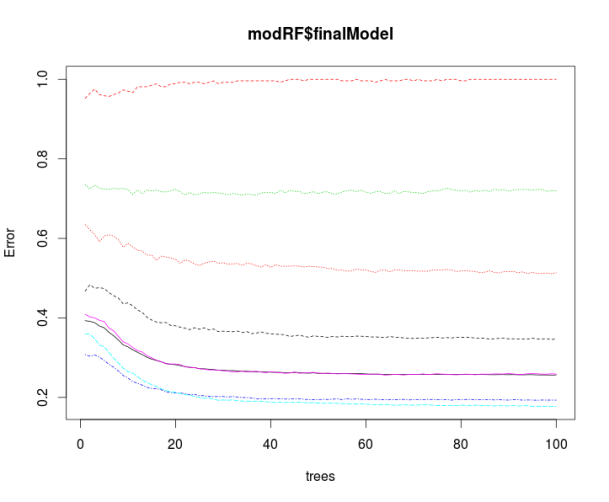
# Random Forest

nTree=100, TrainFull

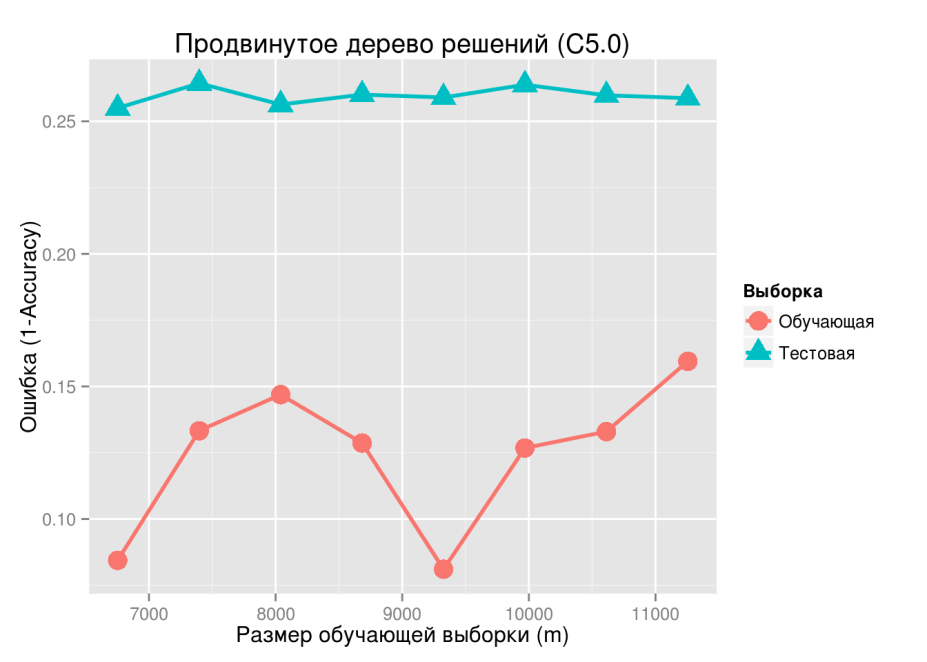


Видим, что в принципе достаточно 50 деревьев.

# Boosted DT (C5.0)

## На сокращенной выборке

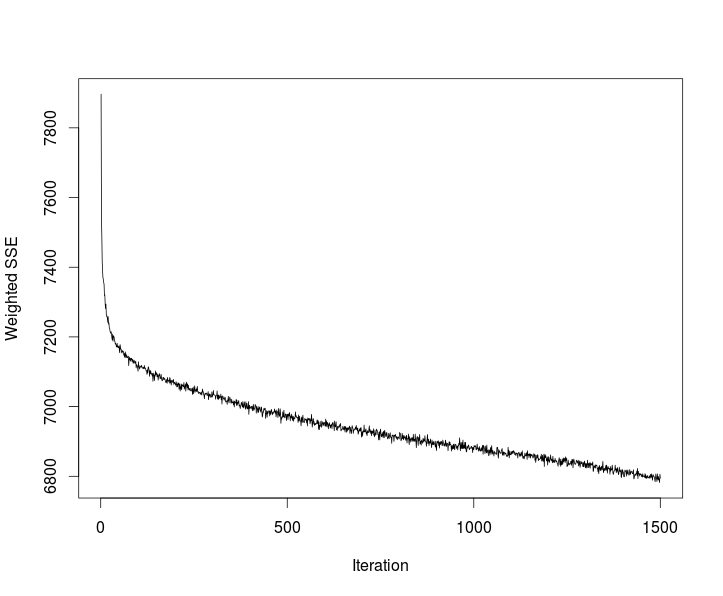
* model = rules,
* winnow = FALSE,
* trials = 60.



