**Learning Journal: Linear models for predicting success in marketing campaigns**

**1. Describing the dependent variable (column 3) in terms of the count, standard deviation, mean and median values, describing the general shape and whether it can be considered a normal distribution.**

Feature mean: 619260.213355

Feature std dev: 2460481.30781

Feature std error of the mean: 27540.0172749

Feature median: 160190.0

7982 values for Y train -unique: 7912

An idea for finding if data can be considered a normal distribution is doing a 2 hypothesis t-test to compare means. However, an issue that affects a 2 Hypothesis test is whether the data are censored, ie. values from the sample are somehow restricted. Censoring occurs if the range of potential values are limited such that values from one or both tails of the distribution are unavailable (e.g., right and/or left censoring where high and/or low values are missing [3]). This could be an issue to be aware here, since we may cut a campaign from running if it is underperforming for max X days, since we do not want to waste marketing campaign costs so we do not allow it running live more than X days).

[Normality tests](http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R4_One-TwoSampleTests-ANOVA/R4_One-TwoSampleTests-ANOVA2.html) [5]:

1) Plotting the histogram for feature in column 3 could give a good picture of normality if we had more data, but a better visual test is the normal probability plot. If the plot is linear, it is consistent with being sampled from a normal distribution (this test help the human eye at detecting linearity versus detecting bell curve)[1].

2) Using both visualisation and scipy.stats.normaltest(data) in method *test\_normality*, we can see that Y\_train cannot be considered a normal distribution (confirmed by a p-value of 0.0 given that H0 = “Both distributions are drawn from a normal distribution”. This means that there is statistical significance between the normal distribution and our sample distribution). This method in scipy is based on D’Agostino and Pearson’s test that combines skew and kurtosis to produce an omnibus test of normality. p-value returned is a 2-sided chi squared probability for the hypothesis test.

3) We may conduct a formal test, the Shapiro-Wilk test, to see whether the data come from a normal distribution.

H0: The data are sampled from a normal distribution

H1: The data are not sampled from a normal distribution.

A small significant p-value (< 0.05), as shown below, means that we reject the null hypothesis (and conclude that the data is probably not normally distributed).

> shapiro.test(rm)

W = 0.9609, p-value = 2.412e-10

We see that we reject the null hypothesis that the rm variable was sampled from a normal distribution (W = 0.9609, p-value = 2.412e-10).

[What do we do if the normality assumption fails?](http://www.mayo.edu/mayo-edu-docs/center-for-translational-science-activities-documents/berd-5-6.pdf)

1) Here is where the non-parametric test comes in. The Wilcoxon Signed rank test can compare the median to a hypothesized median value. For example, in our case,

wilcox.test(dis, mu=3.5) or with with continuity correction:

wilcox.test(dis, mu = 3.5, alternative="less")

wilcox.test(dis, mu=3.5, alternative= "greater")

[1] Just because a population has a normal distribution doesn't mean that a random sample drawn from the population will "look" (in a histogram) normal. Of course for large samples the representativeness is more accurate. If the sample is not large enough, examining the histogram may not give the whole picture or allow full detection of "bell-shaped" curve. The normal probability plot tackles this issue.

**- Are other columns correlated to each other? To Y? Which ones and how to determine the correlation?**

Depending on outliers, unequal variances, normality and linearity of features, there are different tests to compute correlation [12]. Pearson correlation coefficient R is a common one used for finding correlation among features and consequently, eliminate redundant features. We could set a rule to use only in our model those features that have absolute value of Pearson correlation coefficient r > 0.1 (both among original features and a linear combination of features). The sign (close to -1 or close to 1) will indicate the positive or negative correlation on the dependent variable Y.

