Extra Trees*.

Houve grande melhoria na generalidade das métricas. Seguimos para o processamento individual do modelo com melhores métricas, o *Random Forest*. Uma

forma de explicar o melhoramento geral das métricas, após o tratamento de *outliers*, pode ser a redução bem conseguida do ruído, nesse tratamento. Isto vem comfirmar que o método utilizado no tratamento se mostrou adequado. Mantemos o cenário 2 para este modelo.

Comparação entre os cenários 2 e 3 Aqui experimentou-se normalização, a qual não alterou as métricas. Mantivemos o cenário 2, como melhor cenário.

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.353381	0.009281	0.050714	0.044551	0.209968	0.9372	0.089815

fit_time	score_time	test_MAE	test_MSE	test_RMSE	test_r2_Score	test_RMSLE
0.346922	0.009467	0.050714	0.044551	0.209968	0.9372	0.089815

Fig. 16: Resultados entre o Cenário 2 e 3, para o modelo *Random Forest*.

De um modo geral, este melhor modelo experimentou um melhoramento de 7.45% (as métricas apresentadas resultam da técnica de CV, com STK).

5 Conclusões e trabalho futuro

Dado por concluido o projeto, faz sentido apresentar uma visão crıtica do trabalho, o qual nos permitiu consolidar a matéria lecionada na cadeira de DAA, especialmente no que diz respeito ao tratamento de dados e desenvolvimento de modelos de ML. Apesar de termos experimentado Redes Neuronais Artificias, seria interessante aprofundar esta temática, experimentando alterações de configurações.

No presene trabalho, destacamos o uso de diversos modelos de ML e técnicas para tratamento de dados desde *encoding, feature selection e feature engeneering*, até a normalização e tratamento de *outliers*. Em adição, recorremos a tecnologias novas, não lecionadas no contexto curricular, como AutoML, e também envergamos por técnicas diferentes de deteção e tratamento de dados, como transformações logarítmicas e a efetuação de uma análise unibivariada, o que demonstra o nosso interesse e dedicação pelo trabalho realizado. Tentamos ao máximo apresentar variedade na seleção do tipo de *dataset* (regressão e classificação), demodo a mostrar que conseguimos lidar com diferentes tipos de problemas.

Finalmente, consideramos que o presente documento se encontra explicativo, estando de acordo com as etapas estabelecidas na metodologia adotada, CRISP-DM. Além disso, o conjunto de modelos implementados é completo e a exploração realizada consegue responder de forma eficaz e correta aos problemas evidenciados nos *datasets*. Em suma, consideramos que o balanço do trabalho é positivo, as dificuldades sentidas foram superadas e os requisitos propostos foram cumpridos.

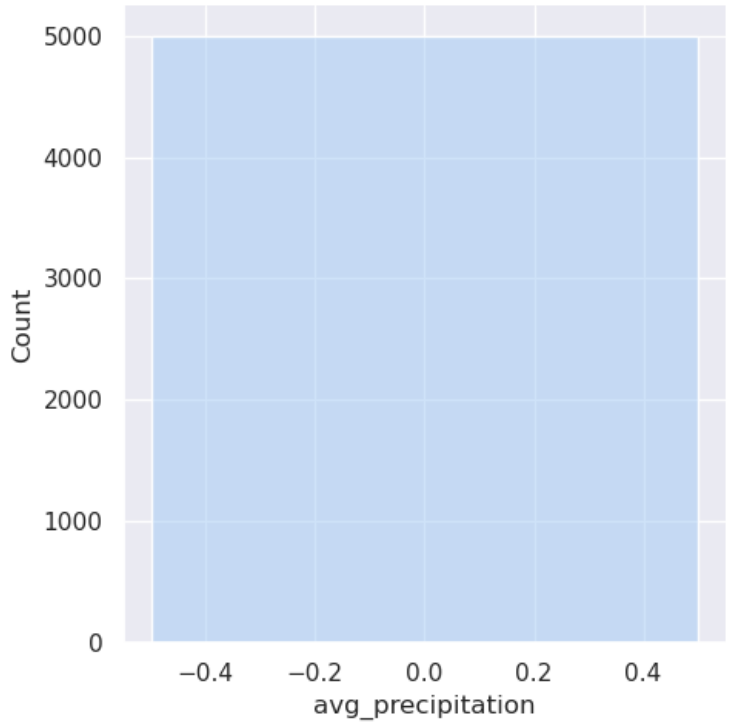
References

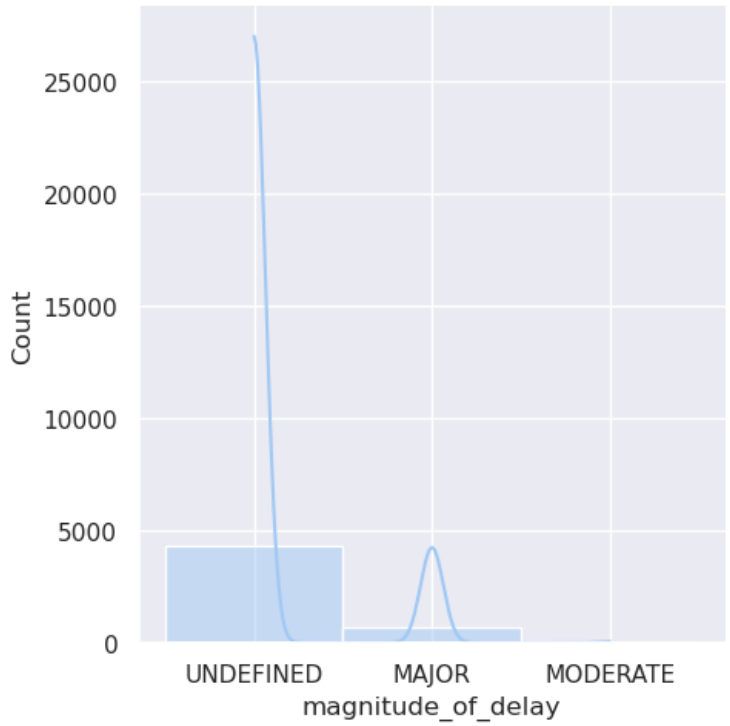
1. Advanced outlier handling methods — kaggle, https://www.kaggle.com/code/navinmundhra/advanced-outlier-handling-methods
2. Correlation coefficient — types, formulas examples, https://www.scribbr.com/statistics/correlation-coefficient/
3. Detecting and treating outliers in python — part 2 — by alicia horsch — towards data science, https://towardsdatascience.com/detecting-and-treating-outliers-in-python-part-2-3a3319ec2c33
4. Feature engineering — deep dive into encoding and binning techniques — by satyam kumar — towards data science, https://towardsdatascience.com/feature-engineering-deep-dive-into-encoding-and-binning-techniques-5618d55a6b38
5. Introduction to pycaret - build ml models faster w/ less code – learndatasci, https://www.learndatasci.com/tutorials/introduction-pycaret-machine-learning/
6. Multicollinearity — multicollinearity in data science models, https://www.analyticsvidhya.com/blog/2021/03/multicollinearity-in-data-science/
7. Normalization — machine learning — google developers, https://developers.google.com/machine-learning/data-prep/transform/normalization
8. R: Smogn algorithm for imbalanced regression problems, https://search.r-project.org/CRAN/refmans/UBL/html/SMOGNRegress.html
9. Which models require normalized data? — by gianluca malato — towards data science, https://towardsdatascience.com/which-models-require-normalized-data-d85ca3c85388
10. Hubert, M., Debruyne, M., Rousseeuw, P.J.: Minimum covariance determinant and extensions. Wiley Interdisciplinary Reviews: Computational Statistics **10** (5 2018). https://doi.org/10.1002/WICS.1421

