eAppendix 1

We used as predictors of life expectancy at birth 60 publicly available variables that are not directly controlled by health managers. The list, the formula and the source are presented below.

Variable	Formula	Source
Automobile ownership	Number of households with an automobile / Total households	2010 Census
Bolsa Familia coverage	Number of families receiving Bolsa Familia / Total families	Health Ministry, 2010
Child labor	Number of residents from 10 to 15 years old that work / Residents from 10 to 15	2010 Census
College education	Number of residents over 25 that completed college / Residents over 25	2010 Census
Commuting	Number of residents > 10 years old that work outside of municipality / Total residents	2010 Census
Computer ownership	Number of households with a computer / Total households	2010 Census
Dependency ratio	Number of residents < 15 + > 60 years old / Residents between 15 and 59 years old	2010 Census
Disability rate	Number of residents that are disabled / Total residents	2010 Census
Elderly	Number of residents > 59 years old / Total residents	2010 Census
Electricity	Number of houselholds with electricity / Total households	2010 Census
Evangelicals	Number of evangelical residents / Total residents	2010 Census
Favela (slums) residents	Number of residents living in favelas / Total residents	2010 Census
Foreigners	Number of residents that were not born in Brazil / Total residents	2010 Census
Fridge	Number of households with fridges / Total households	2010 Census
Gini Coefficient	Calculated and adjusted by the Instituto de Pesquisas Econômicas e Aplicadas (IPEA)	IPEA, 2010
Green spaces	Number of residents living in arborized areas / Total residents	2010 Census
Highschool completion	Number of residents that completed high school (over 25 years old) / Residents over 25	2010 Census
Household density	Number of residents / Total households	2010 Census
Illiteracy race	Number of illiterate residents (over 10 years old) / Residents over 10 years old	2010 Census
Married	Number of married residents / Total residents	2010 Census
Median age	Median age of residents	2010 Census
Median income	Median income of residents > 10 years old	2010 Census
Migrants	Number of residents living for less than 10 years at the municipality / Total residents	2010 Census
Municipal density	Number of residents / Total km2 of municipality	2010 Census
Overworking	Number of residents that work 45+ hours a week / Total residents	2010 Census
Paved street	Number of residents living in paved streets / Total residents	2010 Census
Per capita births	Number of births / Total residents	Health Ministry, 2010
Per capita GDP	GDP of the municipality / Total residents	IBGE, 2010
Poor children	Number of poor children (under 15 years old) / Residents under 15	2010 Census
Residents	Total residents	2010 Census
Retired residents	Number of retired residents / Total residents	2010 Census
State of residence *	Dummy variable for each State	2010 Census
Unemployment	Number of residents > 15 years old unemployed / Residents > 15 years old	2010 Census
Urban area	Number of residents living in urban areas / Total residents	2010 Census
Whites	Number of white residents / Total residents	2010 Census
Women	Number of women / Total residents	2010 Census

* 25 dummy variables were included, representing each state. The most populous, São Paulo, was the reference group and the Federal District was included within Goiás.

eAppendix 2

We tested the predictive performance of 16 popular machine learning algorithms to predict life expectancy of Brazilian municipalities using only socioeconomic characteristics. All were trained and had their hyperparameters tuned using the packages *caret* and *SuperLearner* from R (Kuhn & Johnson, 2013). The list of algorithms and their original libraries are presented below.

Algorithm	Original method/library
Artificial neural networks	nnet
Cubist trees	cubist
Elastic Net	enet/elasticnet
Gradient boosted trees	xgbTree/xgboost
K-nearest neighbors	knn
Lasso regression	lasso/elasticnet
Linear regression	lm
Multivariate adaptive regression spline	earth
Partial least squares	pls
Random forest	rf
Regression tree with complexity tuning	rpart
Regression tree with depth tuning	rpart2
Ridge regression	ridge/elasticnet
SuperLearner	SuperLearner
Support vector machines with linear kernel	svmLinear/kernlab
Support vector machines with polynomial ker	svmPoly/kernlab
Support vector machines with radial kernel	svmRadial/kernlab

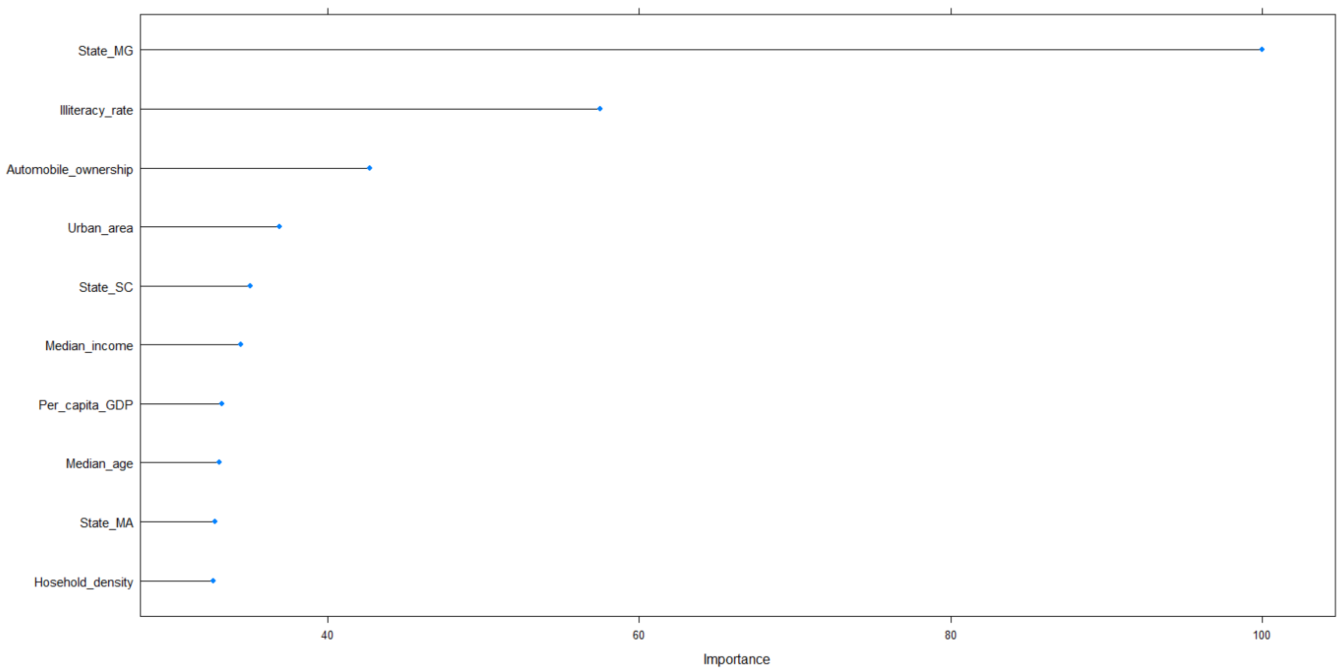
eAppendix 3

After predicting life expectancy at birth for each of the 3,052 municipalities using only socioeconomic characteristics, we separated them into tertiles of life expectancy at birth. We then identified the Top 100 underachievers and the Top 100 overachievers (i.e. the 100 municipalities that had lowest life expectancy than predicted, and the 100 that had highest than predicted) for each tertile, and compared differences in healthcare characteristics, listed below.

Variable	Formula	Source
Caesarean deliveries	Live births by caesarean delivery / Total live births	Health Ministry, 201
Family Health Strategy teams per 10,000 resi	Number of Family Health Strategy teams / Total residents	Health Ministry, 201
Hospital beds per 10,000 residents	Total hospital beds / Total residents	Health Ministry, 201
Life support equipment per 10,000 residents	Total life support equipments / Total residents	Health Ministry, 201
Low birth weight (%)	Number of live births < 2,500g / Total live births	Health Ministry, 201
Mammographies per 100 women	Mammographies performed on women aged 50 to 69 / Women aged 50 to 69	Health Ministry, 201
Oral Health Strategy coverage	Oral Health teams from the Family Health Strategy / Total residents	Health Ministry, 201
Primary health coverage for poor residents	Families on Bolsa Família with primary care follow-up (last year) / Families on Bolsa F	Health Ministry, 201
Ultrasound machines per 10,000 live births	Ultrasound machines / Total live births	Health Ministry, 201
Vaccination coverage	Residents on par with the national immunization list / Target population for the vaccine	Health Ministry, 201
X-ray machines per 10,000 residents	X-ray machines / Total residents	Health Ministry, 201

eAppendix 4

Variable importance for predictive models should not be mistaken for causation or even association as applied to inference problems. For random forests, variable importance is calculated by the linear combination of the usage of each variable in the rule conditions and the model. We assessed variable importance with the varImp() function of the caret package (Kuhn & Johnson, 2013).



eAppendix 5

Top 100 under and overachievers for the **first** tertile of life expectancy at birth.

Underachievers					Overachievers				
Municipality	State	Observed value	SuperLearner predicted value	Difference	Municipality	State	Observed value	SuperLearner predicted value	Difference
Joaquim Nabuco	PE	65.55	69.72	-4.17	Boa Viagem	CE	71.09	70.08	1.01
Saúde	BA	66.12	70.25	-4.13	Arataca	BA	70.46	69.45	1.01
Terra Nova	BA	67.02	71.01	-3.99	Guajeru	BA	71.20	70.18	1.02
Várzea Nova	BA	66.12	70.04	-3.92	Major Isidoro	AL	70.31	69.29	1.02
Neópolis	SE	67.13	71.02	-3.89	Garrafão do Norte	PA	70.75	69.72	1.03
Custódia	PE	67.33	71.20	-3.87	Buriti Bravo	MA	70.10	69.06	1.04
Monteiro	PB	67.51	71.37	-3.86	Potengi	CE	70.54	69.50	1.04
Iaçu	BA	66.81	70.59	-3.78	Buíque	PE	69.75	68.71	1.04
Itapetim	PE	67.25	70.91	-3.66	Santa Maria da Boa Vista	PE	71.39	70.35	1.04
Una	BA	67.61	71.26	-3.65	Anajatuba	MA	70.69	69.64	1.05
Ribeirão	PE	67.68	71.29	-3.61	Santana do Ipanema	AL	71.17	70.09	1.08
Juripiranga	PB	65.64	69.20	-3.56	Guimarães	MA	71.31	70.22	1.09
Mirangaba	BA	66.12	69.67	-3.55	Ouricuri	PE	71.40	70.30	1.10
Bonito	BA	66.12	69.65	-3.53	Pilar	AL	71.26	70.15	1.11
Japoatã	SE	67.01	70.53	-3.52	Araripe	CE	70.54	69.42	1.12
Mari	PB	66.50	69.95	-3.45	Araci	BA	69.79	68.67	1.12
Tobias Barreto	SE	67.13	70.48	-3.35	Pedras de Fogo	PB	71.23	70.11	1.12
Lagoa Grande	PE	67.28	70.60	-3.32	Joselândia	MA	69.59	68.46	1.13
Ribeira do Pombal	BA	67.25	70.45	-3.20	Exu	PE	71.47	70.34	1.13
Jurema	PE	65.87	69.02	-3.15	Anajás	PA	71.43	70.30	1.13
Fátima	BA	66.16	69.30	-3.14	Santa Luzia	BA	71.07	69.93	1.14
Utinga	BA	67.38	70.50	-3.12	Atalaia	AL	70.13	68.99	1.14
Valente	BA	68.75	71.85	-3.10	Brejo	MA	70.45	69.31	1.14
Jijoca de Jericoacoara	CE	68.35	71.45	-3.10	Porto	PI	70.47	69.32	1.15
Dona Inês	PB	66.41	69.49	-3.08	Banzaê	BA	71.15	70.00	1.15
Paulo Ramos	MA	65.64	68.70	-3.06	Ingá	PB	71.07	69.92	1.15
Acarape	CE	67.56	70.60	-3.04	Serra do Ramalho	BA	71.51	70.36	1.15
Olivença	AL	65.63	68.65	-3.02	Machados	PE	70.99	69.83	1.16
Quijingue	BA	66.36	69.33	-2.97	Bom Jardim	MA	69.98	68.79	1.19
Murici	AL	66.11	69.08	-2.97	Nova Olinda do Maranhão	MA	70.32	69.12	1.20
Itagi	BA	66.62	69.56	-2.94	Viçosa	AL	70.46	69.26	1.20
Paripiranga	BA	68.02	70.96	-2.94	Lagoa de Itaenga	PE	71.44	70.22	1.22
Princesa Isabel	PB	68.19	71.12	-2.93	União dos Palmares	AL	70.83	69.61	1.22
João Dourado	BA	67.52	70.44	-2.92	Humberto de Campos	MA	70.55	69.33	1.22
Condado	PE	67.79	70.71	-2.92	Jaqueira	PE	70.07	68.83	1.24
Conde	BA	67.31	70.22	-2.91	Ipecaetá	BA	71.53	70.28	1.25
Umburanas	BA	67.31	70.22	-2.91	Crisópolis	BA	70.87	69.59	1.28

