Université Paris XIII – Ecole d'Ingénieurs
SupGalilée

Mathématiques Appliquées et Calcul Scientifique

Product Analytics – Global Nielsen Approach

Author : ELISTRATOV Danila, MACS3

Supervisor : RUZANOV Behzod

INTRODUCTION

AC Nielsen is a globally present corporation which is a major player in a great number of markets that are related to collection of statistics and its further processing. Fields of activity include collecting a great amount of statistical data, selling its representation and analysis to clients and performing tasks related to prognosis. AC Nielsen has remained one of the key players in this industrial sector for several decades and keeps its strategy of expanding and conquering new markets throughout all the continents.

Due to its size and complexity of its inner structure it is impossible to describe it in full volume or make a visual representation. In this article only organigram close to the Department of Statistical Operations will be presented.

Due to the non-disclosure agreement there will be no real figures or inner procedures presented, only descriptive examples giving a general idea about how Nielsen Analytics work.

Chapter 1. Nielsen in Kazakhstan.

Just like any other country audited by AC Nielsen, Kazakhstan includes a complex structure of territorial and categorical segmentation. The territory consists of 6 major zones (Center, West, East, South, North, Almaty). All together they include 21 big city and over 100 minor settlements, many of which also have sub-segmentations. There is also difference between Urban and Rural zones (which are defined by their Population Groups).

Here is a graph that in general represents the principle of territorial division and sub-division:

As well as other countries Kazakhstan requires the same set of monthly, weekly or bi-monthly procedures in order sustain statistics and prepare it for selling to clients. The approximate algorithm is presented in Chapter 2.

Kazakhstan is classified as a “developing country” in the terms of Nielsen ranging system.

Chapter 2. Procedures.

Data bases are continuously sold to clients (most of them every month), so Statistical Operations Department has a number of repetitive procedures that imply verification, smoothing and reallocation of data. A set of procedures for every Index consist of roughly 30 steps (these are referred to as BAU – Business As Usual procedures). Most common statistical methods used in this analysis are confidence intervals, smoothing of trends, sample inspection and correction of automated procedures that might make errors while estimating and allocating data.

AC Nielsen has a vast range of software utilities that are either used in inner production or sold to clients as a means to scrutinize audited data.

Here is an example of SCS (Sample Clients Service) which is used primarily in Statistical Department procedures:

(KZA - AUG/AUG17MNTL31)

ll	D	NC	BO	EL	DS	OUTLET	NAME	VIS. DATE	REGION	TYPE	C06	CHAIN	P03	T03	TRD	TURNOVER2
	☐	☐	☐	☐	☑	010013	Azhar	02/08/2017	CEN	OXM	01	NON	7	7	TT	32563.20
	☐	☐	☐	☐	☑	010039	Kiosk	24/08/2017	CEN	PKS	01	NON	7	7	TT	5074.31
	☐	☐	☐	☐	☑	010049	Kiosk	17/08/2017	CEN	PKS	01	NON	8	8	TT	3788.42
	☐	☐	☐	☐	☑	010052	Bars	16/08/2017	CEN	OSM	01	NON	8	8	TT	9634.86
	☐	☐	☐	☐	☑	010058	Airtaulyk	17/08/2017	NOT	OSM	14	NON	8	8	TT	21633.97
	☐	☐	☐	☐	☑	010062	Gorodskoy Tsentralny	11/08/2017	CEN	RMF	01	NON	7	7	TT	5.56
	☐	☐	☐	☐	☑	010067	Ryabinushka	17/08/2017	CEN	OXS	01	NON	8	8	TT	9544.69
	☐	☐	☐	☐	☑	010069	Kazakhstan	07/08/2017	CEN	OSM	01	NON	7	7	TT	19093.80
	☐	☐	☐	☐	☑	010071	Kiosk	21/08/2017	CEN	PKS	01	NON	7	7	TT	9048.18
	☐	☐	☐	☐	☑	010072	Kiosk	24/08/2017	CEN	PKS	01	NON	7	7	TT	5058.07
	☐	☐	☐	☐	☑	010076	San	02/08/2017	CEN	OXM	01	NON	7	7	MT	59145.49
	☐	☐	☐	☐	☑	010086	Asan	21/08/2017	CEN	OME	01	NON	7	7	TT	14300.73
	☐	☐	☐	☐	☑	010087	Bars	08/08/2017	CEN	OSM	01	NON	7	7	TT	12724.46
	☐	☐	☐	☐	☑	010092	Impuls	08/08/2017	CEN	OXS	01	NON	7	7	TT	14739.83
	☐	☐	☐	☐	☑	010094	Gurman	11/08/2017	CEN	OXS	01	NON	8	8	TT	27385.69
	☐	☐	☐	☐	☑	010098	Magazin	11/08/2017	CEN	OSM	01	NON	8	8	TT	7926.65
	☐	☐	☐	☐	☑	010107	Zhannet	02/08/2017	CEN	RMF	01	NON	7	7	TT	103.32
	☐	☐	☐	☐	☑	010110	Olzha	16/08/2017	CEN	OXM	01	NON	8	8	TT	7942.44
	☐	☐	☐	☐	☑	010111	Pavilion	07/08/2017	CEN	OPF	01	NON	7	7	TT	18492.74
	☐	☐	☐	☐	☑	010112	Kamila	12/08/2017	CEN	OME	01	NON	7	7	TT	37498.53
	☐	☐	☐	☐	☑	010113	Orion	10/08/2017	NOT	OSM	11	NON	8	8	TT	15611.75
	☐	☐	☐	☐	☑	010114	Aynura	10/08/2017	NOT	OSM	11	NON	8	8	TT	10426.51
	☐	☐	☐	☐	☑	010115	Olvi2	24/08/2017	NOT	OSM	11	NON	8	8	TT	32448.58