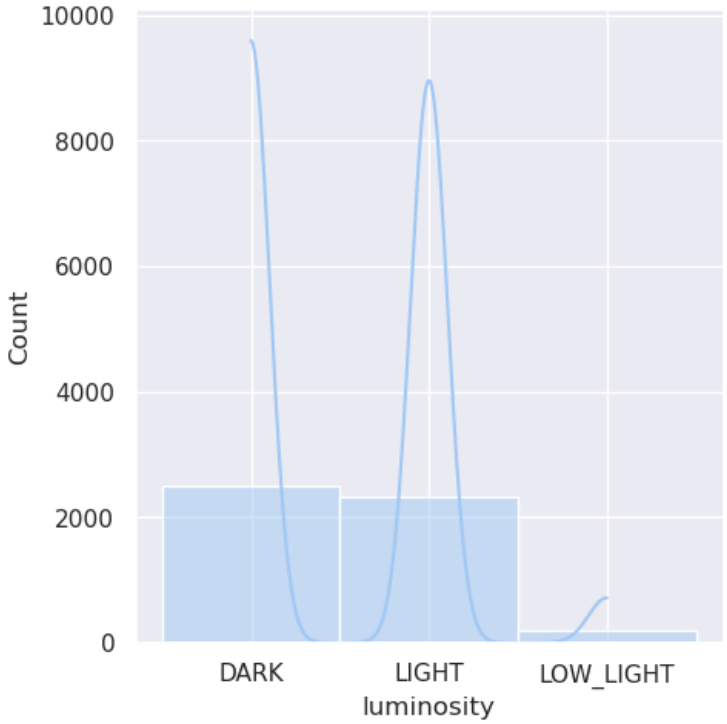
6 Anexos - *Dataset* de Competição

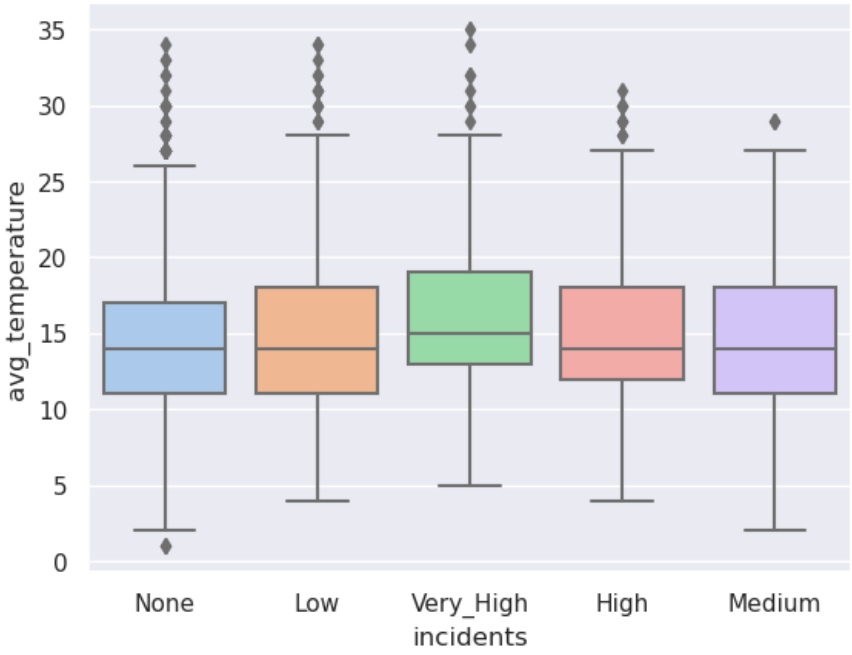
Atributo	**Descrição**	***DType***	**Tipo**	**Exemplo**
city_name	Cidade onde ocorreu a extração dos dados em causa.	*object*	Categórico Nominal	"Guimaras"
magnitude_of_delay	Magnitude do atraso, provocado pelos incidentes.	*object*	Categórico Ordinal	"UNDEFINED", "MODERATE", etc.
delay_in_seconds	Atraso, em segundos, provocado pelos incidentes.	*int64*	Numérico Discreto	"10", "162", etc.
affected_roads	Estradas afectadas pelos incidentes.	*object*	Categórico	"N101,N101"
record_date	O timestamp associado ao registo.	*object*	Categórico	"2021-03-15 23:00"
luminosity	O nível de luminosidade.	*object*	Categórico Ordinal	"DARK", "LIGHT", etc.
avg_temperature	Valor médio da temperatura.	*float64*	Numérico Discreto	"13.0", "7.0", etc.
avg_atm_pressure	Valor médio da pressão atmosférica.	*float64*	Numérico Discreto	"1024.0", "999.0", etc.
avg_humidity	Valor médio de humidade.	*float64*	Numérico Discreto	"8.0", "92.2", etc.
avg_wind_speed	Valor médio da velocidade do vento.	*float64*	Numérico Discreto	"10.0", "2.0", etc.
avg_precipitation	Valor médio de precipitação.	*float64*	Numérico Discreto	"0.0"
avg_rain	Avaliação qualitativa do nível de precipitação.	*object*	Categórico Ordinal	"Sem chuva", "chuva forte", etc.
incidents (target)	Nível de incidentes rodoviários.	*object*	Categórico Ordinal	"Low", "High", etc.

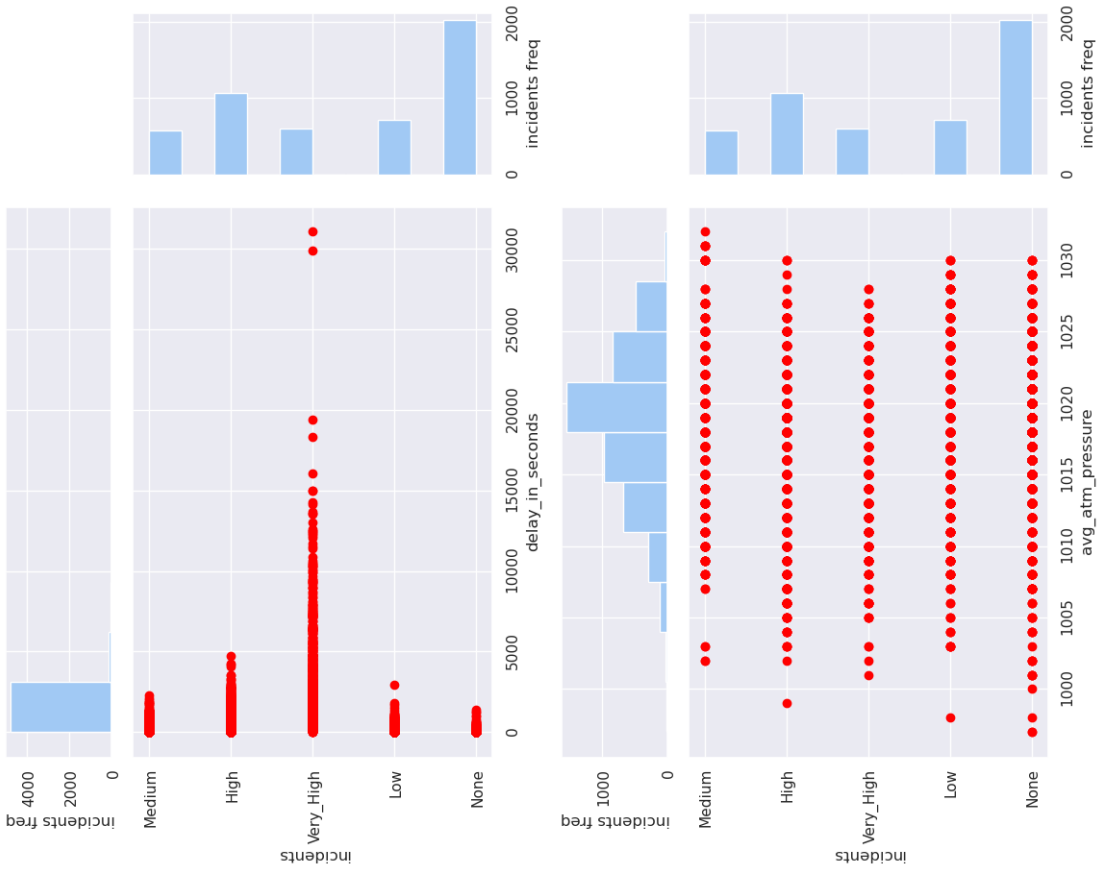
Table 3: Features & Target.