Penalva	MA	66.57	69.48	-2.91	Dário Meira	BA	71.27	69.99	1.28
Igarapé do Meio	MA	66.21	69.10	-2.89	Pariconha	AL	70.23	68.94	1.29
Arame	MA	67.06	69.93	-2.87	Simões	PI	71.24	69.93	1.31
São Gabriel	BA	67.21	70.07	-2.86	Capitão de Campos	PI	71.35	70.04	1.31
Duque Bacelar	MA	66.56	69.41	-2.85	Baixa Grande	BA	71.49	70.18	1.31
Tapauá	AM	66.61	69.43	-2.82	Barra do Corda	MA	70.75	69.44	1.31
Governador Archer	MA	66.21	69.03	-2.82	Chapadinha	MA	71.21	69.88	1.33
Jitaúna	BA	67.25	70.03	-2.78	Itapiúna	CE	71.49	70.14	1.35
Tacaratu	PE	67.63	70.41	-2.78	Loreto	MA	71.27	69.92	1.35
Peixoto de Azevedo	MT	70.65	73.41	-2.76	Augusto Corrêa	PA	71.30	69.92	1.38
Luzilândia	PI	67.00	69.76	-2.76	Santa Bárbara	BA	71.50	70.12	1.38
Montanhas	RN	67.02	69.76	-2.74	Caldeirão Grande	BA	70.46	69.07	1.39
São João do Carú	MA	66.01	68.75	-2.74	Parnarama	MA	70.50	69.11	1.39
Caém	BA	67.08	69.80	-2.72	Olho dÁgua das Cunhãs	MA	70.24	68.84	1.40
Bom Conselho	PE	67.22	69.90	-2.68	Araruna	PB	70.45	69.02	1.43
Cedro	PE	68.00	70.68	-2.68	Sítio do Mato	BA	71.46	70.02	1.44
Afonso Bezerra	RN	67.76	70.44	-2.68	Conceição do LagoAçu	MA	69.26	67.82	1.44
Salinas da Margarida	BA	68.74	71.41	-2.67	Guaratinga	BA	70.95	69.51	1.44
Jeremoabo	BA	66.36	69.03	-2.67	Presidente Vargas	MA	71.10	69.64	1.46
Sapé	PB	67.64	70.31	-2.67	Quiterianópolis	CE	71.46	70.00	1.46
Vitorino Freire	MA	66.26	68.92	-2.66	Mulungu do Morro	BA	70.18	68.71	1.47
Barra da Estiva	BA	67.99	70.63	-2.64	Urbano Santos	MA	71.21	69.72	1.49
Canto do Buriti	PI	68.41	71.01	-2.60	Santana do Mundaú	AL	70.13	68.64	1.49
Santa Inês	BA	67.25	69.83	-2.58	Santana do Maranhão	MA	70.45	68.94	1.51
Senador Alexandre Costa	MA	66.32	68.89	-2.57	Brejões	BA	71.49	69.98	1.51
Itaíba	PE	66.25	68.81	-2.56	Ipixuna	AM	71.32	69.78	1.54
Inácio Martins	PR	70.91	73.46	-2.55	São Bernardo	MA	70.70	69.15	1.55
Manaíra	PB	66.85	69.40	-2.55	Paratinga	BA	71.46	69.91	1.55
Camamu	BA	67.61	70.14	-2.53	Anagé	BA	71.27	69.70	1.57
Ourolândia	BA	67.03	69.54	-2.51	Poço Redondo	SE	70.58	68.99	1.59
Poção	PE	65.59	68.09	-2.50	Várzea da Roça	BA	71.50	69.88	1.62
Tucano	BA	67.69	70.19	-2.50	Pilar	PB	71.52	69.89	1.63
Jucati	PE	65.87	68.36	-2.49	Santa Quitéria do Maranhão	MA	70.61	68.97	1.64
Craíbas	AL	66.22	68.67	-2.45	Salgado de São Félix	PB	71.20	69.55	1.65
Saboeiro	CE	67.87	70.31	-2.44	Presidente Jânio Quadros	BA	71.29	69.63	1.66
Sítio do Quinto	BA	66.69	69.13	-2.44	Matões do Norte	MA	71.10	69.44	1.66
Teotônio Vilela	AL	67.01	69.42	-2.41	São João	PE	71.09	69.42	1.67
Touros	RN	67.95	70.36	-2.41	Anapurus	MA	70.85	69.16	1.69
Cajueiro	AL	66.45	68.86	-2.41	Primavera	PA	71.30	69.61	1.69
Cortês	PE	67.42	69.83	-2.41	Jutaí	AM	70.93	69.23	1.70
Itaquitinga	PE	67.87	70.27	-2.40	Morro do Chapéu	BA	70.98	69.27	1.71
Pedro do Rosário	MA	66.77	69.17	-2.40	Igreja Nova	AL	71.23	69.46	1.77

Itaquiraí	MS	71.30	73.70	-2.40	Dois Riachos	AL	70.74	68.95	1.79
Wanderlândia	TO	69.89	72.26	-2.37	Quebrangulo	AL	71.52	69.70	1.82
Ipaumirim	PE	65.89	68.22	-2.33	Vargem Grande	MA	71.10	69.28	1.82
Regeneração	AL	66.09	68.41	-2.32	Itiúba	BA	71.47	69.65	1.82
São Luís do Curu	PR	71.05	73.37	-2.32	Gonçalves Dias	MA	70.46	68.61	1.85
Sapeaçu	AL	67.85	70.17	-2.32	Pimenteiras	PI	71.43	69.53	1.90
Costa Marques	AM	66.61	68.91	-2.30	Jequiá da Praia	AL	71.32	69.40	1.92
Buritizeiro	AL	67.01	69.30	-2.29	Jenipapo dos Vieiras	MA	70.96	68.95	2.01
Avelino Lopes	MA	66.81	69.10	-2.29	Cantanhede	MA	71.45	69.42	2.03
Manari	PI	68.30	70.57	-2.27	Maragogi	AL	70.96	68.92	2.04
Águas Belas	BA	69.07	71.33	-2.26	Traipu	AL	70.92	68.84	2.08
Mata Grande	PE	67.87	70.13	-2.26	Caracol	PI	71.23	69.13	2.10
Carnaíba	MS	71.30	73.54	-2.24	Alto Alegre do Pindaré	MA	71.44	69.33	2.11
Curimatá	BA	68.08	70.32	-2.24	Nina Rodrigues	MA	71.45	69.22	2.23
Ribeira do Amparo	PE	68.62	70.84	-2.22	Barra do Choça	BA	71.21	68.87	2.34
Varjota	RN	67.95	70.16	-2.21	Peritoró	MA	71.44	69.03	2.41
Ibateguara	AL	66.06	68.26	-2.20	Girau do Ponciano	AL	70.72	68.31	2.41
Groaíras	MA	67.33	69.53	-2.20	Iati	PE	71.09	68.46	2.63
Umbaúba	BA	66.66	68.85	-2.19	Igaci	AL	71.45	68.79	2.66
Cantagalo	AM	68.13	70.32	-2.19	Joaquim Gomes	AL	70.82	67.72	3.10
Jaicós	CE	68.72	70.91	-2.19	Turilândia	MA	71.45	68.21	3.24

Top 100 under and overachievers for the **second** tertile of life expectancy at birth.

