Prevendo a resistência à compressão do concreto com técnicas de machine learning

Pedro Bernardino Alves Moreira

Abril 22, 2020

Resumo

A resistência à compressão é a principal característica do concreto. A previsão correta desse parâmetro significa redução de custo e tempo. Esse trabalho construiu modelos de previsão em 6 idades diferentes de amostras de concreto (3, 7, 14, 28, 56, e 100 dias). Foi utilizado um conjunto de dados obtido em estudos anteriores, um total de 1030 amostras, com 9 variáveis: resistência à compressão, idade e 7 ingredientes (água, cimento, agregado miúdo, agregado graúdo, cinza volante, escória granulada de alto forno e superplastificantes). Outras 6 variáveis foram adicionadas para representar as proporções dos principais ingredientes em cada amostra (água/cimento, agregado miúdo/cimento, agregado graúdo/cimento, agregado miúdo/agregado graúdo, água/agregado graúdo e água/agregado miúdo). Os modelos de previsão foram desenvolvidos em linguagem *R*, utilizando o pacote *caret* com o algoritmo *Parallel Random Forest* e técnica de validação cruzada repetida para otimização dos parâmetros. Os resultados foram satisfatórios e compatíveis com outros estudos utilizando o mesmo conjunto de dados. O modelo mais importante, de 28 dias, obteve *RMSE* de 4,717. O modelo de 3 dias obteve o melhor resultado, *RMSE* de 3,310. O pior resultado foi o modelo de 56 dias, com *RMSE* de 5,939. O trabalho mostrou que a resistência à compressão do concreto pode ser prevista. A escolha de criar um modelo para cada idade, ao invés de utilizar a idade como característica para previsão, permitiu chegar a resultados compatíveis com os dados disponíveis de cada idade. Foi uma alternativa promissora, visto que bons resultados foram atingidos treinando com apenas um algoritmo. Esse trabalho facilita a exploração e novos esforços para previsão da resistência à compressão do concreto, ele pode ser replicado utilizando diferentes algoritmos ou a combinação de diversos.

Sumário

Lista de figuras

Lista de tabelas

1 Introdução

A resistência à compressão é a principal característica do concreto, medida por testes de padrões internacionais que consistem na quebra de corpos de prova. A medição aos 28 dias é obrigatória e representa a classe do concreto. Saber com antecêdencia qual o resultado será obtido para uma determinada idade, a partir das proporções de seus ingredientes, é de grande interesse para os fabricantes de concreto, construtoras e engenheiros civis.

Essa resistência à compressão é uma função não linear de seus ingredientes e idade, tornando difícil o estabelecimento de uma fórmula analítica, apesar de algumas fórmulas já haveram sido propostas. Hasan (2011) propós um modelo matemático para prever a partir dos resultados de testes de 7 e 14 dias, e Kabir (2012) a partir de 7 dias. Porém técnicas de machine learning podem ser utilizadas para modelar essa característica a partir de dados reais de amostras, utilizando apenas os ingredientes.

Muitos estudos anteriores utilizam o mesmo conjunto de dados utilizado por Yeh (1998) para prever a resistência à compressão do concreto. Alshamiri (2020) obteve bons resultados com a técnica de regularized extreme learning machine (RELM), e Hameed (2020) obteve resultados ainda melhores com a técnica de Artificial Neural Networks e cross-validation. Esse conjunto de amostras é tão conhecido que há ainda páginas na internet de estudos não publicados que o utilizam e possuem bons resultados, como Abban (2016), Raj (2018), Modukuru (2020) e Pierobon (2018). Ao final do trabalho os resultados encontrados são comparados aos trabalhos citados aqui.

Diferente dos estudos anteriores com esse conjunto de amostras, este trabalho faz a preparação dos dados de forma diferente. A idade do concreto é a variável mais singular que contribui para sua resistência à compressão, por esse motivo, a idade é tratada separadamente nos modelos de machine learning, criando modelos para cada faixa de idade.

2 Metodologia

2.1 Materiais utilizados

A metodologia foi realizada utilizando o software RStudio (RStudio Team 2020), ambiente virtual integrado para desenvolvimento de código na linguagem *R* (R Core Team 2020). Ao longo do processo, todo código executado foi documentado na mesma ordem de sua execução no Apêndice 3, e foi sempre realizada referência aos códigos ao longo do texto. Todas as informações relevantes relacionados ao sistema operacional e pacotes instalados foram apresentados no Apêndice 1. Além disso foi criado um repositorio online e de código aberto no *Github*, abrigando todo o código utilizado para gerar esse trabalho, o link foi diponibilizado no Apêndice 2.

2.2 Reprodutibilidade

Para garantir a reprodutibilidade, sempre que houve algum código que podesse utilizar operações probabilísticas, foi definido um *seed* antes da sua execução, garantindo que quando rodado em outra máquina, com a mesma versão de *R* e o mesmo *seed*, chegue ao mesmo resultado. Os *seeds* podem ser conferidos ao longo do apêndice 3 ou diretamente no *Github*.

2.3 Obtenção dos dados

O download dos dados foi realizado no website da Universidade da California Irvine ("Concrete Compressive Strength Data Set" 2008) (8.1.1). No total são 1030 amostras com 9 colunas. As amostras foram renomeadas e foi adicionado uma coluna de id para facilitar na manipulação dos dados (8.1.2). As colunas foram reordenadas para colocar a nova coluna id em primeira posição (8.1.3). As primeiras amostras podem ser visualizadas na tabela 1.

Tabela 1: Primeiras 6 amostras

ID	Cimento kg/m^3	E.G.A.F kg/m^3	C.Volante kg/m^3	Água kg/m^3	Superp. kg/m^3	A.Graúdo kg/m^3	A.Miúdo kg/m^3	Dia	Comp.Str. MPa
1	540.0	0.0	0	162	2.5	1040.0	676.0	28	79.99
2	540.0	0.0	0	162	2.5	1055.0	676.0	28	61.89
3	332.5	142.5	0	228	0.0	932.0	594.0	270	40.27
4	332.5	142.5	0	228	0.0	932.0	594.0	365	41.05
5	198.6	132.4	0	192	0.0	978.4	825.5	360	44.30
6	266.0	114.0	0	228	0.0	932.0	670.0	90	47.03

2.4 Preparação dos dados

A preparação dos dados consistiu em transformar o conjunto de amostra afim de manter apenas dados relevantes para os estudos subsequêntes. Foram retirandos dados considerados irrelevantes ou que com potencial de adicionar um ruído indesejavel nas análises. Além disso, os dados relevântes foram transformados para melhor se enquadrar para os estudos nas próximas etapas.

2.4.1 Limpeza inicial dos dados

Inicialmente foram observadas a existência de 25 amostras duplicadas que foram retiradas, resultando em um novo total de 1005 amostras (8.2.1).

Os dados apresentam as variaveis nas colunas e amostras nas linhas. Porém foi verificado que algumas amostras são identicas em proporções de ingredientes, alterando apenas o valor da idade e resistência à compressão, por exemplo as amostras 653, 654, 678 e 681, mostradas na tabela 2.

Tabela 2: Amostras com a mesma composição

ID	Cimento kg/m^3	E.G.A.F kg/m^3	C.Volante kg/m^3	Água kg/m^3	Superp. kg/m^3	A.Graúdo kg/m^3	A.Miúdo kg/m^3	Dia	Comp.Str. MPa
653	102	153	0	192	0	887	942	3	4.57
678	102	153	0	192	0	887	942	7	7.68
681	102	153	0	192	0	887	942	28	17.28
654	102	153	0	192	0	887	942	90	25.46

Além disso, também existem amostras com os mesmos valores e proporções de ingredientes, mas com resistência à compressão diferente, provavelmente devido a diferenças na execução. É o caso por exemplo das amostras 472, 473 e 474, mostradas na tabela 3.

Para facilitar a analise das amostras, todas as amostras que são iguais em relação aos ingredientes, foram atribuidos o mesmo *id*. Além disso, como a resistência à compressão aos 28 dias é o parâmetro para determinar a classe do concreto, foi mantido apenas os elementos que contenham esse dia entre suas amostras. No caso das amostras iguais mas com resultado diferentes de resistência à compressão, foi calculado a média dos valores. Após todas essas alterações (8.2.4), o novo total de amostras foi reduzido para 970, contendo 416 configurações diferentes das proporções de ingredientes.

Tabela 3: Amostras iguais com resultados diferentes

ID	Cimento kg/m^3	E.G.A.F kg/m^3	C.Volante kg/m^3	Água kg/m^3	Superp. kg/m^3	A.Graúdo kg/m^3	A.Miúdo kg/m^3	Dia	Comp.Str. MPa
472	446	24	79	162	11.6	967	712	28	57.03
473	446	24	79	162	11.6	967	712	28	44.42
474	446	24	79	162	11.6	967	712	28	51.02

O resultado pode ser conferido na tabela 4. Todas as amostras com configurações iguais de ingredientes possuem o mesmo id, e quando possuiam resultados diferentes para os mesmos dias, foram transformadas em apenas uma amostra, com a média aritmética na resistência à compressão.

Tabela 4: Amostras anteriores após processamento

ID	Cimento kg/m^3	E.G.A.F kg/m^3	C.Volante kg/m^3	Água kg/m^3	Superp. kg/m^3	A.Graúdo kg/m^3	A.Miúdo kg/m^3	Dia	Comp.Str. MPa
472	446	24	79	162	11.6	967	712	28	50.82
653	102	153	0	192	0.0	887	942	3	4.57
653	102	153	0	192	0.0	887	942	7	7.68
653	102	153	0	192	0.0	887	942	28	17.28
653	102	153	0	192	0.0	887	942	90	25.46

2.4.2 Seleção das idades

Como descrito anteriormente, a principal idade para análise da resistência à compressão é aos 28 dias, mas as outras idades também podem ser utilizadas para construir modelos de previsão. Porém é necessário verificar o quanto relevante os dados dessas outras idades são. Iniciando pela distribuição das amostras em relação a cada idade (8.2.6) mostrado na figura 1.

Foi observado que as idades de 90, 91 e 100 dias provavelmente representam extremos entre si das configurações de ingredientes, uma vez que são idades relativamente próximas porém com valores muito diferentes, especialmente para 90 e 91.

Essa hipótese foi verificada utilizando o método de análise de componente principal, aplicado às amostras dessas 3 idades (8.2.7). A fígura 2 mostra como as amostras se relacionam umas com as outras (quais são parecidas ou diferentes) e revelou como cada variável contribui para a análise. As duas primeiras dimensões representam 37% e 24% respectivamente da variância.

Outro ponto importante considerado, da própria natureza do concreto, é o fato da taxa de crescimento de sua resistência à compressão diminuir com o tempo, chegando a um certo valor de establidade. A figura 3 mostra a resistência à compressão ao longo dos dias para amostras com mais de 5 dados, ou seja, dados disponiveis para no mínimo 6 idades distintas (8.2.8).

Pelos motivos apresentados nas figuras 1, 2 e 3, foi considerado que as idades de 90, 91 e 100 dias podem ser agrupadas para melhorar a leitura e diminuir o ruído das amostras. Elas foram convertidas para o mesmo valor, que no caso foi escolhido a idade de 100 dias (8.2.9), pois como mostrado na figura 3, a resistência apenas aumenta, logo aos 100 dias a resistência à compressão será maior ou igual ao valor de 90 ou 91 dias.

Figura 1: Boxplot - Resistência à compressão (MPa) vs idade (dias)