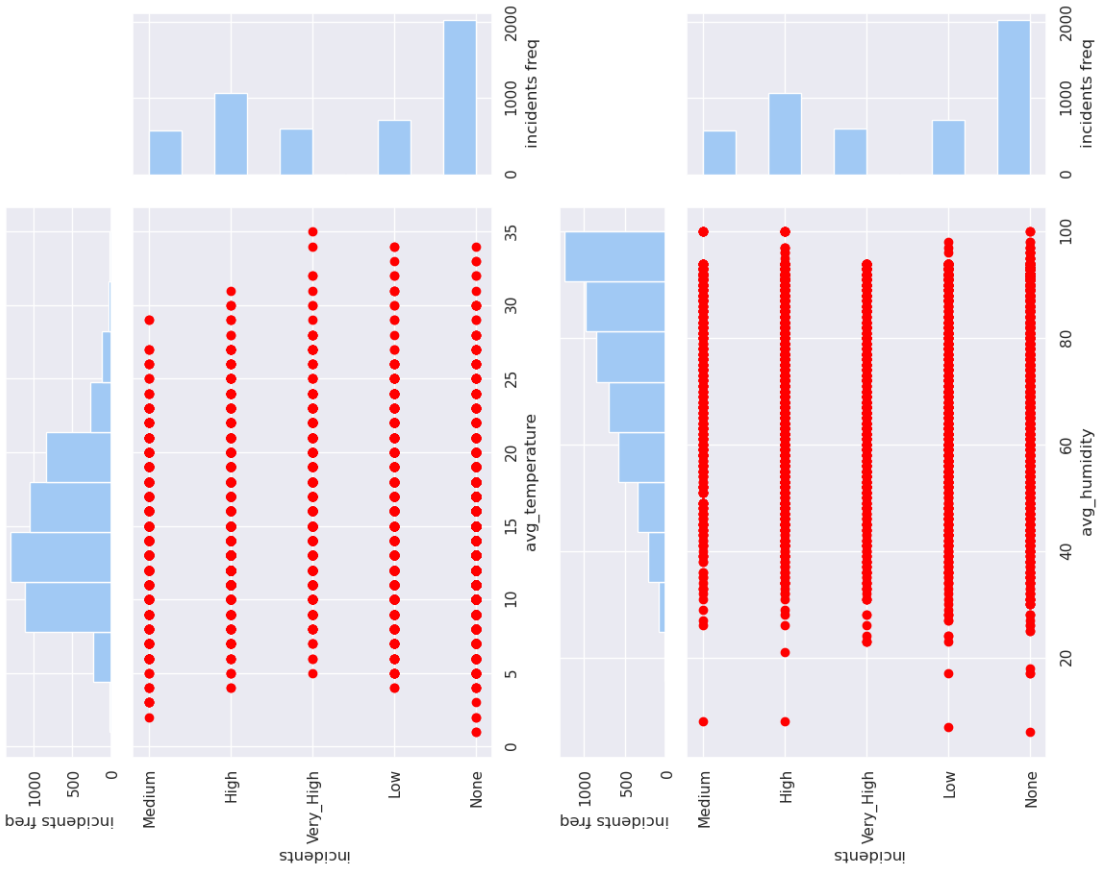

5000
4000
3000
2000
1000
0
Count
−0.4
−0.2
0.0
0.2
0.4
avg_precipitation

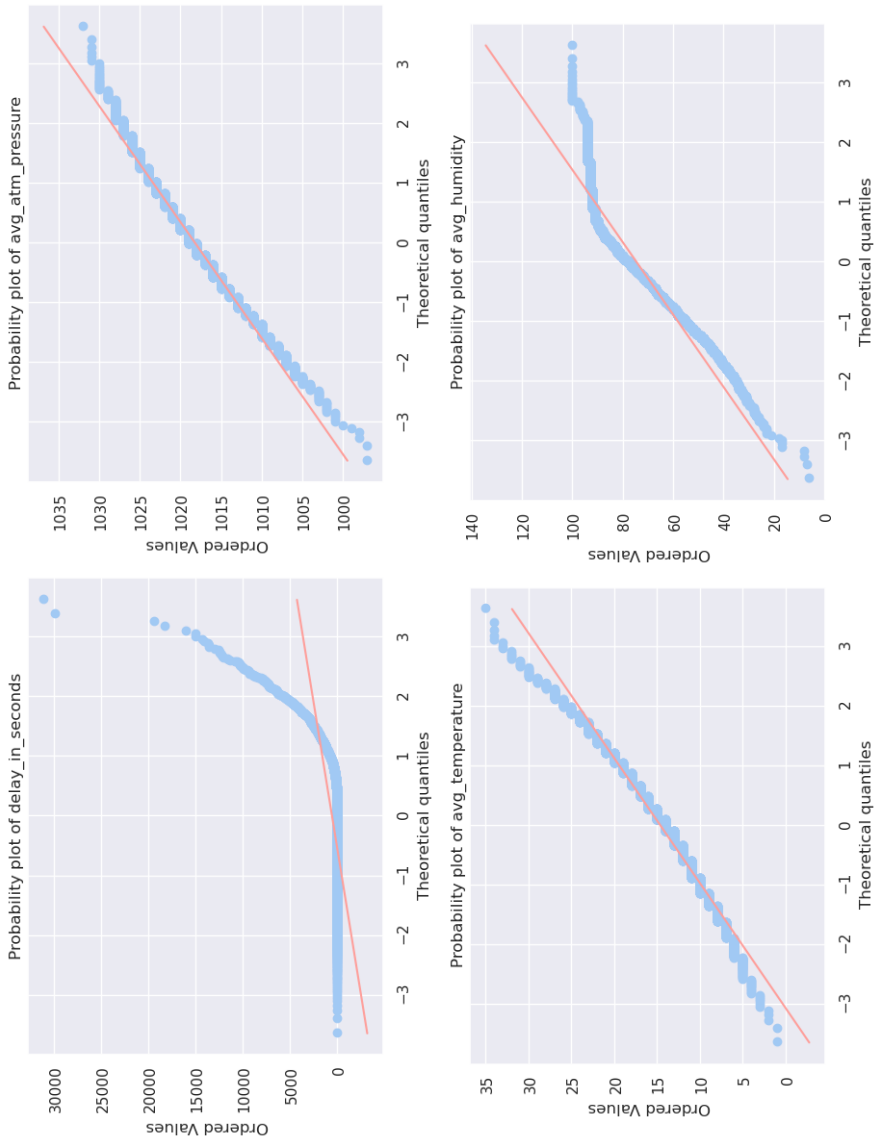

25000
20000
15000
10000
5000
0
Count
UNDEFINED
MAJOR
MODERATE
magnitude_of_delay


10000
8000
6000
4000
2000
0
Count
DARK
LIGHT
LOW_LIGHT
luminosity


35
30
25
20
15
10
5
0
avg_temperature
None
Low
Very_High
High
Medium
incidents

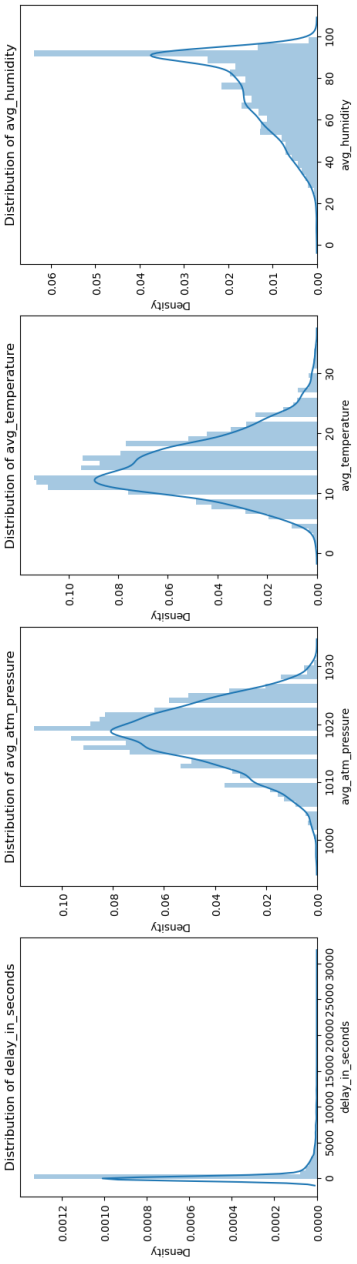

incidents freq
0
1000
2000
delay_in_seconds
0
5000
10000
15000
20000
25000
30000
avg_atm_pressure
1000
1005
1010
1015
1020
1025
1030
incidents
Medium
High
Very_High
Low
None
incidents freq
0
2000
4000

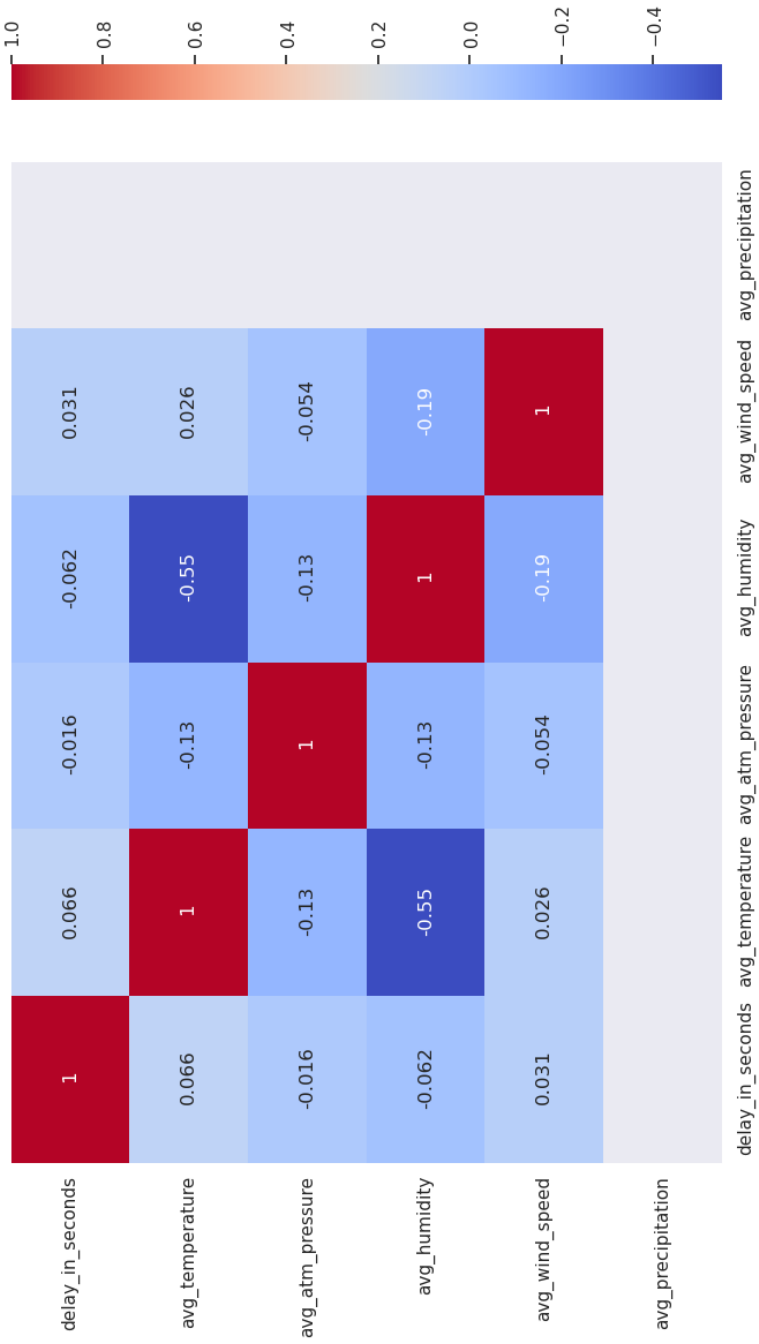

incidents freq
0
1000
2000
avg_temperature
0
5
10
15
20
25
30
35
avg_humidity
20
40
60
80
100
incidents
Medium
High
Very_High
Low
None
incidents freq
0
500
1000