This statistical software allows to stock data and apply any corrections approved by StatOps Department. It also has efficient utilities that allow to download different templates of data under FoxPro or Excel

format. It undergoes multiple updates throughout the production process.

Apart from BAU (which take about 2 weeks every month) there are PE (Product Enhancement) procedures. These are designed to upgrade BAU procedures and maybe imply new principles of data processing. It usually takes place once in half a year.

These are repetitive procedures that are connected to the continuous production. There are also projects that are treated separately from every month routine. They are usually projects ordered by specific clients to make research on a specific trading unit. For example, Coca-Cola who is interested in whether it was a nice idea to start selling 0,33 Cola cans in rural areas of western Kazakhstan. These projects may differ in its parameters and even principles, but very often they are brought to a standard t-test procedure that evaluates binary alterations in a statistical row (sometimes the technique may differ – either Chow test for a trend break or logistic regressions).

In most cases these verifications are performed with the same data sets as BAU procedures but sometimes it is necessary to require optional data sets for more detailed tests.

The last part of repetitive procedures is sustaining connection to the so-called Field Department. It consists of a vast number of employees who directly address the trading points and acquire data.

Due to the constant rotation of our panel (described in details in Chapter 4), Field Department needs support from StatOps department giving them directions on which types and geographical locations are to be included in the panel so that the general picture seen by the client does not get distorted.

Chapter 3. Organigram.

Obviously, full organigram of AC Nielsen is practically impossible to present due to its enormous size and quite an intricate department structure. Here only a few words will be said about departments that contact closely StatOps Department.

First of all, the above-mentioned Field Department. This is one of the most densely filled department which has hundreds of people searching for panel stores in all the countries of the world.

It is important (and of this consists one of the objectives of StatOps) to support the Field by continuously providing them with characteristics for shops that are needed for the panel in the current period (see Image 2. Optimal and Actual Panels, Chapter 4).

Shops themselves can also undergo changes (changing their size, type, region, name, trading range, etc.) which also needs to be marked in data bases.

Apart from Field Department there is also CIPO department which communicates directly to the clients' representatives. Employees of StatOps do not get the orders from the clients – they communicate with CIPO, which deals with putting in place the VPS system (see Image 3. VPS in Chapter 5). They try to grasp the situation from the client's perspective, form questions and requirements related to data and send these questions to StatOps team.

Data Science Department is also worth mentioning as they continuously work on the development and substitution of methods that lie in the basis of Product Enhancement procedures. It is up to them to define the method of smoothing for Universe Update Procedure, or reconsider the principle of territorial sub-division, or carry out more complex particular projects.

