Rural broadband coverage project report

Harald Lie

2020-Feb-24

1. Introduction and datasets

This report describes the Rural broadband coverage project that is the second of two projects required for the last course in the "Professional Certificate in Data Science" programme at Harvard University.
Broadband internet access became commercially available in many countries in the late 1990s, and has become a very popular service among households and businesses. In Norway, more than 95% of households have some sort of Internet access at home. There is a large gap, however, between the haves and the have-nots: The average download speed for Norwegian residential users is currently more than 150 Mbit/s, but the variation is high and many households have only access to a 20 Mbit/s service or even less. Most of the homes and businesses with access to low broadband speeds are located in rural areas, and rural Internet coverage is the scope of this report. The main data set that I have used for the project includes Internet coverage information about almost 458 000 buildings in rural Norway. The data set is publicly available and is published annually by Nkom, the Norwegian telecoms regulator[1]. It has data for all buildings - also urban - but I filtered out the urban buildings since Internet coverage in general is rather good and it is normally profitable to build high-speed networks in urban areas.
This is not always the case in rural areas: The cost of deploying a broadband network is highly correlated with distance between customers, and almost per definition this distance is higher in rural areas. Statistics Norway defines an urban areas as an area with at least 200 inhabitants (ca 100 homes) and less than 50 metres between houses. Among rural buildings in the dataset, ca 45% have access to a 100 Mbit/s, while the average for all buildings in Norway is close to 90%.

Project objective

Given that ca. 45% of rural buildings actually have 100 Mbit/s coverage, quite a bit of deployment have happened over the years. The project objective is to develop a high-quality model that can predict whether a rural building in Norway has 100 Mbit/s coverage or not, and to understand important drivers for rural high-speed broadband access.

Datsets

I have combined data from several sources for the analysis. The main dataset, the Nkom dataset, has eight columns and looks like this:

```
## # A tibble: 6 x 9
##   komnr building_nr boliger    nord      ost `10mbit` `30mbit` `100mbit` rural
##   <dbl>       <dbl>   <dbl>   <dbl>    <dbl>    <dbl>    <dbl>     <dbl> <dbl>
## 1   402        1570       1 336226. 6677907.        1        0         0     1
## 2   402        2445       1 350507. 6665967.        1        1         1     1
```

[1]The dataset and coverage maps can be found at www.nkom.no/teknisk/bredb%C3%A5nd/utbygging/dekningsinformasjon

```
## 3   402       13323       1 350601. 6665938.        1        1        1     1
## 4   402       13471       1 336831. 6671458.        1        0        0     1
## 5   402       13498       1 338025. 6669928.        1        0        0     1
## 6   402       13560       1 336636. 6671392.        0        0        0     1
```

It includes:
* *komnr* - the municipality ID which uniquely identifies ca. 370 municipalities in Norway
* *byggnr* - the unique building ID from the official property register
* *boliger* - the number of apartments in each building
* *nord and ost* - building geographical coordinates
* *10mbit, 30mbit, and 100mbit* - Flags for whether the building has 10, 30 or 100 Mbit/s broadband coverage

The Nkom dataset was combined with the following datasets[2]:
* **rural_b** - a list of building numbers that are located in rural areas.
* **dat_muni** - data from Statistics Norway that includes various municipality-level information such as population, size, income, and employment levels.
* **dat_hist** - data from Nkom regarding municipality-level historical coverage information from 2001 and 2013.
* **dat_support** - yearly data from Nkom that shows which municipalities that were the recipient of broadband deployment subsidies from the national government.

Data cleaning and wrangling

I wrangled the five datasets into one dataset - dat_all - and used that for analysis and modeling. The wrangling part was challenging since I have used data from different years and Norwegian municipalities tend to merge and split up every now and then. In addition, each municipality belong to one of ca. 15 counties, but the county structure has also been changed in recent years. In any case, dat_all has almost 458 000 rows and 28 variables. One of them, called Y100, is the outcome variable. When Y100 has the value '1' it means that the building has 100 Mbit/s broadband coverage, and a '0' means that it does not have coverage. The remaining 27 variables are predictors. Here's a glimpse of the dat_all dataset:

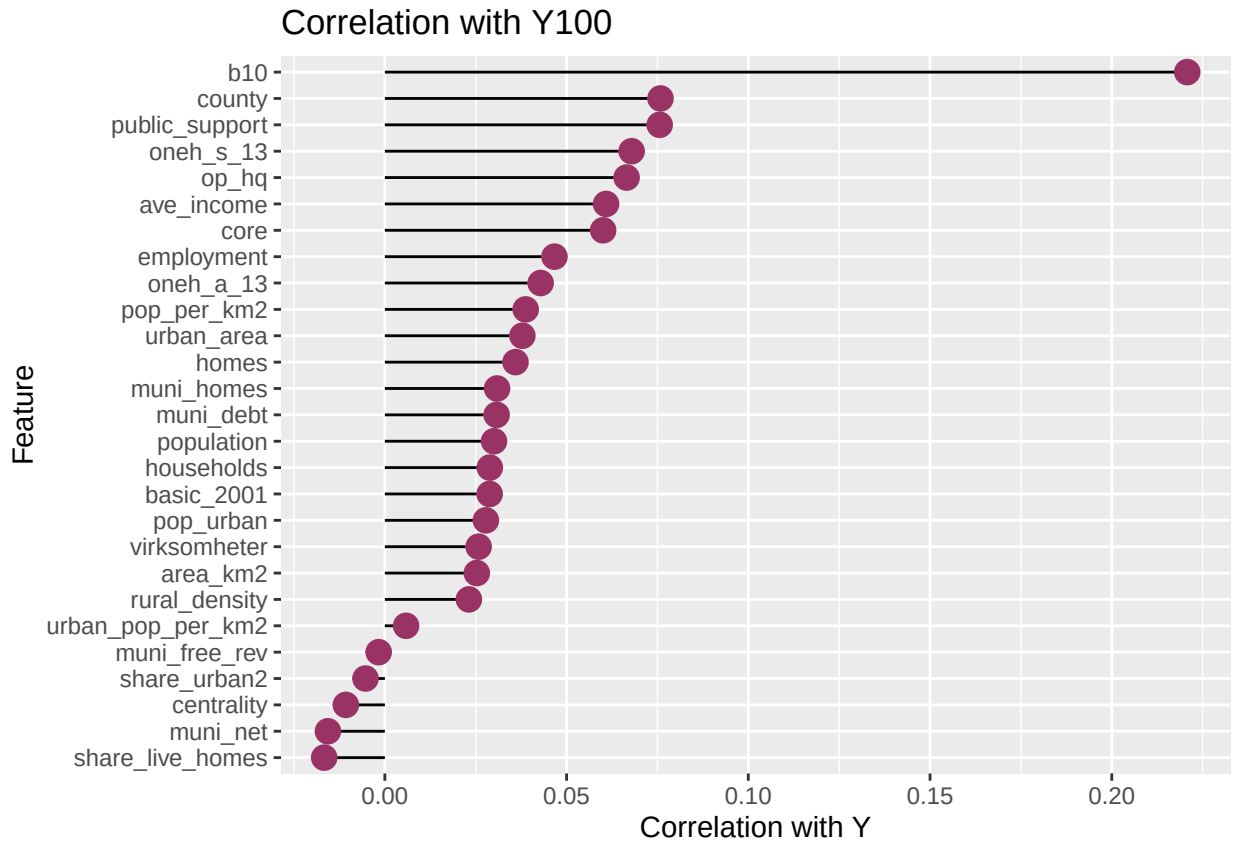
```
## Observations: 457,763
## Variables: 28
## $ Y100             <dbl> 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ homes            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ b10              <dbl> 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ population       <dbl> 17823, 17823, 17823, 17823, 17823, 17823, 17823, ...
## $ muni_homes       <dbl> 9088, 9088, 9088, 9088, 9088, 9088, 9088, 9088, 9...
## $ households       <dbl> 8741, 8741, 8741, 8741, 8741, 8741, 8741, 8741, 8...
## $ area_km2         <dbl> 952.78, 952.78, 952.78, 952.78, 952.78, 952.78, 9...
## $ urban_area       <dbl> 9.33, 9.33, 9.33, 9.33, 9.33, 9.33, 9.33, 9.33, 9...
## $ pop_urban        <dbl> 13078, 13078, 13078, 13078, 13078, 13078, 13078, ...
## $ share_urban2     <dbl> 0.733771, 0.733771, 0.733771, 0.733771, 0.733771,...
## $ centrality       <dbl> 799, 799, 799, 799, 799, 799, 799, 799, 799, 799,...
## $ ave_income       <dbl> 415900, 415900, 415900, 415900, 415900, 415900, 4...
## $ muni_net         <dbl> 2.1, 2.1, 2.1, 2.1, 2.1, 2.1, 2.1, 2.1, 2.1, 2.1,...
## $ muni_debt        <dbl> 118.2, 118.2, 118.2, 118.2, 118.2, 118.2, 118.2, ...
## $ muni_free_rev    <dbl> 55215, 55215, 55215, 55215, 55215, 55215, 55215, ...
## $ employment       <dbl> 58.6, 58.6, 58.6, 58.6, 58.6, 58.6, 58.6, 58.6, 5...
## $ virksomheter     <dbl> 692, 692, 692, 692, 692, 692, 692, 692, 692, 692,...
## $ op_hq            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ county           <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4...
```