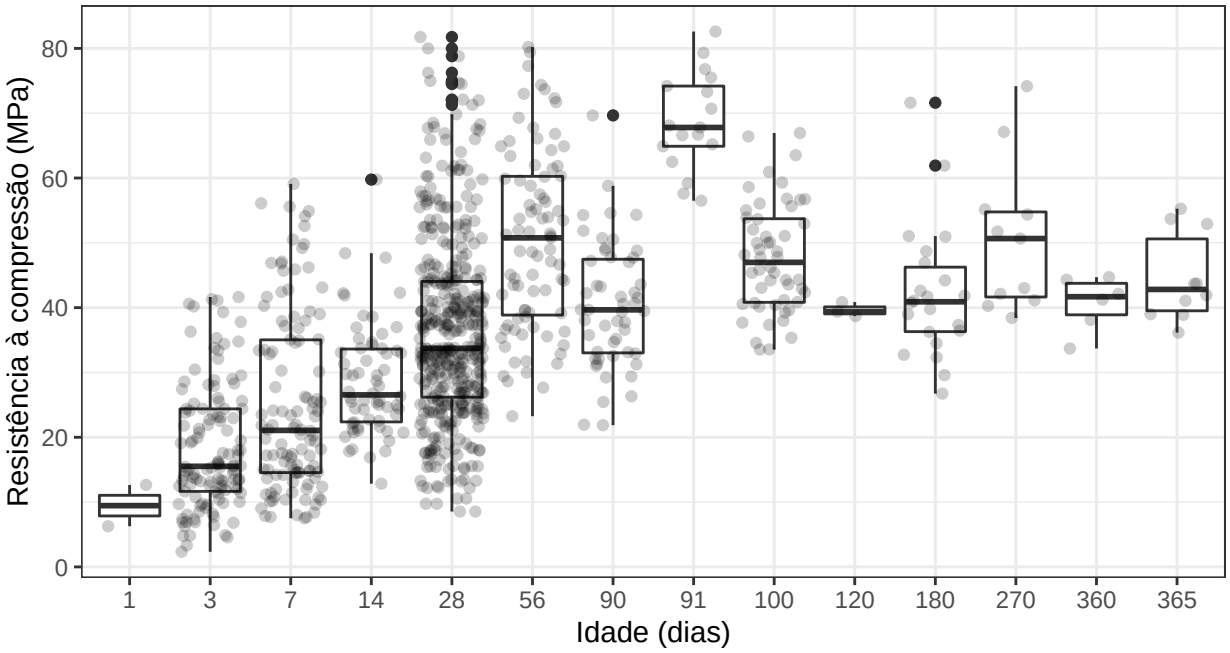


Figura 2: Análise componente principal - 90, 91 e 100 dias

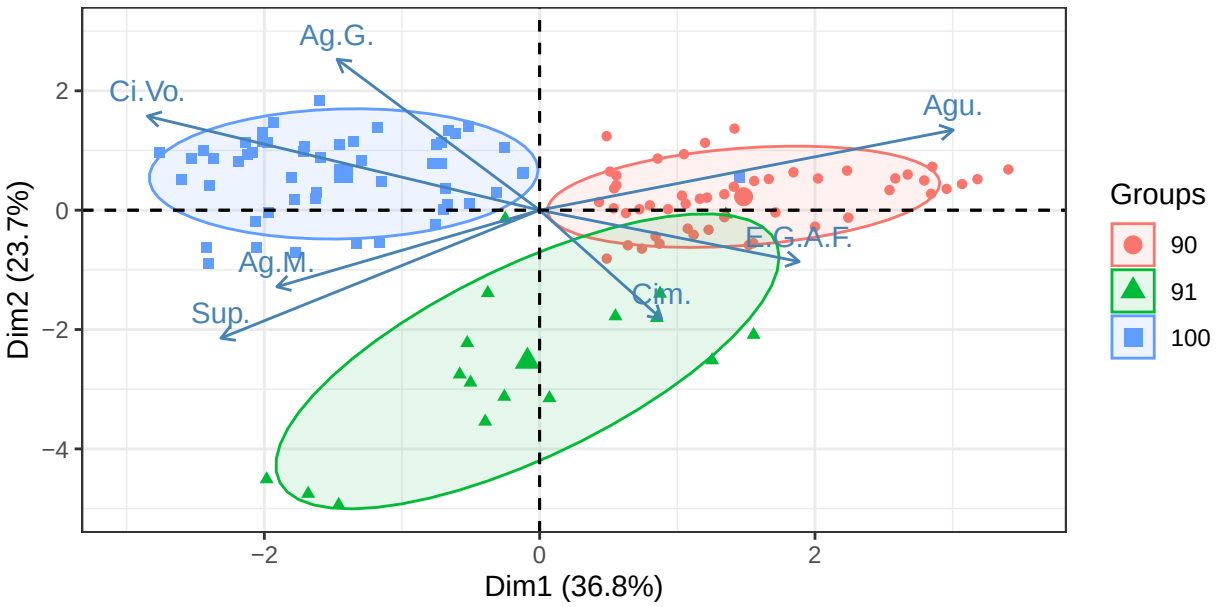
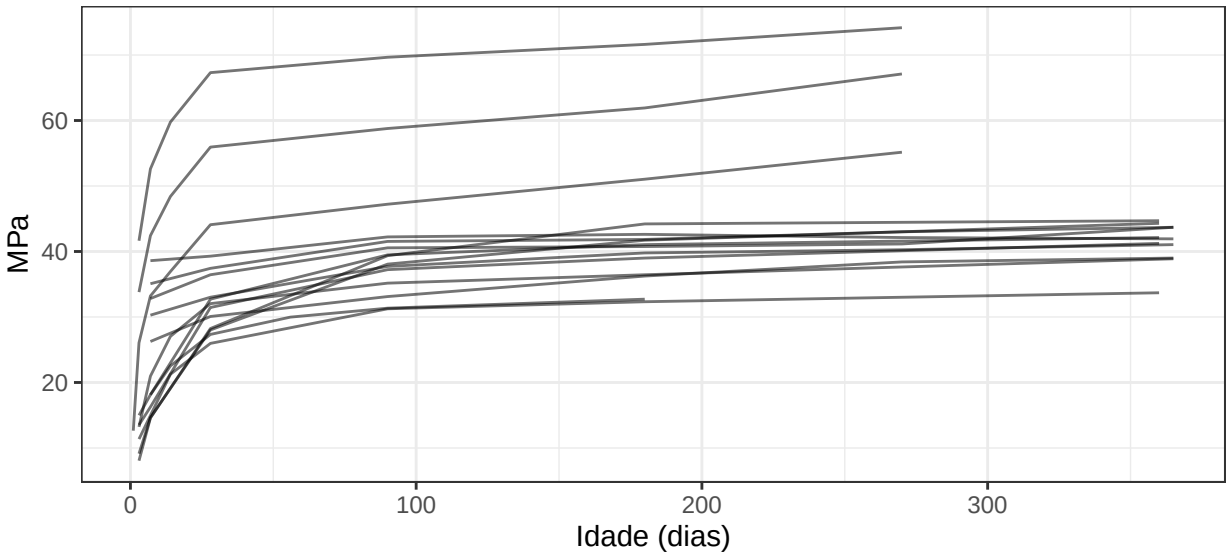
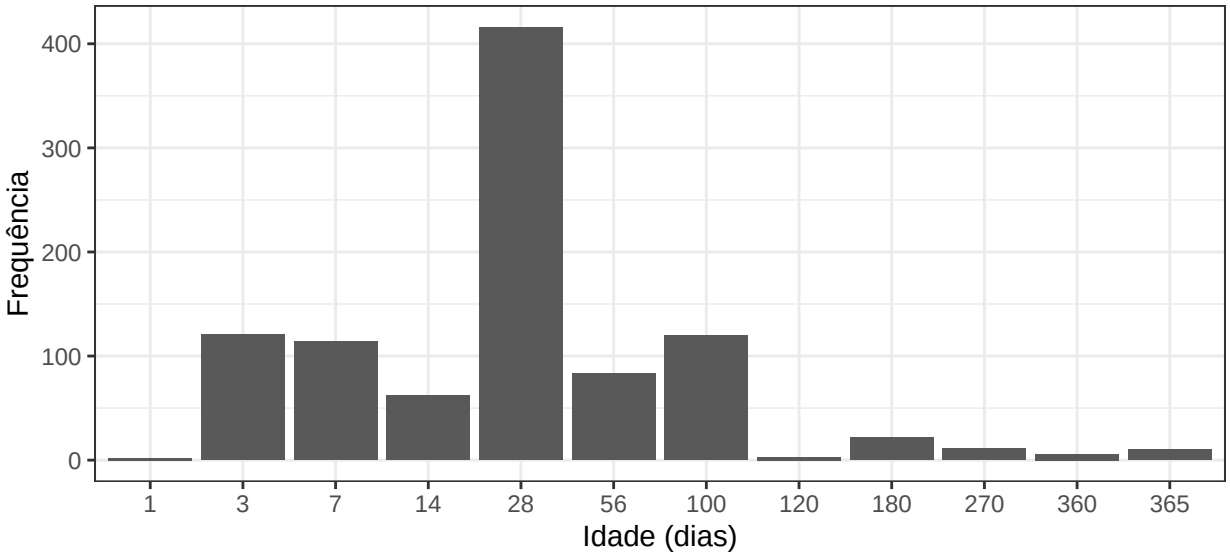


Figura 3: Resistência à compressão ao longo do tempo



Mais um tópico analisado na seleção das idades foi a frequência observada de cada valor de idade após essa transformação dos 90, 91 em 100 dias, mostrada na figura 4. Alguns valores de dias apresentam concentrações muito baixas de amostras, correndo o risco de prejudicar mais do que ajudar na criação dos modelos, logo elas foram removidas (8.2.11). O critério adotado foi manter apenas idades com frequência maior que 50, ou seja, apenas os valores de 3, 7, 14, 28, 56 e 100 dias.

Figura 4: Frequência das idades



2.4.3 Reorganização dos dados

As amostras foram agrupadas para manter apenas uma amostra distinta de cada conjunto de configuração das proporções dos ingredientes, adicionando novas variáveis/colunas para a resistência em cada idade

(8.2.12). O resultado nas primeiras amostras após esse processamento é mostrado na tabela 5.

Tabela 5: Primeiras 6 amostras reorganizadas

ID	Cimento kg/m^3	E.G.A.F. kg/m^3	C.Volante kg/m^3	Água kg/m^3	Superp. kg/m^3	A.Graúdo kg/m^3	A.Miúdo kg/m^3	3 dias MPa	7 dias MPa	14 dias MPa	28 dias MPa	56 dias MPa	100 dias MPa
1	540.0	0.0	0	162	2.5	1040.0	676.0				79.99		
2	540.0	0.0	0	162	2.5	1055.0	676.0				61.89		
3	332.5	142.5	0	228	0.0	932.0	594.0		30.28		33.02		37.72
5	198.6	132.4	0	192	0.0	978.4	825.5	9.13	14.64		28.02		38.07
6	266.0	114.0	0	228	0.0	932.0	670.0				45.85		47.03
7	380.0	95.0	0	228	0.0	932.0	594.0		32.82		36.45		40.56

O número de amostras e amostras distintas após toda essa manipulação permaneceu o mesmo, um total de 416 (8.2.14).

2.4.4 Adicionando novas variáveis

Para finalizar a preparação dos dados, novas colunas foram adicionadas ao conjunto de amostras (8.2.15). Iniciando pela classe do concreto, por exemplo se a resistência à compressão está entre 25 e 30, recebe a classe *C25*. A inclusão da classe foi importânte pois a resistência em MPa é uma variável contínua, que será utilizada nos modelos de regressão, mas a classe como variável discreta pode fornecer outro ângulo de visualização dos dados. Também foi adicionado o traço aproximado do concreto, que representa as proporções de agregados (miúdo e graúdo) para o cimento. Outras proporções entre os principais ingredientes também foram adicionadas. As novas variáveis são apresentadas na tabela 6.

Tabela 6: Novas variáveis

ID	Classe	Traço aproximado	Água / Cimento	A.Miúdo / Cimento	A.Graúdo / Cimento	A.Miúdo / A.Graúdo	Água / A.Graúdo	Água / A.Miúdo
1	C75	1:1:2	0.3000	1.2519	1.9259	0.6500	0.1558	0.2396
2	C60	1:1:2	0.3000	1.2519	1.9537	0.6408	0.1536	0.2396
3	C30	1:2:3	0.6857	1.7865	2.8030	0.6373	0.2446	0.3838
5	C25	1:4:5	0.9668	4.1566	4.9265	0.8437	0.1962	0.2326
6	C45	1:3:4	0.8571	2.5188	3.5038	0.7189	0.2446	0.3403
7	C35	1:2:2	0.6000	1.5632	2.4526	0.6373	0.2446	0.3838

2.5 Visualização dos dados

Para avaliar a necessidade de mais manipulações antes da construção dos modelos, nesta etapa as 416 amostras já processadas foram visualizadas e analisadas.

2.5.1 Estatistica descritiva

A tabela 7 apresenta os dados estatísticos das variáveis contínuas (8.3.1). A linha de *Null* representa o número de valores zerados para os ingredientes, e a linha *NA* representa o número de dados faltando. Como as amostras foram filtradas para manter apenas conjuntos de amostras com valores conhecidos da resistência à compressão aos 28 dias, o número de *NAs* é zero para essa idade. A figura 5 apresenta os dados estatísticos das variáveis discretas (8.3.2).

Tabela 7: Estatística descritiva - variáveis contínuas

	Amostras	Null	NA	Min	Max	Intervalo	Soma	Mediana	Média	Erro padrão da média	Intervalo de confiança da média	Variância	Desvio Padrão	Coeficiênte de variação
Cimento	416	0	0	102.00	540.00	438.00	109373.10	257.70	262.92	5.10	10.02	10817.50	104.01	0.40
E.G.A.F.	416	174	0	0.00	359.40	359.40	35824.60	94.25	86.12	4.32	8.49	7755.00	88.06	1.02
Cinza Volante	416	202	0	0.00	200.10	200.10	26389.00	71.25	63.44	3.26	6.40	4407.81	66.39	1.05
Água	416	0	0	121.80	247.00	125.20	76335.60	185.00	183.50	0.94	1.86	370.73	19.25	0.10
Superplast.	416	107	0	0.00	32.20	32.20	2871.30	7.60	6.90	0.26	0.52	28.85	5.37	0.78
A.Graúdo	416	0	0	801.00	1145.00	344.00	397799.90	953.35	956.25	4.12	8.10	7063.06	84.04	0.09
A.Miúdo	416	0	0	594.00	992.60	398.60	317809.80	769.65	763.97	3.59	7.06	5371.89	73.29	0.10
3 dias	121	0	295	2.33	41.64	39.31	2210.82	15.52	18.27	0.87	1.72	91.64	9.57	0.52
7 dias	114	0	302	7.51	59.09	51.58	2845.52	21.06	24.96	1.29	2.55	188.81	13.74	0.55
14 dias	62	0	354	12.84	59.76	46.92	1782.56	26.54	28.75	1.10	2.19	74.62	8.64	0.30
28 dias	416	0	0	8.54	81.75	73.21	15101.13	33.72	36.30	0.70	1.38	206.30	14.36	0.40
56 dias	83	0	333	23.25	80.20	56.95	4178.77	50.77	50.35	1.52	3.02	190.82	13.81	0.27
100 dias	120	0	296	21.86	82.60	60.74	5701.90	45.61	47.52	1.17	2.31	163.40	12.78	0.27
Água/ Cimento	416	0	0	0.27	1.88	1.62	340.60	0.73	0.82	0.02	0.03	0.11	0.34	0.41
A.Miúdo/ Cimento	416	0	0	1.14	9.24	8.10	1415.82	2.94	3.40	0.07	0.14	1.96	1.40	0.41
A.Graúdo/ Cimento	416	0	0	1.55	8.70	7.14	1761.06	3.67	4.23	0.08	0.16	2.70	1.64	0.39
A.Miúdo / A.Graúdo	416	0	0	0.53	1.16	0.63	335.42	0.80	0.81	0.01	0.01	0.01	0.11	0.14
Água / A.Graúdo	416	0	0	0.12	0.29	0.17	80.66	0.19	0.19	0.00	0.00	0.00	0.03	0.16
Água / A.Miúdo	416	0	0	0.13	0.38	0.26	101.26	0.24	0.24	0.00	0.00	0.00	0.04	0.17

2.5.2 Correlação dos ingredientes e resistência à compressão

A figura 6 apresenta a correlação das variáveis para cada conjunto de idades (8.3.3). A figura 7 apresenta os mesmos dados, mas em vez de correlacionar todos, correlaciona apenas com a resistência à compressão, mostrando os valores mais detalhadamente (8.3.4).

A interpretação da figura 7 sugere que a resistência do concreto está relacionada positivamente principalmente com os ingredientes cimento e superplastificante e negativamente com a água e agregado miúdo. Quanto menor a quantidade de cimento para os agregados e para água, mais negativamente estão correlacionados com a resistência à compressão.

As figuras 8 e 9 mostram a relação entre os principais ingredientes (conhecida como traço) em relação à resistência à compressão (8.3.5 e 8.3.6). A interpretação dessas figuras mostra que quanto maior a quantidade de cimento em relação aos outros ingredientes, maior será a resistência à compressão.

2.5.3 Distribuição das variáveis

A figura 10 mostra a distribuição das variáveis nas amostras (8.3.7). Foi calculado utilizando apenas os dados aos 28 dias.

De outra forma, a figura 11 mostra a distribuição dos ingredientes e da resistência à compressão para cada conjunto de idades (8.3.8), ou seja, no caso dos 28 dias apresenta a mesma informação que a figura 10. Através dela é visualizado que como esperado, a resistência à compressão gradualmente aumenta ao longo do tempo. Além disso é visualizado que a concentração dos ingredientes podem variar muito quando estratíficado pelas idades.