There is also MEFF (Market Effectiveness) Department. Their job consists mainly of preparing the presentations, re-processing clients'

non-repetitive orders and selling statistical results acquired by StatOps team. Sometimes they go to meetings, sometimes they just sell the results supported by reports and presentations, sometimes they form complementary list of questions and remarks that is later treated by StatOps and Data Science teams.

Chapter 4. Universes and Panels.

AC Nielsen processes a huge amount of numerical and qualitative data in order to prepare data bases that are sold to clients. One of the main objectives is to have such a numerical panel that would be capable of representing a real territory possessing a certain amount of shops.

Obviously, it is impossible to collect data from all the shops existing throughout audited countries. That is why the technique of extrapolation is used – for each territorial sector (or anyhow else defined segmentation) the optimal panels are calculated. The procedure is based on predefined values of standard error that must not be overstepped. This process is carried out by Data Science Department along with Design Center Department and StatOps Department.

Actual panel is counted in thousands and the process of its approximation to the optimal panel is still ongoing which represents an important part of StatOps role in Nielsen.

Here is a fragment of data representing the segmentation and its optimal panels.

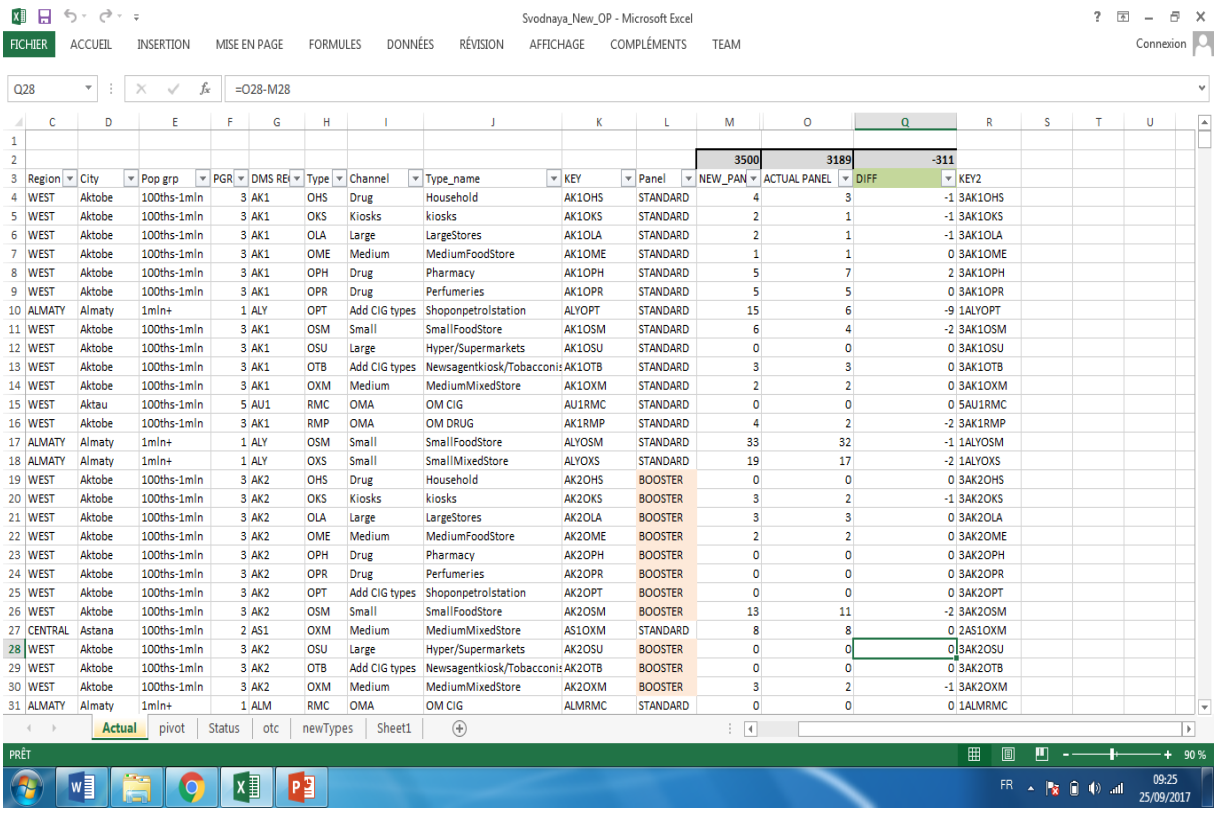


Image 2. Optimal and actual panels

Major complexities include constant change of the universe (bi-monthly procedure of universe update), monthly rotation of the panel due to the changes in shops' characteristics and disproportional changes in Z- and X-factors.