Underachievers					Overachievers				
Municipality	State	Observed value	SuperLearner predicted value	Difference	Municipality	State	Observed value	SuperLearner predicted value	Difference
Cunha Porã	SC	73.24	76.32	-3.08	Varzelândia	MG	73.37	71.78	1.59
São Lourenço do Oeste	SC	73.43	76.41	-2.98	Nova Timboteua	PA	72.43	70.84	1.59
João Pinheiro	MG	72.25	75.15	-2.90	Cotriguaçu	MT	74.51	72.91	1.60
Paraopeba	MG	72.55	75.16	-2.61	Pedra Branca	CE	71.73	70.12	1.61
Bady Bassitt	SP	73.31	75.90	-2.59	Axixá	MA	71.54	69.93	1.61
Itaperuçu	PR	71.72	74.30	-2.58	Maraú	BA	73.08	71.47	1.61
São Manuel	SP	73.29	75.87	-2.58	Abaetetuba	PA	72.89	71.27	1.62
Araruna	PR	72.08	74.64	-2.56	Acrelândia	AC	73.45	71.83	1.62
Rio Piracicaba	MG	72.63	75.16	-2.53	Vera Cruz	BA	74.02	72.40	1.62
Ponte Serrada	SC	72.40	74.88	-2.48	Candeias do Jamari	RO	74.11	72.49	1.62
Lontras	SC	74.20	76.51	-2.31	Ubatã	BA	72.29	70.66	1.63
Rio dos Cedros	SC	74.61	76.89	-2.28	Amélia Rodrigues	BA	73.59	71.95	1.64
Correia Pinto	SC	73.21	75.44	-2.23	Massapê	CE	71.70	70.05	1.65
Sinimbu	RS	72.50	74.73	-2.23	Brasil Novo	PA	73.55	71.90	1.65
Planalto	PR	72.16	74.32	-2.16	Tavares	PB	71.59	69.93	1.66
Jandaia do Sul	PR	73.04	75.19	-2.15	Alhandra	PB	71.69	70.03	1.66
Bom Jesus dos Perdões	SP	73.10	75.24	-2.14	Orobó	PE	72.19	70.52	1.67
Tocantins	MG	72.99	75.08	-2.09	Queimadas	PB	72.73	71.06	1.67
General Salgado	SP	73.50	75.55	-2.05	Santana do Cariri	CE	71.72	70.04	1.68
BiritibaMirim	SP	72.70	74.75	-2.05	Umarizal	RN	72.49	70.81	1.68
Morungaba	SP	73.10	75.14	-2.04	Alto Paraíso	RO	73.22	71.54	1.68
Palmital	SP	73.83	75.84	-2.01	Itatim	BA	71.93	70.24	1.69
Vespasiano	MG	73.65	75.65	-2.00	Miraíma	CE	71.70	70.00	1.70
Cajamar	SP	73.59	75.58	-1.99	Macaparana	PE	71.70	70.00	1.70
Anastácio	MS	72.33	74.31	-1.98	Boninal	BA	72.35	70.64	1.71
Pederneiras	SP	73.74	75.72	-1.98	Chapada Gaúcha	MG	74.22	72.50	1.72
Carmo da Cachoeira	MG	73.03	74.98	-1.95	Tamboril	CE	71.71	69.97	1.74
Bela Vista de Minas	MG	72.72	74.66	-1.94	Jacareacanga	PA	73.00	71.24	1.76
Itapeva	SP	73.18	75.11	-1.93	Santo Antônio	RN	71.94	70.16	1.78
Candói	PR	71.69	73.61	-1.92	Canavieiras	BA	72.74	70.96	1.78
Joaquim Távora	PR	73.04	74.94	-1.90	Campinápolis	MT	73.17	71.39	1.78
Matipó	MG	71.80	73.68	-1.88	Tartarugalzinho	AP	72.63	70.84	1.79
Pedralva	MG	73.39	75.27	-1.88	Maués	AM	72.97	71.17	1.80
Morro da Fumaça	SC	74.51	76.36	-1.85	Condeúba	BA	72.32	70.52	1.80
Entre Rios de Minas	MG	73.14	74.98	-1.84	João Lisboa	MA	72.07	70.26	1.81
Cesário Lange	SP	73.23	75.07	-1.84	Nossa Senhora do Livramento	MT	74.20	72.38	1.82
Naviraí	MS	73.15	74.99	-1.84	Salvaterra	PA	72.60	70.76	1.84
Sete Quedas	MS	71.70	73.54	-1.84	Barbalha	CE	74.02	72.18	1.84
Venâncio Aires	RS	74.09	75.91	-1.82	Itaparica	BA	74.56	72.68	1.88

Lauro Muller	SC	74.32	76.13	-1.81
Senhora dos Remédios	MG	71.56	73.34	-1.78
Teixeiras	MG	73.26	75.03	-1.77
Santa Margarida	MG	72.07	73.81	-1.74
São José do Norte	RS	72.52	74.24	-1.72
Tijucas do Sul	PR	72.54	74.26	-1.72
Brotas	SP	73.90	75.62	-1.72
São Luís de Montes Belos	GO	73.55	75.27	-1.72
Arroio Grande	RS	73.43	75.15	-1.72
Wenceslau Braz	PR	72.42	74.13	-1.71
Cacequi	RS	73.39	75.10	-1.71
São Joaquim	SC	74.01	75.71	-1.70
Camaquã	RS	74.14	75.80	-1.66
Catalão	GO	74.12	75.78	-1.66
Corbélia	PR	73.44	75.08	-1.64
Juquitiba	SP	72.48	74.12	-1.64
Aparecida do Taboado	MS	73.23	74.87	-1.64
São Miguel Arcanjo	SP	72.95	74.57	-1.62
Capanema	PR	73.04	74.64	-1.60
Barros Cassal	RS	72.71	74.28	-1.57
Mantena	MG	73.16	74.71	-1.55
Guaíra	SP	74.46	76.01	-1.55
São Bento do Sapucaí	SP	73.69	75.24	-1.55
Nova Alvorada do Sul	MS	73.52	75.07	-1.55
Nova Xavantina	MT	73.00	74.55	-1.55
Guararema	SP	74.03	75.57	-1.54
Três Rios	RJ	73.03	74.56	-1.53
Pinhalzinho	SP	73.49	75.02	-1.53
Colorado	PR	73.25	74.77	-1.52
Itamogi	MG	73.61	75.13	-1.52
Sertanópolis	PR	73.24	74.75	-1.51
Centenário do Sul	PR	72.48	73.99	-1.51
Mar de Espanha	MG	74.51	76.02	-1.51
Bom Sucesso	MG	73.49	75.00	-1.51
Valparaíso de Goiás	GO	73.91	75.41	-1.50
São Pedro da Aldeia	RJ	73.03	74.51	-1.48
Alto Araguaia	MT	73.13	74.61	-1.48
Faxinal	PR	72.75	74.23	-1.48
Crixás	GO	73.31	74.79	-1.48
Mococa	SP	74.63	76.10	-1.47
Taquarituba	SP	73.65	75.12	-1.47
Goiás	GO	73.27	74.73	-1.46
Arraial do Cabo	RJ	73.31	74.77	-1.46

Nova Cruz	RN	72.28	70.38	1.90
Cruz do Espírito Santo	PB	71.66	69.74	1.92
Matias Olímpio	PI	71.74	69.81	1.93
Orós	CE	72.58	70.64	1.94
Macaúbas	BA	72.43	70.48	1.95
Milagres	BA	72.32	70.37	1.95
Autazes	AM	72.91	70.96	1.95
Santa Cruz da Baixa Verde	PE	72.14	70.18	1.96
Afogados da Ingazeira	PE	73.39	71.38	2.01
Maragogipe	BA	72.58	70.55	2.03
Bonito de Santa Fé	PB	72.18	70.15	2.03
Amargosa	BA	73.40	71.36	2.04
Sairé	PE	72.05	70.01	2.04
Granja	CE	71.68	69.64	2.04
GuajaráMirim	RO	74.39	72.32	2.07
Igrapiúna	BA	72.32	70.22	2.10
São Bento do Una	PE	72.34	70.23	2.11
Mundo Novo	BA	72.43	70.31	2.12
São Sebastião de Lagoa de Roça	PB	72.67	70.52	2.15
São Sebastião do Maranhão	MG	73.94	71.78	2.16
Taipu	RN	71.68	69.51	2.17
Mata Roma	MA	71.56	69.38	2.18
Iaciara	GO	74.53	72.34	2.19
Itacaré	BA	73.42	71.23	2.19
Cacimba de Dentro	PB	71.59	69.35	2.24
Triunfo	PE	74.00	71.75	2.25
São José do Campestre	RN	72.34	70.09	2.25
Aliança	PE	72.93	70.67	2.26
Conceição do Almeida	BA	73.31	71.02	2.29
Santo Estêvão	BA	73.47	71.18	2.29
Macajuba	BA	71.81	69.50	2.31
Carira	SE	72.45	70.12	2.33
Paranhos	MS	73.65	71.31	2.34
Coremas	PB	72.65	70.31	2.34
Iguaí	BA	72.18	69.84	2.34
Curionópolis	PA	73.53	71.18	2.35
Poções	BA	72.85	70.49	2.36
Piranhas	AL	72.14	69.75	2.39
Remígio	PB	72.83	70.40	2.43
Altinho	PE	72.54	70.10	2.44
Nova Canaã	BA	72.35	69.89	2.46
Turiaçu	MA	71.54	69.07	2.47
Ibirataia	BA	72.33	69.84	2.49

Jacinto Machado	SC	74.06	75.52	-1.46	Sena Madureira	AC	73.59	71.04	2.55
Valentim Gentil	SP	73.62	75.08	-1.46	Chã Grande	PE	73.18	70.53	2.65
Santo Cristo	RS	74.20	75.66	-1.46	Cândido Sales	BA	72.90	70.24	2.66
Quedas do Iguaçu	PR	72.65	74.10	-1.45	Cocos	BA	73.28	70.62	2.66
Aimorés	MG	73.58	75.01	-1.43	Brejo do Cruz	PB	72.85	70.15	2.70
Maceió	AL	72.94	74.37	-1.43	Panelas	PE	71.68	68.96	2.72
Porto Ferreira	SP	74.63	76.06	-1.43	Alagoa Grande	PB	72.83	70.01	2.82
Miranda	MS	71.91	73.34	-1.43	Alagoinha	PE	73.04	70.21	2.83
Santa Cruz das Palmeiras	SP	73.87	75.29	-1.42	Coribe	BA	73.44	70.54	2.90
Cachoeira de Minas	MG	74.25	75.67	-1.42	Alagoinha	PB	72.72	69.81	2.91
Regente Feijó	SP	74.10	75.52	-1.42	Pedro Alexandre	BA	71.68	68.77	2.91
São Pedro do Sul	RS	74.45	75.85	-1.40	Irará	BA	73.53	70.59	2.94
Céu Azul	PR	73.69	75.08	-1.39	Rio Formoso	PE	73.56	70.60	2.96
Itirapina	SP	73.77	75.15	-1.38	Pão de Açúcar	AL	72.58	69.57	3.01
Rosana	SP	74.10	75.48	-1.38	Matriz de Camaragibe	AL	71.62	68.59	3.03
São Roque do Canaã	ES	73.16	74.53	-1.37	Feira Nova	PE	73.43	70.02	3.41
Guariba	SP	73.63	75.00	-1.37	Olindina	BA	73.60	70.02	3.58
Janaúba	MG	72.74	74.11	-1.37	Alto Alegre	RR	73.78	70.02	3.76

Top 100 under and overachievers for the **third** tertile of life expectancy at birth.