Figura 5: Estatística descritiva - variáveis discretas

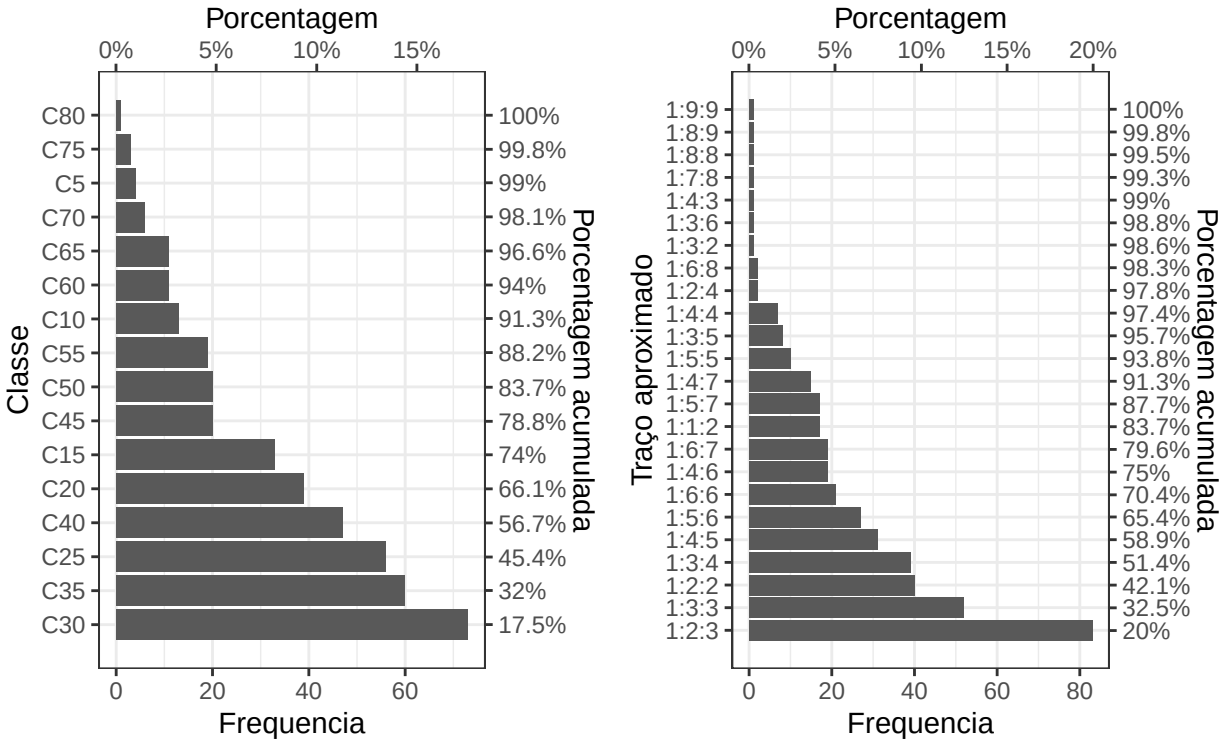


Figura 6: Correlações em cada idade

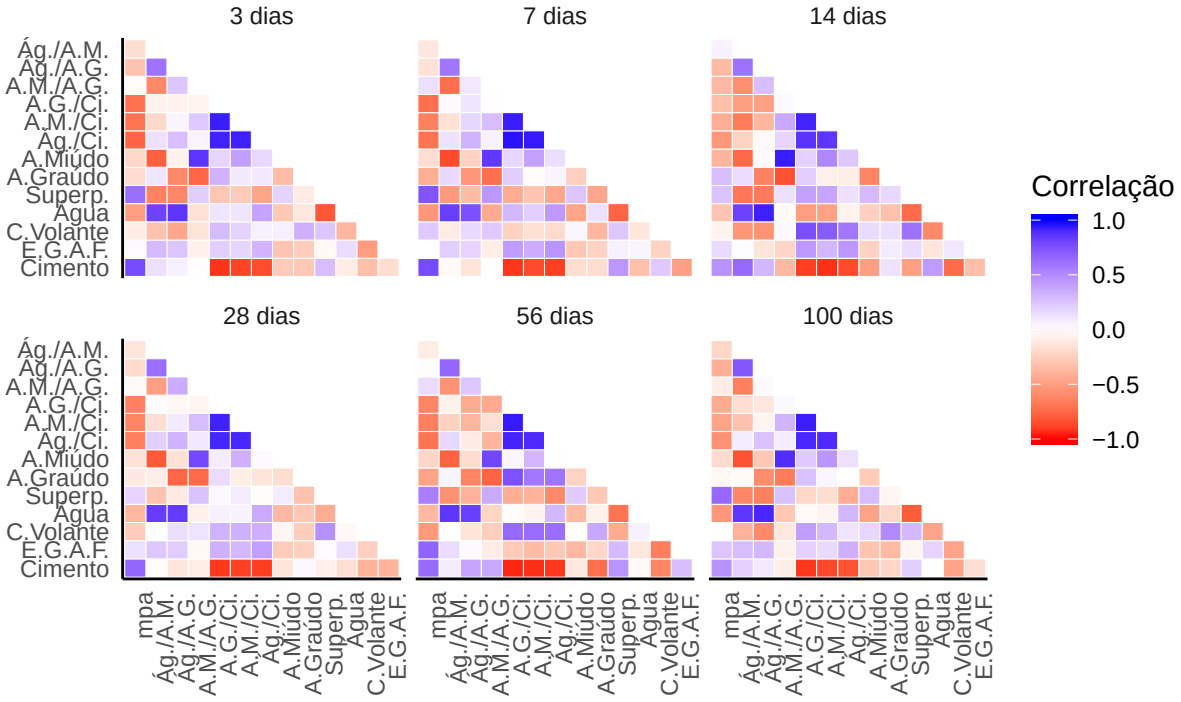


Figura 7: Correlação das variáveis com a resistência à compressão no tempo

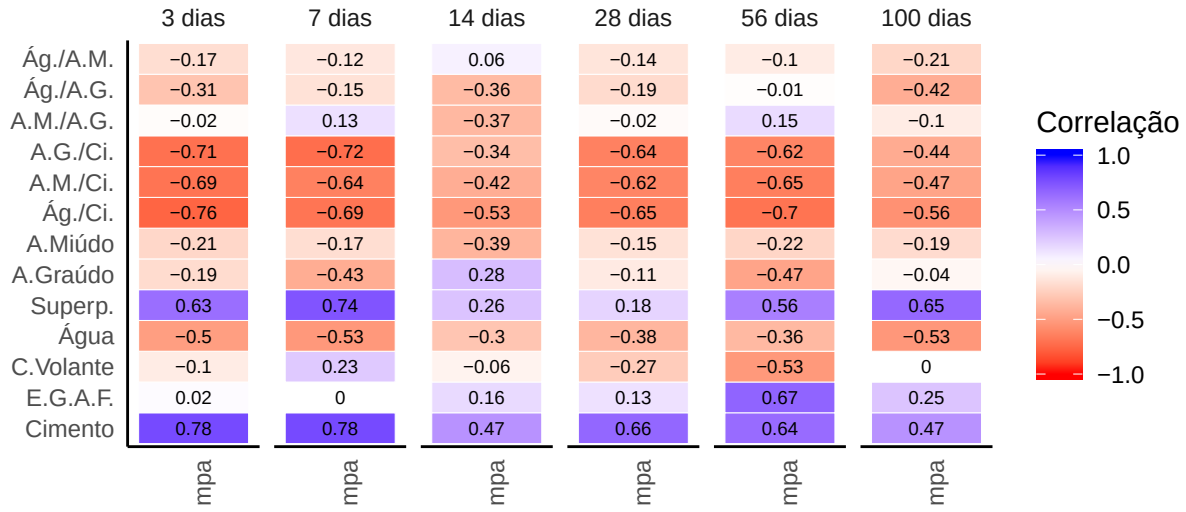


Figura 8: Relação entre o traço aproximado, água, resistência à compressão e idade

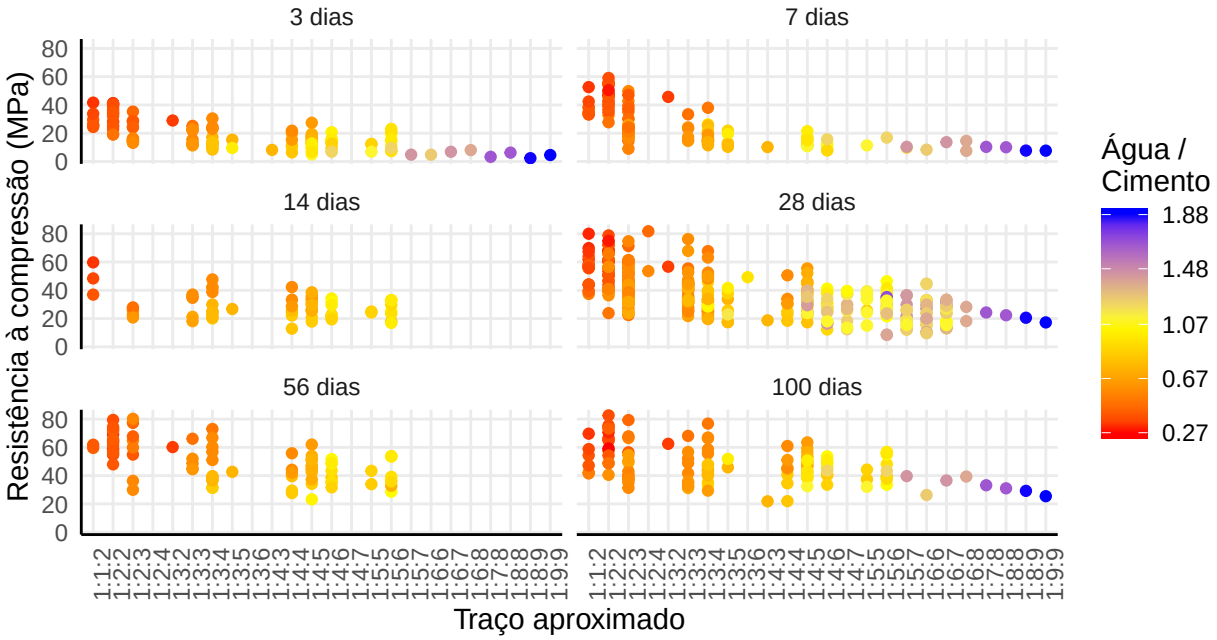
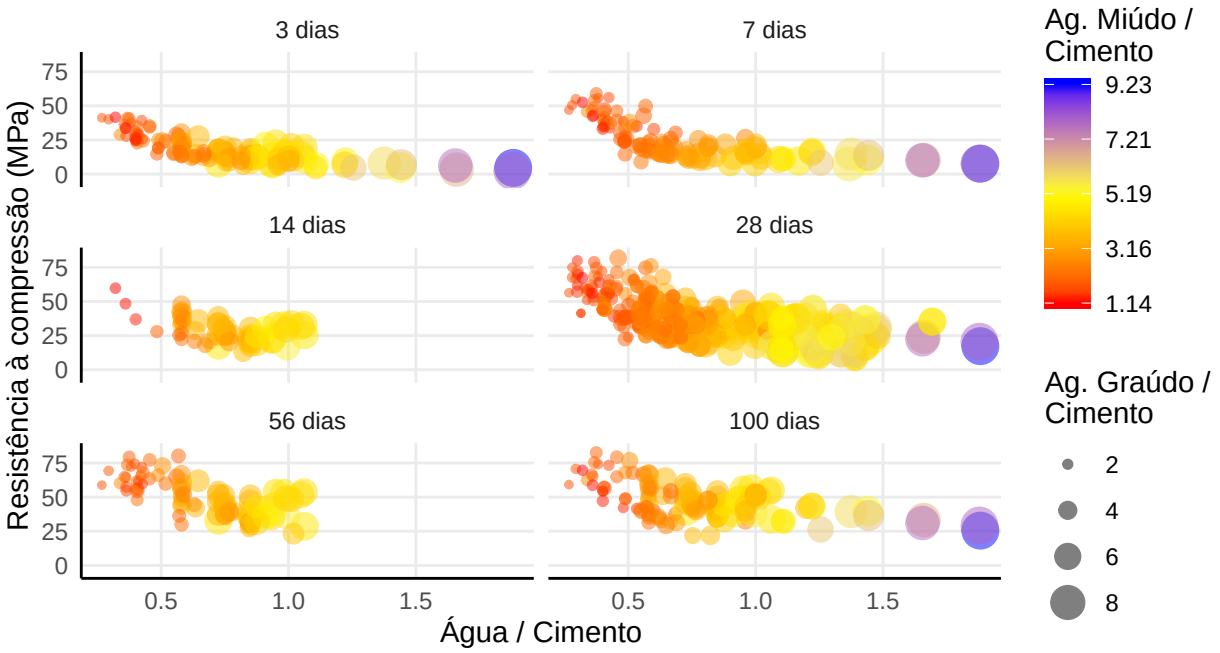


Figura 9: Relação das principais proporções do concreto



2.5.4 Análise de componente principal

Na figura 12, utilizando uma classificação alternativa, foi realizada a análise de componente principal nos ingredientes das amostras (8.3.9). A classificação separa o concreto em 4 grupos de resistência a compressão diferentes, baixo até *20 MPa*, normal até *40 MPa*, médio até *70 MPa* e alto acima disso. É possivel perceber que os grupos se sobrepõem, mas existem uma diferenciação entre o grupo alto e baixo.

2.6 Modelos de machine learning

O desenvolvimento dos modelos de machine learning foi realizado com o pacote *caret* (Kuhn 2020) e baseado em Irizarry (2019) e Kuhn (2008).

2.6.1 Pre processamento e separação dos dados

Como existe a variável categórica para o traço aproximado do concreto, foi realizada a conversão dessa variável em variáveis fictiícias (*dummy vars*) (8.4.1), passando de 22 colunas (id, classe, resistência à compressão e mais 19 *features*) para 45 colunas, uma adição de 23 variáveis, uma para cada traço aproximado.

As amostras foram separadas baseado nas idades. Foram criados um conjunto de dados para cada valor de idade, totalizando 6 conjuntos diferentes (8.4.2). Para fins ilustrativos, as primeiras 18 de 45 colunas das primeiras 6 amostras do conjunto de 28 dias são mostradas na tabela 8.