Thorough analysis in order to conserve the structure of the panel is necessary due to clients' requirement to see the data in a vast number of sections or representations (all the changes the clients see must be proportional to the real changes in shops' data and must not be a consequence of inner production structure alterations.

Chapter 5. Baskets and Categories

All goods traded in panel shops throughout the country have their p-codes which allows to analyze statistics more efficiently. The territory of Kazakhstan is represented by roughly 110 categories which are stocked in 6 major indexes, each defined by a corresponding Basket (allocation of categories in indexes). In Kazakhstan Indexes are: 'Monthly-Food', 'Bi-monthly Food', 'Monthly Drug&Utilities', 'Monthly Beer', 'Monthly Cigarettes', 'Bi-monthly Drug&Utilities'.

The definition of Indexes depends, first of all, on clients. Some of them (like cigarette brands) require more thorough analysis and that's why they require their personal Index with a set of specific procedures.

Each of them requires BAU-updates and PE-treatment so usually indexes are allocated to the members of StatOps team (2-3 for a person).

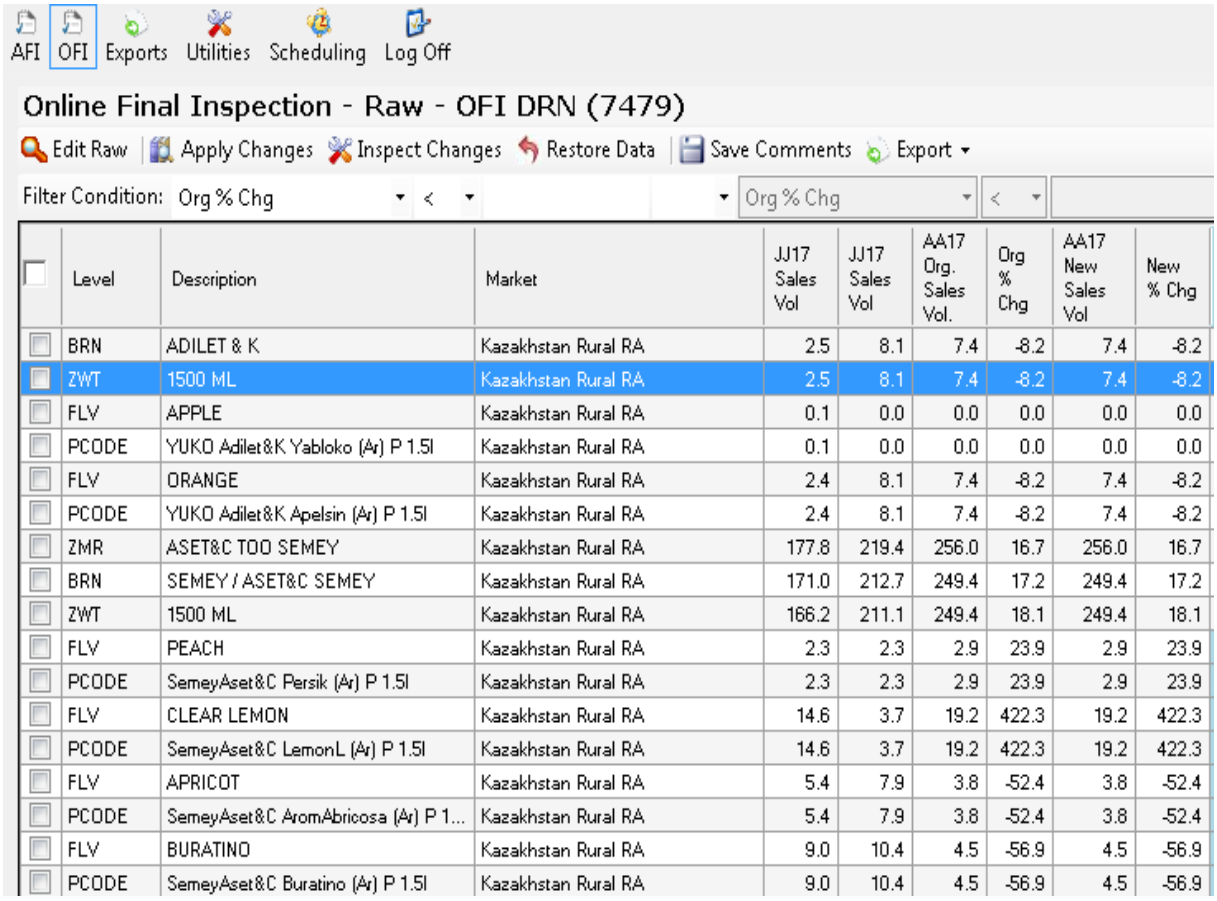
Smoothing is applied on the level of each category and it consists of two parts: correction of automated procedure and building confidence intervals.