[2]All of these datasets are available at my GitHub repository: github.com/harrall/coverage-ml

```
## $ rural_density     <dbl> 6.645917, 6.645917, 6.645917, 6.645917, 6.645917,...
## $ basic_2001        <dbl> 0.3747897, 0.3747897, 0.3747897, 0.3747897, 0.374...
## $ oneh_s_13         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ oneh_a_13         <dbl> 0.5879397, 0.5879397, 0.5879397, 0.5879397, 0.587...
## $ public_support    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ pop_per_km2       <dbl> 18.70631, 18.70631, 18.70631, 18.70631, 18.70631,...
## $ urban_pop_per_km2 <dbl> 1401.715, 1401.715, 1401.715, 1401.715, 1401.715,...
## $ core              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ share_live_homes  <dbl> 0.9618178, 0.9618178, 0.9618178, 0.9618178, 0.961...
```

2. Analysis

This chapter describes data wrangling, data exploration, and the modeling approach I decided on. First, I looked at the correlation between Y100 and the predictor variables. The chart looks like this:



Most variables are not very correlated with Y100, but some variables stand out:
* **b10:** Whether the building has 10 Mbit/s coverage - primarily from mobile 4G networks
* **county:** The county ID. As we will see later, this variable is quite important. Is is not a random ID: Rather, county 1 starts in the south-eastern part of Norway and it goes all the way to Finnmark county (no 18) in the far north of the country.
* **public_support:** The number of times that a municipality has received national support for broadband projects.
* **oneh_s_13:** The municipal coverage of symmetrical 100 Mbit/s networks (in practise, fiber access networks) in 2013. The 'core' variable has value 1 of this coverage was larger than 10% and 0 if not.
* **op_hq:** There are some 80 fiber access network operators in Norway that provide 100 Mbit/s networks. The op_hq variable has a value of 1 if the building is located in a municipiality where one of these 80

operators has their main office.
* **ave_income:** The average gross income per adult in the municipality.

Also, some variables are highly correlated with each other. The table below shows that the "households" variable has a higher than 0.8 correlation with several other variables that I removed from the dataset.

```
# Correlation table
corr %>% filter(households > 0.8) %>% subset(select = c(rowname, households))
```

```
## # A tibble: 5 x 2
##   rowname      households
##   <chr>             <dbl>
## 1 population        0.999
## 2 muni_homes        1.00
## 3 urban_area        0.924
## 4 pop_urban         0.997
## 5 virksomheter      0.995
```

```
rm(corr)
```

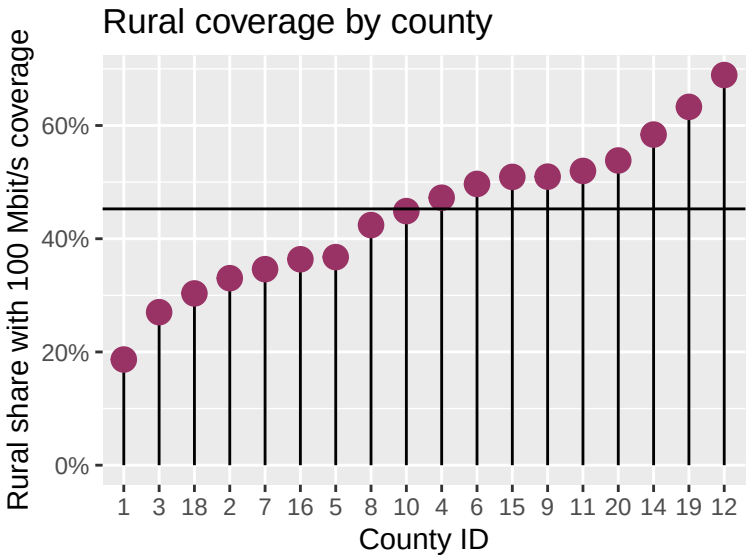
Next, I looked at the average 100 Mbit/s coverage for rural buildings:

```
# Find average coverage
y_hat <- 1
average <- mean(y_hat == dat_all$Y100)
```

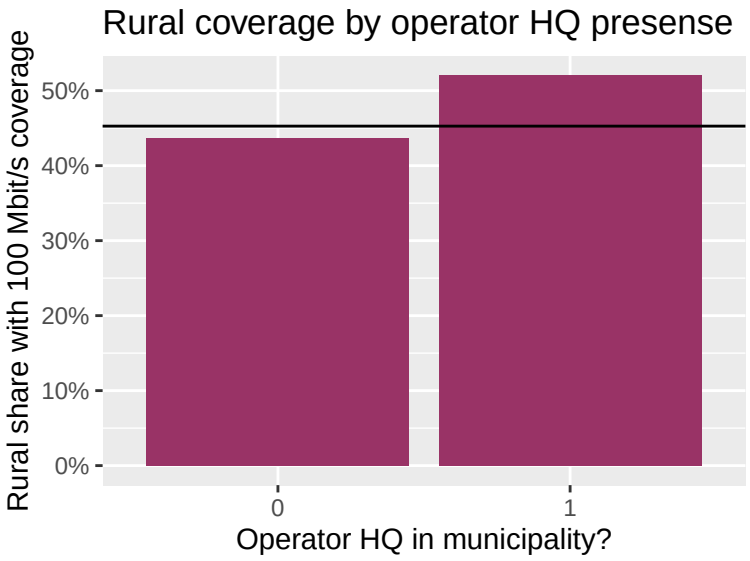
The average 100 Mbit/s coverage is 0.452719. This means that by guessing that a building will *not* have coverage, we would be correct ca 55% of the time.