Figura 10: Distribuição das variáveis

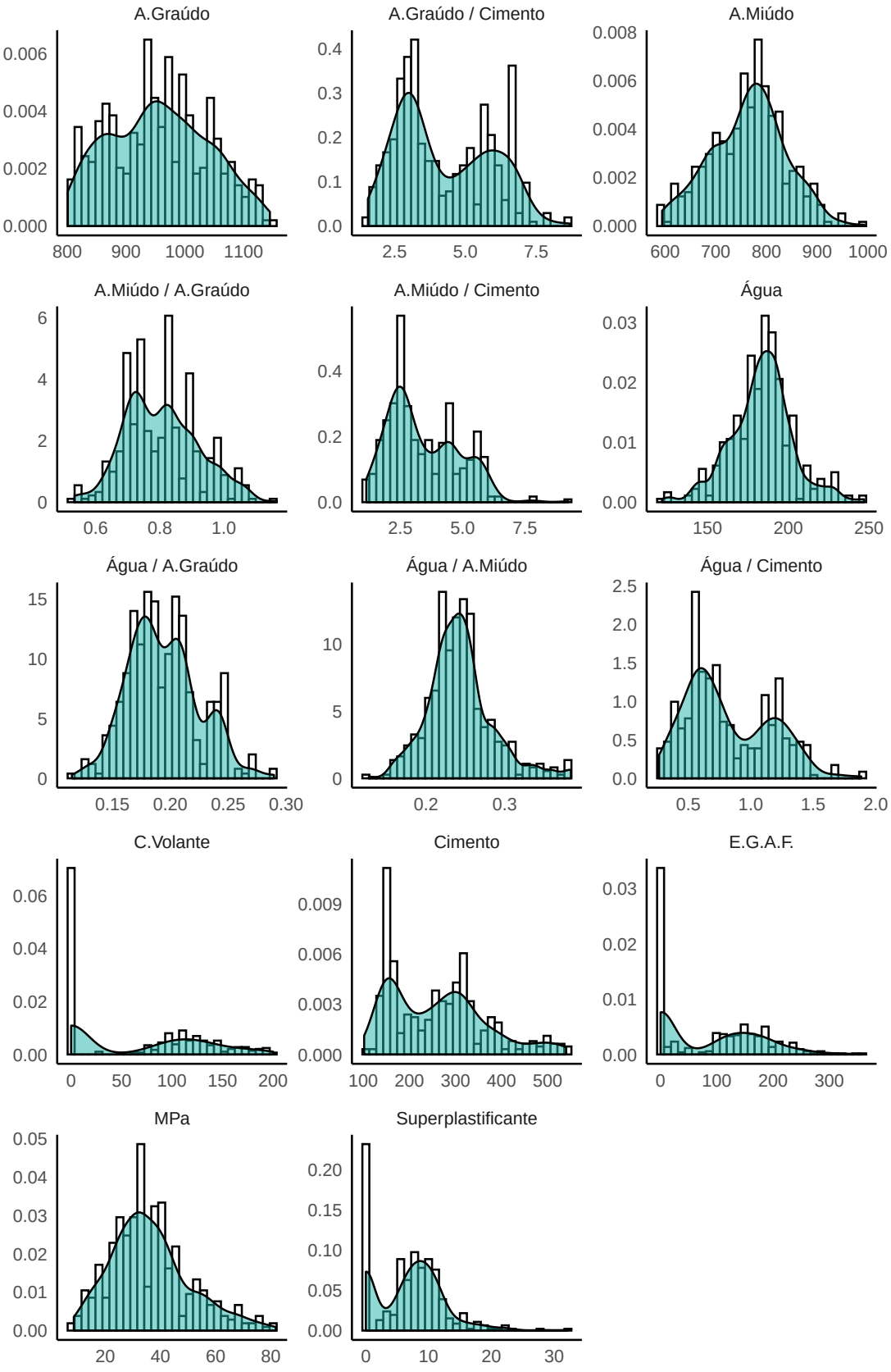


Figura 11: Distribuição das variáveis em relação a idade

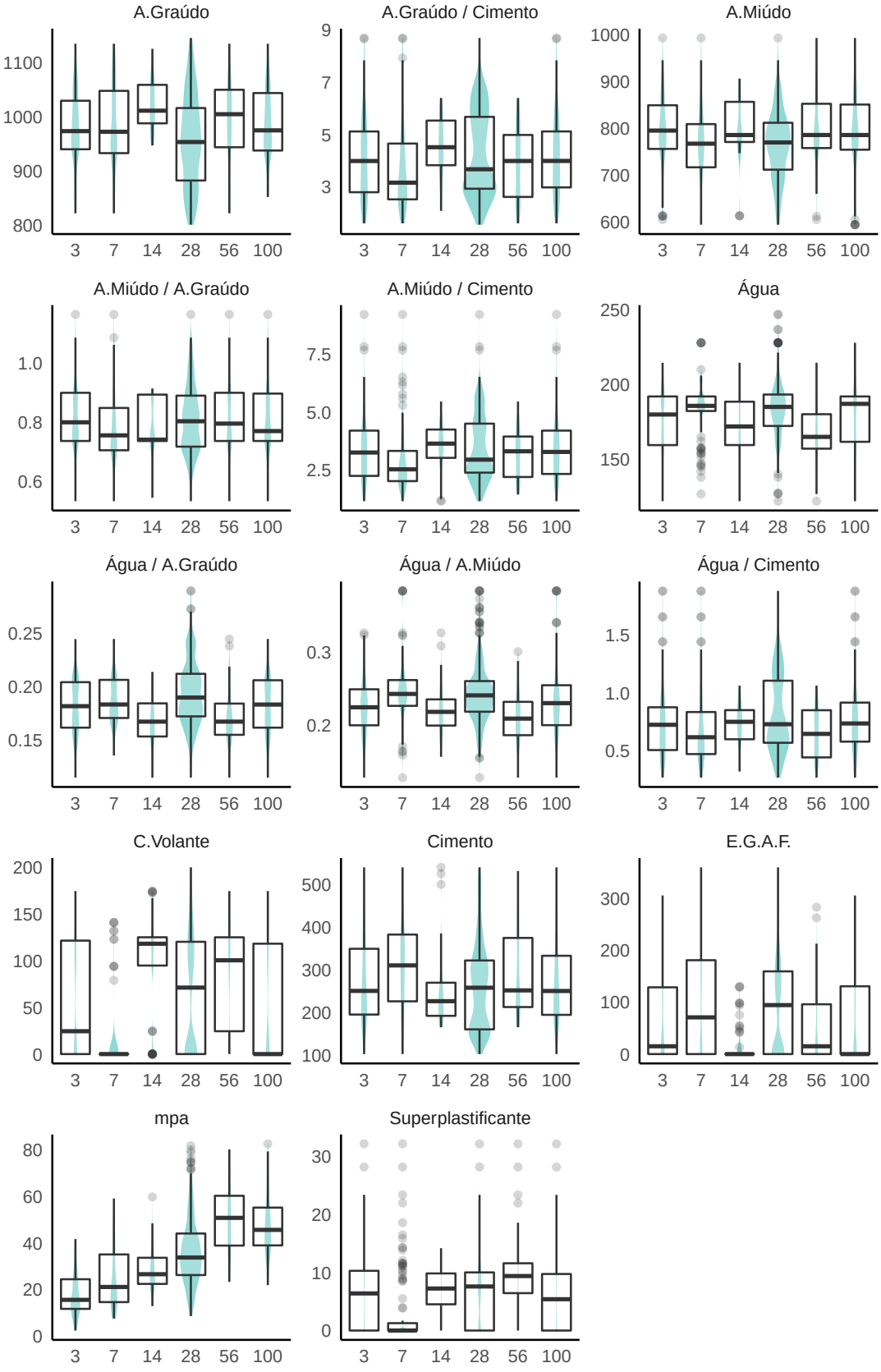


Figura 12: Análise componente principal nos ingredientes

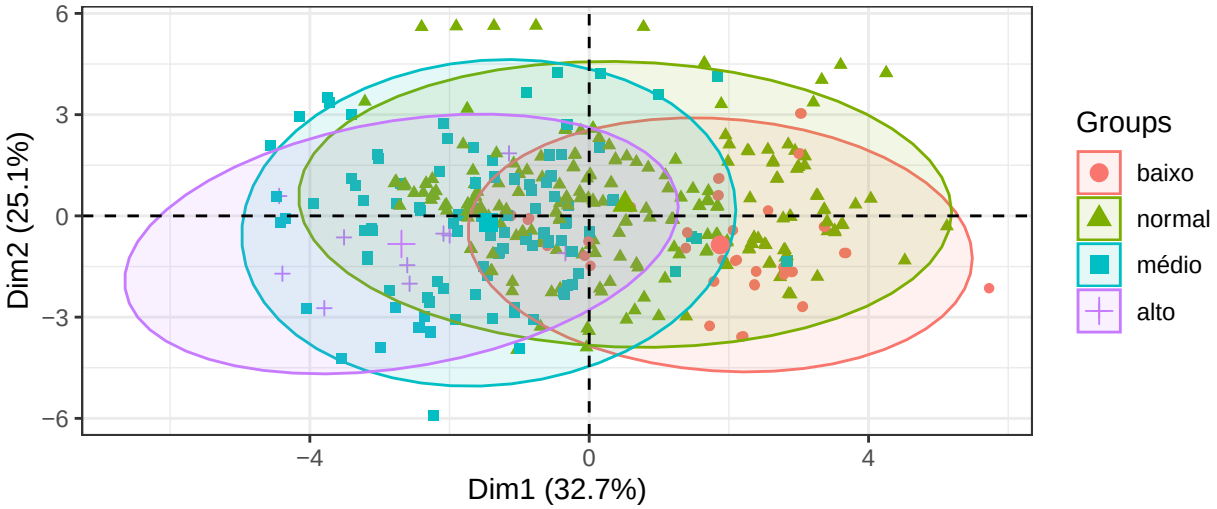


Tabela 8: Primeiras 18 colunas das primeiras 6 amostras de 28 dias

ID	Cimento kg/m^3	E.G.A.F. kg/m^3	C.Vol. kg/m^3	Água kg/m^3	Superp. kg/m^3	A.Graúdo kg/m^3	A.Miúdo kg/m^3	MPa MPa	Classe	Ág./ Ci.	A.M./ Ci.	A.G./ Ci.	A.M./ A.G.	Ág./ A.G.	Ág./ Ag.M.	Traço 1:1:2	Apox. 1:2:2
1	540.0	0.0	0	162	2.5	1040.0	676.0	79.99	C75	0.30	1.25	1.93	0.65	0.16	0.24	1	0
2	540.0	0.0	0	162	2.5	1055.0	676.0	61.89	C60	0.30	1.25	1.95	0.64	0.15	0.24	1	0
3	332.5	142.5	0	228	0.0	932.0	594.0	33.02	C30	0.69	1.79	2.80	0.64	0.24	0.38	0	0
5	198.6	132.4	0	192	0.0	978.4	825.5	28.02	C25	0.97	4.16	4.93	0.84	0.20	0.23	0	0
6	266.0	114.0	0	228	0.0	932.0	670.0	45.85	C45	0.86	2.52	3.50	0.72	0.24	0.34	0	0
7	380.0	95.0	0	228	0.0	932.0	594.0	36.45	C35	0.60	1.56	2.45	0.64	0.24	0.38	0	1

Para cada um dos 6 conjuntos foi verificado a existência ou não de variáveis com variância próxima a zero e sua subsequênte remoção (8.4.4). Muitas das 23 variáveis adicionadas referentes ao traço aproximado do concreto foram removidas devido a esse fato. Além delas, no caso do conjunto de 7 dias, a variável cinza volante também foi removida. Depois foi verificado que não existem variáveis com alta correlação, acima de 0,999 em nenhum dos 6 conjuntos de dados (8.4.5). Após essas etapas, os conjuntos de amostras apresentaram 24, 21, 23, 23, 24 e 25 colunas respectivamente para as idades em sequência crescente.

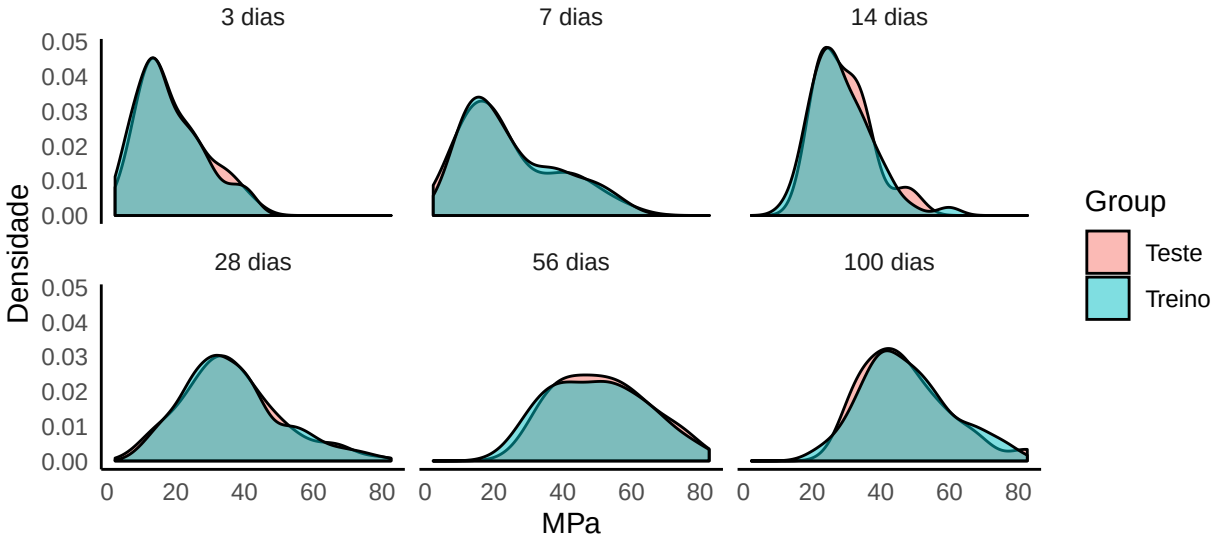
A etapa de centralização e normalização das variáveis foi realizada mais a frente, junto com a aplicação dos modelos, pois é mais simples fazer dessa forma com o pacote *caret*. Se fosse realizada nesse momento, seria nescessário manualmente desfazer essas transformações nas previsões. O *caret* permite transformar antes do treino dos modelos e já transforma de volta os resultados.

Cada um dos conjuntos de dados foi separado em conjuntos de teste e treino, *20%* e *80%* respectivamente (8.4.6). A figura 13 mostra como ficou a distribuição dos dados entre os conjuntos em relação a resistência à compressão para cada modelo (8.4.7).

2.6.2 Medidas de performance

A avaliação da performance dos modelos foi realizada pela Raiz do Erro Quadrático Médio (*RMSE*). O *RMSE* é a medida utilizada em todos os trabalhos citados na introdução e permitirá a comparação dos modelos na discussão.

Figura 13: Distribuição dos conjuntos de teste e treino



2.6.3 Modelos ingênuos

Antes de criar os modelos verdadeiros, para fins de comparação, foram criados modelos ingênuos. Eles simplesmente prevêem que a resistência à compressão do conjunto de teste, é a média da resistência à compressão do conjunto de treino (8.4.8). Em outras palavras, os modelos ingênuos são simplesmente o melhor palpite possível. Os resultados podem ser conferidos na tabela 9.

Tabela 9: Modelos ingênuos

Idade	Média MPa (treino)	RMSE (treino)	RMSE (teste)
3	18.08887	9.591344	9.303229
7	25.12383	13.731569	13.443646
14	28.63980	8.786823	7.593319
28	36.33605	14.361021	14.283824
56	50.06555	13.968077	12.702112
100	47.57000	12.758042	12.614652

2.6.4 Escolha do algorítimo

O pacote *caret* (Kuhn 2020) expõe mais de 200 algorítimos diferentes para criar modelos de *machine learning*. A documentação do pacote apresenta um código inicial ("Models Clustered by Tag Similarity" 2020) como sugestão para selecionar um portifólio de algóritmos mais distintos possíveis em relação à algum algorítimo pré-selecionado, mas para agilidade e devido a limitações técnicas, foi escolhido utilizar um algorítimo com a maior probabilidade de atingir o melhor resultado possível. Segundo Fernandez-Delgado et al. (2014), que comparou 179 algorítimos em 121 banco de dados diferentes, o algorítimo mais provável de atingir os melhores resultados possíveis é o *Parallel Random Forest* (denominado *prRF* no *caret*).

2.6.5 Modelos de regressão

Como ao longo do processamento foram adicionadas novas variáveis (as relações entre os ingredientes e mais algumas *dummy vars* para cada conjunto de idade), foram estudadas 5 possibilidades de configurações

das *features* para os modelos:

1. Todas as *features*;
2. Sem as *dummy vars*;
3. Apenas *features* originais;
4. Apenas *features* adicionadas;
5. Apenas *features* adicionadas, sem as *dummy vars*;

Construindo um modelo para cada uma dessas configurações utilizando o conjunto de amostras aos 28 dias (8.4.10), mostrou que a melhor opção é a configuração 2, ou seja, as *dummy vars* foram completamente descartadas, porém foram mantidas as outras novas varáveis. Para fins ilustrativos, a tabela 10 mostra as primeiras 6 amostras do conjunto de treino do modelo de 28 dias. As amostras dos outros modelos, dos conjuntos de teste e treino são similares, sendo a única diferença no modelo de 7 dias, que exlcui a cinza volante devido a variância próxima a zero, realizado anteriormente.

Tabela 10: Primeiras 6 amostras do conjunto de treino do modelo de 28 dias

Features													Outcome
Cimento kg/m^3	E.G.A.F. kg/m^3	C.Vol. kg/m^3	Água kg/m^3	Superp. kg/m^3	A.Graúdo kg/m^3	A.Miúdo kg/m^3	Ág./ Ci.	A.M./ Ci.	A.G./ Ci.	A.M./ A.G.	Ág./ A.G.	Ág./ Ag.M.	y MPa
540.0	0.0	0	162	2.5	1040.0	676.0	0.30	1.25	1.93	0.65	0.16	0.24	79.99
540.0	0.0	0	162	2.5	1055.0	676.0	0.30	1.25	1.95	0.64	0.15	0.24	61.89
266.0	114.0	0	228	0.0	932.0	670.0	0.86	2.52	3.50	0.72	0.24	0.34	45.85
380.0	95.0	0	228	0.0	932.0	594.0	0.60	1.56	2.45	0.64	0.24	0.38	36.45
475.0	0.0	0	228	0.0	932.0	594.0	0.48	1.25	1.96	0.64	0.24	0.38	39.29
198.6	132.4	0	192	0.0	978.4	825.5	0.97	4.16	4.93	0.84	0.20	0.23	28.02

Para cada conjunto de idade foi criado um modelo utilizando o algorítimo *Parallel Random Forest*, definido anteriormente (8.4.12). Para cada um dos 6 modelos, o parâmetro *mtry* foi otimizado, e foi realizado *repeated cross-validation*, dividindo em 10 ou 30 partes e repetindo 10 vezes.

3 Resultados

O *RMSE* de teste de cada modelo em ordem crescente de idade foi respectivamente 3.31, 4.36, 4.62, 4.72, 5.94 e 5.85. A tabela 11 apresenta os detalhes e resultados de cada modelo (8.5.1). A figura 14 compara os valores reais e previstos (8.5.2), e as tabelas seguintes mostram os melhores e piores resultados de cada modelo (8.5.3).