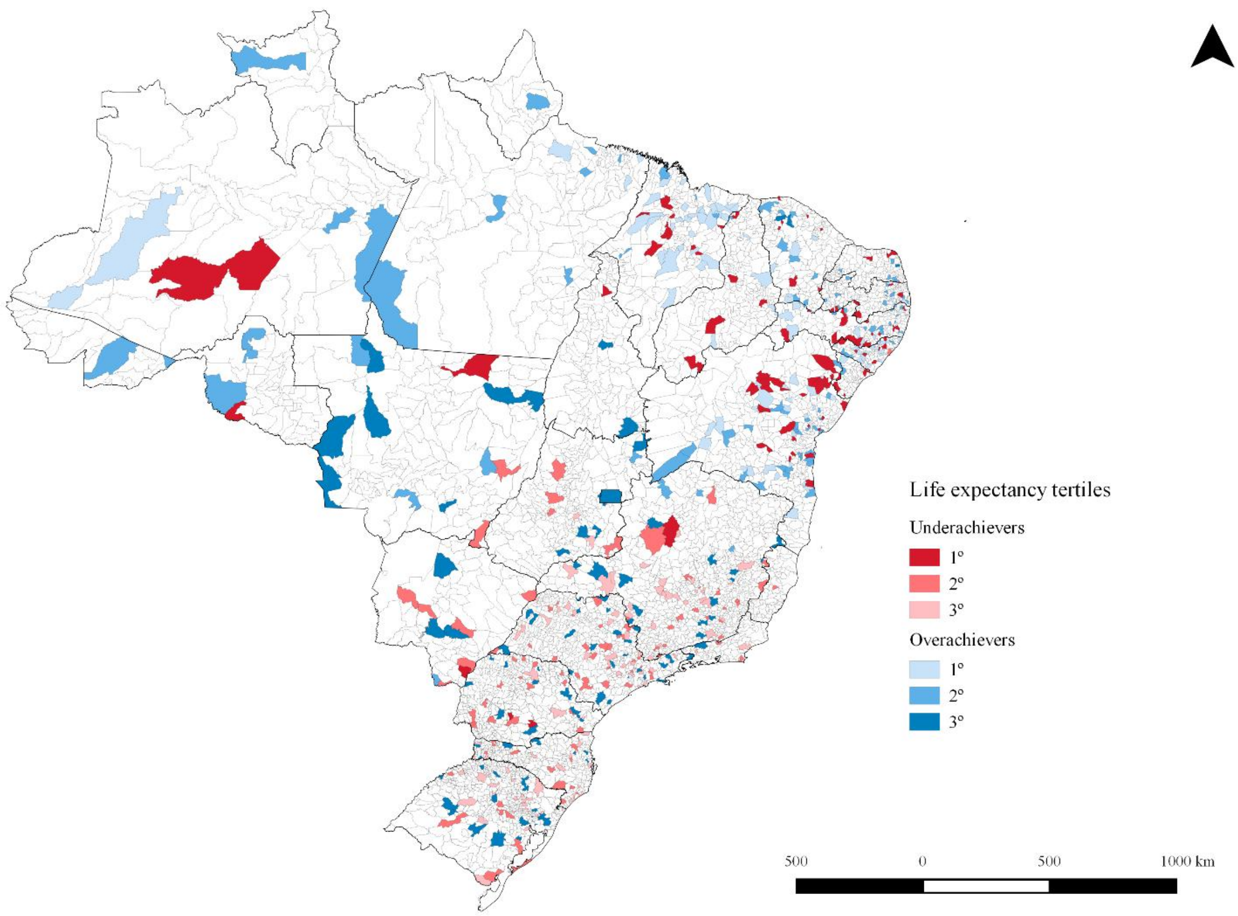
Underachievers					Overachievers				
Municipality	State	Observed value	SuperLearner predicted value	Difference	Municipality	State	Observed value	SuperLearner predicted value	Difference
Itabirito	MG	74.68	76.69	-2.01	Uberlândia	MG	78.09	76.61	1.48
São Ludgero	SC	75.01	76.71	-1.70	Comodoro	MT	75.37	73.89	1.48
Santa Rita do Sapucaí	MG	74.80	76.49	-1.69	Rodeio	SC	78.34	76.85	1.49
São Sebastião do Paraíso	MG	75.15	76.80	-1.65	Guapiaçu	SP	76.93	75.43	1.50
Criciúma	SC	75.76	77.36	-1.60	Colombo	PR	77.17	75.67	1.50
Lagoa da Prata	MG	74.89	76.47	-1.58	Caconde	SP	76.45	74.94	1.51
Biguaçu	SC	75.17	76.73	-1.56	Aguaí	SP	76.49	74.98	1.51
Erechim	RS	74.95	76.50	-1.55	Lima Duarte	MG	76.68	75.16	1.52
Nova Ponte	MG	75.08	76.56	-1.48	Nova Londrina	PR	76.36	74.84	1.52
Timbó	SC	76.36	77.80	-1.44	Charqueadas	RS	77.22	75.70	1.52
Ouro Preto	MG	75.03	76.45	-1.42	Perdizes	MG	77.80	76.27	1.53
Juiz de Fora	MG	75.65	77.06	-1.41	Alto Paraná	PR	75.76	74.23	1.53
Itapiranga	SC	75.50	76.89	-1.39	Brasília	DF	77.35	75.82	1.53
Luz	MG	75.00	76.35	-1.35	São Sepé	RS	77.23	75.69	1.54
Garuva	SC	74.82	76.16	-1.34	Santo Antônio da Patrulha	RS	76.97	75.41	1.56
Indaiatuba	SP	75.22	76.55	-1.33	Ilhota	SC	77.95	76.39	1.56
Uberaba	MG	75.71	77.03	-1.32	Barbacena	MG	77.88	76.31	1.57
Contagem	MG	74.94	76.24	-1.30	Valença	RJ	75.85	74.28	1.57
Alfenas	MG	75.45	76.74	-1.29	Mariana	MG	77.43	75.85	1.58
Timóteo	MG	75.14	76.43	-1.29	Guaratinguetá	SP	78.17	76.59	1.58
Três de Maio	RS	75.02	76.26	-1.24	Mirandópolis	SP	77.12	75.54	1.58
Ituiutaba	MG	75.37	76.60	-1.23	Porto União	SC	78.43	76.85	1.58
Três Coroas	RS	74.75	75.96	-1.21	Taubaté	SP	77.98	76.40	1.58
Laranjal Paulista	SP	74.73	75.91	-1.18	Juscimeira	MT	75.54	73.96	1.58
São José do Cedro	SC	74.79	75.97	-1.18	Brasnorte	MT	76.00	74.41	1.59
Jaguaruna	SC	74.80	75.97	-1.17	Serro	MG	75.01	73.42	1.59
Aparecida	SP	74.67	75.83	-1.16	Ronda Alta	RS	77.03	75.43	1.60
Divinópolis	MG	75.63	76.78	-1.15	Seberi	RS	76.91	75.31	1.60
Guaíba	RS	74.99	76.13	-1.14	Fronteira	MG	77.28	75.66	1.62
Pinhais	PR	75.15	76.29	-1.14	Fartura	SP	77.02	75.40	1.62
Ubá	MG	75.45	76.56	-1.11	Jataizinho	PR	75.70	74.08	1.62
Veranópolis	RS	75.29	76.39	-1.10	Canelinha	SC	76.89	75.27	1.62
Carlos Barbosa	RS	75.07	76.16	-1.09	Mairiporã	SP	77.86	76.24	1.62
Jacareí	SP	75.20	76.29	-1.09	Abelardo Luz	SC	76.10	74.47	1.63
Santo Antônio do Monte	MG	75.37	76.46	-1.09	Jaguari	RS	76.93	75.29	1.64
Guarulhos	SP	74.83	75.91	-1.08	Teodoro Sampaio	SP	76.38	74.74	1.64
Lajeado	RS	75.41	76.48	-1.07	Igrejinha	RS	77.53	75.89	1.64
Salto	SP	75.19	76.26	-1.07	São Caetano do Sul	SP	78.20	76.55	1.65
Tapera	RS	75.07	76.13	-1.06	Brasilândia de Minas	MG	75.87	74.21	1.66

Sete Lagoas	MG	75.37	76.42	-1.05	Pedro Afonso	TO	75.77	74.11	1.66
Jaguarão	RS	74.89	75.93	-1.04	Mondaí	SC	77.52	75.84	1.68
Palmitos	SC	75.13	76.15	-1.02	Tietê	SP	78.04	76.36	1.68
Videira	SC	76.42	77.43	-1.01	Guaraciaba	SC	77.28	75.59	1.69
Itatiba	SP	75.64	76.65	-1.01	Suzano	SP	77.36	75.66	1.70
Valinhos	SP	76.01	77.02	-1.01	Nova Bandeirantes	MT	75.51	73.81	1.70
Socorro	SP	74.66	75.67	-1.01	Alvorada	RS	77.41	75.71	1.70
Carazinho	RS	75.42	76.43	-1.01	Holambra	SP	77.66	75.95	1.71
Antônio Prado	RS	75.07	76.07	-1.00	Alto Rio Doce	MG	74.70	72.97	1.73
Batatais	SP	75.41	76.41	-1.00	Passos	MG	78.15	76.42	1.73
Praia Grande	SP	75.04	76.02	-0.98	Águas Lindas de Goiás	GO	75.85	74.12	1.73
São José do Rio Preto	SP	75.74	76.71	-0.97	Canápolis	MG	77.32	75.58	1.74
Casa Branca	SP	75.08	76.05	-0.97	Bom Jesus de Goiás	GO	76.59	74.84	1.75
Camanducaia	MG	75.22	76.19	-0.97	Rio Branco do Sul	PR	75.84	74.08	1.76
Otacílio Costa	SC	75.12	76.09	-0.97	PariqueraAçu	SP	76.70	74.94	1.76
Sapiranga	RS	74.93	75.88	-0.95	Nanuque	MG	76.02	74.24	1.78
Araçatuba	SP	75.46	76.41	-0.95	Cosmópolis	SP	77.57	75.78	1.79
Tupanciretã	RS	74.67	75.61	-0.94	Novo Gama	GO	76.06	74.27	1.79
Guaporé	RS	75.22	76.16	-0.94	Altônia	PR	75.88	74.09	1.79
Ibirama	SC	75.87	76.81	-0.94	Taquara	RS	77.82	76.01	1.81
Araranguá	SC	76.17	77.09	-0.92	Itabirinha	MG	75.30	73.48	1.82
Vacaria	RS	75.25	76.16	-0.91	Santiago	RS	77.93	76.06	1.87
Raposos	MG	74.90	75.80	-0.90	Orlândia	SP	77.89	76.02	1.87
Sabará	MG	74.99	75.89	-0.90	Tamarana	PR	74.76	72.86	1.90
Nova Prata	RS	75.36	76.25	-0.89	Andradina	SP	78.07	76.15	1.92
Palmeira das Missões	RS	74.83	75.71	-0.88	Vargem Grande Paulista	SP	78.02	76.10	1.92
Tupi Paulista	SP	75.08	75.96	-0.88	Triunfo	RS	77.35	75.43	1.92
Esteio	RS	75.57	76.44	-0.87	Jaú	SP	78.13	76.20	1.93
Governador Valadares	MG	75.06	75.93	-0.87	Rio Brilhante	MS	76.67	74.74	1.93
Osasco	SP	75.42	76.29	-0.87	Rio Verde de Mato Grosso	MS	76.11	74.17	1.94
Lençóis Paulista	SP	75.19	76.06	-0.87	Miguelópolis	SP	76.78	74.84	1.94
Cássia	MG	75.22	76.09	-0.87	Estância Velha	RS	78.23	76.28	1.95
Lagoa Formosa	MG	74.91	75.77	-0.86	Jaguariaíva	PR	76.54	74.59	1.95
Santa Fé do Sul	SP	75.41	76.27	-0.86	Cafelândia	SP	77.01	75.04	1.97
Sorocaba	SP	75.59	76.44	-0.85	Monte Santo de Minas	MG	77.46	75.46	2.00
Ribeirão Preto	SP	75.65	76.50	-0.85	São Sebastião	SP	77.51	75.51	2.00
Santos	SP	76.13	76.98	-0.85	Sobral	CE	74.93	72.92	2.01
Tanabi	SP	74.73	75.58	-0.85	Santa Rita do Passa Quatro	SP	78.22	76.17	2.05
Piracicaba	SP	75.88	76.71	-0.83	Arraias	TO	74.73	72.67	2.06
Bom Princípio	RS	75.18	76.01	-0.83	Tapiratiba	SP	77.18	75.12	2.06
Cachoeira Paulista	SP	75.19	76.01	-0.82	Piracanjuba	GO	77.21	75.09	2.12
Santana de Parnaíba	SP	75.92	76.74	-0.82	Cruz Machado	PR	75.67	73.53	2.14
Quatro Barras	PR	74.87	75.68	-0.81	São Félix do Araguaia	MT	75.86	73.70	2.16