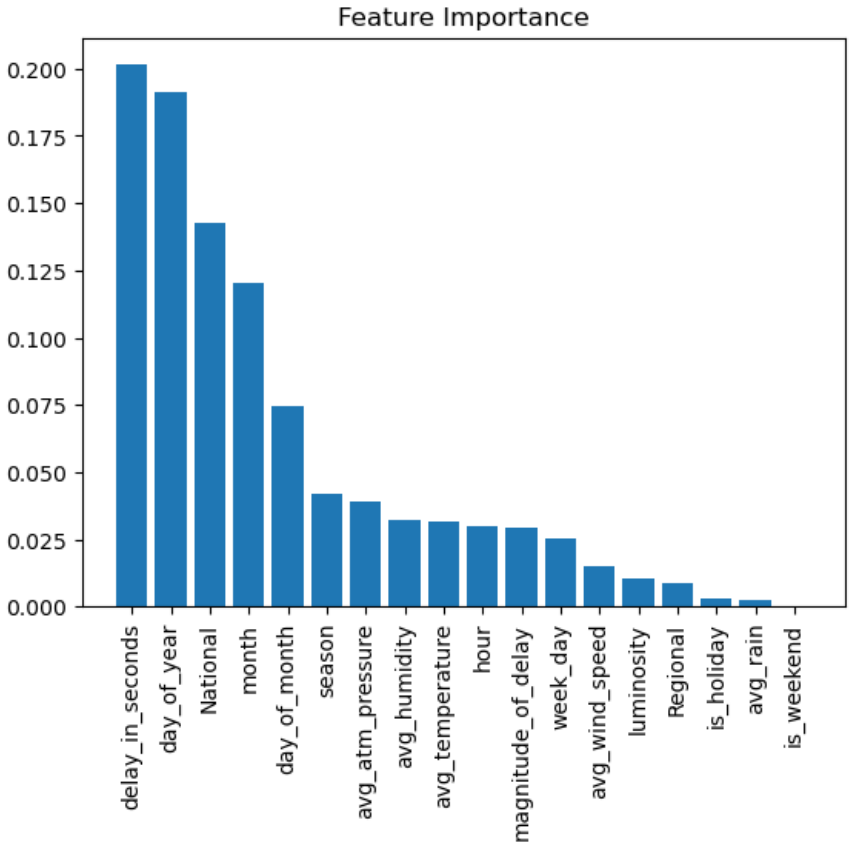

Probability plot of avg_atm_pressure
Ordered Values
Theoretical quantiles
Probability plot of avg_humidity
Ordered Values
Theoretical quantiles
Probability plot of delay_in_seconds
Ordered Values
Theoretical quantiles
Probability plot of avg_temperature
Ordered Values
Theoretical quantiles

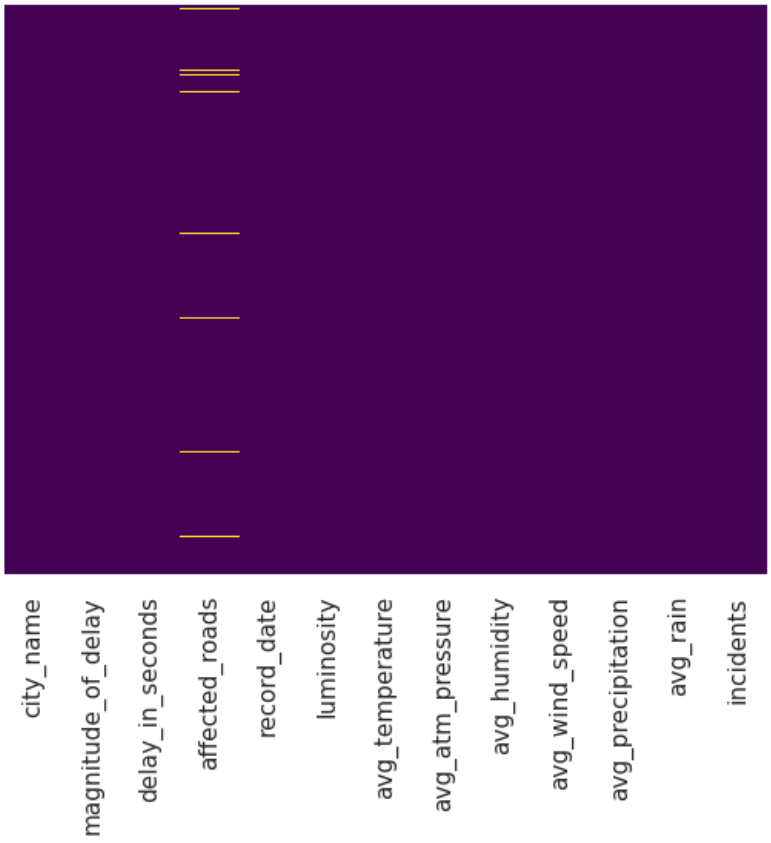

avg_precipitation
avg_wind_speed
avg_humidity
avg_atm_pressure
avg_temperature
delay_in_seconds
avg_precipitation
avg_wind_speed
avg_humidity
avg_atm_pressure
avg_temperature
delay_in_seconds


30000
25000
20000
15000
10000
5000
0
delay_in_seconds
avg_temperature
avg_atm_pressure
avg_humidity
avg_wind_speed
avg_precipitation


Distribution of avg_humidity
avg_humidity
Density
Distribution of avg_temperature
avg_temperature
Density
Distribution of avg_atm_pressure
avg_atm_pressure
Density
Distribution of delay_in_seconds
delay_in_seconds
Density


1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
delay_in_seconds
avg_temperature
avg_atm_pressure
avg_humidity
avg_wind_speed
avg_precipitation
1 0.066 -0.016 -0.062 0.031
0.066 1 -0.13 -0.55 0.026
-0.016 -0.13 1 -0.13 -0.054
-0.062 -0.55 -0.13 1 -0.19
0.031 0.026 -0.054 -0.19 1


Feature Importance
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
delay_in_seconds
day_of_year
National
month
day_of_month
season
avg_atm_pressure
avg_humidity
avg_temperature
hour
magnitude_of_delay
week_day
avg_wind_speed
luminosity
Regional
is_holiday
avg_rain
is_weekend


city_name
magnitude_of_delay
delay_in_seconds
affected_roads
record_date
luminosity
avg_temperature
avg_atm_pressure
avg_humidity
avg_wind_speed
avg_precipitation
avg_rain
incidents

6.1 *RoadMap* Temporal das várias submissões

Estas submissões foram um instrumento muito relevante para despiste de alguns casos, que poderiam estar a causar *overfitting*, nomeadamente, o caso do tratamento de *outliers* por remoção. Esta remoção leva a que muitas linhas sejam removidas, fazendo com que se perca muito conhecimento, o modelo não consegue generalizar para todos os casos. Isto gerou uma *accuracy* muito elevada de 0.97%, o que causou a tal suspeita.

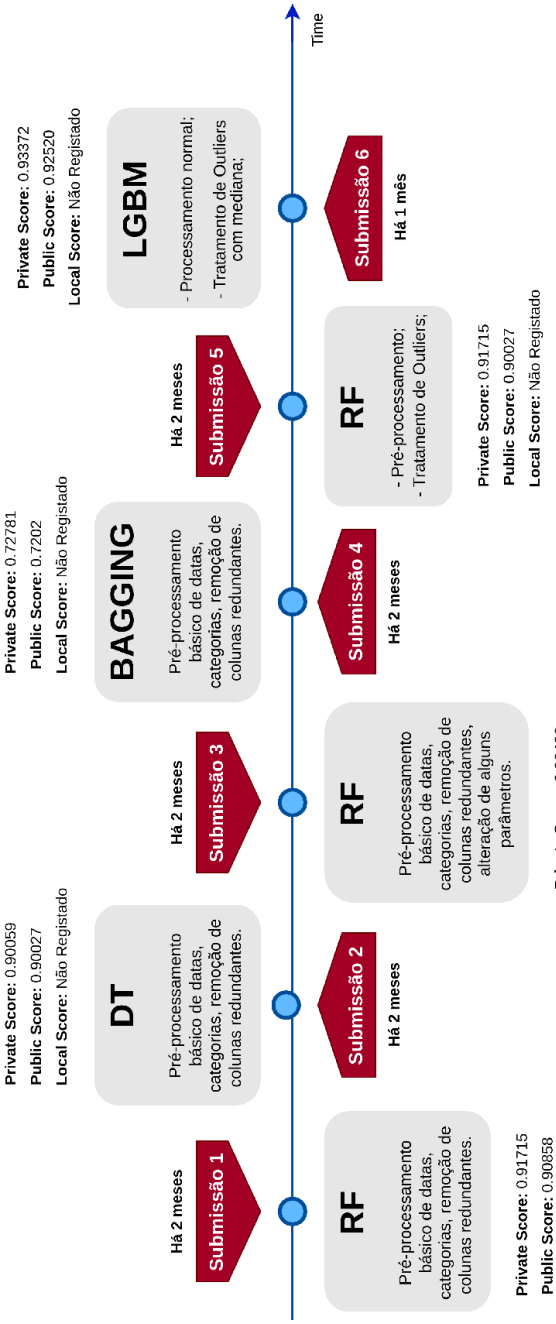
Para além disso, ao longo do processamento, fomos vendo os *scores* que iam sendo obtidos, interpretando se o método utilizado se mostrava adequado face aos resultados obtidos.