Tabela 11: Resultados dos modelos de regressão

Modelo	mtry	CV	Repetições	RMSE (treino)	RMSE (teste)
3 dias	6	30	10	3.905196	3.310370
7 dias	2	10	10	4.475981	4.361987
14 dias	13	30	10	5.136687	4.620515
28 dias	11	30	10	5.847334	4.716698
56 dias	8	30	10	6.702565	5.939163
100 dias	8	10	10	6.381940	5.851088

Figura 14: Comparação dos valores reais e previstos em cada modelo

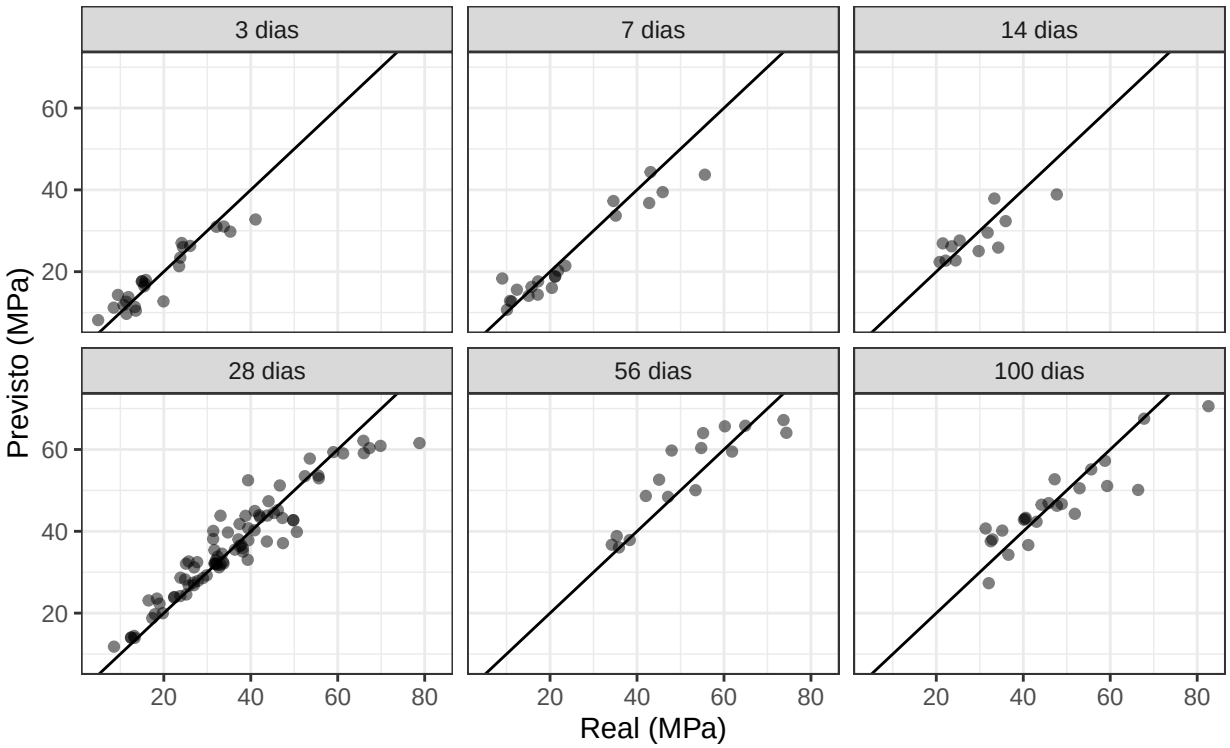


Tabela 12: Modelo de 3 dias

10 melhores			10 piores		
Real	Previsto	Erro	Real	Previsto	Erro
26.06	26.297746	0.2377457	41.10	32.758262	-8.341738
23.80	23.432367	-0.3676333	19.93	12.732172	-7.197828
15.52	16.436292	0.9162920	35.30	29.781989	-5.518011
10.76	11.830651	1.0706507	9.45	14.316742	4.866742
32.11	30.977619	-1.1323807	4.90	8.154245	3.254245
11.36	12.660044	1.3000440	13.57	10.464572	-3.105428
15.62	17.079698	1.4596980	24.10	27.016321	2.916321
24.39	26.018261	1.6282610	33.80	31.045796	-2.754204
11.41	9.688879	-1.7211213	8.49	11.196910	2.706910
11.85	13.788864	1.9388640	14.99	17.677379	2.687379

Tabela 13: Modelo de 7 dias

10 melhores			10 piores		
Real	Previsto	Erro	Real	Previsto	Erro
17.24	17.61947	0.3794733	55.60	43.71082	-11.889177
15.75	16.29318	0.5431847	9.01	18.32999	9.319987
10.09	10.67438	0.5843837	45.90	39.45277	-6.447230
15.07	14.10010	-0.9698955	42.80	36.78526	-6.014737
43.11	44.35190	1.2419040	20.42	16.03647	-4.383533
35.10	33.69951	-1.4004926	12.37	15.58829	3.218292
11.17	12.71080	1.5408013	17.17	14.38220	-2.787795
21.86	20.23260	-1.6274033	34.57	37.26745	2.697450
10.79	12.85092	2.0609233	21.16	18.79017	-2.369827
23.52	21.42631	-2.0936880	21.18	18.84231	-2.337694

Tabela 14: Modelo de 14 dias

10 melhores			10 piores		
Real	Previsto	Erro	Real	Previsto	Erro
22.14	22.69248	0.5524823	47.71	38.87416	-8.835845
20.77	22.36561	1.5956093	34.24	25.88502	-8.354984
24.45	22.74412	-1.7058757	21.50	26.91740	5.417403
25.37	27.62261	2.2526130	29.75	25.02926	-4.720744
31.81	29.53499	-2.2750130	33.36	37.88168	4.521675
23.51	26.19615	2.6861493	35.96	32.35358	-3.606417
35.96	32.35358	-3.6064167	23.51	26.19615	2.686149
33.36	37.88168	4.5216750	31.81	29.53499	-2.275013
29.75	25.02926	-4.7207443	25.37	27.62261	2.252613
21.50	26.91740	5.4174030	24.45	22.74412	-1.705876

Tabela 15: Modelo de 28 dias

10 melhores			10 piores		
Real	Previsto	Erro	Real	Previsto	Erro
27.83	27.86776	0.0377573	78.80	61.57401	-17.225986
43.80	43.84951	0.0495087	39.38	52.47227	13.092267
26.92	26.82812	-0.0918800	33.04	43.81819	10.778193
31.87	32.06794	0.1979360	50.60	39.85258	-10.747415
19.77	19.97517	0.2051693	47.40	37.11150	-10.288501
31.88	32.13467	0.2546680	69.84	60.88636	-8.953635
59.00	59.36857	0.3685727	31.38	40.09934	8.719345
23.79	24.17376	0.3837583	49.77	42.71779	-7.052213
28.99	28.55592	-0.4340790	49.77	42.72394	-7.046057
32.24	31.79166	-0.4483390	25.10	32.07616	6.976158

Tabela 16: Modelo de 56 dias

10 melhores			10 piores		
Real	Previsto	Erro	Real	Previsto	Erro
35.85	36.06413	0.2141317	47.97	59.74774	11.777736
38.33	37.86955	-0.4604540	74.36	64.08547	-10.274529
64.90	65.81326	0.9132646	55.20	64.00472	8.804718
47.13	48.41224	1.2822413	45.08	52.62057	7.540568
61.86	59.49031	-2.3696857	42.03	48.63652	6.606522
34.20	36.72639	2.5263850	73.70	67.19523	-6.504768
53.46	50.04835	-3.4116453	54.77	60.36449	5.594488
35.34	38.82656	3.4865560	60.20	65.67068	5.470680
60.20	65.67068	5.4706798	35.34	38.82656	3.486556
54.77	60.36449	5.5944884	53.46	50.04835	-3.411645

Tabela 17: Modelo de 100 dias

10 melhores			10 piores		
Real	Previsto	Erro	Real	Previsto	Erro
67.80	67.53752	-0.2624793	66.42	50.11266	-16.307345
55.64	55.13289	-0.5071103	82.60	70.59383	-12.006168
43.06	42.33515	-0.7248493	31.35	40.69152	9.341519
45.84	46.89673	1.0567320	59.30	51.07318	-8.226824
47.74	46.24704	-1.4929567	51.86	44.27586	-7.584144
58.78	57.23738	-1.5426163	47.22	52.72019	5.500191
48.85	46.67286	-2.1771403	32.92	38.02416	5.104161
44.21	46.50377	2.2937710	32.53	37.56541	5.035406
36.59	34.26929	-2.3207093	35.17	40.18011	5.010112
52.96	50.52831	-2.4316947	32.07	27.31722	-4.752779

4 Discussão

Os modelos construidos apresentam resultados satisfatórios e provam que a resistência à compressão do concreto pode ser prevista de forma relativamente fácil. A alternativa adotada de criar um modelo para cada conjunto de idade se mostrou como uma alternativa válida, conseguindo estratificar para obter resultados específicos de cada conjunto. Os estudos citados na introdução utilizando o mesmo conjunto de dados possuem resultados similares, como esperado. A tabela 18 apresenta os resultados desses trabalhos (8.6.1), e a tabela 19 apresenta os valores encontrados (8.6.2) para fácil comparação.

Tabela 18: Comparação dos estudos de outros autores

Autor	Ano	Algorítimo	RMSE
Pierobon	2018	Ensemble com 5 algorítimos	4.150
Hameed	2020	Artificial Neural Networks	4.736
Raj	2018	Gradient Boosting Regressor	4.957
Modukuru	2020	Random Forest Regressor	5.080
Alshamiri	2020	Regularized Extreme Learning Machine	5.508
Abban	2016	Support Vector Machines with Radial Basis Function Kernel	6.105

Tabela 19: Resultados finais

Modelo	RMSE
3 dias	3.310370
7 dias	4.361987
14 dias	4.620515
28 dias	4.716698
56 dias	5.939163
100 dias	5.851088

Seguindo a linha de raciocínio desse trabalho, ele pode ser realizado com diferentes algorítimos, os resultados aqui encontrados utilizaram apenas um único (*Parallel Random Forest*), mesmo que tenha sido teoricamente o "melhor" encontrado, outros algorítmos podem apresentar resultados ainda melhores. Outra opção é criar um *ensemble* com diversos algorítimos, como realizado por Pierobon (2018), mas com a separação de conjuntos de idade aqui proposto. Além disso, pode ser realizado com um conjunto maior de dados, idealmente com o mesmo número de amostras em cada conjunto de idade, distribuição mais homgênia da resistência à compressão e menor variância entre as amostras.

5 Bibliografia

Abban, Daniel. 2016. "Concrete Compresive Strength." October 2016. https://rpubs.com/brother_abban/220101.

Alshamiri, Tian-Feng e Kim, Abobakr e Yuan. 2020. "Non-Tuned Machine Learning Approach for Predicting the Compressive Strength of High-Performance Concrete." *Materials* 13 (February): 1023. https://doi.org/10.3390/ma13051023.

"Concrete Compressive Strength Data Set." 2008. University of California Irvine. March 2008. https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength.

Fernandez-Delgado, Manuel, E. Cernadas, S. Barro, and Dinani Amorim. 2014. "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" *Journal of Machine Learning Research* 15 (October): 3133–81.

Hameed, Mohamed, Mohammed e Khalid. 2020. "Prediction of Compressive Strength of High-Performance Concrete: Hybrid Artificial Intelligence Technique." In, 323–35. https://doi.org/10.1007/978-3-030-38752-5_26.

Hasan, Ahsanul, Md e Kabir. 2011. "Prediction of Compressive Strength of Concrete from Early Age Test Result." In. https://doi.org/10.13140/RG.2.1.3270.7684.

Irizarry, R. A. 2019. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R*. Chapman & Hall/Crc Data Science Series. CRC Press. https://books.google.com.br/books?id=xb29DwAAQBAJ.

Kabir, Md e Miah, Ahsanul e Hasan. 2012. "Predicting 28 Days Compressive Strength of Concrete from 7 Days Test Result," January, 18–22. https://www.researchgate.net/publication/258255513_Predicting_28_Days_Compressive_Strength_of_Concrete_from_7_Days_Test_Result.

Kuhn, Max. 2008. "Building Predictive Models in R Using the Caret Package." *Journal of Statistical Software, Articles* 28 (5). https://doi.org/10.18637/jss.v028.i05.

Kuhn, Max et al. 2020. *Caret: Classification E Regression Training*. https://cran.r-project.org/web/packages/caret/index.html.

"Models Clustered by Tag Similarity." 2020. 2020. http://topepo.github.io/caret/models-clustered-by-tag-similarity.html.

Modukuru, Pranay. 2020. "Concrete Compressive Strength Prediction Using Machine Learning." 2020. https://towardsdatascience.com/concrete-compressive-strength-prediction-using-machine-learning-4a531b3c43f3.

Pierobon, Gabriel. 2018. "A Comprehensive Machine Learning Workflow with Multiple Modelling Using Caret and caretEnsemble in R." September 2018. https://towardsdatascience.com/a-comprehensive-machine-learning-workflow-with-multiple-modelling-using-caret-and-caretensemble-in-fcbf6d80b5f2.

Raj, Pavan. 2018. "Predicting Compressive Strength of Concrete." June 2018. https://www.kaggle.com/pavanraj159/predicting-compressive-strength-of-concrete.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

RStudio Team. 2020. *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc. http://www.rstudio.com/.

Yeh, I-Cheng. 1998. "Modeling of Strength of High-Performance Concrete Using Artificial Neural Networks." Cement and Concrete Research, 28(12), 1797-1808." *Cement and Concrete Research* 28 (December): 1797–1808. https://doi.org/10.1016/S0008-8846(98)00165-3.

6 Appendix 1 - Ambiente virtual

6.1 Sistema operacional

platform	x86_64-apple-darwin15.6.0
arch	x86_64
os	darwin15.6.0
system	x86_64, darwin15.6.0
status	
major	3
minor	6.2
year	2019
month	12
day	12
svn rev	77560
language	R
version.string	R version 3.6.2 (2019-12-12)
nickname	Dark and Stormy Night

6.2 Pacotes utilizados

caret	6.0.85
cowplot	1.0.0
dplyr	0.8.4
factoextra	1.0.6
gdata	2.18.0
ggplot2	3.2.1
gridExtra	2.3
kableExtra	1.1.0
knitr	1.28
pastecs	1.3.21
purrr	0.3.3
questionr	0.7.0
reshape2	1.4.3
tidyr	1.0.2
tidyverse	1.3.0

7 Appendix 2 - Repositório online

https://github.com/pedrobern/concrete-compressive-strength-prediction

8 Appendix 3 - Código

8.1 Obtenção dos dados

8.1.1 Download dos dados

```
# Download dos dados
url_base <- "https://archive.ics.uci.edu"
url <-  "/ml/machine-learning-databases/concrete/compressive/Concrete_Data.xls"
download.file(paste0(url_base, url), "data.xls")
dat <- read.xls("data.xls")
n_inicial_samples <- nrow(dat)
colnames(dat)
```

8.1.2 Renomeando as colunas

```
# Renomeando as colunas
colnames(dat) <- c(
  "cement",
  "blast_furnace_slag",
  "fly_ash",
  "water",
  "superplasticizers",
  "coarse_aggregate",
  "fine_aggregate",
  "day",
  "mpa"
  )
dat$id <- seq.int(nrow(dat))
```

8.1.3 Reordenando os dados

```
# Reordenando os dados
col_order <- c(
  "id",
  "cement",
  "blast_furnace_slag",
  "fly_ash",
  "water",
  "superplasticizers",
  "coarse_aggregate",
  "fine_aggregate",
  "day",
  "mpa"
)
dat <- dat[, col_order]
```

8.1.4 Definindo nomes e unidades das colunas

```r
# Definindo nomes e unidades das colunas
colNames <- c("ID", "Cimento", "E.G.A.F", "C.Volante", "Água",
              "Superp.", "A.Graúdo", "A.Miúdo", "Dia", "Comp.Str.")
dfUnits <- c("", "$kg/m^3$", "$kg/m^3$","$kg/m^3$","$kg/m^3$",
              "$kg/m^3$","$kg/m^3$","$kg/m^3$","","$MPa$")
```

8.1.5 Tabela - Primeiras amostras

```r
# Tabela - Primeiras amostras
caption <- "Primeiras 6 amostras"
kable(
    dat[1:6,],
    col.names = dfUnits,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    align = "c"
    ) %>%
    add_header_above(header = colNames, line = F, align = "c") %>%
    kable_styling(latex_options = c("HOLD_position", "scale_down"))
```