Analysis - categorical variables

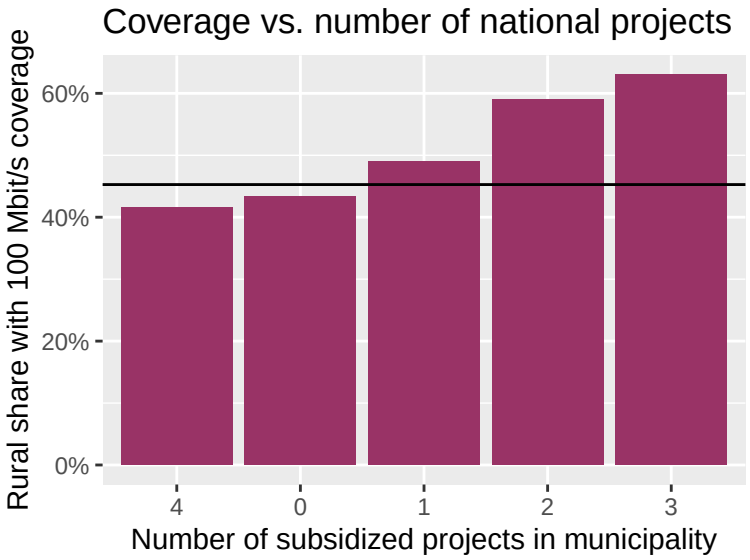
The chart below shows coverage by county. It is clear that the likelyhood of having 100 Mbit/s coverage will vary quite a bit with the county that the building is located in.



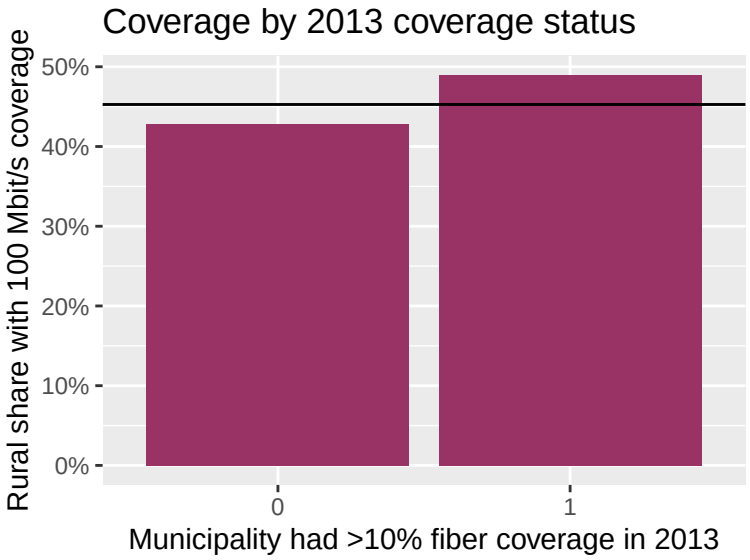
The chart below show coverage by precence of operator headquartes. The rural coverage is slightly higher in municipalities that hosts the main office of a fiber network.



The chart below shows rural coverage by the number of times that a municipality has received national support. It is interesting that the municipalities with the highest number of supported projects have the lowest coverage.



The chart below shows that rural buildings located in municipalities with some (>10%) fiber broadband coverage in 2013 has higher chance of having 100 Mbit/s coverage today.



Analysis - continous variables

The charts below shows density plots for some of the predictors that are continous. For most of them, it is hard to spot any meaningful differences between the predictors in terms of 100 Mbit/s rural coverage for rural buildings. But buildings located in municipalities with higher average income seems to have slightly better 100 Mbit/s coverage than others.