After normalising categorical feature in column 1, we have 7224 columns. Out of those, 38 have a correlation coefficient of Pearson P larger than abs(0.05) with the dependent variable Y (all of these are the categorical feature). However, Pearson correlation test is only applicable when each dataset is normally distributed, which is not the case for our Y variable [9]. However, Spearman correlation coefficient, on the contrary, is non parametric (its exact sampling distribution can be obtained without requiring knowledge about the parameters of the joint probability distribution of X and Y). Spearman is applicable when X and Y are related by any monotonic function (in contrast to the Pearson correlation, which only is precise when X and Y are related by a linear function)[10]. Spearman’s Rho is considered as the regular Pearson’s correlation coefficient in terms of the proportion of variability accounted for, whereas Kendall’s Tau represents a probability, i.e., the difference between the probability that the observed data are in the same order versus the probability that the observed data are not in the same order [12]. Therefore I provide a grid plot on both Kendall and Spearman correlation coefficients.

For the categorical features, correlation per se is not adequate, and we compute for each pair of features, its covariance matrix using a vectorized implementation that allows to compute instead the the mutual information coefficient (MIC). This method MIC\_plot shows features with high correlation for those values of MIC closer to 1.

Once detected highly correlated features, one can do:

1. PCA (using grid search on hyper parameters of the model selection like the # of components in PCA) and they will be naturally gone, or
2. Choose the conjugates coordinates, meaning if x and y are highly correlated choose r= (x+y)/2 and s = (x-y)/2 then r is effectively representing one of them while s is representing the deviation of one from the other. Otherwise, the main feature to attend is how removing or adding these redundant features affects the CV score.

**2. Looking at Y (Column 3 in the test data), what regression models should one first and what are the pros and cons of each?**

In terms of regression algorithms, options are, as a summary of the most popular regression algorithms:

* Ordinary Least Squares Regression (OLSR)
* Linear Regression
* Logistic Regression
* Stepwise Regression
* Multivariate Adaptive Regression Splines (MARS)
* Locally Estimated Scatterplot Smoothing (LOESS)
* Polynomial Regression
* Ridge Regression
* Lasso Regression
* ElasticNet Regression

1) Linear regression: alone suffers from multicollinearity and is sensitive to outliers. Solved with ordinary least squares.

2) Mixed model regression with condition as a fixed effect on different features and maybe date as a random effect (to estimate and remove variability due to events going on outside of our control, like school holidays or weekends, etc. to improve the chance of observing a difference and to better estimate the predictor Y).

3) Polynomial regression: makes a linear estimation by fitting a nonlinear model to the data (when linear relations among variables may not hold and we will have a more complex model with a second term for x such as y = a0 + bx + cx^2). Non-linear least squares can be used.

4) Maximum likelihood estimation (MLE), a special case of Maximum a posteriori (MAP, which ignores the priors and therefore is unregularized). It is a parametric method requiring mean and variance of an assumed normally distributed variable.

5) Gradient descent and stochastic gradient descent (including random samples out of all the set). In order to model a set of points using a line defined by two parameters, gradient descent attempts to find the best values for the line's slope m, and y-intercept b parameters, subject to an error function.

6) Other types of regression: Bayesian (using conditional distribution and errors that have a normal distribution, assumes a prior distribution to obtain a posterior probability distribution), Ecological and Robust regression (designed to be not overly affected by violations of assumptions by the underlying data-generating process of other methods). Bayesian regression uses the conditional probabilities and ordinary least squares to estimate the coefficient vector using the Moore-Penrose pseudoinverse.

I would try linear regression models first of all. In order to prevent overfitting, I would apply three different kinds of regularization techniques:

a) Lasso or L1 regularizer

b) Ridge regression or L2 regularizer

c) a weighted balance of L1 and L2 regularizers through Elastic Net regularizer. This method, although takes longer training time and is less maintainable in time, provides the best compromise of both kinds of L1 and L2 regularization techniques.

Ridge regression can't zero out coefficients and thus, you either end up including all the coefficients in the model, or none of them" [4]. However, LASSO does both parameter shrinkage and variable selection automatically. If some of your covariates are highly correlated, Elastic Net may be better instead of the LASSO. Another approach is using the Non-Negative Garotte (NNG) [1] as it's consistent in terms of estimation and variable selection [2]. Unlike LASSO and ridge regression, NNG requires an initial estimate that is then shrunk towards the origin. In the original paper, Breiman recommends the least-squares solution for the initial estimate (you may however want to start the search from a ridge regression solution and use something like GCV to select the penalty parameter) [3]" [4]

Ridge regression and Lasso are subset selection methods or shrinkage methods that use least squares to fit a linear model that contains a subset of predictors. Shrinking the coefficient estimates towards zero can significantly reduce the model's variance. Lasso uses an L1 norm penalty of the coefficient vector while Ridge uses an L2 norm. Cross validation for an optimal lambda value election is critical for performance of these models.