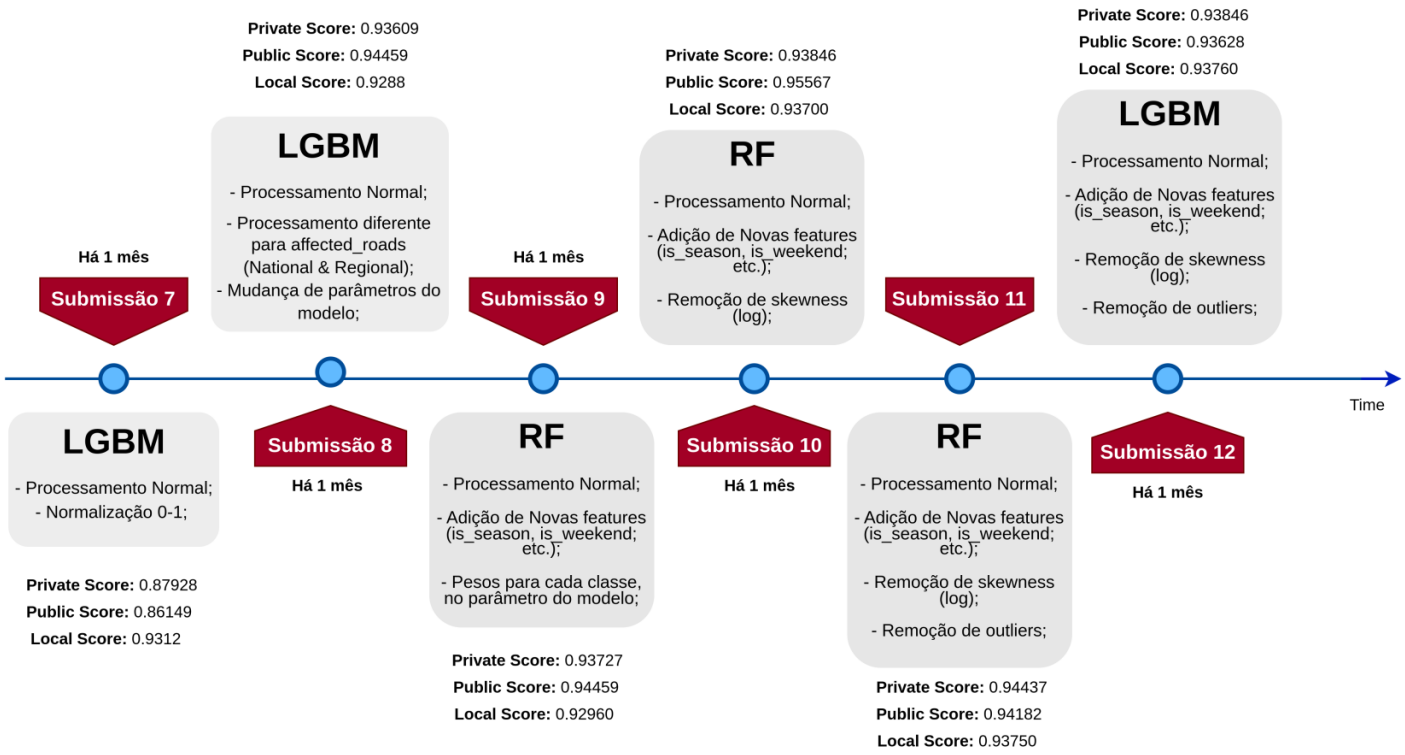
Desta forma, as submissões consistiram como que uma garantia de que o trabalho efetuado, estava coerente e correto.

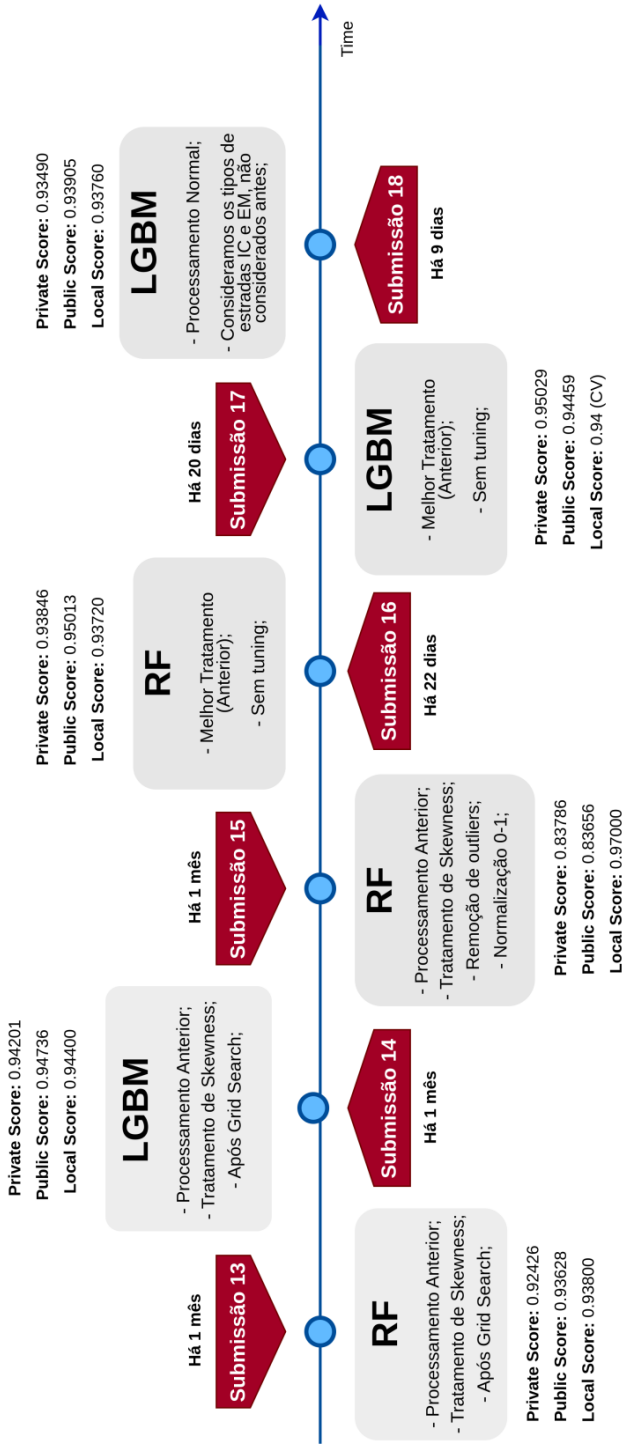
Também é possível observar a generalidade dos cenários e os vários modelos utilizados, ainda que não estejam todos presentes.

A coleta das informações das várias submissões ocorreu no dia 14 de janeiro de 2023, o fluxo temporal tem como referência esse dia.

Seguem as várias submissões.


Time
Há 2 meses
Submissão 1
RF
Pré-processamento básico de datas, categorias, remoção de colunas redundantes.
Private Score: 0.91715
Public Score: 0.90858
Local Score: Não Registado
Há 2 meses
Submissão 2
DT
Pré-processamento básico de datas, categorias, remoção de colunas redundantes.
Private Score: 0.90059
Public Score: 0.90027
Local Score: Não Registado
Há 2 meses
Submissão 3
RF
Pré-processamento básico de datas, categorias, remoção de colunas redundantes, alteração de alguns parâmetros.
Private Score: 0.92426
Public Score: 0.90581
Local Score: Não Registado
Há 2 meses
Submissão 4
BAGGING
Pré-processamento básico de datas, categorias, remoção de colunas redundantes.
Private Score: 0.72781
Public Score: 0.7202
Local Score: Não Registado
Há 2 meses
Submissão 5
RF
- Pré-processamento;
- Tratamento de Outliers;
Private Score: 0.91715
Public Score: 0.90027
Local Score: Não Registado
Há 1 mês
Submissão 6
LGBM
- Processamento normal;
- Tratamento de Outliers com mediana;
Private Score: 0.93372
Public Score: 0.92520
Local Score: Não Registado