8.2 Preparação dos dados

8.2.1 Removendo amostras duplicadas

```r
# Removendo amostras duplicadas
n_distinct_samples <- dat %>% select(-c(id)) %>% n_distinct()
n_duplicated_samples <- n_inicial_samples - n_distinct_samples
dat <- dat[!duplicated(select(dat, -c(id))),]
n_samples <- nrow(dat)
```

8.2.2 Tabela - Amostras com a mesma composição

```r
# Tabela - Amostras com a mesma composição
same_samples <- dat %>%
  filter(id %in% c(653, 678, 654, 681))
caption <- "Amostras com a mesma composição"
kable(
  same_samples[order(same_samples$day),],
  col.names = dfUnits,
  escape = F,
  booktabs = T,
  caption = caption,
  linesep = "\\addlinespace",
  align = "c",
  row.names = FALSE
  ) %>%
  add_header_above(header = colNames, line = F, align = "c") %>%
  kable_styling(latex_options = c("HOLD_position", "scale_down"))
```

8.2.3 Tabela - Amostras iguais com resultados diferentes

```
# Tabela - Amostras iguais com resultados diferentes
same_samples_2 <- dat %>%
  filter(id %in% c(472, 473, 474))
caption <- "Amostras iguais com resultados diferentes"
kable(
  same_samples_2[order(same_samples_2$day),],
  col.names = dfUnits,
  escape = F,
  booktabs = T,
  caption = caption,
  linesep = "\\addlinespace",
  align = "c",
  row.names = FALSE
  ) %>%
  add_header_above(header = colNames, line = F, align = "c") %>%
  kable_styling(latex_options = c("scale_down"))
```

8.2.4 Limpeza inicial das amotras

```
# Limpeza inicial das amotras
dat <- dat %>%
  group_by(
    cement,
    blast_furnace_slag,
    fly_ash,
    water,
    superplasticizers,
    coarse_aggregate,
    fine_aggregate,
  ) %>%
  filter("28" %in% day) %>%
  mutate(id = id[which.min(id)]) %>%
  ungroup() %>%
  group_by(id, day) %>%
  mutate(mpa = mean(mpa)) %>%
  ungroup()
dat <- dat[!duplicated(select(dat, -c(id))),]
dat$id<-factor(dat$id)
n_samples <- nrow(dat)
n_distinct_samples <- n_distinct(dat$id)
```

8.2.5 Tabela - Amostras anteriores após processamento

```
# Tabela - Amostras anteriores após processamento
same_samples <- dat %>%
  filter(id == 653 | id == 472 & day == 28)
caption <- "Amostras anteriores após processamento"
kable(
  same_samples[order(same_samples$id, same_samples$day),],
```

```
  col.names = dfUnits,
  escape = F,
  booktabs = T,
  caption = caption,
  linesep = "\\addlinespace",
  align = "c",
  row.names = FALSE,
  digits = 2
  ) %>%
  add_header_above(header = colNames, line = F, align = "c") %>%
  kable_styling(latex_options = c("HOLD_position", "scale_down"))
```

8.2.6 Figura - Resistência à compressão (MPa) vs idade (dias)

```
# Figura - Resistência à compressão (MPa) vs idade (dias)
cap <- "Boxplot - Resistência à compressão (MPa) vs idade (dias)"
ylabel <- "Resistência à compressão (MPa)"
xlabel <- "Idade (dias)"
dat %>%
  ggplot(aes(x=factor(day), y=mpa)) +
  geom_boxplot() +
  geom_jitter(alpha=0.2)  +
  theme_bw() +
  ylab(ylabel) +
  xlab(xlabel)
```

8.2.7 Figura - Análise componente principal - 90, 91 e 100 dias

```
# Figura - Análise componente principal - 90, 91 e 100 dias
dat_90_91_100 <- dat %>%
  ungroup() %>%
  filter(day %in% c(90, 91, 100)) %>%
  select(-c(id, mpa))
cap <- "Análise componente principal - 90, 91 e 100 dias"
colnames(dat_90_91_100) <- c(
   "Cim.","E.G.A.F.","Ci.Vo.","Agu.","Sup.","Ag.G.","Ag.M.","dia"
)
pca <- prcomp(select(dat_90_91_100, -c(dia)), scale = TRUE)
habillage <- dat_90_91_100$dia
fviz_pca_biplot(
  pca,
  geom.ind = "point",
  habillage=habillage,
  addEllipses = TRUE,
  ellipse.level=0.75) +
  ggtitle("") +
  theme_bw() +
  coord_cartesian(xlim = c(-3, 3.5), ylim = c(3, -5))
```

8.2.8 Figura - Resistência à compressão ao longo do tempo

```
# Figura - Resistência à compressão ao longo do tempo
cap <- "Resistência à compressão ao longo do tempo"
ylabel <- "MPa"
xlabel <- "Idade (dias)"
dat_duplicated_only <- dat %>%
  group_by(id) %>%
  filter(n()>5) %>%
  select(id, mpa, day)
dat_duplicated_only %>%
  ggplot(aes(day, mpa, fill=id, alpha = 0.5)) +
  geom_line() +
  xlab(xlabel) +
  ylab(ylabel) +
  theme_bw() +
  theme(legend.position = "none")
```

8.2.9 Juntando amostras de 90, 91 e 100 dias

```
# Juntando amostras de 90, 91 e 100 dias
ind_90 <- dat$id[which(dat$day == "90")]
ind_91 <- dat$id[which(dat$day == "91")]
ind_100 <- dat$id[which(dat$day == "100")]
sum_duplicated <- sum(duplicated(c(ind_90, ind_91, ind_100))) # 0
dat <- dat %>%
  ungroup() %>%
  mutate(day = ifelse(day %in% c(91, 90), 100, day))
```

8.2.10 Figura - Frequência das idades

```
# Figura - Frequência das idades
cap <- "Frequência das idades"
ylabel <- "Frequência"
xlabel <- "Idade (dias)"
dat %>%
  ggplot(aes(x = factor(day))) +
  geom_bar() +
  theme_bw() +
  xlab(xlabel) +
  ylab(ylabel)
```

8.2.11 Removendo idades com frequência menor que 50

```
# Removendo idades com frequência menor que 50
dat <- dat[dat$day %in% c(3, 7, 14, 28, 56, 100),]
```

8.2.12 Reorganização das amotras

```r
# Reorganização das amotras
dat <- dat %>%
  group_by_at(vars(-mpa)) %>%
  mutate(row_id = 1:n()) %>% ungroup() %>%
  spread(day, mpa, sep = "_") %>%
  select(-row_id)
```

8.2.13 Tabela - Primeiras 6 amostras reorganizadas

```r
# Tabela - Primeiras 6 amostras reorganizadas
caption <- "Primeiras 6 amostras reorganizadas"
colNames2 = c("ID", "Cimento", "E.G.A.F.", "C.Volante", "Água",
              "Superp.", "A.Graúdo", "A.Miúdo", "3 dias", "7 dias",
              "14 dias", "28 dias", "56 dias", "100 dias")
dfUnits2 <- c("", "$kg/m^3$", "$kg/m^3$","$kg/m^3$","$kg/m^3$",
               "$kg/m^3$", "$kg/m^3$","$kg/m^3$","$MPa$","$MPa$",
               "$MPa$","$MPa$","$MPa$","$MPa$")
kable(
    head(dat[order(dat$id),]),
    col.names = dfUnits2,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    align = "c"
    ) %>%
    add_header_above(header = colNames2, line = F, align = "c") %>%
    kable_styling(latex_options = c("HOLD_position", "scale_down"))
```

8.2.14 Total de amostras

```r
# Total de amostras
n_samples <- nrow(dat) # 416
n_distinct_samples <- n_distinct(dat$id) # 416
```

8.2.15 Adicionando novas variáveis

```r
# Adicionando novas variáveis
concrete_class <- function(mpa){
  if (mpa >= 10) {
    s <- as.character(mpa)
    first <- substr(s, start = 1, stop = 1)
    second <- ifelse(substr(s, start = 2, stop = 2) >= 5, 5, 0)
  }
  else {
    first <- ""
    second <- "5"
  }
  paste("C", first, second, sep = "")
}
mix <- function(c, f_ag, c_ag){
```

```
  paste(
    1,
    round(f_ag/c, 0),
    round(c_ag/c, 0),
    sep = ":")
}
dat <- dat %>%
  mutate(class = sapply(day_28, concrete_class)) %>%
  mutate(class = as.factor(class)) %>%
  mutate(mix_app = factor(
    mix(cement, fine_aggregate, coarse_aggregate))) %>%
  mutate(`water_/_cement` = water / cement) %>%
  mutate(`fine_aggregate_/_cement` = fine_aggregate/cement) %>%
  mutate(`coarse_aggregate_/_cement` = coarse_aggregate/cement) %>%
  mutate(`fine_aggregate_/_coarse_aggregate` = fine_aggregate/coarse_aggregate) %>%
  mutate(`water_/_coarse_aggregate` = water/coarse_aggregate) %>%
  mutate(`water_/_fine_aggregate` = water/fine_aggregate)
lvl <- levels(dat$class)
dat$class <- factor(
  dat$class,
  levels=c( "C5", sort(lvl[lvl!="C5"], decreasing=F)))
```

8.2.16 Tabela - Novas variáveis

```
# Tabela - Novas variáveis
caption <- "Novas variáveis"
colNames7 = c("ID", "Classe","Traço aproximado",
              "Água / Cimento", "A.Miúdo / Cimento",
              "A.Graúdo / Cimento", "A.Miúdo / A.Graúdo",
              "Água / A.Graúdo", "Água / A.Miúdo")
kable(
    head(dat[order(dat$id),][,c(1,15:22)]),
    col.names = colNames7,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    align = "c",
    digits = 4
    ) %>%
  kable_styling(latex_options = c("HOLD_position", "scale_down")) %>%
  column_spec(2, width = "1.5cm") %>%
  column_spec(3, width = "2cm") %>%
  column_spec(4:9, width = "1.7cm")
```

8.3 Visualização dos dados

8.3.1 Tabela - Estatística descritiva - variáveis contínuas

```
# Tabela - Estatística descritiva - variáveis contínuas
summ <- t(
  stat.desc(select(dat, -c(id, class, mix_app))))
```

```
caption <- "Estatística descritiva - variáveis contínuas"
colnames(summ) <- c("Amostras", "Null", "NA", "Min", "Max", "Intervalo",
                "Soma", "Mediana", "Média", "Erro padrão  da média",
                "Intervalo de confiânça da média",
                "Variância", "Desvio Padrão", "Coeficiênte de variação")
rownames(summ) = c("Cimento", "E.G.A.F.", "Cinza Volante", "Água",
                "Superplast.", "A.Graúdo", "A.Miúdo", "3 dias", "7 dias",
                "14 dias", "28 dias", "56 dias", "100 dias",
                "Água/ Cimento", "A.Miúdo/ Cimento",
                "A.Graúdo/ Cimento", "A.Miúdo / A.Graúdo",
                "Água / A.Graúdo", "Água / A.Miúdo")
kable(
  summ,
  escape = F,
  booktabs = T,
  caption = caption,
  linesep = "\\addlinespace",
  align = "c",
  digits = c(0,0,0,2,2,2,2,2,2,2,2,2,2,2)
  ) %>%
  kable_styling(latex_options = c("HOLD_position", "scale_down")) %>%
  column_spec(c(1,10:15), width = "1.5cm")
```

8.3.2 Figura - Estatística descritiva - variáveis discretas

```
# Figura - Estatística descritiva - variáveis discretas
cap <- "Estatística descritiva - variáveis discretas"
name1 <- "Porcentagem"
name2 <- "Porcentagem acumulada"
ylabel <- "Frequencia"
xlabel1 <- "Classe"
xlabel2 <- "Traço aproximado"
format_percent = function(n){
  paste(n, "%", sep = "")
}
format_class <- function(cls){
  str_remove_all(cls, "C")
}
f_class <- freq(dat$class, cum = TRUE, sort = "dec", total = F) %>%
  select(n, "%", "%cum") %>%
  mutate(class = row.names(.)) %>%
  mutate(class_n = as.numeric(format_class(class)))
f_cls_labels = function(n) {
  f_class$class[n]
}
f_cls_acc_labels = function(n){
  paste(f_class$`%cum`[n], "%", sep = "")
}
p1 <- f_class %>%
  ggplot(aes(x = as.integer(reorder(class_n, -n)), y = n)) +
  geom_bar(stat = 'identity') +
  scale_y_continuous(
      sec.axis = sec_axis(~./length(dat$class) * 100,
```

```
                          name = name1,
                          labels = format_percent)) +
  scale_x_continuous(labels = f_cls_labels, breaks = 1:16, limits = c(0.5,16.5),
                     sec.axis = sec_axis(~., breaks = 1:16,
                                         name = name2,
                                         labels = f_cls_acc_labels)) +
  theme_bw() +
  theme(panel.grid.minor.y = element_blank()) +
  xlab(xlabel1) +
  ylab(ylabel) +
  coord_flip()
format_mix <- function(mix){
  str_remove_all(mix, ":")
}
f_mix <- freq(dat$mix_app, cum = TRUE, sort = "dec", total = F) %>%
  select(n, "%", "%cum") %>%
  mutate(mix = row.names(.)) %>%
  mutate(mix_n = as.numeric(format_mix(mix)))
f_mix_labels = function(n) {
  f_mix$mix[n]
}
f_mix_acc_labels = function(n){
  paste(f_mix$`%cum`[n], "%", sep = "")
}
p2 <- f_mix %>%
  ggplot(aes(x = as.integer(reorder(mix_n, -n)), y = n)) +
  geom_bar(stat = 'identity') +
  scale_y_continuous(
      sec.axis = sec_axis(~./length(dat$mix_app) * 100,
                          name = name1,
                          labels = format_percent)) +
  scale_x_continuous(labels = f_mix_labels, breaks = 1:24, limits = c(0.5,24.5),
                     sec.axis = sec_axis(~., breaks = 1:24,
                                         name = name2,
                                         labels = f_mix_acc_labels)) +
  theme_bw() +
  theme(panel.grid.minor.y = element_blank()) +
  xlab(xlabel2) +
  ylab(ylabel) +
  coord_flip()
grid.arrange(p1, p2, ncol=2)
```

8.3.3 Figura - Correlações em cada idade

```
# Figura - Correlações em cada idade
cor_dat <- dat %>% select(-c(id))
cap <- "Correlações em cada idade"
f_lvl <- c("3 dias", "7 dias", "14 dias","28 dias", "56 dias", "100 dias")
name <- "Correlação"
colnames_dat <- c("Cimento", "E.G.A.F.", "C.Volante", "Água",
                  "Superp.", "A.Graúdo", "A.Miúdo",
                  "3","7","14","28","56","100",
                  "class","mix_app" ,"Ág./Ci.",
```

```
                  "A.M./Ci.", "A.G./Ci.", "A.M./A.G.",
                  "Ág./A.G.", "Ág./A.M.")
colnames(cor_dat) <- colnames_dat
cor_dat <- cor_dat %>%
  gather("day", "mpa", c("3", "7", "14", "28", "56", "100")) %>%
  drop_na()
cor_day <- function(d){
  res <- cor_dat %>%
    filter(day == d) %>%
    select(-c(day, class, mix_app)) %>%
    cor(.)
  res[upper.tri(res)] <- NA
  return(res)
}
cor_dats <- list(cor_day(3), cor_day(7), cor_day(14),
                 cor_day(28), cor_day(56), cor_day(100))
melt_day <- function(df, d){
  df %>%
    melt() %>%
    mutate(day = d)
}
melt_dats <- list(melt_day(cor_dats[1], 3), melt_day(cor_dats[2], 7),
                  melt_day(cor_dats[3], 14), melt_day(cor_dats[4],28),
                  melt_day(cor_dats[5], 56), melt_day(cor_dats[6],100))
melt_dat_final <- melt_dats %>%
  reduce(rbind) %>%
  filter(value != 1)
melt_dat_final$day <- factor(melt_dat_final$day)
levels(melt_dat_final$day) <- f_lvl
melt_dat_final %>%
  ggplot(aes(x=reorder(Var1, desc(Var1)), y=Var2, fill=value)) +
  geom_tile(color = "white") +
  facet_wrap(~day, ncol=3) +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
   midpoint = 0, limit = c(-1,1),name=name, na.value="white") +
  xlab("") +
  ylab("") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))
```