Because the standard least squares coefficient estimates are scale equivariant, the ridge regression coefficient estimates can change substantially when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function. Therefore, it is best to apply ridge regression after standardizing the predictors [7]. Both methods shrink the coefficient estimates towards zero, however, in Lasso, the L1 penalty forces some coefficient estimates to be zero when the tuning parameter λ is large enough.

Apart from regular grid-search fine-tuning for optimal configuration of the regularization parameters, I would also run a random forest regressor and use the built-in out-of-bag scoring feature for random forest regressor as a manner of applying bootstrapping to the model [6]

Other models one could try are mixed effect models where there would be some random components to capture external factors that are not possible to observe or quantify,

and account for seasonality effects, holiday "playing" times, holiday periods when users are more receptive to play games, outside school time/exams time fluctuations, etc.

**- How to measure the accuracy and validate the robustness of a model?**

Cross validation is the general assessment model procedure. Depending on the model one or other metric will be looked at.

1. In linear regression models normally one computes the coefficient of determination R^2, but also the adjusted R^2, error term, the statistical significance of each feature and Mallow’s Cp criterion can be used. The latter compares precision and bias of the full model to models with a subset of predictors. R^2 provides a way to quantitatively measure how well the model explains the data [11]. To compare the ground truth target values with predicted target values by the random forest regressor, scikit-learn doesn't use the MSE but R2 or coefficient of determination #r2\_score() computes the coefficient of determination R2, whose best possible score is 1.0, and lower values are worse.

R-squared is the “percent of variance explained” by the model [8] or fraction by which the variance of the errors is less than the variance of the dependent variable. Therefore, from all models tried with different regularization methods, I would keep the one producing best (closer to 1) coefficient of determination R^2 [4].

Other criteria for measuring the relative quality of statistical models are Akaike Information criterion (AIC) (assessed by likelihood function) and the Bayesian Information Criterion (AIC) or Schwarz criterion (SBIC), which is used for model selection among a set of them (lower BIC is preferred).

[1] Breiman, L. Better Subset Regression Using the Nonnegative Garrote Technometrics, 1995, 37, 373-384

[2] Yuan, M. & Lin, Y. On the non-negative garrotte estimator Journal of the Royal Statistical Society (Series B), 2007, 69, 143-161

[3] Zou, H. & Hastie, T. Regularization and variable selection via the elastic net Journal of the Royal Statistical Society (Series B), 2005, 67, 301-320

[4] http://stats.stackexchange.com/questions/866/when-should-i-use-lasso-vs-ridge

[5] Breiman, Leo. Out-of-bag estimation. Technical report, Statistics Department, University of California Berkeley, 1996b. 33, 34.

[6] OOB estimation http://scikit-learn.org/stable/auto\_examples/ensemble/plot\_ensemble\_oob.html

[7] https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/model\_selection.pdf

[8] RandomForestRegressor score() returns the coefficient of determination R^2 of the prediction, defined as (1 - u/v), where u is the regression sum of squares ((y\_true - y\_pred) \*\* 2).sum() and v is the residual sum of squares ((y\_true - y\_true.mean()) \*\* 2).sum(). Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

[9] http://www.statsoft.com/textbook/glosp.html#Pearson%20Correlation

[10] Pearson's ρ does not assume normality, but is only an exhaustive measure of association if the joint distribution is multivariate normal. Given the failed normality test in our case, another option can be using Spearman’s s or Kendall’s tau. Spearman’s rho can be less reliable and interpretable than Kendall’s while Spearman rho is easier to compute) [Kendall & Gibbons, 1990].