Jaraguá do Sul	SC	76.92	77.73	-0.81	Nova Petrópolis	RS	78.38	76.21	2.17
Bento Gonçalves	RS	75.52	76.31	-0.79	Lagoa Vermelha	RS	78.10	75.90	2.20
Caldas Novas	GO	74.89	75.68	-0.79	Maracaju	MS	77.35	75.13	2.22
Londrina	PR	75.19	75.97	-0.78	Cerro Grande do Sul	RS	76.28	74.04	2.24
Tapejara	RS	75.23	76.01	-0.78	Açucena	MG	75.66	73.38	2.28
Restinga Seca	RS	74.70	75.47	-0.77	São Gonçalo do Amarante	RN	74.72	72.42	2.30
Macaé	RJ	74.66	75.43	-0.77	Eldorado	SP	75.82	73.51	2.31
Itaguara	MG	75.14	75.90	-0.76	Piraquara	PR	77.15	74.80	2.35
Pirapora	MG	74.65	75.41	-0.76	Bananal	SP	77.29	74.94	2.35
Ponta Grossa	PR	75.22	75.97	-0.75	Espumoso	RS	78.04	75.65	2.39
Nhandeara	SP	75.08	75.81	-0.73	Nova Laranjeiras	PR	74.96	72.56	2.40
Campo Limpo Paulista	SP	75.40	76.13	-0.73	Pires do Rio	GO	77.79	75.31	2.48
Goiânia	GO	75.28	76.00	-0.72	Vila Bela da Santíssima Trindade	MT	75.57	73.07	2.50
Dores do Indaiá	MG	75.40	76.11	-0.71	Encruzilhada do Sul	RS	77.50	74.75	2.75
Dracena	SP	75.51	76.22	-0.71	Parobé	RS	78.18	75.41	2.77
Arapongas	PR	75.02	75.72	-0.70	Santana do Paraíso	MG	77.66	74.88	2.78
Presidente Venceslau	SP	75.19	75.89	-0.70	Porto Xavier	RS	77.53	74.69	2.84
Volta Redonda	RJ	74.98	75.67	-0.69	São Domingos	GO	74.77	71.85	2.92

eAppendix 6

Spatial distribution of the top 100 under and overachievers by tertile of life expectancy at birth.



eAppendix 7

We performed the same analyses for identifying the under and overachievers, but instead of using the SuperLearner to predict the expected value of life expectancy at birth, we used the individual algorithm with the highest predictive performance (random forests). The difference between the final results is small, and the results for random forests are presented below.

Healthcare variables	1st Tertile				2nd Tertile				3rd Tertile			
	Over	Under	Difference*	CI (95%)	Over	Under	Difference*	CI (95%)	Over	Under	Difference*	CI (95%)
Caesarean deliveries (%)	30.24	31.87	-2.97	-6.52;0.65	35.86	59.27	-22.01	-26.20;-17.94	56.76	62.44	-6.62	-10.71;-2.60
Family Health Strategy teams per 10,000 residents	3.50	3.10	0.10	-0.10;0.40	3.00	2.05	1.10	0.70;1.40	2.00	1.40	0.60	0.20;0.90
Hospital beds per 10,000 residents	12.50	14.25	-0.80	-4.10;1.20	15.35	16.40	-1.80	-5.40;0.90	15.75	14.35	0.80	-1.90;3.60
Life support equipment per 10,000 residents	0.80	1.20	-0.40	-0.80;-0.00	1.40	3.60	-2.00	-2.60;-1.50	2.90	3.00	0.10	-0.50;0.60
Low birth weight (%)	6.88	6.74	0.43	-0.14;0.99	6.96	8.05	-0.85	-1.54;-0.21	8.35	8.70	-0.58	-1.14;0.00
Mammographies per 100 women	2.00	2.50	-0.00	-1.00;0.00	3.00	12.50	-8.00	-10.00;-7.00	10.50	14.50	-4.00	-6.00;-2.00
Oral health coverage	88.18	84.38	-0.00	-2.06;0.16	81.23	32.37	34.72	23.28;44.69	42.88	16.88	14.29	3.87;25.00
Primary health coverage for poor residents (last year)	100.00	93.32	0.00	-0.00;0.15	87.28	70.39	15.00	6.85;21.99	64.85	51.17	13.89	5.09;21.96
Ultrasound machines per 10,000 live births	23.35	27.10	-0.00	-9.80;1.90	25.80	48.20	-21.10	-35.30;-6.20	37.05	37.55	0.00	-7.80;10.60
Vaccination coverage (%)	81.30	78.73	1.80	-1.44;4.82	78.08	75.12	2.41	0.16;4.77	75.61	72.99	2.04	0.25;3.75
X-ray machines per 10,000 residents	0.45	0.45	0.00	-0.00;0.01	0.60	1.00	-0.50	-0.70;-0.30	1.00	1.10	-0.00	-0.20;0.20

* For Mann-Whitney U tests, the difference in this case is the median of the difference between a sample from the underachievers and a sample from the overachievers, not the difference of the median between the two groups.

eAppendix 8

We also performed another sensitivity test, for which we ran the same analyses but excluding the first, second and third variables that most contributed to improve predictive performance, i.e. residing in Minas Gerais State, the illiteracy rate and proportion of households with automobile ownership. The MSE on each of these cases were 0.186, 0.183 and 0.178, respectively, which was similar to the one with all the variables included 0.175.

eAppendix 9

title: "Life_Expectancy_SL_prediction"

date: "4th June 2018"

output: html_document

```
# Packages
library(tidyverse)
library(caret)
library(hydroGOF)
library(SuperLearner)
library(ggplot2)
library(dummies)
library(reshape)
library(dplyr)
library(plyr)
library(RhpcBLASctl)
```

```
# Data set to develop predictive life expectancy model

data_set1 <- read.table ("http://bit.ly/ 2KP2aC7",
                header = T ,sep=';', dec='.',colClass=c(rep("numeric",37), "factor"))
str(data_set1)

#Filter: municipalities with more than 10,000 residents
filter_data <- data_set1 %>%
        filter(Residents > 10000)

summary(filter_data)
```

````
```


# Pre-process


#Standardization
nums <- sapply(filter_data, is.numeric)
quantis <- filter_data[,nums]
quantis_filter <- select(quantis, -c(Municipality_code))
scale_variables <- preProcess(quantis_filter, method = c("center", "scale"))
quantis_scale <- predict(scale_variables,quantis_filter)
head(quantis_scale)


#Indicator variables
State<-filter_data$State_of_residence
df_State <- dummy(State, sep="_")
df_State<-as.data.frame(df_State)
head(df_State)


#final_data
final_data <- cbind(quantis$Municipality_code,quantis$Life_expectance,quantis_scale,
dplyr::select(df_State, -State_SP))
names(final_data)[1]<-"Municipality_code"
names(final_data)[2]<-"original_LE"


#-----------------------------------------------------------------------------------------------------
set.seed(1)
final_data_sort<-sample_n(final_data, nrow(final_data), replace=FALSE)
final_data_sort$position<-seq(1:3052)
head(final_data_sort)
```
````

Training steps using SuperLearner package

- Step 1: combining Super Learner with the caret package

#Observations: the SuperLearner package provides the function SL.caret which allows you to include algorithms in the Super Learner that implicitly use cross-validation.

#I used a slight modification to this function, written by David Benkeser(http://benkeser.github.io/sllecture/).

#Additionally,I modified "tuneLength" to "tuneGrid" and, for neural network, I add "linout=TRUE" when regression is of interest.

```
SL.caret1 <- function (Y, X, newX, family, obsWeights, method = "rf", tuneGrid = tuneGrid,
            trControl = trainControl(method = "cv",
                         number = 10,
                         verboseIter = FALSE,
                         savePredictions=TRUE),
            metric,...)
{
 if (length(unique(Y))>2){
  if(is.matrix(Y)) Y <- as.numeric(Y)
  metric <- "RMSE"
  if(method=="gbm"){
   suppressWarnings(
    # pass verbose==FALSE directly to train (verboseIter doesn't suppress output)
    fit.train <- caret::train(x = X, y = Y, weights = obsWeights,
                 metric = metric, method = method,
                 tuneGrid = tuneGrid,
                 trControl = trControl,verbose=FALSE)
   )
  }else{
```

```
    suppressWarnings(
      fit.train <- caret::train(x = X, y = Y, weights = obsWeights,
                               metric = metric, method = method,
                               tuneGrid = tuneGrid,
                               trControl = trControl)
    )
  }
  if(method=="nnet"){
      fit.train <- caret::train(x = X, y = Y, weights = obsWeights,
                               metric = metric, method = method,
                               tuneGrid = tuneGrid,
                               trControl = trControl,linout=TRUE)
  }else{
    fit.train <- caret::train(x = X, y = Y, weights = obsWeights,
                               metric = metric, method = method,
                               tuneGrid = tuneGrid,
                               trControl = trControl)
  }
  pred <- predict(fit.train, newdata = newX, type = "raw")
}
if (length(unique(Y))<=2) {
  metric <- "Accuracy"
  Y.f <- as.factor(Y)
  levels(Y.f) <- c("A0", "A1")
  if(method=="gbm"){
    suppressWarnings(
      # pass verbose==FALSE directly to train (verboseIter doesn't
      # suppress output)
      fit.train <- caret::train(x = X, y = Y.f, weights = obsWeights,
                               metric = metric, method = method,
```

```
                         tuneGrid = tuneGrid,
                         trControl = trControl, verbose = FALSE)
      )
    }else{
      suppressWarnings(
        fit.train <- caret::train(x = X, y = Y, weights = obsWeights,
                         metric = metric, method = method,
                         tuneGrid = tuneGrid,
                         trControl = trControl)
      )
    }
    pred <- predict(fit.train, newdata = newX, type = "prob")[,2]
  }
  fit <- list(object = fit.train)
  out <- list(pred = pred, fit = fit)
  class(out$fit) <- c("SL.caret")
  return(out)
}
```