3. Modeling

This chapter desribes the modeling work I did with the dat_all data. First, I removed highly correlated variables and normalized the continous variables so that they have mean=0 and a standard deviation of 1. Then, I sampled 50,000 rows from the dat_all dataset in order to reduce model training time. Then, I split the reduced dataset into a training set and test set. Here's the code:

```
######### Part 3 ###############
# Modeling
################################

## Remove highly correlated and redundant variables
dat_all <- dat_all %>%
  subset(select = -c(pop_urban, population, muni_homes, urban_area, urban_pop_per_km2,
                     virksomheter, muni_free_rev, share_urban2))

## Normalize continous variables with mean=0 and sd=1
dat_all <- dat_all %>%
  mutate(ave_income = (ave_income - mean(ave_income))/sd(ave_income),
         centrality = (centrality - mean(centrality))/sd(centrality),
         employment = (employment - mean(employment))/sd(employment),
         muni_debt = (muni_debt - mean(muni_debt))/sd(muni_debt),
         muni_net = (muni_net - mean(muni_net))/sd(muni_net),
         area_km2 = (area_km2 - mean(area_km2))/sd(area_km2),
         households = (households - mean(households))/sd(households),
         rural_density = (rural_density - mean(rural_density))/sd(rural_density),
         pop_per_km2 = (pop_per_km2 - mean(pop_per_km2))/sd(pop_per_km2))

## Make test and training set
# Development set - reduce dat_all to reduce computation time
set.seed(1, sample.kind = "Rounding")
dat_small <- sample_n(dat_all,50000,replace = FALSE)
rm(dat_all)

dat_small <- dat_small %>% subset(select = -c(Y100))

test_index <- createDataPartition(y = dat_small$y100f, times = 1, p = 0.1, list = FALSE)
train_set <- dat_small[-test_index,]
test_set <- dat_small[test_index,]

train_set <- as.data.frame(train_set)
test_set <- as.data.frame(test_set)
rm(dat_small, test_index)
```

A first model - just the average

In the first model I simply took the chance of not having coverage and used that as the predictor.

```
## The first model - accuracy = 0.55
y_hat <- 0
mu_naive <- mean(y_hat == test_set$y100f)
mu_naive
```

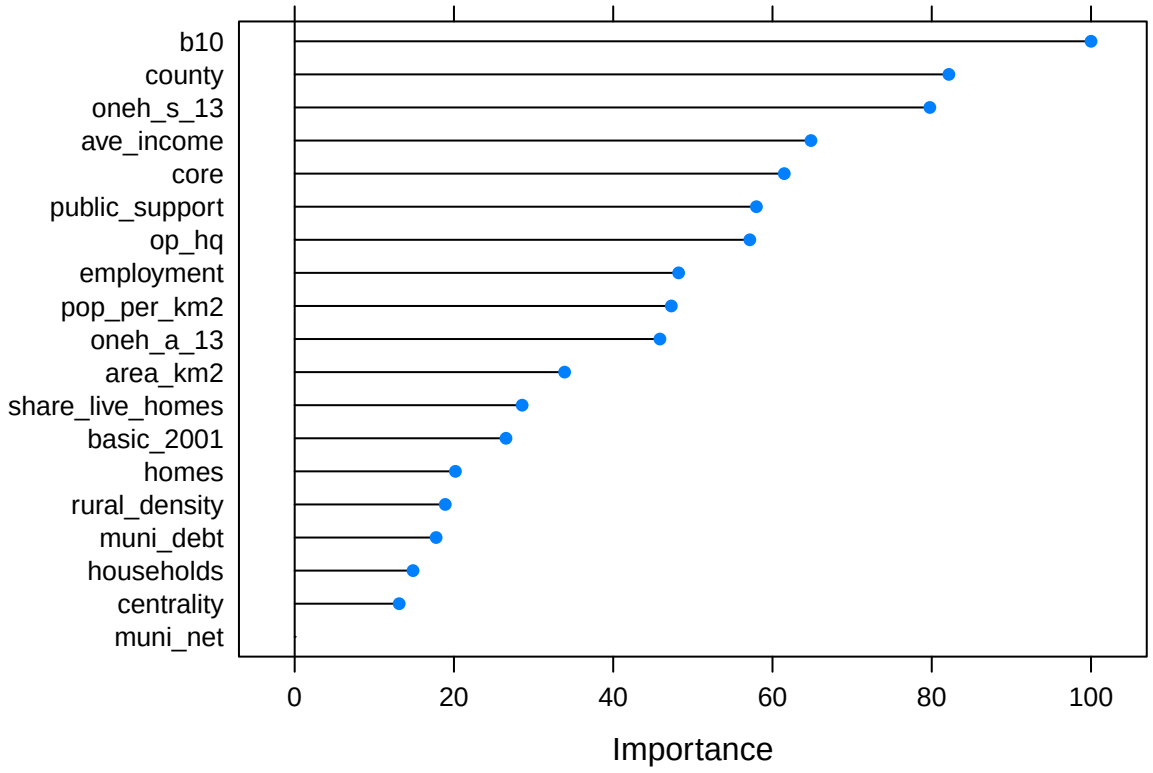
```
## [1] 0.5504
```

The chance of *not* having 100 Mbit/s coverage is 0.5504 on the training set. If we were to guess, our best guess would be "this building does not have coverage", and we would be right ca 55% of the time.
But can we do better? Let's try a few machine-learning models.