8.3.4 Figura - Correlações no tempo

```
# Figura - Correlações no tempo
cap <- "Correlação das variáveis com a resistência à compressão no tempo"
name <- "Correlação"
melt_dat_final %>%
  filter(Var1 == "mpa" | Var2 == "mpa") %>%
  ggplot(aes(x=reorder(Var1, desc(Var1)), y=Var2, fill=value)) +
  geom_tile(color = "white") +
  facet_wrap(~day, ncol=6) +
```

```
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
   midpoint = 0, limit = c(-1,1),name=name, na.value="white") +
  xlab("") +
  ylab("") +
  geom_text(aes(label = round(value, 2)), size = 2.5) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))
```

8.3.5 Figura - Relação entre o traço aproximado, água, MPa e idade

```
# Figura - Relação entre o traço aproximado, água, MPa e idade
cap <- "Relação entre o traço aproximado, água, resistência à compressão e idade"
d <- " dias"
xlabel <- "Traço aproximado"
ylabel <- "Resistência à compressão (MPa)"
label <- "Água /\nCimento"
mix_dat <- dat %>%
  select(c(day_3,day_7,day_14,day_28,day_56,day_100,
           mix_app, `water_/_cement`,
           `fine_aggregate_/_cement`,
           `coarse_aggregate_/_cement`))

labs <- paste(c("3","7","14","28","56","100"), d, sep="")
mix_dat <- mix_dat %>%
  gather("day", "mpa", -c(mix_app, `water_/_cement`,
                          `fine_aggregate_/_cement`,
                          `coarse_aggregate_/_cement`)) %>%
  drop_na()
lvls <- paste("day_",c("3","7","14","28","56","100"), sep="")
mix_dat$day <- factor(mix_dat$day, levels=lvls)
levels(mix_dat$day) <- labs
min_x <- min(mix_dat$`water_/_cement`)
max_x <- max(mix_dat$`water_/_cement`)
s_x <- max_x - min_x
mix_dat %>%
  ggplot(aes(x=mix_app, y=mpa, colour = `water_/_cement`)) +
  geom_point()  +
  facet_wrap(~ day, ncol=2) +
  theme_bw() +
  ylab(ylabel) +
  xlab(xlabel) +
  scale_shape_manual(values=c(16, 2, 8)) +
  scale_colour_gradient2(low = "red", mid = "yellow", high = "blue",
                         midpoint = s_x / 2 + min_x ,limits = c(min_x, max_x),
                         breaks = c(round(min_x, 2),
                                    round(s_x*0.25 + min_x,2),
                                    round(s_x*0.5 + min_x,2),
                                    round(s_x * 0.75 + min_x,2),
                                    round(max_x, 2))) +
  labs(colour = label) +
```

```
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(panel.grid.minor = element_blank(),
      axis.line = element_line(colour = "black"))
```

8.3.6 Figura - Relação das principais características do concreto

```
# Figura - Relação das principais características do concreto
cap <- "Relação das principais proporções do concreto"
d <- " dias"
xlabel <- "Água / Cimento"
ylabel <- "Resistência à compressão (MPa)"
colour <- "Ag. Miúdo /\nCimento"
size <- "Ag. Graúdo /\nCimento"
min_x_2 <- min(mix_dat$`fine_aggregate_/_cement`)
max_x_2 <- max(mix_dat$`fine_aggregate_/_cement`)
s_x_2 <- max_x_2 - min_x_2
mix_dat %>%
  ggplot(aes(x=`water_/_cement`, y=mpa,
             colour = `fine_aggregate_/_cement`,
             size = `coarse_aggregate_/_cement`)) +
  geom_point(alpha = 0.5)  +
  facet_wrap(~ day, ncol=2) +
  theme_bw() +
  ylab(ylabel) +
  xlab(xlabel) +
  scale_colour_gradient2(low = "red", mid = "yellow", high = "blue",
                         midpoint = (s_x_2 / 2) + min_x_2,
                         limits = c(min_x_2, max_x_2),
                         breaks = c(round(min_x_2, 2),
                                    round(s_x_2*0.25 + min_x_2,2),
                                    round(s_x_2*0.5 + min_x_2,2),
                                    round(s_x_2 * 0.75 + min_x_2,2),
                                    round(max_x_2 - 0.01, 2))
                         ) +
  labs(colour = colour, size = size) +
  ylim(c(-5,85)) +
  scale_radius() +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
      axis.line = element_line(colour = "black"))
```

8.3.7 Figura - Distribuição das variáveis

```
# Figura - Distribuição das variáveis
cap <- "Distribuição das variáveis"
colnames_dat <- c("Cimento", "E.G.A.F.", "C.Volante", "Água",
                  "Superplastificante", "A.Graúdo", "A.Miúdo", "MPa",
                  "Água / Cimento", "A.Miúdo / Cimento", "A.Graúdo / Cimento",
                  "A.Miúdo / A.Graúdo","Água / A.Graúdo", "Água / A.Miúdo")
dist_dat <- dat %>%
  select(-c(id, class, day_3, day_7, day_14, day_56, day_100, mix_app))
```

```r
colnames(dist_dat) <- colnames_dat
dist_dat <- dist_dat %>%
  gather("Var", "value") %>%
  mutate(value = as.numeric(value))
dist_dat %>%
  ggplot(aes(value)) +
  geom_histogram(aes(y = ..density..),
                 colour = "black",
                 fill = "white") +
  geom_density(alpha = .5, fill = "lightseagreen") +
  facet_wrap(~ Var, ncol=3, scale = "free") +
  theme_minimal() +
  xlab("") +
  ylab("") +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))
```

8.3.8 Figura - Distribuição das variáveis agrupadas por idade

```r
# Figura - Distribuição das variáveis agrupadas por idade
days_labs <- c("3","7","14","28","56","100")
cap <- "Distribuição das variáveis em relação a idade"
colnames_dat <- c("Cimento", "E.G.A.F.", "C.Volante", "Água",
                        "Superplastificante", "A.Graúdo", "A.Miúdo",
                        days_labs,
                        "Água / Cimento", "A.Miúdo / Cimento", "A.Graúdo / Cimento",
                        "A.Miúdo / A.Graúdo","Água / A.Graúdo", "Água / A.Miúdo")
dist_dat_2 <- dat %>%
  select(-c(id, class, mix_app))
colnames(dist_dat_2) <- colnames_dat
dist_dat_2 <- dist_dat_2 %>%
  gather("day", "mpa", days_labs) %>%
  drop_na() %>%
  gather("Var", "value", -c("day")) %>%
  mutate(day = factor(day, levels = days_labs))
dist_dat_2 %>%
  ggplot(aes(x = day, y = value)) +
  geom_violin(color = NA,
                fill = "lightseagreen",
                alpha = .5,
                na.rm = TRUE,
                scale = "count") +
  geom_boxplot(alpha = 0.2) +
  facet_wrap(~ Var, ncol=3, scale = "free") +
  theme_minimal() +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black")) +
  xlab("") +
  ylab("")
```

8.3.9 Figura - Análise componente principal nos ingredientes

```
# Figura - Análise componente principal nos ingredientes
cap <- "Análise componente principal nos ingredientes"
colnames_dat <- c(
  "Cim.", "E.G.A.F.", "Ci.Vo.", "Águ.", "Sup.", "Ag.G.",
  "Ag.M.","AxC", "MxC","GxC","MxC","AxG","AxM"
)
class_list <- list(baixo="0", normal="20", médio="40", alto="70")
dat_pca <- dat %>%
  select(-c(id, class, mix_app,
            "day_3", "day_7", "day_14", "day_28", "day_56", "day_100"))
colnames(dat_pca) <- colnames_dat
class_2 <- dat$day_28
class_2[class_2 < 20] = "0"
class_2[class_2 >= 20 & class_2 < 40] = "20"
class_2[class_2 >= 40 & class_2 < 70] = "40"
class_2[class_2 >= 70] = "70"
class_2 <- factor(class_2)
levels(class_2) <- class_list
pca <- prcomp(dat_pca, scale = TRUE)
fviz_pca_ind(
  pca,
  geom.ind = "point",
  habillage=class_2,
  addEllipses = T,
  ellipse.level=0.95) +
  ggtitle("") +
  theme_bw()
```

8.4 Modelos de machine learning

8.4.1 Variáveis fictícias - dummy vars

```
# Variáveis fictícias - dummy vars
dummies <- dummyVars( ~ mix_app, data = dat)
dummyDat <- data.frame(predict(dummies, newdata = dat))
dummyDat$id <- dat$id
dat <- dat %>%
  select(-c(mix_app)) %>%
  full_join(., dummyDat)
```

8.4.2 Preparação dos dados

```
# Preparação dos dados
names(dat) <- gsub(x = names(dat), pattern = "/", replacement = ".")
dat_3 <- dat %>%
  select(-c("day_7", "day_14", "day_28", "day_56", "day_100")) %>%
  drop_na() %>%
  rename_at("day_3",~"mpa")
dat_7 <- dat %>%
  select(-c("day_3", "day_14", "day_28", "day_56", "day_100")) %>%
```

```
  drop_na() %>%
  rename_at("day_7",~"mpa")
dat_14 <- dat %>%
  select(-c("day_3", "day_7", "day_28", "day_56", "day_100")) %>%
  drop_na() %>%
  rename_at("day_14",~"mpa")
dat_28 <- dat %>%
  select(-c("day_3", "day_7", "day_14", "day_56", "day_100")) %>%
  drop_na() %>%
  rename_at("day_28",~"mpa")
dat_56 <- dat %>%
  select(-c("day_3", "day_7", "day_14", "day_28", "day_100")) %>%
  drop_na() %>%
  rename_at("day_56",~"mpa")
dat_100 <- dat %>%
  select(-c("day_3", "day_7", "day_14", "day_28", "day_56")) %>%
  drop_na() %>%
  rename_at("day_100",~"mpa")
```

8.4.3 Tabela - Primeiras 18 colunas das primeiras 6 amostras de 28 dias

```
# Tabela - Primeiras 18 colunas das primeiras 6 amostras de 28 dias
colNames = c("ID", "Cimento", "E.G.A.F.", "C.Vol.", "Água",
              "Superp.", "A.Graúdo", "A.Miúdo", "MPa", "Classe",
              "Ág./", "A.M./", "A.G./",
              "A.M./","Ág./", "Ág./", "Traço Apox." = 2)
dfUnits <- c("", "$kg/m^3$", "$kg/m^3$","$kg/m^3$","$kg/m^3$",
              "$kg/m^3$", "$kg/m^3$","$kg/m^3$","$MPa$","", "Ci.",
              "Ci.", "Ci.", "A.G.", "A.G.", "Ag.M.", "1:1:2", "1:2:2")
caption <- "Primeiras 18 colunas das primeiras 6 amostras de 28 dias"
dat_table <- dat_28[1:18]
kable(
    head(dat_table[order(dat_table$id),]),
    col.names = dfUnits,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    digits = 2,
    align = "c"
    ) %>%
    add_header_above(header = colNames, line = F, align = "c") %>%
    kable_styling(latex_options = c("HOLD_position", "scale_down"))
```

8.4.4 Removendo colunas com variância próxima a zero

```
# Removendo colunas com variância próxima a zero
nzv_3 <- nearZeroVar(dat_3)
nzv_7 <- nearZeroVar(dat_7)
nzv_14 <- nearZeroVar(dat_14)
nzv_28 <- nearZeroVar(dat_28)
```

```r
nzv_56 <- nearZeroVar(dat_56)
nzv_100 <- nearZeroVar(dat_100)
dat_3 <- dat_3[,-nzv_3]
dat_7 <- dat_7[,-nzv_7]
dat_14 <- dat_14[,-nzv_14]
dat_28 <- dat_28[,-nzv_28]
dat_56 <- dat_56[,-nzv_56]
dat_100 <- dat_100[,-nzv_100]
```

8.4.5 Verificação de variáveis com alta correlação

```r
# Verificação de variáveis com alta correlação
descr_cor_3 <- cor(select(dat_3, -c(mpa, id, class)))
descr_cor_7 <- cor(select(dat_7, -c(mpa, id, class)))
descr_cor_14 <- cor(select(dat_14, -c(mpa, id, class)))
descr_cor_28 <- cor(select(dat_28, -c(mpa, id, class)))
descr_cor_56 <- cor(select(dat_56, -c(mpa, id, class)))
descr_cor_100 <- cor(select(dat_100, -c(mpa, id, class)))
high_cor_3 <- sum(abs(descr_cor_3[upper.tri(descr_cor_3)]) > .999)
high_cor_7 <- sum(abs(descr_cor_7[upper.tri(descr_cor_7)]) > .999)
high_cor_14 <- sum(abs(descr_cor_14[upper.tri(descr_cor_14)]) > .999)
high_cor_28 <- sum(abs(descr_cor_28[upper.tri(descr_cor_28)]) > .999)
high_cor_56 <- sum(abs(descr_cor_56[upper.tri(descr_cor_56)]) > .999)
high_cor_100 <- sum(abs(descr_cor_100[upper.tri(descr_cor_100)]) > .999)
```

8.4.6 Separação em conjunto de teste e treino

```r
# Separação em conjunto de teste e treino
reg <- list(
  dat_3 %>% select(-c(mpa, class, id)) %>% mutate(y = dat_3$mpa),
  dat_7 %>% select(-c(mpa, class, id)) %>% mutate(y = dat_7$mpa),
  dat_14 %>% select(-c(mpa, class, id)) %>% mutate(y = dat_14$mpa),
  dat_28 %>% select(-c(mpa, class, id)) %>% mutate(y = dat_28$mpa),
  dat_56 %>% select(-c(mpa, class, id)) %>% mutate(y = dat_56$mpa),
  dat_100 %>% select(-c(mpa, class, id)) %>% mutate(y = dat_100$mpa)
)
reg_seed <- c(
  1111, # 3
  1, # 7
  22, # 14
  11111, # 28
  111, # 56
  11 # 100
  )
split_reg <- function(n){
  set.seed(reg_seed[[n]], sample.kind="Rounding")
  createDataPartition(reg[[n]]$y, p = .8, list = F)
}
trainIndex_reg <- lapply(list(1,2,3,4,5,6), split_reg)
gen_dat <- function(n){
  regIndex <- trainIndex_reg[[n]]
  list(train = reg[[n]][regIndex,], test = reg[[n]][-regIndex,])
```

```
}
dats_reg <- lapply(list(1,2,3,4,5,6), gen_dat)
names(dats_reg) <- c("d3", "d7", "d14", "d28", "d56", "d100")
```

8.4.7 Distribuição dos conjuntos de teste e treino

```
# Distribuição dos conjuntos de teste e treino
cap <- "Distribuição dos conjuntos de teste e treino"
d_lab <- " dias"
ylabel <- "Densidade"
g1 <- "Treino"
g2 <- "Teste"
dens <- function(d, n){
  dens <- full_join(
      d$train %>%
        select(y) %>%
        mutate(Group = g1, day = str_sub(n, 2)),
      d$test %>%
        select(y) %>%
        mutate(Group = g2, day = str_sub(n, 2))
    )
  dens
}
densities <- imap(dats_reg, dens) %>%
  reduce(rbind) %>%
  mutate(day_f = factor(day, levels=c('3','7','14','28', '56', '100')))
labs <- paste(c("3","7","14","28","56","100"), d_lab, sep="")
levels(densities$day_f) <- labs
densities %>%
  ggplot(aes(y, fill = Group)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~day_f, ncol=3) +
  xlab("MPa") +
  ylab(ylabel) +
  theme_bw() +
  theme_minimal() +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(colour = "black"))
```