- Step 2: algorithms inside caret

```
#Uses ten-fold cross validation to select tuning parameters
#We must define SL.<algorithm>.caret1

#---------------------------------------
#***Linear models***
#---------------------------------------
#Ridge
SL.ridge.caret <- function(...,method = "ridge",
```

```
                          tuneGrid = expand.grid(.lambda = seq(0, .01, length = 10)),
                          trControl = trainControl(method = "cv",
                                                   repeats = 10,
                                                   verboseIter = FALSE,
                                                   savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#----------------------------------------
#Lasso
SL.lasso.caret <- function(...,method = "lasso",
                          tuneGrid = expand.grid(.fraction = seq(.5, 1, length = 10)),
                          trControl = trainControl(method = "cv",
                                                   number =  10,
                                                   verboseIter = FALSE,
                                                   savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#----------------------------------------
#Elastic Net
SL.enet.caret <- function(...,method = "enet",
                         tuneGrid = expand.grid(.lambda = seq(0, 0.1, length = 10),
                                                .fraction = seq(.5, 1, length = 10)),
                         trControl = trainControl(method = "cv",
                                                  number = 10,
                                                  verboseIter = FALSE,
                                                  savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}
```

```
#---------------------------------------
#Partial Least Squares
SL.pls.caret <- function(...,method ="pls",
                         tuneGrid = expand.grid(.ncomp = c(1:28)),
                         trControl = trainControl(method = "cv",
                                                  number = 10,
                                                  verboseIter = FALSE,
                                                  savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#---------------------------------------
#***Non-linear models***
#---------------------------------------
#Neural network
SL.nnet.caret <- function(...,method ="nnet",
                          tuneGrid = expand.grid(.decay = seq(0,0.1,length=5),
                                                 .size = c(1:5)),
                          linout=TRUE,
                          trControl = trainControl(method = "cv",
                                                   number = 10,
                                                   verboseIter = FALSE,
                                                   savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#---------------------------------------
#MARS
SL.mars.caret <- function(...,method ="earth",
```

```
                    tuneGrid = expand.grid(.degree = 1, .nprune = 2:24),
                    trControl = trainControl(method = "cv",
                                             number = 10,
                                             verboseIter = FALSE,
                                             savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#---------------------------------------
#SVM linear
SL.svmL.caret <- function(...,method = "svmLinear",
                    tuneGrid = expand.grid(.C= c(.25,.50,1,2,4)),
                    trControl = trainControl(method = "cv",
                                             number = 10,
                                             verboseIter = FALSE,
                                             savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#---------------------------------------
#SVM polynomial
SL.svmP.caret <- function(...,method = "svmPoly",
                    tuneGrid = expand.grid(.degree=c(1,2,3,4,5),
                                           .scale=c(0,0.0001,0.001,0.01,0.1),
                                           .C=c(.25,.50,1,2,4)),
                    trControl = trainControl(method = "cv",
                                             number = 10,
                                             verboseIter = FALSE,
                                             savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
```

```
}

#-----------------------------------------
#SVM radial
SL.svmR.caret <- function(...,method = "svmRadial",
                          tuneGrid = expand.grid(.C = c(.25,.50,1,2,4),
                                                 .sigma= c(0.001,0.002,0.004,0.006,0.01)),
                          trControl = trainControl(method = "cv",
                                                   number = 10,
                                                   verboseIter = FALSE,
                                                   savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#-----------------------------------------
#Knn
SL.knn.caret <- function(...,method = "knn",
                         tuneGrid = expand.grid(.k = 1:20),
                         trControl = trainControl(method = "cv",
                                                  number = 10,
                                                  verboseIter = FALSE,
                                                  savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#-----------------------------------------
#***Based-tree models***
#-----------------------------------------
#Regression tree (.maxdepth method)
SL.rpart.caret <- function(...,method = "rpart",
```

```
                    tuneGrid = expand.grid(.cp=seq(0.0001,0.01,length=20)),
                    trControl = trainControl(method = "cv",
                                             number = 10,
                                             verboseIter = FALSE,
                                             savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#---------------------------------------
#Regression tree (.cp method)
SL.rpart2.caret <- function(...,method = "rpart2",
                     tuneGrid = expand.grid(.maxdepth=c(3,6,7,9,12)),
                     trControl = trainControl(method = "cv",
                                              number = 10,
                                              verboseIter = FALSE,
                                              savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#---------------------------------------
#Random forest
SL.rf.caret <- function(...,method = "rf",
                  tuneGrid = expand.grid(.mtry= c(10,20,30,35,40)),
                  trControl = trainControl(method = "cv",
                                           number = 10,
                                           verboseIter = FALSE,
                                           savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}
```

```
#----------------------------------------
#Boosting
SL.xgb.caret <- function(...,method = "xgbTree",
                    tuneGrid = expand.grid(.nrounds = c(50,100,150,200,250),
                                           .max_depth = c(2,3,4,5,6),
                                           .eta = c(0.01,0.03,0.05,0.1,0.3,0.5),
                                           .gamma = 0, #default
                                           .colsample_bytree = 1,#default
                                           .min_child_weight = 1, #default
                                           .subsample = 1), #default
                    trControl = trainControl(method = "cv",
                                             number = 10,
                                             verboseIter = FALSE,
                                             savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}

#----------------------------------------
#Cubist
SL.cub.caret <- function(...,method = "cubist",
                    tuneGrid = expand.grid(.committees=c(10,17,20,22,25),
                                           .neighbors=c(0:9)),
                    trControl = trainControl(method = "cv",
                                             number = 10,
                                             verboseIter = FALSE,
                                             savePredictions = TRUE)){
  SL.caret1(...,method = method,tuneGrid = tuneGrid, trControl = trControl)
}
```
```
```

- Specifying the SuperLearner library of candidate algorithms

```
#We can now implement the Super Learner.
#Note that the run time on this Super Learner will be considerably longer due
#to the additional layers of cross validation.
sl.lib <- c("SL.glm","SL.ridge.caret","SL.lasso.caret","SL.enet.caret",
        "SL.pls.caret","SL.nnet.caret","SL.mars.caret","SL.svmL.caret",
        "SL.svmP.caret","SL.svmR.caret","SL.knn.caret","SL.rpart.caret",
        "SL.rpart2.caret","SL.rf.caret","SL.xgb.caret","SL.cub.caret")
```

- Training and test

```
#We use CV.SuperLearner function to objectively evaluate the performance of
#the SuperLearner predictions relative to those from its component methods.
X <- select(final_data_sort,-c(Municipality_code,original_LE,Life_expectance,position))
Y <- final_data_sort$Life_expectance

#fit cross-validated Super Learner
set.seed(1)
cvSL <- CV.SuperLearner(Y = Y, X = X,
                #V specifies the number of outer CV layers used to evalute
                #the Super Learner (which by default uses 10-fold CV)
                V = 10,
                parallel = "multicore",
                family = gaussian(),
                method = "method.NNLS",
                SL.library = sl.lib)
```

- Performance evaluation

```
plot(cvSL)
summary(cvSL)

#We can extract the weights at each CV.SuperLearner iteration and summarize its distribution.
#function to review meta-weights (coefficients) from a CV.SuperLearner object
review_weights = function(cvSL) {
  meta_weights = coef(cvSL)
  means = colMeans(meta_weights)
  sds = apply(meta_weights, MARGIN = 2,  FUN = sd)
  mins = apply(meta_weights, MARGIN = 2, FUN = min)
  maxs = apply(meta_weights, MARGIN = 2, FUN = max)
  #Combine the stats into a single matrix.
  sl_stats = cbind("mean(weight)" = means, "sd" = sds, "min" = mins, "max" = maxs)
  #Sort by decreasing mean weight.
  sl_stats[order(sl_stats[, 1], decreasing = T), ]
}

review_weights(cvSL)

#Review the distribution of the best single learner as external CV folds
table(simplify2array(cvSL$whichDiscreteSL))
```

- Predicted values

```
position<-seq(1:3052)

SL_pred<-cvSL$SL.predict
```

~~~
Discrete_pred<-cvSL$discreteSL.predict
RLinear_pred<-cvSL$library.predict[,1]
ridge_pred<-cvSL$library.predict[,2]
lasso_pred<-cvSL$library.predict[,3]
enet_pred<-cvSL$library.predict[,4]
pls_pred<-cvSL$library.predict[,5]
nnet_pred<-cvSL$library.predict[,6]
mars_pred<-cvSL$library.predict[,7]
svmL_pred<-cvSL$library.predict[,8]
svmP_pred<-cvSL$library.predict[,9]
svmR_pred<-cvSL$library.predict[,10]
knn_pred<-cvSL$library.predict[,11]
rpart_pred<-cvSL$library.predict[,12]
rpart2_pred<-cvSL$library.predict[,13]
rf_pred<-cvSL$library.predict[,14]
xgb_pred<-cvSL$library.predict[,15]
cub_pred<-cvSL$library.predict[,16]

#predicted values data set
data_pred<-cbind(position,SL_pred,Discrete_pred,RLinear_pred,ridge_pred,
          lasso_pred,enet_pred,pls_pred,nnet_pred,mars_pred,svmL_pred,
          svmP_pred,svmR_pred,knn_pred,rpart_pred,rpart2_pred,rf_pred,
          xgb_pred,cub_pred)

#merge predicted values with data set used to estimate it wich contains
#the predictors and the original variable response
data_all<-merge(final_data_sort,data_pred, by="position")
head(data_all)
```
~~~

Identifying over and underachievers

- Organization of the data set

#health characteristics data set

health_variables<-read.csv2("http://bit.ly/ 2KPfqXg", sep=";",header = TRUE, na.strings="NA")

names(health_variables)

#merge data_all and health_variables

newdata<-merge(data_all,health_variables,by="Municipality_code")

#mean and standard deviation of variable response (life expectancy)

meanLE<-mean(as.numeric(newdata$original_LE))

sdLE<-sd(as.numeric(newdata$original_LE))

#predicted SL values in original scale of variable response

newdata$SL_original_value<-((newdata$SL_pred)*sdLE)+meanLE

#predicted Random Forest values in original scale of variable response

newdata$RF_original_value<-((newdata$rf_pred)*sdLE)+meanLE

```

- Organization of the tertiles data sets

#difference between observed and predicted values according SL algorithm

newdata$dif_O_P_SL<-newdata$original_LE-newdata$SL_original_value

#difference between observed and predicted values according Random Forest algorithm

newdata$dif_O_P_RF<-newdata$original_LE-newdata$RF_original_value
```

```
#tercile of observed Life Expectancy
quantile(newdata$original_LE, c(1/3, 2/3))

newdata$tertileY[newdata$original_LE<71.53 | newdata$original_LE==71.53]<-"<=T1"
newdata$tertileY[newdata$original_LE>71.53 & (newdata$original_LE<74.63 |
newdata$original_LE==74.63)]<-">T1 e <=T2"
newdata$tertileY[newdata$original_LE>74.63]<-">T2"

table(newdata$tertileY)

#tertiles data sets
data_YT1<-newdata %>%
  filter(tertileY=="<=T1")

data_YT1T2<-newdata %>%
  filter(tertileY==">T1 e <=T2")

data_YT2<-newdata %>%
  filter(tertileY==">T2")
```

- For results based on predicted Super Learner values

```
#order tertiles data sets by observed-expected difference for SL algorithm
data_YT1<-data_YT1[order(data_YT1$dif_O_P_SL),]
data_YT1T2<-data_YT1T2[order(data_YT1T2$dif_O_P_SL),]
data_YT2<-data_YT2[order(data_YT2$dif_O_P_SL),]
```

- eAppendix: results based on predicted Random Forest values