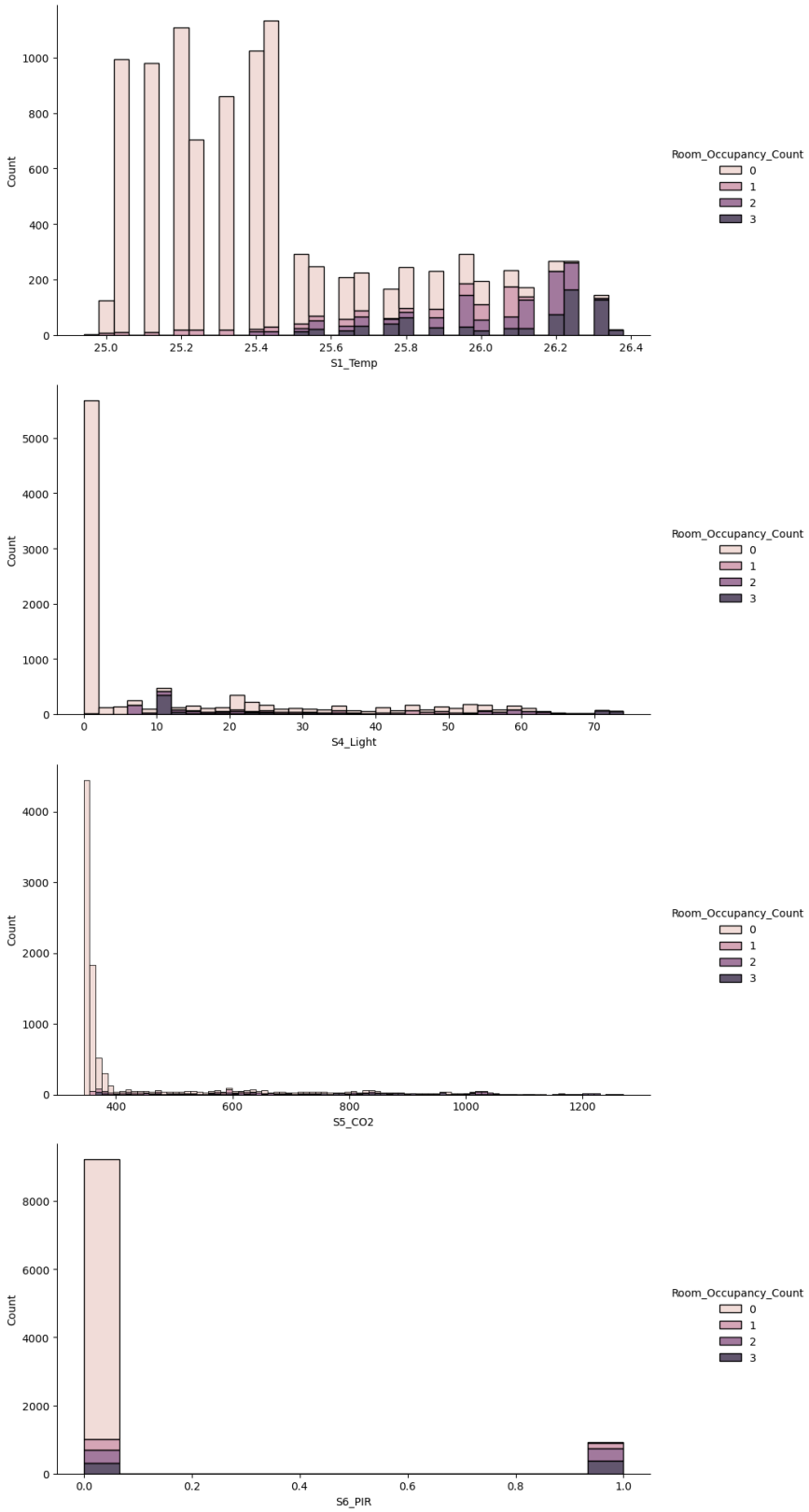

Há 1 mês
Submissão 7
LGBM
- Processamento Normal;
- Normalização 0-1;
Private Score: 0.87928
Public Score: 0.86149
Local Score: 0.9312
Private Score: 0.93609
Public Score: 0.94459
Local Score: 0.9288
LGBM
- Processamento Normal;
- Processamento diferente para affected_roads (National & Regional);
- Mudança de parâmetros do modelo;
Submissão 8
Há 1 mês
Há 1 mês
Submissão 9
RF
- Processamento Normal;
- Adição de Novas features (is_season, is_weekend; etc.);
- Pesos para cada classe, no parâmetro do modelo;
Private Score: 0.93727
Public Score: 0.94459
Local Score: 0.92960
Private Score: 0.93846
Public Score: 0.95567
Local Score: 0.93700
RF
- Processamento Normal;
- Adição de Novas features (is_season, is_weekend; etc.);
- Remoção de skewness (log);
Submissão 10
Há 1 mês
Submissão 11
RF
- Processamento Normal;
- Adição de Novas features (is_season, is_weekend; etc.);
- Remoção de skewness (log);
- Remoção de outliers;
Private Score: 0.94437
Public Score: 0.94182
Local Score: 0.93750
Private Score: 0.93846
Public Score: 0.93628
Local Score: 0.93760
LGBM
- Processamento Normal;
- Adição de Novas features (is_season, is_weekend; etc.);
- Remoção de skewness (log);
- Remoção de outliers;
Submissão 12
Há 1 mês
Time

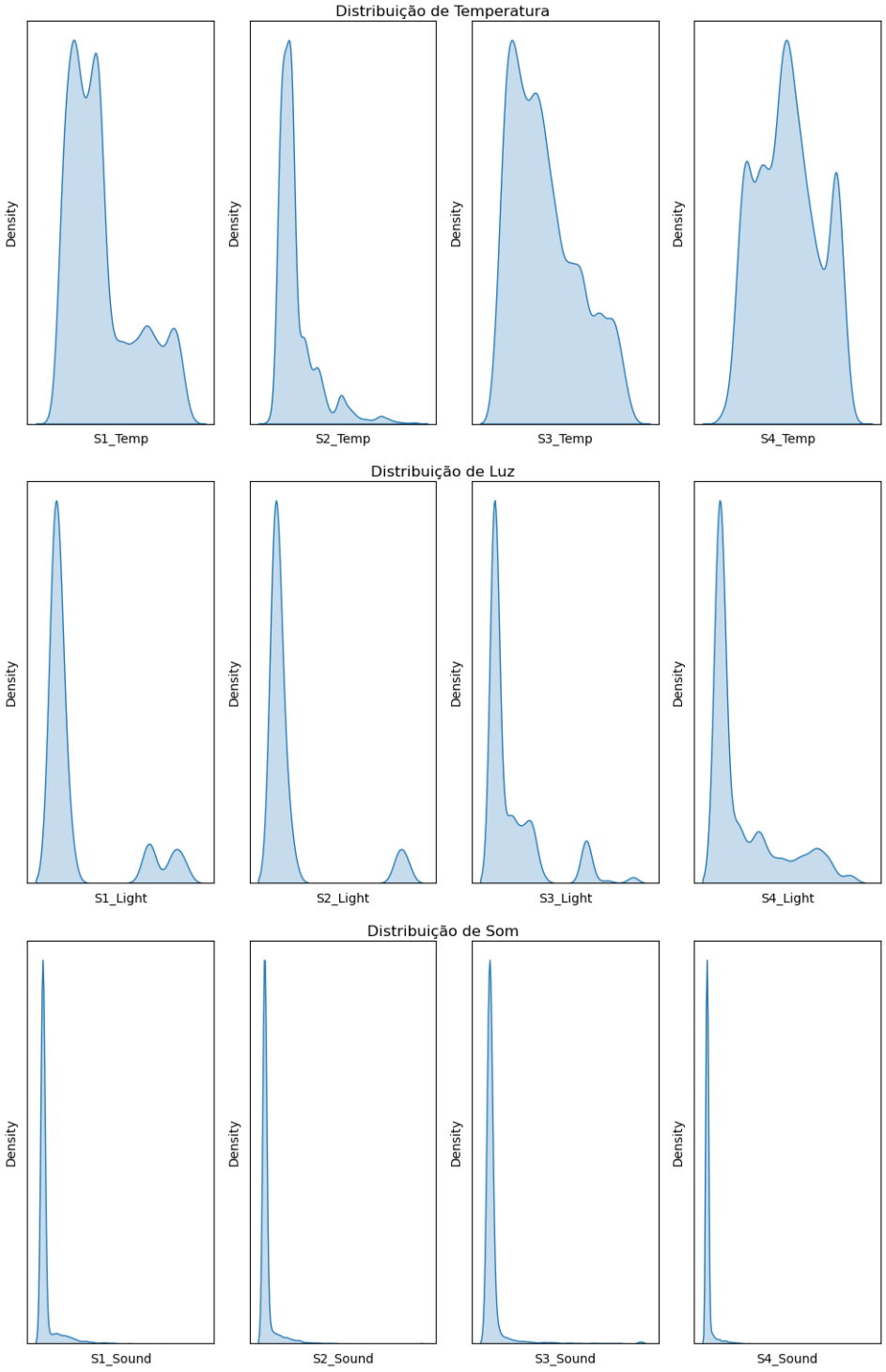

Time
Há 1 mês
Submissão 13
RF
- Processamento Anterior;
- Tratamento de Skewness;
- Após Grid Search;
Private Score: 0.92426
Public Score: 0.93628
Local Score: 0.93800
Há 1 mês
Submissão 14
RF
- Processamento Anterior;
- Tratamento de Skewness;
- Remoção de outliers;
- Normalização 0-1;
Private Score: 0.83786
Public Score: 0.83656
Local Score: 0.97000
Há 22 dias
Submissão 16
RF
- Melhor Tratamento (Anterior);
- Sem tuning;
Private Score: 0.93846
Public Score: 0.95013
Local Score: 0.93720
Há 20 dias
Submissão 17
LGBM
- Melhor Tratamento (Anterior);
- Sem tuning;
Private Score: 0.95029
Public Score: 0.94459
Local Score: 0.94 (CV)
Há 9 dias
Submissão 18
LGBM
- Processamento Normal;
- Consideramos os tipos de estradas IC e EM, não considerados antes;
Private Score: 0.93490
Public Score: 0.93905
Local Score: 0.93760
Há 1 mês
Submissão 15
LGBM
- Processamento Anterior;
- Tratamento de Skewness;
- Após Grid Search;
Private Score: 0.94201
Public Score: 0.94736
Local Score: 0.94400