[11] R^2 is measurable using scoring function 'r2' as follows:

# LINEAR REGRESSION

lr = LinearRegression(fit\_intercept=True, normalize=False, copy\_X=True, n\_jobs=-1)

score\_lr = cross\_val\_score(lr, x, y, cv=5, scoring='r2')

alpha\_vals = [0.1, 0.3, 1.0] # lasso, ridge (alpha in [0,1])

# LINEAR REGRESSION LASSO Regularization

la = Lasso(alpha=a, fit\_intercept=True, normalize=False, precompute=False, copy\_X=True, max\_iter=1000, tol=0.0001, warm\_start=False, positive=False, random\_state=None, selection='cyclic')

score\_la = cross\_val\_score(la, train, y, cv=5, scoring='r2')

# LINEAR REGRESSION RIDGE Regression

ri = Ridge(alpha=a, fit\_intercept=True, normalize=False, copy\_X=True, max\_iter=None, tol=0.001, solver='auto', random\_state=None)

score\_ri = cross\_val\_score(ri, train\_all, y, cv=5, scoring='r2')

[12] http://www.unesco.org/webworld/idams/advguide/Chapt4\_2.htm

2. To measure the accuracy and validate the robustness of the model, some goodness-of-fit tests are:

-Chi-square test for continuous and discrete distributions;

-Kolmogorov-Smirnov test for continuous distributions based on the empirical distribution function (EDF);

-Anderson-Darling test for continuous distributions: http://www.oswego.edu/~srp/stats/npp\_small.htm

Specially the latter, is well suited when the sample size is small, and a histogram does not provide very accurate insights regarding normality (see histograms in notebook).

[1] The Standard Normal curve, shown here, has mean 0 and standard deviation 1. If a dataset follows a normal distribution, then about 68% of the observations will fall within of the mean , which in this case is with the interval (-1,1). About 95% of the observations will fall within 2 standard deviations of the mean, which is the interval (-2,2) for the standard normal, and about 99.7% of the observations will fall within 3 standard deviations of the mean, which corresponds to the interval (-3,3) in this case. Although it may appear as if a normal distribution does not include any values beyond a certain interval, the density is actually positive for all values, . Data from any normal distribution may be transformed into data following the standard normal distribution by subtracting the mean and dividing by the standard deviation. http://www.stat.yale.edu/Courses/1997-98/101/normal.htm

[2] https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.mstats.normaltest.html

[3] Goodness-of-fit techniques D'Agostino & Stephens (1986).

[4] R2 is a statistic that will give some information about the goodness of fit of a model. In regression, the R2 coefficient of determination is a statistical measure of how well the regression line approximates the real data points. An R2 of 1 indicates that the regression line perfectly fits the data. Values of R2 outside the range 0 to 1 can occur where it is used to measure the agreement between observed and modeled values and where the "modeled" values are not obtained by linear regression and depending on which formulation of R2 is used.

[5] Normality tests: http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R4\_One-TwoSampleTests-ANOVA/R4\_One-TwoSampleTests-ANOVA2.html

*Other methods for model accuracy improvement*

In order to improve the accuracy of the model, I would use some of the common feature selection strategies below to identify and remove unneeded irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model:

a) Filter Methods to score-ranking each feature: Chi squared test, information gain and correlation coefficient scores.

b) Wrapper Methods that select a set of features as a search problem, to find the best set of features with e.g. stochastic methods such as random hill-climbing algorithm, for instance, the recursive feature elimination algorithm.

c) Embedded Methods (into the model), which learn the contribution of each attribute within the model fitting phase, e.g. regularization methods.

d) Perform feature analysis for feature significance and feature relevance using University of Washington Lime package

More concretely in Python scikit-learn, I would use

-Recursive Feature Elimination (obtains support and ranking for each feature via rfe.support\_ and rfe.ranking\_ where RFE is a sklearn.features\_selection RFE class and model is a (logistic) regression model)

-Feature importance ranking (model.feature\_importances\_ where model is a ExtraTreesClassifier object rfe = RFE(model, n\_features)).

After applying correlation coefficient score and removing features highly correlated, we can improve the linear regressor by including extra features by multiplying some of the most predictive ones and creating non-linear features in this way that can improve the predictive power of the model. Other strategies to try are removing correlated features among themselves or less predictive, visualising the data to find the right model or apply PCA to remove co-linearity.