```
#order tertiles data sets by observed-expected difference for Radon Forest algorithm
data_YT1<-data_YT1[order(data_YT1$dif_O_P_RF),]
data_YT1T2<-data_YT1T2[order(data_YT1T2$dif_O_P_RF),]
data_YT2<-data_YT2[order(data_YT2$dif_O_P_RF),]
```

Choose 100 over and 100 underachievers within tertiles data sets and test differences in heath characteristics

- 1Âº Tercile

```
overunder<-rbind(data_YT1[1:100,],data_YT1[922:1021,])
overunder$category<-c(rep(1,100),rep(0,100))
overunder$category<-as.factor(overunder$category)
overunder$category<-revalue(overunder$category,c("1"="under","0"="over"))
overunder_T1<-overunder

data<-overunder_T1

a<-data %>%
  filter(category=="over")

b<-data %>%
  filter(category=="under")

#Caesarean deliveries (%)
median(a$cesarean_deliveries)
median(b$cesarean_deliveries)
```

```
wilcox.test(as.numeric(as.character(a$cesarean_deliveries)),
        as.numeric(as.character(b$cesarean_deliveries)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Family Health Strategy teams per 10,000 residents
median(a$Family_Health_Strategy_teams_per_10.000_residents)
median(b$Family_Health_Strategy_teams_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Family_Health_Strategy_teams_per_10.000_residents)),
        as.numeric(as.character(b$Family_Health_Strategy_teams_per_10.000_residents)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Hospital beds per 10,000 residents
median(a$Hospital_beds_per_10.000_residents)
median(b$Hospital_beds_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Hospital_beds_per_10.000_residents)),
        as.numeric(as.character(b$Hospital_beds_per_10.000_residents)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Life support equipment per 10,000 residents
median(a$Life_support_equipment_per_10.000_residents)
median(b$Life_support_equipment_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Life_support_equipment_per_10.000_residents)),
        as.numeric(as.character(b$Life_support_equipment_per_10.000_residents)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Low birth weight (%)
median(a$Low_birth_weight_...)
median(b$Low_birth_weight_...)
wilcox.test(as.numeric(as.character(a$Low_birth_weight_...)),
        as.numeric(as.character(b$Low_birth_weight_...)),
```

```
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Mammographies per 100 women
median(a$Mammographies_per_100_women)
median(b$Mammographies_per_100_women)
wilcox.test(as.numeric(as.character(a$Mammographies_per_100_women)),
    as.numeric(as.character(b$Mammographies_per_100_women)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Oral health coverage
median(a$Oral_Health_Strategy_coverage)
median(b$Oral_Health_Strategy_coverage)
wilcox.test(as.numeric(as.character(a$Oral_Health_Strategy_coverage)),
    as.numeric(as.character(b$Oral_Health_Strategy_coverage)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Primary health coverage for poor residents (last year)
median(a$Primary_health_coverage_for_poor_residents_.last.year.)
median(b$Primary_health_coverage_for_poor_residents_.last.year.)
wilcox.test(as.numeric(as.character(a$Primary_health_coverage_for_poor_residents_.last.year.)),
    as.numeric(as.character(b$Primary_health_coverage_for_poor_residents_.last.year.)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Ultrasound machines per 10,000 live births
median(a$Ultrasound_machines._per_10.000_live_births)
median(b$Ultrasound_machines._per_10.000_live_births)
wilcox.test(as.numeric(as.character(a$Ultrasound_machines._per_10.000_live_births)),
    as.numeric(as.character(b$Ultrasound_machines._per_10.000_live_births)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)
```

```
#Vaccination coverage (%)
median(a$Vaccination_coverage_...)
median(b$Vaccination_coverage_...)
wilcox.test(as.numeric(as.character(a$Vaccination_coverage_...)),
            as.numeric(as.character(b$Vaccination_coverage_...)),
            paired=FALSE,conf.int = TRUE, conf.level = 0.95)

# X-ray machines per 10,000 residents
median(a$Xray_machines_per_10.000_residents)
median(b$Xray_machines_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Xray_machines_per_10.000_residents)),
            as.numeric(as.character(b$Xray_machines_per_10.000_residents)),
            paired=FALSE,conf.int = TRUE, conf.level = 0.95)
```

- 2nd Tertile

```
overunder<-rbind(data_YT1T2[1:100,],data_YT1T2[919:1018,])
overunder$category<-c(rep(1,100),rep(0,100))
overunder$category<-as.factor(overunder$category)
overunder$category<-revalue(overunder$category,c("1"="under","0"="over"))

overunder_T1T2<-overunder

data<-overunder_T1T2

a<-data %>%
  filter(category=="over")

b<-data %>%
```

```
  filter(category=="under")

#Caesarean deliveries (%)
median(a$cesarean_deliveries)
median(b$cesarean_deliveries)
wilcox.test(as.numeric(as.character(a$cesarean_deliveries)),
        as.numeric(as.character(b$cesarean_deliveries)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Family Health Strategy teams per 10,000 residents
median(a$Family_Health_Strategy_teams_per_10.000_residents)
median(b$Family_Health_Strategy_teams_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Family_Health_Strategy_teams_per_10.000_residents)),
        as.numeric(as.character(b$Family_Health_Strategy_teams_per_10.000_residents)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Hospital beds per 10,000 residents
median(a$Hospital_beds_per_10.000_residents)
median(b$Hospital_beds_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Hospital_beds_per_10.000_residents)),
        as.numeric(as.character(b$Hospital_beds_per_10.000_residents)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Life support equipment per 10,000 residents
median(a$Life_support_equipment_per_10.000_residents)
median(b$Life_support_equipment_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Life_support_equipment_per_10.000_residents)),
        as.numeric(as.character(b$Life_support_equipment_per_10.000_residents)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)
```

```
#Low birth weight (%)
median(a$Low_birth_weight_...)
median(b$Low_birth_weight_...)
wilcox.test(as.numeric(as.character(a$Low_birth_weight_...)),
        as.numeric(as.character(b$Low_birth_weight_...)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Mammographies per 100 women
median(a$Mammographies_per_100_women)
median(b$Mammographies_per_100_women)
wilcox.test(as.numeric(as.character(a$Mammographies_per_100_women)),
        as.numeric(as.character(b$Mammographies_per_100_women)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Oral health coverage
median(a$Oral_Health_Strategy_coverage)
median(b$Oral_Health_Strategy_coverage)
wilcox.test(as.numeric(as.character(a$Oral_Health_Strategy_coverage)),
        as.numeric(as.character(b$Oral_Health_Strategy_coverage)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Primary health coverage for poor residents (last year)
median(a$Primary_health_coverage_for_poor_residents_.last.year.)
median(b$Primary_health_coverage_for_poor_residents_.last.year.)
wilcox.test(as.numeric(as.character(a$Primary_health_coverage_for_poor_residents_.last.year.)),
        as.numeric(as.character(b$Primary_health_coverage_for_poor_residents_.last.year.)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Ultrasound machines per 10,000 live births
median(a$Ultrasound_machines._per_10.000_live_births)
```

```
median(b$Ultrasound_machines._per_10.000_live_births)
wilcox.test(as.numeric(as.character(a$Ultrasound_machines._per_10.000_live_births)),
        as.numeric(as.character(b$Ultrasound_machines._per_10.000_live_births)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Vaccination coverage (%)
median(a$Vaccination_coverage_...)
median(b$Vaccination_coverage_...)
wilcox.test(as.numeric(as.character(a$Vaccination_coverage_...)),
        as.numeric(as.character(b$Vaccination_coverage_...)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)

# X-ray machines per 10,000 residents
median(a$Xray_machines_per_10.000_residents)
median(b$Xray_machines_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Xray_machines_per_10.000_residents)),
        as.numeric(as.character(b$Xray_machines_per_10.000_residents)),
        paired=FALSE,conf.int = TRUE, conf.level = 0.95)
```

3rd Tertile

```
overunder<-rbind(data_YT2[1:100,],data_YT2[914:1013,])
overunder$category<-c(rep(1,100),rep(0,100))
overunder$category<-as.factor(overunder$category)
overunder$category<-revalue(overunder$category,c("1"="under","0"="over"))
overunder_T2<-overunder

data<-overunder_T2
```

```
a<-data %>%
  filter(category=="over")

b<-data %>%
  filter(category=="under")

#Caesarean deliveries (%)
median(a$cesarean_deliveries)
median(b$cesarean_deliveries, na.rm = T)
wilcox.test(as.numeric(as.character(a$cesarean_deliveries)),
            as.numeric(as.character(b$cesarean_deliveries)),
            paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Family Health Strategy teams per 10,000 residents
median(a$Family_Health_Strategy_teams_per_10.000_residents)
median(b$Family_Health_Strategy_teams_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Family_Health_Strategy_teams_per_10.000_residents)),
            as.numeric(as.character(b$Family_Health_Strategy_teams_per_10.000_residents)),
            paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Hospital beds per 10,000 residents
median(a$Hospital_beds_per_10.000_residents)
median(b$Hospital_beds_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Hospital_beds_per_10.000_residents)),
            as.numeric(as.character(b$Hospital_beds_per_10.000_residents)),
            paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Life support equipment per 10,000 residents
median(a$Life_support_equipment_per_10.000_residents)
median(b$Life_support_equipment_per_10.000_residents)
```

```
wilcox.test(as.numeric(as.character(a$Life_support_equipment_per_10.000_residents)),
    as.numeric(as.character(b$Life_support_equipment_per_10.000_residents)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Low birth weight (%)
median(a$Low_birth_weight_...)
median(b$Low_birth_weight_...)
wilcox.test(as.numeric(as.character(a$Low_birth_weight_...)),
    as.numeric(as.character(b$Low_birth_weight_...)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Mammographies per 100 women
median(a$Mammographies_per_100_women)
median(b$Mammographies_per_100_women)
wilcox.test(as.numeric(as.character(a$Mammographies_per_100_women)),
    as.numeric(as.character(b$Mammographies_per_100_women)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Oral health coverage
median(a$Oral_Health_Strategy_coverage)
median(b$Oral_Health_Strategy_coverage)
wilcox.test(as.numeric(as.character(a$Oral_Health_Strategy_coverage)),
    as.numeric(as.character(b$Oral_Health_Strategy_coverage)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Primary health coverage for poor residents (last year)
median(a$Primary_health_coverage_for_poor_residents_.last.year.)
median(b$Primary_health_coverage_for_poor_residents_.last.year.)
wilcox.test(as.numeric(as.character(a$Primary_health_coverage_for_poor_residents_.last.year.)),
    as.numeric(as.character(b$Primary_health_coverage_for_poor_residents_.last.year.)),
```

```
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Ultrasound machines per 10,000 live births
median(a$Ultrasound_machines._per_10.000_live_births)
median(b$Ultrasound_machines._per_10.000_live_births)
wilcox.test(as.numeric(as.character(a$Ultrasound_machines._per_10.000_live_births)),
    as.numeric(as.character(b$Ultrasound_machines._per_10.000_live_births)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

#Vaccination coverage (%)
median(a$Vaccination_coverage_...)
median(b$Vaccination_coverage_...)
wilcox.test(as.numeric(as.character(a$Vaccination_coverage_...)),
    as.numeric(as.character(b$Vaccination_coverage_...)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)