8.4.8 Modelo ingênuo

```
# Modelo ingênuo
res <- function(d, n){
  data.frame(
    day = str_sub(n, 2),
    mean_mpa = mean(d$train$y)
    )
}
ing_model <- imap(dats_reg, res)
get_rmse <- function(d, n){
   data.frame(
```

```
    rmse_train = RMSE(d$train$y, ing_model[[n]]$mean_mpa),
    rmse_test =  RMSE(d$test$y, ing_model[[n]]$mean_mpa),
    day = str_sub(n, 2)
    )
}
rmses <- imap(dats_reg, get_rmse) %>%
  reduce(rbind)
ing_model_df <- ing_model %>% reduce(rbind)
df_performance_reg <- full_join(ing_model_df, rmses)
```

8.4.9 Tabela - Modelos ingênuo

```
# Tabela - Modelos ingênuo
colNames = c("Idade", "Média $MPa$ (treino)", "RMSE (treino)", "RMSE (teste)")
caption <- "Modelos ingênuos"
kable(
    df_performance_reg,
    col.names = colNames,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    align = "c"
    )  %>%
  kable_styling(latex_options = c("HOLD_position"))
```

8.4.10 Escolha das caracterísitcas (features)

```
# Escolha das caracterísitcas (features)
data1 <- dats_reg$d28$train # full
data2 <- dats_reg$d28$train[c(1:13, 21)] # no dummy vars
data3 <- dats_reg$d28$train[c(1:7, 21)] # only original variables
data4 <- dats_reg$d28$train[c(8:21)] # only new variables
data5 <- dats_reg$d28$train[c(8:13, 21)] # only new varibels, no dummy vars
test1 <- dats_reg$d28$test # full
test2 <- dats_reg$d28$test[c(1:13, 21)] # no dummy vars
test3 <- dats_reg$d28$test[c(1:7, 21)] # only original variables
test4 <- dats_reg$d28$test[c(8:21)] # only new variables
test5 <- dats_reg$d28$test[c(8:13, 21)] # only new varibels, no dummy vars
# parRF
modelLookup("parRF")
control_parRF <- trainControl(method='repeatedcv',number=10,repeats=5,search='grid')
fit_parRF<- function(data, n) {
  set.seed(1, sample.kind = "Rounding")
  tunegrid_parRF<- expand.grid(mtry = seq(1, length(data), n))
  train(y ~ .,
    data = data,
    preProcess = c("center","scale"),
    method='parRF',
    tuneGrid=tunegrid_parRF,
    trControl=control_parRF)
```

```
}
fit_parRF1 <- fit_parRF(data1, 3)
ggplot(fit_parRF1)
min(fit_parRF1$results$RMSE) # 6.26108
p1_parRF1 <- predict(fit_parRF1, newdata = test1)
RMSE(p1_parRF1, test1$y) # 4.887264
fit_parRF2 <- fit_parRF(data2, 1)
ggplot(fit_parRF2)
min(fit_parRF2$results$RMSE) # 6.268789
p_parRF2 <- predict(fit_parRF2, newdata = test2)
RMSE(p_parRF2, test2$y) # 4.812994
fit_parRF3 <- fit_parRF(data3, 1)
ggplot(fit_parRF3)
min(fit_parRF3$results$RMSE) # 6.361905
p_parRF3 <- predict(fit_parRF3, newdata = test3)
RMSE(p_parRF3, test3$y) # 5.083467
fit_parRF4 <- fit_parRF(data4, 1)
ggplot(fit_parRF4)
min(fit_parRF4$results$RMSE) # 8.705984
p_parRF4 <- predict(fit_parRF4, newdata = test4)
RMSE(p_parRF4, test4$y) # 8.083962
fit_parRF5 <- fit_parRF(data5, 1)
ggplot(fit_parRF5)
min(fit_parRF5$results$RMSE) # 8.691755
p_parRF5 <- predict(fit_parRF5, newdata = test5)
RMSE(p_parRF5, test5$y) # 7.991537
```

8.4.11 Tabela - Primeiras 6 amostras do conjunto de treino do modelo de 28 dias

```
# Tabela - Primeiras 6 amostras do conjunto de treino do modelo de 28 dias
caption <- "Primeiras 6 amostras do conjunto de treino do modelo de 28 dias"
colNames = c("Cimento", "E.G.A.F.", "C.Vol.", "Água",
             "Superp.", "A.Graúdo", "A.Miúdo",
             "Ág./", "A.M./", "A.G./",
             "A.M./","Ág./", "Ág./", "y")
dfUnits <- c("$kg/m^3$", "$kg/m^3$","$kg/m^3$","$kg/m^3$",
             "$kg/m^3$", "$kg/m^3$","$kg/m^3$", "Ci.",
             "Ci.", "Ci.", "A.G.", "A.G.", "Ag.M.", "$MPa$")
colNames2 = c("Features"=13, "Outcome"=1)
kable(
    head(data2),
    col.names = dfUnits,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    digits = 2,
    align = "c"
    ) %>%
    add_header_above(header = colNames, line = F, align = "c") %>%
    add_header_above(header = colNames2, line = T, align = "c") %>%
    kable_styling(latex_options = c("HOLD_position", "scale_down"))
```

8.4.12 Modelos de regressão

```
# Modelos de regressão
# no dummy vars:
data3 <- dats_reg$d3$train[c(1:13, 22)]
data7 <- dats_reg$d7$train[c(1:12, 19)]
data14 <- dats_reg$d14$train[c(1:13, 21)]
data28 <- dats_reg$d28$train[c(1:13, 21)]
data56 <- dats_reg$d56$train[c(1:13, 22)]
data100 <- dats_reg$d100$train[c(1:13, 23)]
test3 <- dats_reg$d3$test[c(1:13, 22)]
test7 <- dats_reg$d7$test[c(1:12, 19)]
test14 <- dats_reg$d14$test[c(1:13, 21)]
test28 <- dats_reg$d28$test[c(1:13, 21)]
test56 <- dats_reg$d56$test[c(1:13, 22)]
test100 <- dats_reg$d100$test[c(1:13, 23)]
get_trControl <- function(n, r){
  trainControl(method='repeatedcv',number=n,repeats=r,search='grid')
}
trControl3 <- get_trControl(30, 10)
trControl7 <- get_trControl(10, 10)
trControl14 <- get_trControl(30, 10)
trControl28 <- get_trControl(30, 10)
trControl56 <- get_trControl(30, 10)
trControl100 <- get_trControl(10, 10)
get_tuneGrid <- function(data){
  expand.grid(mtry = seq(1, length(data), 1))
}
tuneGrid3 <- get_tuneGrid(data3)
tuneGrid7 <- get_tuneGrid(data7)
tuneGrid14 <- get_tuneGrid(data14)
tuneGrid28 <- get_tuneGrid(data28)
tuneGrid56 <- get_tuneGrid(data56)
tuneGrid100 <- get_tuneGrid(data100)
set.seed(1, sample.kind = "Rounding")
fit_3 <- train(y ~ .,
               data = data3,
               preProcess = c("center","scale"),
               method='parRF',
               tuneGrid=tuneGrid3,
               trControl=trControl3)
p_3 <- predict(fit_3, newdata = test3)
RMSE_test_3 <- RMSE(p_3, test3$y)
RMSE_test_3 # 3.31037
fit_3$bestTune # mtry = 6
set.seed(1, sample.kind = "Rounding")
fit_7 <- train(y ~ .,
               data = data7,
               preProcess = c("center","scale"),
               method='parRF',
               tuneGrid=tuneGrid7,
               trControl=trControl7)
p_7 <- predict(fit_7, newdata = test7)
```

```
RMSE_test_7 <- RMSE(p_7, test7$y)
RMSE_test_7 # 4.361987
fit_7$bestTune # mtry = 2
set.seed(1, sample.kind = "Rounding")
fit_14 <- train(y ~ .,
            data = data14,
            preProcess = c("center","scale"),
            method='parRF',
            tuneGrid=tuneGrid14,
            trControl=trControl14)
p_14 <- predict(fit_14, newdata = test14)
RMSE_test_14 <- RMSE(p_14, test14$y)
RMSE_test_14 # 4.620515
fit_14$bestTune # mtry = 13
set.seed(1, sample.kind = "Rounding")
fit_28 <- train(y ~ .,
            data = data28,
            preProcess = c("center","scale"),
            method='parRF',
            tuneGrid=tuneGrid28,
            trControl=trControl28)
p_28 <- predict(fit_28, newdata = test28)
RMSE_test_28 <- RMSE(p_28, test28$y)
RMSE_test_28 # 4.716698
fit_28$bestTune # mtry = 11
set.seed(1, sample.kind = "Rounding")
fit_56 <- train(y ~ .,
            data = data56,
            preProcess = c("center","scale"),
            method='parRF',
            tuneGrid=tuneGrid56,
            trControl=trControl56)
p_56 <- predict(fit_56, newdata = test56)
RMSE_test_56 <- RMSE(p_56, test56$y)
RMSE_test_56 # 5.939163
fit_56$bestTune # mtry = 8
set.seed(1, sample.kind = "Rounding")
fit_100 <- train(y ~ .,
            data = data100,
            preProcess = c("center","scale"),
            method='parRF',
            tuneGrid=tuneGrid100,
            trControl=trControl100)
p_100 <- predict(fit_100, newdata = test100)
RMSE_test_100 <- RMSE(p_100, test100$y)
RMSE_test_100 # 5.851088
fit_100$bestTune # mtry = 8
```

8.5 Resultados

8.5.1 Tabela - Detalhes dos modelos

```
# Tabela - Detalhes dos modelos
colNames <- c("Modelo", "mtry", "CV", "Repetições",
              "RMSE (treino)", "RMSE (teste)")
caption <- "Resultados dos modelos de regressão"
day <- c("3 dias", "7 dias", "14 dias", "28 dias", "56 dias", "100 dias")
dat_reg_models <- data.frame(
  dia = day,
  mtry = c(fit_3$bestTune$mtry, fit_7$bestTune$mtry, fit_14$bestTune$mtry,
           fit_28$bestTune$mtry, fit_56$bestTune$mtry, fit_100$bestTune$mtry),
  number = c(trControl3$number, trControl7$number, trControl14$number,
             trControl28$number,trControl56$number,trControl100$number),
  repeats = c(trControl3$repeats, trControl7$repeats, trControl14$repeats,
              trControl28$repeats,trControl56$repeats,trControl100$repeats),
  RMSE_train = c(min(fit_3$results$RMSE), min(fit_7$results$RMSE),
                 min(fit_14$results$RMSE), min(fit_28$results$RMSE),
                 min(fit_56$results$RMSE), min(fit_100$results$RMSE)),
  RMSE_test = c(RMSE_test_3, RMSE_test_7, RMSE_test_14,
                RMSE_test_28, RMSE_test_56, RMSE_test_100)
)
kable(
    dat_reg_models,
    col.names = colNames,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    align = "c"
    )  %>%
  kable_styling(latex_options = c("HOLD_position"))
```

8.5.2 Figura - Comparação dos modelos

```
# Figura - Comparação dos modelos
cap <- "Comparação dos valores reais e previstos em cada modelo"
models <- c("3 dias", "7 dias", "14 dias", "28 dias", "56 dias", "100 dias")
xlabel <- "Real (MPa)"
ylabel <- "Previsto (MPa)"
preds <- list(p_3, p_7, p_14, p_28, p_56, p_100)
actuals <- list(test3$y, test7$y, test14$y, test28$y, test56$y, test100$y)
gen_res_df <- function(ind){
  data.frame(actual = actuals[[ind]], pred = preds[[ind]], model = models[[ind]])
}
res_df <- lapply(1:6, gen_res_df)
res_df <- bind_rows(res_df)
res_df$model <- factor(res_df$model,levels=models)
res_df %>%
  ggplot(aes(actual, pred)) +
  facet_wrap(~ model, ncol=3) +
  geom_point(alpha=0.5) +
```

```
  theme_bw() +
  geom_abline(slope=1, intercept=0) +
  xlab(xlabel) +
  ylab(ylabel)
```

8.5.3 Tabelas dos 10 melhores e piores resultados

```
# Tabelas dos 10 melhores e piores resultados
colNames1 = c("Real", "Previsto", "Erro", "","Real", "Previsto", "Erro")
colNames2 = c("10 melhores"=3,"", "10 piores"=3)
caption_0 <- "Modelo de "
get_X <- function(pred, actual, X=10){
  diff <- abs(pred - actual)
  diff_2 <- pred - actual
  ind_min <- which(diff <= max(sort(diff, decreasing = F)[1:X]), arr.ind = T)
  ind_max <- which(diff >= min(sort(diff, decreasing = T)[1:X]), arr.ind = T)
  df_min <- data.frame(
    actual_min=actual[ind_min],
    pred_min=pred[ind_min],
    diff_min=diff[ind_min],
    diff_min_2=diff_2[ind_min]
  )
  df_max <- data.frame(
    actual_max=actual[ind_max],
    pred_max=pred[ind_max],
    diff_max=diff[ind_max],
    diff_max_2=diff_2[ind_max]
  )
  df_max <- df_max[order(-df_max$diff_max),]
  df_min <- df_min[order(df_min$diff_min),]
  df_null <- data.frame(null_col = rep(c(""), X))
  res <- cbind(df_min,df_null, df_max)
  res <- res %>% select(-c(diff_max, diff_min))
  rownames(res) <- c()
  res
}
gen_kable <- function(ind){
  df <- get_X(preds[[ind]], actuals[[ind]])
  caption <- paste0(caption_0,models[ind])
  kable(
    df,
    col.names = colNames1,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    align = "c"
    ) %>%
  column_spec(4, width = "1cm",) %>%
  add_header_above(header = colNames2, line = T, align = "c") %>%
  kable_styling(latex_options = c("HOLD_position"))
}
gen_kable(1)
```

```
gen_kable(2)
gen_kable(3)
gen_kable(4)
gen_kable(5)
gen_kable(6)
```

8.6 Discussão

8.6.1 Tabela - "Comparação dos estudos de outros autores"

```
# Tabela - "Comparação dos estudos de outros autores"
colNames = c("Autor","Ano" ,"Algorítimo", "RMSE")
caption <- "Comparação dos estudos de outros autores"
works <- data.frame(
  autor = c("Pierobon", "Hameed", "Raj","Modukuru" ,"Alshamiri", "Abban"),
  ano = c("2018", "2020", "2018","2020" ,"2020", "2016"),
  modelo = c("Ensemble com 5 algorítimos", "Artificial Neural Networks",
          "Gradient Boosting Regressor","Random Forest Regressor" ,
          "Regularized Extreme Learning Machine",
          "Support Vector Machines with Radial Basis Function Kernel"),
  RMSE = c("4.150", "4.736", "4.957","5.080","5.508", "6.105")
)
kable(
  works,
  col.names = colNames,
  escape = F,
  booktabs = T,
  caption = caption,
  linesep = "\\addlinespace",
  align = "l"
  )  %>%
    kable_styling(latex_options = c("HOLD_position"))
```

8.6.2 Tabela - Resultados finais

```
# Tabela - Resultados finais
colNames <- c("Modelo", "RMSE")
caption <- "Resultados finais"
day <- c("3 dias", "7 dias", "14 dias", "28 dias", "56 dias", "100 dias")
dat_reg_models <- data.frame(
  dia = day,
  RMSE_test = c(RMSE_test_3, RMSE_test_7, RMSE_test_14,
                RMSE_test_28, RMSE_test_56, RMSE_test_100)
)
kable(
    dat_reg_models,
    col.names = colNames,
    escape = F,
    booktabs = T,
    caption = caption,
    linesep = "\\addlinespace",
    align = "c"
```

```
  )  %>%
kable_styling(latex_options = c("HOLD_position"))
```