The Caret package and the LDA method

The Caret package is a framework for building machine learning models in R and supports a number of modeling methods. I started out with the Linear Discriminant Analysis (LDA) method. Linear Discriminant Analysis (LDA) is a well-established machine learning technique and classification method for predicting categories. Its main advantages, compared to other classification algorithms such as neural networks and random forests, are that the model is interpretable and that prediction is easy. Linear Discriminant Analysis is frequently used as a dimensionality reduction technique for pattern recognition or classification and machine learning[3]. Here's the code and a plot that shows the relative importance of the variables:

```
# LDA model
train_lda <- train(y100f ~ .,method = "lda",data = train_set)
acc_lda <- confusionMatrix(predict(train_lda, test_set), test_set$y100f)$overall["Accuracy"]
lda_imp <- varImp(train_lda)
plot(lda_imp)
```



The accuracy achieved with the LDA method is 0.6044. This result is not much higher than what plain guessing achieved. Can we do better with other methods?

[3]Source: DisplayR blog - Linear Discriminant Analysis in R - An introduction

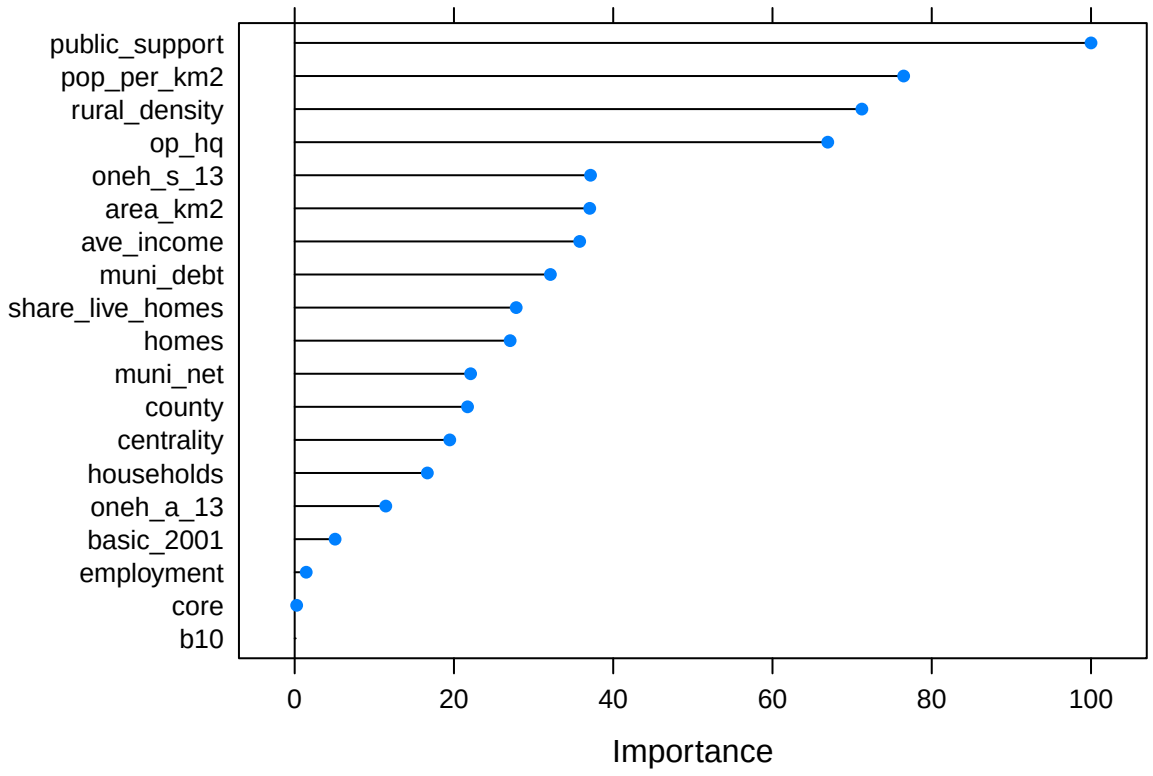
The logistic regression model (GLM)

Logistic regression is a technique that is well suited for examining the relationship between a categorical response variable and one or more categorical or continuous predictor variables. The model is generally presented in the following format, where beta refers to the parameters and x represents the independent variables:

$$log(odds) = \beta_0 + \beta_1 * x_1 + ...\beta_n * x_n$$

The log(odds), or log-odds ratio, is defined by ln[p/(1-p)] and expresses the natural logarithm of the ratio between the probability that an event will occur, p(Y=1), to the probability that it will not occur[4]. Here's the code and a plot that shows the relative importance of the variables:

```
# GLM model
train_glm <- train(y100f ~ .,method = "glm",data = train_set)
acc_glm <- confusionMatrix(predict(train_glm, test_set), test_set$y100f)$overall["Accuracy"]
glm_imp <- varImp(train_glm)
plot(glm_imp)
```



The accuracy achieved with the GLM method is 0.605. This result is a little better than the LDA method, but it is not much better than guessing. What about other models?