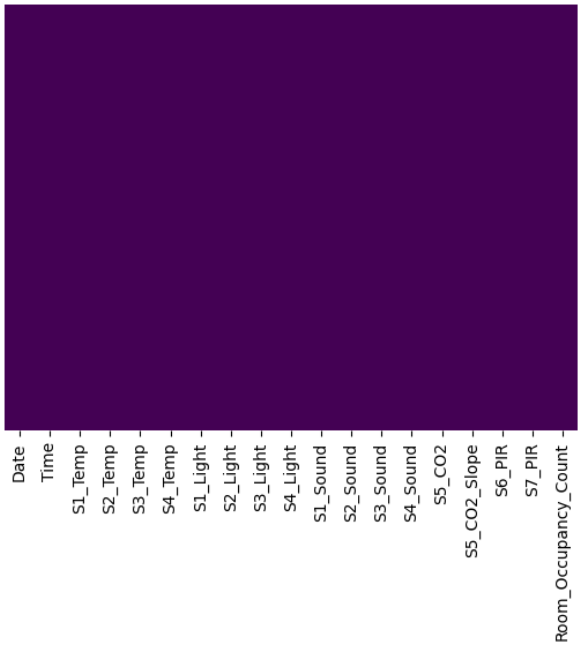
7 Anexos - *Dataset* de Grupo

Atributo	Descrição	DType	Tipo	Exemplo
Date	Data da recolha dos valores captados pelos sensores.	object	Categórico Ordinal	”2018/01/11”
Time	Momento temporal de captação.	object		”09:00:09”
S[1,2,3,4]_Temp	Temperatura captada pelo sensor [1,2,3,4] em graus Celsius.	int64	Numérico Contínuo	”24.94”
S[1,2,3,4]_Light	Luminosidade captada pelo sensor [1,2,3,4] em LUX (unidade SI de fluxo luminoso por unidade de área, ou seja da densidade de intensidade luminosa conhecida por iluminância).			”121”
S[1,2,3,4]_Sound	Som captado pelo sensor 1 em Volts (o que determina a diferença entre sinais de áudio é a sua voltagem).			”0.08”, ”3.16”
S5_CO2	Concentração de CO2 captada pelo sensor 5 em PPM.			”355”
S5_CO2_Slope	Inclinação de CO2 captada pelo sensor 5.			”4.873077”
S[6,7]_PIR	Sensor [6,7] PIR (digital passive infrared) para deteção de movimento (através de infravermelhos).			”0” ou ”1”
Room_Occupancy_Count (target)	Número de pessoas na sala (determinado manualmente por uma pessoa).		Numérico Discreto	”0”, ”4”

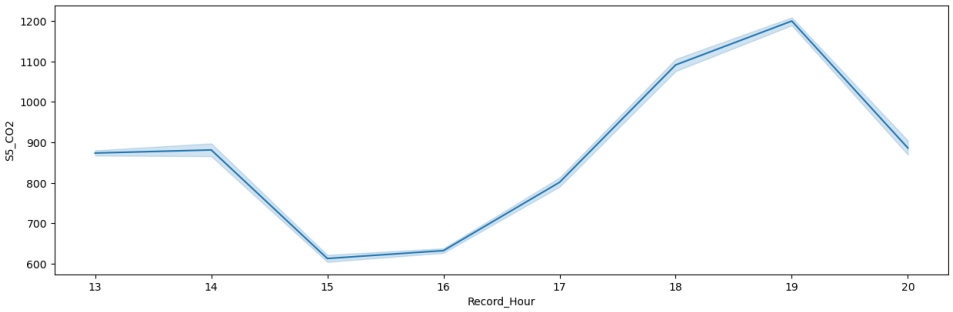
Table 4: Features Target.

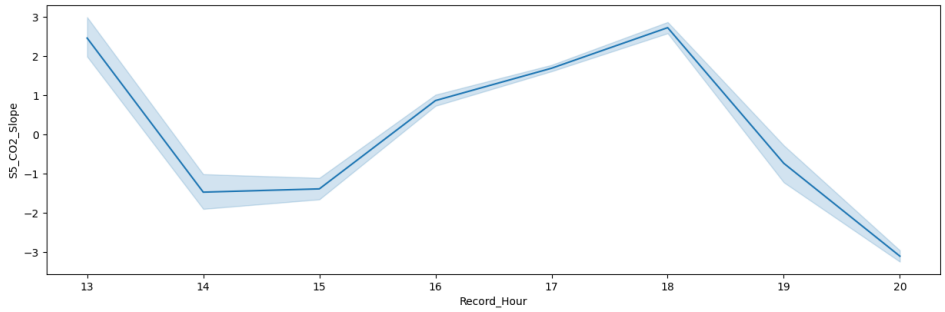

Count
S1_Temp
S4_Light
S5_CO2
S6_PIR
Room_Occupancy_Count
0
1
2
3

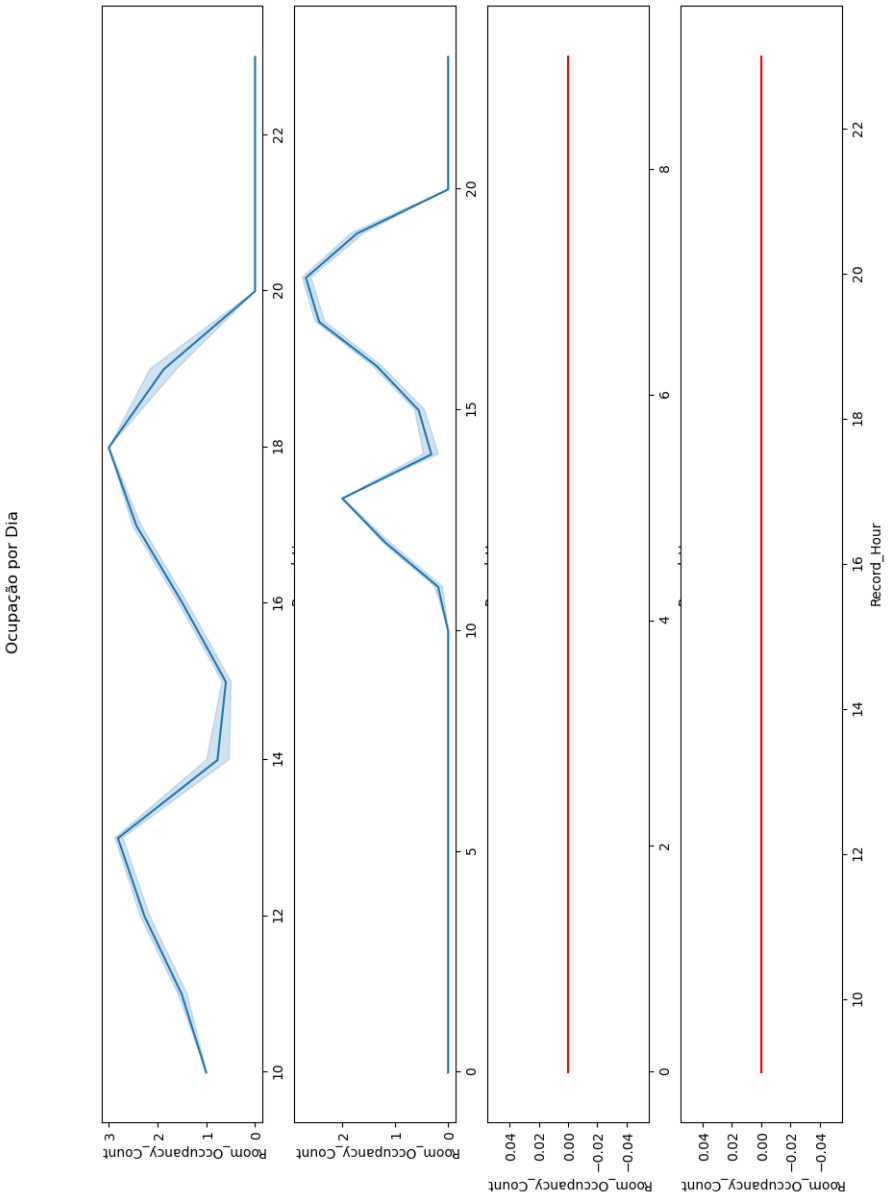

Distribuição de Temperatura
Density
S1_Temp
S2_Temp
S3_Temp
S4_Temp
Distribuição de Luz
S1_Light
S2_Light
S3_Light
S4_Light
Distribuição de Som
S1_Sound
S2_Sound
S3_Sound
S4_Sound