# X-ray machines per 10,000 residents
median(a$Xray_machines_per_10.000_residents)
median(b$Xray_machines_per_10.000_residents)
wilcox.test(as.numeric(as.character(a$Xray_machines_per_10.000_residents)),
    as.numeric(as.character(b$Xray_machines_per_10.000_residents)),
    paired=FALSE,conf.int = TRUE, conf.level = 0.95)
```

eAppendix: Random Forest results - Nested Cross Validation (CV) Analysis

- Folds for nested CV using index folds from CV.SuperLearner

```
final_data_sort2<-cbind(as.numeric(as.character(row.names(final_data_sort))),
        select(final_data_sort,-c(Municipality_code,original_LE,position)))
```

```
names(final_data_sort2)[1]<-"index_SL"

data_list <- lapply(1:10, function(x){paste0("data", x)})
i=1
for(i in 1:10){
  index_SL<-as.data.frame(cvSL$folds[[i]])
  names(index_SL)[1]<-"index_SL"
  data_list[[i]]<-merge(index_SL,final_data_sort2,by="index_SL")
}

data_list_all<-select(melt(data_list,id=1:ncol(final_data_sort2)),-c(L1,index_SL))
names(data_list_all)
```

- Random Forest algorithm to visualize variable importance

```
RF_fit<-list()
importvar_RF<-list()
pred_rf<-list()

j=1
for (j in 1:10){

  #training set
  data_training <- select(melt(data_list[-j],id=1:ncol(final_data_sort2)),-c(L1,index_SL))

  X_training = select(data_training,-Life_expectance)

  Y_training = data_training$Life_expectance
```

```
  trainData<-cbind(Y_training,X_training)

  #------------------------------------------------------------------------------------------
  #test set
  data_teste <- select(data_list[[j]],-index_SL)

  X_test = select(data_teste,-Life_expectance)

  Y_test = data_teste$Life_expectance

  testData<-cbind(Y_test,X_test)

#----------------------------------------------------------------------------------------------
#Fit using Random Forest algorithm
set.seed(1)
rf_fit <- train(X_training,Y_training,
        method = "rf",
        tuneGrid = expand.grid(.mtry= c(10,20,30,35,40)),
        trControl = trainControl(method = "cv",
                    number = 10,
                    verboseIter = FALSE,
                    savePredictions = TRUE),
        importance=TRUE)

RF_fit[[j]]<-rf_fit

#----------------------------------------------------------------------------------------------
#variable importance
importvar_RF[[j]]<-varImp(rf_fit)
```

```
#-----------------------------------------------------------------------------------------------
#Predictions
pred_rf[[j]]<-predict(rf_fit, newdata = X_test)
}
```

- Variable importance to each iteration

```
plot(importvar_RF[[1]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[2]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[3]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[4]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[5]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[6]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[7]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[8]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[9]],top=10,scales=list(y=list(cex=.95)))
plot(importvar_RF[[10]],top=10,scales=list(y=list(cex=.95)))
```

- Random Forest analysis without the three most important variables

```
RF_fit_1<-list()
pred_rf_1<-list()

RF_fit_2<-list()
pred_rf_2<-list()

RF_fit_3<-list()
pred_rf_3<-list()
```

```
j=1
for (j in 1:10){
#-----------------------------------------------------------------------------------------------
##Without State_MG
#-----------------------------------------------------------------------------------------------
#training set
data_training <- select(melt(data_list[-j],id=1:ncol(final_data_sort2)),-c(L1,index_SL,State_MG))

X_training = select(data_training,-Life_expectance)

Y_training = data_training$Life_expectance

trainData<-cbind(Y_training,X_training)

#-----------------------------------------------------------------------------------------------
#test set
data_teste <- select(data_list[[j]],-c(index_SL,State_MG))

X_test = select(data_teste,-Life_expectance)

Y_test = data_teste$Life_expectance

testData<-cbind(Y_test,X_test)

#-----------------------------------------------------------------------------------------------
#Fit using Random Forest algorithm
set.seed(1)
rf_fit_1 <- train(X_training,Y_training,
            method = "rf",
```

```
        tuneGrid = expand.grid(.mtry= c(10,20,30,35,40)),
        trControl = trainControl(method = "cv",
                     number = 10,
                     verboseIter = FALSE,
                     savePredictions = TRUE),
        importance=TRUE)

RF_fit_1[[j]]<-rf_fit_1

#-----------------------------------------------------------------------------------------------
#Predictions
pred_rf_1[[j]]<-predict(rf_fit_1, newdata = X_test)

#-----------------------------------------------------------------------------------------------
##Without illiteracy_rate
#-----------------------------------------------------------------------------------------------
#training set
data_training <- select(melt(data_list[-j],id=1:ncol(final_data_sort2)),-c(L1,index_SL,Illiteracy_rate))

X_training = select(data_training,-Life_expectance)

Y_training = data_training$Life_expectance

trainData<-cbind(Y_training,X_training)

#-----------------------------------------------------------------------------------------------
#test set
data_teste <- select(data_list[[j]],-c(index_SL,Illiteracy_rate))

X_test = select(data_teste,-Life_expectance)
```

```
Y_test = data_teste$Life_expectance

testData<-cbind(Y_test,X_test)

#------------------------------------------------------------------------------------------------
#Fit using Random Forest algorithm
set.seed(1)
rf_fit_2 <- train(X_training,Y_training,
            method = "rf",
            tuneGrid = expand.grid(.mtry= c(10,20,30,35,40)),
            trControl = trainControl(method = "cv",
                          number = 10,
                          verboseIter = FALSE,
                          savePredictions = TRUE),
            importance=TRUE)

RF_fit_2[[j]]<-rf_fit_2

#------------------------------------------------------------------------------------------------
#Predictions
pred_rf_2[[j]]<-predict(rf_fit_2, newdata = X_test)

#------------------------------------------------------------------------------------------------
##without Automobile_ownership
#------------------------------------------------------------------------------------------------
#training set
data_training <- select(melt(data_list[-j],id=1:ncol(final_data_sort2)),-
c(L1,index_SL,Automobile_ownership))
```

```
X_training = select(data_training,-Life_expectance)

Y_training = data_training$Life_expectance

trainData<-cbind(Y_training,X_training)

#-----------------------------------------------------------------------------------------------
#test set
data_teste <- select(data_list[[j]],-c(index_SL,Automobile_ownership))

X_test = select(data_teste,-Life_expectance)

Y_test = data_teste$Life_expectance

testData<-cbind(Y_test,X_test)

#-----------------------------------------------------------------------------------------------
#Fit using Random Forest algorithm
set.seed(1)
rf_fit_3 <- train(X_training,Y_training,
        method = "rf",
        tuneGrid = expand.grid(.mtry= c(10,20,30,35,40)),
        trControl = trainControl(method = "cv",
                    number = 10,
                    verboseIter = FALSE,
                    savePredictions = TRUE),
        importance=TRUE)

RF_fit_3[[j]]<-rf_fit_3
```

```
#----------------------------------------------------------------------------------------------------
#Predictions
pred_rf_3[[j]]<-predict(rf_fit_3, newdata = X_test)
}
```

- MSE and RMSE for Random forest models

```
##Without State_MG
pred_rf_1_all<-c(pred_rf_1[[1]],pred_rf_1[[2]],pred_rf_1[[3]],pred_rf_1[[4]],pred_rf_1[[5]],
            pred_rf_1[[6]],pred_rf_1[[7]],pred_rf_1[[8]],pred_rf_1[[9]],pred_rf_1[[10]])

rmse_RF_1<-rmse(data_list_all$Life_expectance,pred_rf_1_all)
rmse_RF_1
mse_RF_1<-rmse_RF_1^2
mse_RF_1

##Without illiteracy_rate
pred_rf_2_all<-c(pred_rf_2[[1]],pred_rf_2[[2]],pred_rf_2[[3]],pred_rf_2[[4]],pred_rf_2[[5]],
            pred_rf_2[[6]],pred_rf_2[[7]],pred_rf_2[[8]],pred_rf_2[[9]],pred_rf_2[[10]])

rmse_RF_2<-rmse(data_list_all$Life_expectance,pred_rf_2_all)
rmse_RF_2
mse_RF_2<-rmse_RF_2^2
mse_RF_2

##without Automobile_ownership
pred_rf_3_all<-c(pred_rf_3[[1]],pred_rf_3[[2]],pred_rf_3[[3]],pred_rf_3[[4]],pred_rf_3[[5]],
            pred_rf_3[[6]],pred_rf_3[[7]],pred_rf_3[[8]],pred_rf_3[[9]],pred_rf_3[[10]])
```

rmse_RF_3<-rmse(data_list_all$Life_expectance,pred_rf_3_all)

rmse_RF_3

mse_RF_3<-rmse_RF_3^2

mse_RF_3

```

# Top 100 under and over for each tertile

mun_code<-read.csv2("mun_code.csv",header = TRUE)

T1<-merge(overunder_T1,mun_code,by="Municipality_code")

write.table(T1,"1tertile.csv",row.names = FALSE, dec = ".",sep = ";")

T1T2<-merge(overunder_T1T2,mun_code,by="Municipality_code")

write.table(T1T2,"2tertile.csv",row.names = FALSE,dec = ".",sep = ";")

T2<-merge(overunder_T2,mun_code,by="Municipality_code")

write.table(T2,"3tertile.csv",row.names = FALSE,dec = ".",sep = ";")

```

Graph: CV risk with mean and square error associated to each candidate algorithm

#Figure 1

data_cv_risk<-read.csv2("Data_cv_risk.csv",header=TRUE)

data_cv_risk$mean<-as.numeric(as.character(data_cv_risk$mean))

data_cv_risk$sd<-as.numeric(as.character(data_cv_risk$sd))

data_cv_risk$Algorithm <- factor(data_cv_risk$Algorithm,

levels = data_cv_risk$Algorithm[order(data_cv_risk$mean, decreasing = T)])

library(ggplot2)

````
g1<-ggplot(data_cv_risk,aes(x=mean, y=Algorithm,order=Algorithm))+
  geom_errorbarh(aes(xmin=mean-sd, xmax=mean+sd), height=.2)+
  geom_line() +
  geom_point() +
  scale_x_continuous(limits=c(0.16,0.28),breaks = seq(0.16, 0.28, 0.02)) +
  xlab("10-fold CV Mean Square Error") +
  ylab("Method") +
  theme(panel.background = element_rect(fill = "white", colour = "grey50"))

tiff("cv_risk.tiff", height = 12, width = 17, units = 'cm',compression = "lzw", res = 300)
    plot(g1)
    dev.off()
```


# Predicted SL versus observed life expectancy


#Figure 2
g2<-qplot(newdata$original_LE,
    newdata$SL_original_value,
    alpha = I(1/4)) +
    theme(panel.background = element_rect(fill = "white", colour = "grey50")) +
    xlim(65,80) +
    ylim(65,80) +
    labs(x="Observed life expectancy",y="Predicted SL life expectancy")

tiff("obs_vs_pred.tiff", height = 12, width = 17, units = 'cm',compression = "lzw", res = 300)
plot(g2)
dev.off()
```
````