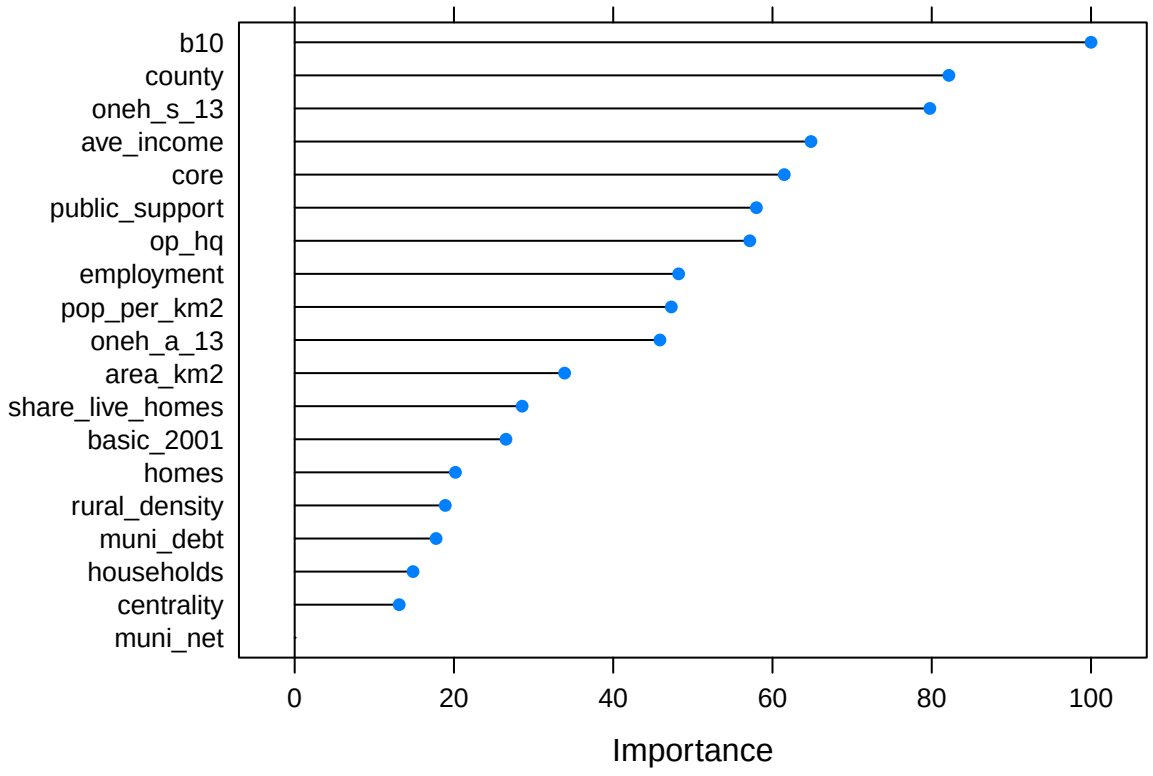
[4]Source: R-bloggers: Evaluating Logistic Regression Models, August 2015

The k-nearest neighbors model (kNN)

Unlike the methods I have used until now, the kNN algorithm is non-parametric since it uses the neighbor points information to predict the outcome.

At first, the kNN model defines the distance between all observations based on the features. Then, for any point (x1,x2) for which we want an estimate of p(x1,x2), we look for the k nearest points to (x1,x2) and then take an average of the 0s and 1s associated with these points. We refer to the set of points used to compute the average as the neighborhood[5]. Here's the code and a plot that shows the relative importance of the variables:

```
# KNN model
train_knn <- train(y100f ~ ., method = "knn",data = train_set, use.all=FALSE)
acc_knn <- confusionMatrix(predict(train_knn, test_set), test_set$y100f)$overall["Accuracy"]
imp_knn <- varImp(train_knn)
plot(imp_knn)
```



The accuracy achieved with the kNN method is 0.6896. This result is better than the parametric LDS and GLM methods, and almost 15 percentage points better than guessing. Not bad! But can we do even better?

