REPORT

Assumptions

- Provided dataset has no information what type of data it represents. There is no explanation for dependent variable nor predictors. I assume this is done on purpose. With that being said, I treat the model from strictly theoretical perspective.

- Since there is a lack of context, it is impossible to benchmark the model against any industry standard. I establish arbitrary RMSE baseline (57.00891) given by Random Forest model which uses all available variables as predictors. The baseline model is trained using *rpart* library on the same training data set as any other. I assume, every model with lower RMSE than the baseline has better performance.

- I assume simpler model with less predictive variables is preferred over more complex one, whenever both models evaluate to similar RMSE.

- I work with two families of predictive models. GLM and Random Forest. As per theoretical knowledge Random Forests tend to over-fit more than GLM. Given two models with similar performance, I assume GLM is preferred over the Random Forest.

- Provided dataset contains variables with multiple undefined (NA) values. In most occasions undefined values represent more than 30% observations. I assume these variables does not contribute significantly to the predictions accuracy. For simplicity of the analysis, I decided to remove all variables which contains undefined variables. In my method, I study how this assumption impacts the prediction accuracy.

- I assume that if for some case the prediction cannot be generated, it should be considered as incorrect prediction while computing the accuracy.

Methods and Algorithms

Step 1. Introduction.

- I build predictive model in the iterative process. The first step is the explanatory analysis of the data set. It is useful to understand the distributions of depended variable as well as few predictors. There are too many potential predictors to explore all of them manually.

- I split the available data set in two portions: train and test. I use random uniform sampling to generate 70% - 30% split. I train every model on the train dataset. I evaluate every model on the test dataset.

- In every step, I compare GLM and Random Forest models trained and evaluated on the same data sets.

Step 2. Benchmark.

- I start building the model from establishing a benchmark. I consider GLM with all available variables and Random Forest with all available variables.

- Given that, at this stage the Random Forest performs better by factor of 10, it is my choice for the benchmark.

Step 3. Undefined variables.

- My objective is to reduce the number of the variables which do not contribute to the prediction accuracy.

- I observe that, there are two types of variables. The first type are variables which are defined for all observations. The second type are variables which contains undefined values for many observations. For the simplicity, I decide to remove all variables that contain undefined observations.

- I train and evaluate the model, which uses only these variables which are properly informed for all the observations. The GLM model represents similar performance than the benchmarked Random Forest.

Step 4. Variable scaling.

- Explanatory analysis showed that: (1) the numerous predictor variables are highly left skewed, (2) the distribution of the dependent variable is slightly right skewed.

- In such situation I consider useful to employ the variable transformation. Given that almost all variables contain zeroes I prefer to use root transformation rather than log or 1/x. For negative values I maintain the sign.

- Given that the number of predictors in my model is still big, I decide to apply transformation uniformly to all the predictors.

Step 5. Choose appropriate root scaling.

- I apply the square root scaling and I evaluate the resulting model. Since the performance improves I decide to check the cube root.

- Given that, the performance improves with cube root, I choose the scaling factor algorithmically. Thus I verify the performance of 10 different models. Each of this model uses different scaling by root from 1 to 10.

- I observe that, the model has best performance when predictors are scaled by 5th root, that is: *x^(1/5)*. The model with such scaling is my best choice.

Step 6. Remove skewed predictors.

- I check whether removing predictors with high skewness improves the performance of the model.

- I consider the variable is high skewed, if the skew of its distribution is higher than one. I remove such variables, and construct the model only with low skewed predictors.

- Since the variable scaling from the pervious step gave me a good result, I decide to maintain the variables scaled by 5th root, while evaluating the model with low skewed predictors.

- The models (GLM nor Random Forest) do not seem to improve, thus I decide do not continue employing this technique.

Step 7. Remove insignificant variables.

- My objective is to reduce number of redundant predictors if possible. Given the GLM model with scaled variables, I examine the p-value of the predictors.

- I decide to evaluate the model, which contains only these variables which p-value is lower than 0.01.

- Unfortunately, I do not observe any positive result of the variable reduction. Rather than that the model performs slightly worst.

- I examine the rank of the matrix generated by predictors and observations. The rank shows me there are 5 vectors which might be linearly dependent. Given that the number of predictors is high: 263, I do not consider the linear dependency is any significant issue. Thus, I decide to ignore the warning about the rank deficient fit.

Step 8. Final model.

- Resuming all the trials and iterations I decide to chose my final model as GLM with identity link function created in step 5. My model uses 263 predictors scaled uniformly by root 5. The predictors are not expected to have undefined values.

- I use the strip function to pack and save model to the .rds file.

Tools and frameworks

- For the entire analysis and the model construction, I use R programing language version 3.2.4.

- R runs in the interactive IDE environment: RStudio on in OS X El Capitan 10.11.6

- In addition to standard R functions I use external R packages:
    - rpart
    - e1071

  available in CRAN repository.

Results and evaluation

- **My best choice model has RMSE = 37.5 which represents around 34% performance uplift against the benchmarked model. The model accuracy is 7.42%.**

- The following table shows the evaluation of all my models.

Model	RMSE	Accuracy %
First-try model: GLM with all variables as predictors.	508.3635	0.2300
Benchmark model: RF with all variables as predictors.	57.00891	5.2
GLM, variables with only complete observations	54.69914	4.92
RF, variables with only complete observations.	58.80077	4.86
GLM, squared root variable scaling.	41.38065	6.86
GLM, cubic root variable scaling.	38.25813	7.25
Best Choice: GLM, 5th root variable scaling.	37.53679	7.42
RF, 5th root variable scaling.	59.0666	4.80
GLM, without skewed variables	42.45525	6.44
RF, without skewed variables	65.19326	4.19
GLM, low p-value predictors	36.30616	7.18
RF, low p-value predictors	59.0666	4.8

Instruction to run the model

- The model should be run in an environment with installed **R** version 3.2.4 or higher.

- To run the code, clone GitHub repository:
  https://github.com/rafalszota/datascienceexcercise
  and navigate to the repository folder.

- Alternatively, if GitHub cannot be used, refer to the attached .7z file, unzip and navigate to the desired folder

- The model can be run directly from command line using following command:

  ```
  $ Rscript final_model.R [input_file.csv]
  ```

  where the *input_file.csv* is the name of the file with dataset to do predictions.

- The script installs required R packages for CRAN repository whenever it is run.

- In order to run the script properly the environment needs the Internet connection and eventually proxy settings so that the url:
  http://cran.rstudio.com can be reached.

- The script outputs result similar to this:

  ```
  [1] "RMSE: "            "54.8751332141919"
  [1] "Accuracy %: " "4.95"
  [1] "Results written to: " "result.txt"
  ```

- The script requires read/write permissions in folder it is executed.

- The script runs fairly smooth in environment with Intel Core i5 2.7 GHz and 8GB RAM. Any environment with equal or higher characteristics should be satisfactory.

- The script running time (including the library installation) for 100.000 observations should be less than 60 seconds.