Посмотреть на нее на полной выборке!

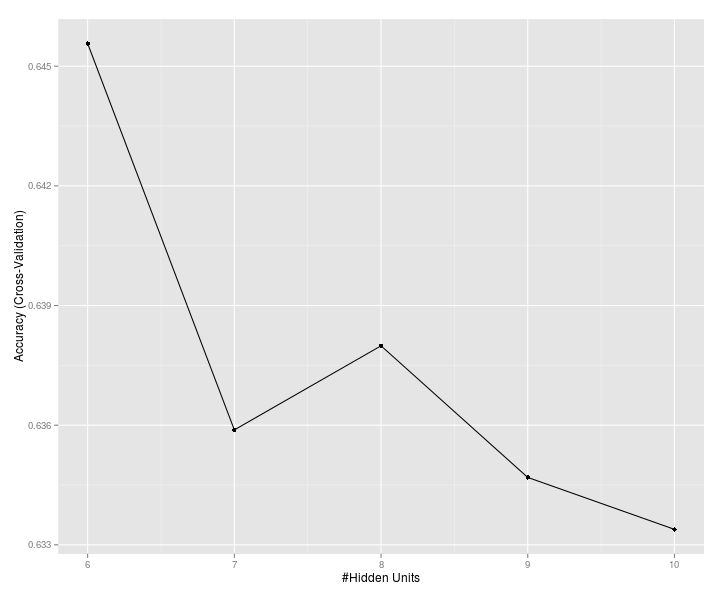
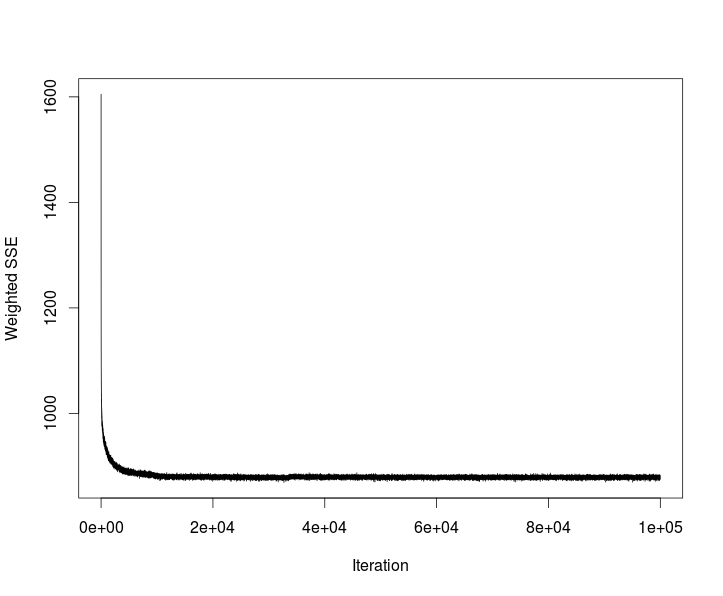
# RSNNS