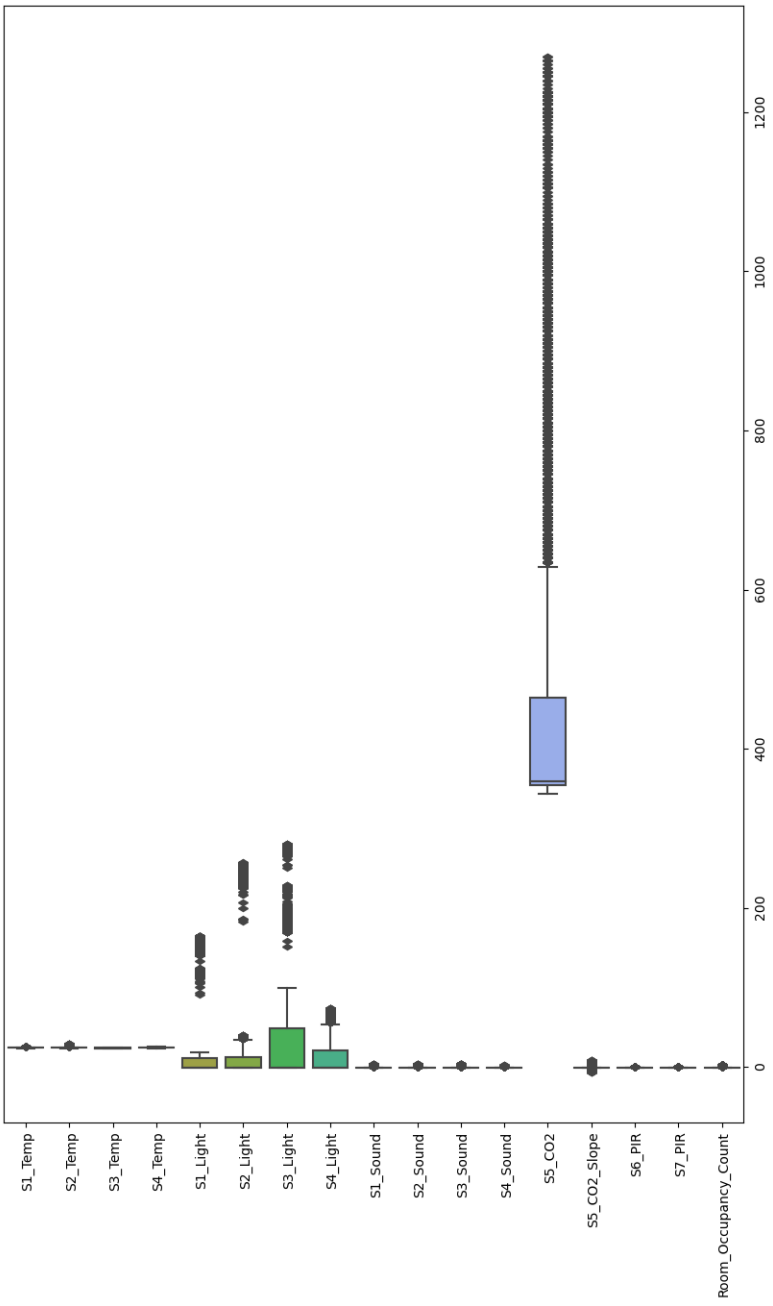


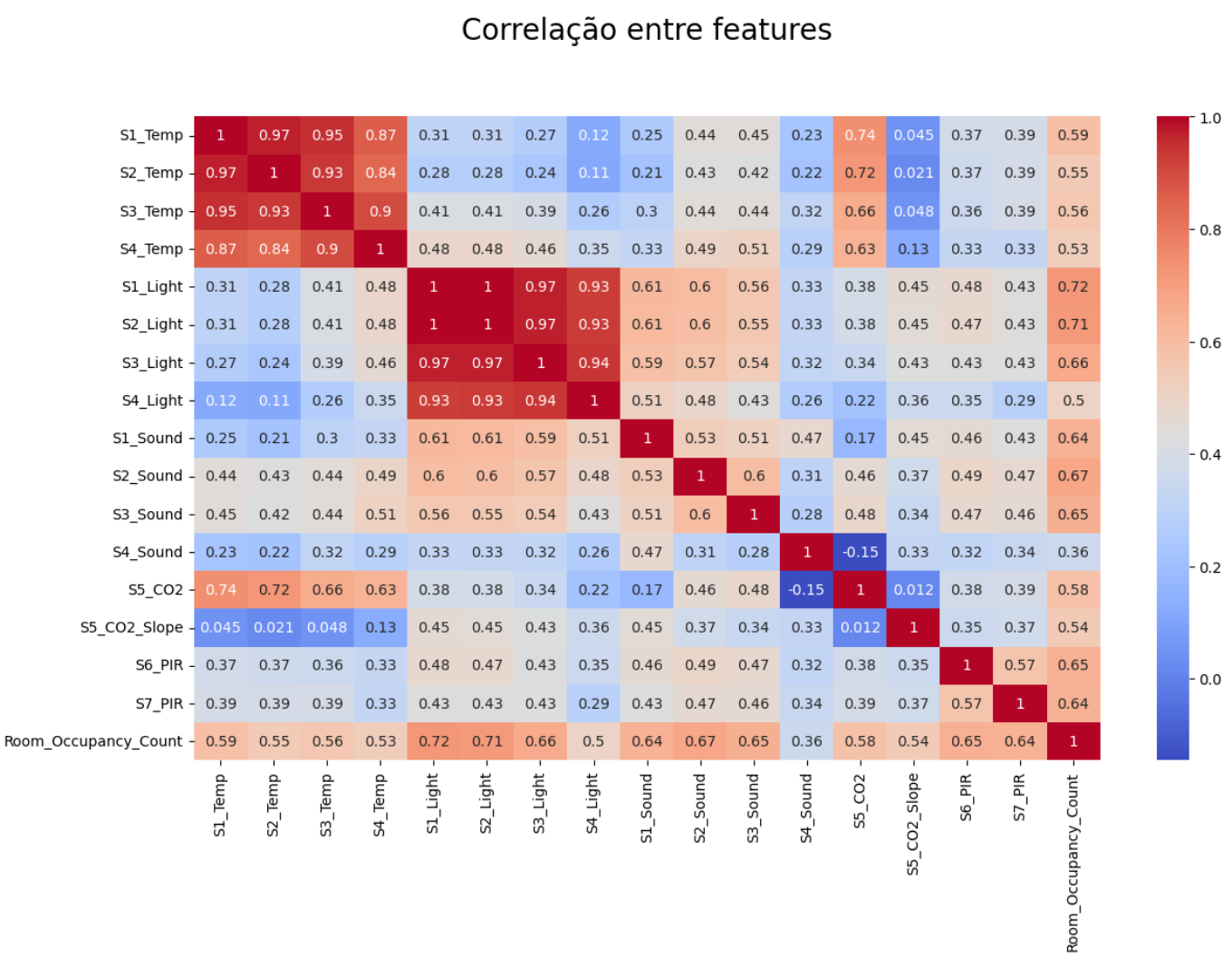
Sensor 5, Dia 1






Ocupação por Dia
Room_Occupancy_Count
Record_Hour


0
200
400
600
800
1000
1200
S1_Temp
S2_Temp
S3_Temp
S4_Temp
S1_Light
S2_Light
S3_Light
S4_Light
S1_Sound
S2_Sound
S3_Sound
S4_Sound
S5_CO2
S5_CO2_Slope
S6_PIR
S7_PIR
Room_Occupancy_Count


Correlação entre features
S1_Temp 1 0.97 0.95 0.87 0.31 0.31 0.27 0.12 0.25 0.44 0.45 0.23 0.74 0.045 0.37 0.39 0.59
S2_Temp 0.97 1 0.93 0.84 0.28 0.28 0.24 0.11 0.21 0.43 0.42 0.22 0.72 0.021 0.37 0.39 0.55
S3_Temp 0.95 0.93 1 0.9 0.41 0.41 0.39 0.26 0.3 0.44 0.44 0.32 0.66 0.048 0.36 0.39 0.56
S4_Temp 0.87 0.84 0.9 1 0.48 0.48 0.46 0.35 0.33 0.49 0.51 0.29 0.63 0.13 0.33 0.33 0.53
S1_Light 0.31 0.28 0.41 0.48 1 1 0.97 0.93 0.61 0.6 0.56 0.33 0.38 0.45 0.48 0.43 0.72
S2_Light 0.31 0.28 0.41 0.48 1 1 0.97 0.93 0.61 0.6 0.55 0.33 0.38 0.45 0.47 0.43 0.71
S3_Light 0.27 0.24 0.39 0.46 0.97 0.97 1 0.94 0.59 0.57 0.54 0.32 0.34 0.43 0.43 0.43 0.66
S4_Light 0.12 0.11 0.26 0.35 0.93 0.93 0.94 1 0.51 0.48 0.43 0.26 0.22 0.36 0.35 0.29 0.5
S1_Sound 0.25 0.21 0.3 0.33 0.61 0.61 0.59 0.51 1 0.53 0.51 0.47 0.17 0.45 0.46 0.43 0.64
S2_Sound 0.44 0.43 0.44 0.49 0.6 0.6 0.57 0.48 0.53 1 0.6 0.31 0.46 0.37 0.49 0.47 0.67
S3_Sound 0.45 0.42 0.44 0.51 0.56 0.55 0.54 0.43 0.51 0.6 1 0.28 0.48 0.34 0.47 0.46 0.65
S4_Sound 0.23 0.22 0.32 0.29 0.33 0.33 0.32 0.26 0.47 0.31 0.28 1 -0.15 0.33 0.32 0.34 0.36
S5_CO2 0.74 0.72 0.66 0.63 0.38 0.38 0.34 0.22 0.17 0.46 0.48 -0.15 1 0.012 0.38 0.39 0.58
S5_CO2_Slope 0.045 0.021 0.048 0.13 0.45 0.45 0.43 0.36 0.45 0.37 0.34 0.33 0.012 1 0.35 0.37 0.54
S6_PIR 0.37 0.37 0.36 0.33 0.48 0.47 0.43 0.35 0.46 0.49 0.47 0.32 0.38 0.35 1 0.57 0.65
S7_PIR 0.39 0.39 0.39 0.33 0.43 0.43 0.43 0.29 0.43 0.47 0.46 0.34 0.39 0.37 0.57 1 0.64
Room_Occupancy_Count 0.59 0.55 0.56 0.53 0.72 0.71 0.66 0.5 0.64 0.67 0.65 0.36 0.58 0.54 0.65 0.64 1
S1_Temp
S2_Temp
S3_Temp
S4_Temp
S1_Light
S2_Light
S3_Light
S4_Light
S1_Sound
S2_Sound
S3_Sound
S4_Sound
S5_CO2
S5_CO2_Slope
S6_PIR
S7_PIR
Room_Occupancy_Count
1.0
0.8
0.6
0.4
0.2
0.0

7.1 Autores

Daniel Xavier

Duarte Cerquido

Henrique Alvelos

Diogo Rebelo