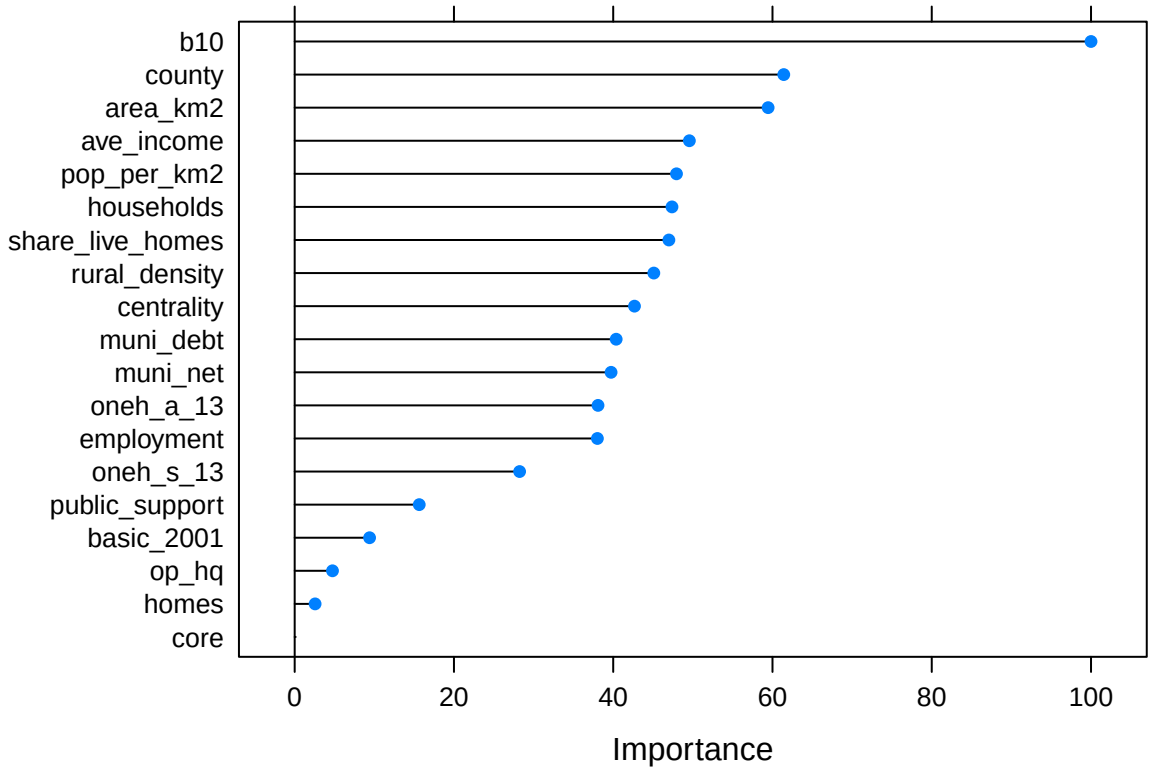
The random forests model (RF)

According to the author Anish Walia, ensemble learning is a type of supervised learning technique where the basic idea is to generate multiple models on a training dataset and then combine them to generate a

[5]Source: Introduction to Data Science, Rafael A. Irizarry, Feb-2020

well-performing model which aviods overfitting.
Random Forests is an ensemble learning technique, and in the data science class we have learned that "Random forests are a very popular machine learning approach that addresses the shortcomings of decision trees using a clever idea. The goal is to improve prediction performance and reduce instability by averaging multiple decision trees (a forest of trees constructed with randomness)."[6]
Here's the code for the Random Forest model and a plot that shows the relative importance of the variables:

```
# Random forests
train_rf <- train(y100f ~ ., method = "rf", data = train_set)
acc_rf <- confusionMatrix(predict(train_rf, test_set), test_set$y100f)$overall["Accuracy"]
rf_imp <- varImp(train_rf)
plot(rf_imp)
```



The accuracy achieved with the Randon Forests method is 0.7138. This result is the best so far, and we're now far from 'simply-guessing' territory. There is little question that we can increase the quality of rural broadband coverage predictions using machine learning methods.

[6]Source: Introduction to Data Science, Rafael A. Irizarry, Feb-2020

4. Results and conclusion

Results table

The table below summarizes the accuracy achieved from the various models applied to the validation dataset called "train_set".

Table 1: Accuracy results

model	Accuracy
Naive method	0.5504
LDA	0.6044
GLM	0.6050
KNN	0.6896
Random Forest	0.7138

Clearly, the choice of machine-learning method really matters. The quality of predictions varies quite a bit with the algorithm, and it is important to try out different models before making a final decision on which model to use.

Conclusion

The report describes my work with various rural broadband prediction models. I have documented how I wrangled the data, explored the data, made different models using the dat_all dataset, and also how I applied the final model to the validation set.
But is the project objective - to develop a model for high-quality rural broadband predictions - met? In my view, no. The final model has an accuracy of 0.7138 which is much better than simply guessing. But it is not good enough to make investment decisions or to inform public policy.
More pre-processing of the data, such as creating creating dummy variables from categorical variables, could possibly increase the accuracy. But in order to make signifiant model improvements I think it is necessary to have a wider dataset with more information about the each building. With a few exceptions, my feature variables are on a municipal level: We know quite a bit about the municipality that each building is located in, but we know very little about each building. For example, it would be valuable to have more geographic, building-level information such as distance to nearest existing fiber node. This would be valuable variables for future work.