# 11255, IF1, maxIter=1500, 8 min, erorrs, size=6, acc= 51.75%

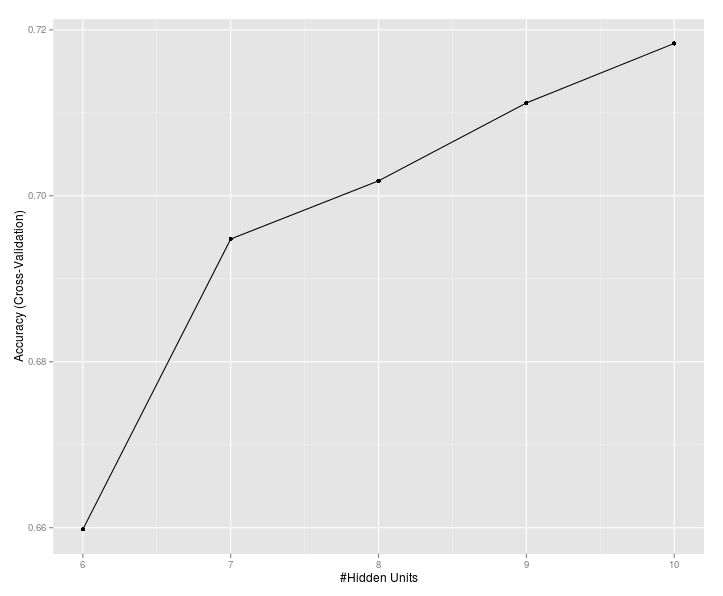
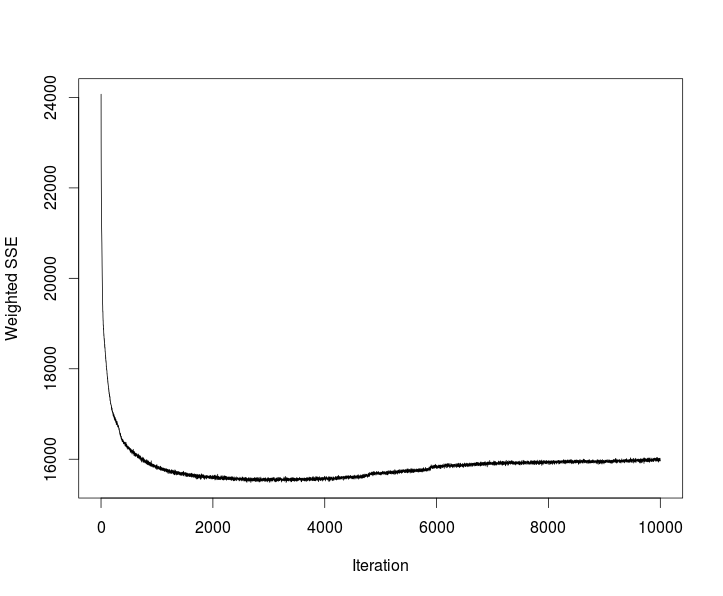


Можно увеличивать число итераций!

На нормализованном наборе данных CMSS, IF3, 100 000 итераций:

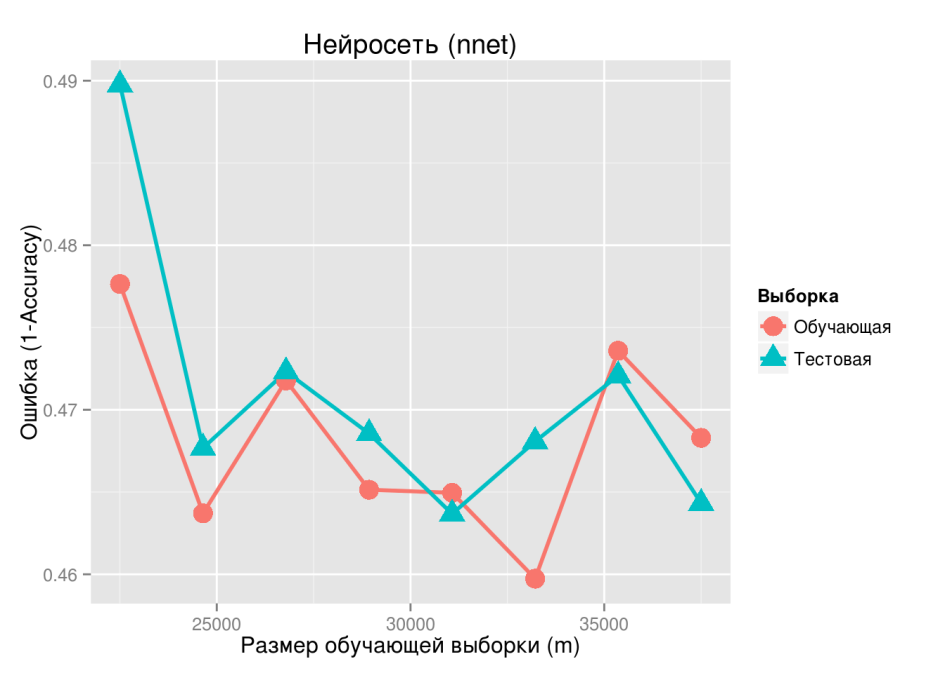


На нормализованном наборе CMYJR, IF3, 10 000 итераций, лучше предыдущего:



# Nnet

* maxit = 500,
* size = 8,
* decay = 10^{-4}.



Модель сильно сдвинута. Странно, что настолько сильно, что-то тут не так..

* Может помочь нормализация!
* Больше итераций

# TODO:

1. Нейросети
   1. (+) Построить RSNNS и посмотреть, какое количество итераций нужно
      1. ?Попробовать на 10 000 итерациях.
   2. (+) Взять VarImp от RF и потренировать нейросети на существенных переменных
   3. (+) Попробовать nnet на 10 000 итерациях и большем числе нейронов при IF3. Дает небольшой прирост точности (~0,5%)
   4. (+) Попробовать с нелинейной нормализацией. – Стало хуже!
2. Линейная регрессия?
3. Случайный лес
   1. Похоже, что переобучается. Побороться с этим:
      1. Bagging (чтобы снизить разброс)
         1. (+) Bagged AdaBoost. method = 'AdaBag'. Tuning Parameters: mfinal (#Trees), maxdepth (Max Tree Depth). Нет улучшения
         2. (+) Bagged CART. method = 'treebag'. No Tuning Parameters. Нет улучшения
      2. Регуляризация. Сомнение в том, что увеличится сдвиг, а этого не хочется. Нет улучшения
      3. Оставить меньше параметров (feature selection).
4. (+) SVM
5. Категорийные параметры
   1. (+) Вставка пустых категорий – пропорционально существующим
   2. (+) Замена каждой категории на номер наиболее вероятного класса
      1. (+)Довести до оценки – прогресс!
   3. Разложить многокатегорийные переменные по малому числу категорий (~5 шт)
   4. mice <http://www.r-bloggers.com/imputing-missing-data-with-r-mice-package/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+RBloggers+%28R+bloggers%29>
6. (+) Нелинейная нормализация числовых категорий
   1. (+) YeoJohnson + range
   2. (+) YeoJohnson (без range)
   3. (+) tanh
7. (+)#guided random forest (help RRF) – нет улучшения
8. Сквозная нормализация (объединить тренировочный и тес товый датасеты)
9. Построить модель на редкий класс, который плохо предсказывается RF и добавить такую фичу в датасет на вход RF
10. У многих числовых параметров количество NA одинаковое или близкое. Посмотреть на NA значения в числовых параметрах – как они распределены по строкам. Если окажется, что NA в разных столбцах принадлежат одним и тем же строкам, то выкинуть эти строки из обучающей выборки. Mice?
11. Для возрастной группы 0 слишком мало тренировочных данных. Объединить ее с возрастной группой 1.
12. (+)Предсказывать не категорию, а линейную величину от 0 до 7 и выделять классы по диапазонам. Или добавить ее как новую фичу и потом предсказывать категорию.
13. Все предсказания делать на полной тренировочной выборке (50к записей)
14. Ylin предсказывать после y0-y7. Имеет ли смысл?????
15. При финальном моделировании RF использовать сбалансированный датасет (oversampled)
16. VarImp
    1. Подумать, где сокращение переменных может помочь. Выделить модели, у которых ошибка на обучающей выборке ~0, а на тестовой >0 (высокий разброс).
       1. Random Forest
17. Ночью
    1. ? RSNNS, maxIter = 10 000.
    2. modSVM, TrainFull, IF2
    3. LogitBoost, allFeatures
    4. C50 на полной выборке.

# Выводы

1. Начинать с простой модели (линейная регрессия, случайный лес)
2. Не тратить время на SVM
3. Сразу тренировать нейросеть и смотреть, достаточно ли данных для нее. Если да, то тюнить ее.