Baskets and categories are, first of all, designed for the clients. It enables more detailed analysis on the level of p-codes (personal codes of unique goods).

Software which is sold to the clients allows to see the dynamics on every level of representation.

It is called VPS and here is an abstract of its environment:

Image 3. VPS



Level	Description	Market	JJ17 Sales Vol	JJ17 Sales Vol	AA17 Org. Sales Vol.	Org % Chg	AA17 New Sales Vol	New % Chg
BRN	ADILET & K	Kazakhstan Rural RA	2.5	8.1	7.4	-8.2	7.4	-8.2
ZWT	1500 ML	Kazakhstan Rural RA	2.5	8.1	7.4	-8.2	7.4	-8.2
FLV	APPLE	Kazakhstan Rural RA	0.1	0.0	0.0	0.0	0.0	0.0
PCODE	YUKO Adilet&K Yabloko (Ar) P 1.5l	Kazakhstan Rural RA	0.1	0.0	0.0	0.0	0.0	0.0
FLV	ORANGE	Kazakhstan Rural RA	2.4	8.1	7.4	-8.2	7.4	-8.2
PCODE	YUKO Adilet&K Apelsin (Ar) P 1.5l	Kazakhstan Rural RA	2.4	8.1	7.4	-8.2	7.4	-8.2
ZMR	ASET&C TOO SEMEY	Kazakhstan Rural RA	177.8	219.4	256.0	16.7	256.0	16.7
BRN	SEMEY / ASET&C SEMEY	Kazakhstan Rural RA	171.0	212.7	249.4	17.2	249.4	17.2
ZWT	1500 ML	Kazakhstan Rural RA	166.2	211.1	249.4	18.1	249.4	18.1
FLV	PEACH	Kazakhstan Rural RA	2.3	2.3	2.9	23.9	2.9	23.9
PCODE	SemeyAset&C Persik (Ar) P 1.5l	Kazakhstan Rural RA	2.3	2.3	2.9	23.9	2.9	23.9
FLV	CLEAR LEMON	Kazakhstan Rural RA	14.6	3.7	19.2	422.3	19.2	422.3
PCODE	SemeyAset&C LemonL (Ar) P 1.5l	Kazakhstan Rural RA	14.6	3.7	19.2	422.3	19.2	422.3
FLV	APRICOT	Kazakhstan Rural RA	5.4	7.9	3.8	-52.4	3.8	-52.4
PCODE	SemeyAset&C AromAbricosa (Ar) P 1...	Kazakhstan Rural RA	5.4	7.9	3.8	-52.4	3.8	-52.4
FLV	BURATINO	Kazakhstan Rural RA	9.0	10.4	4.5	-56.9	4.5	-56.9
PCODE	SemeyAset&C Buratino (Ar) P 1.5l	Kazakhstan Rural RA	9.0	10.4	4.5	-56.9	4.5	-56.9

Chapter 6. Programming skills

Software procedure in StatOps are divided into three major groups: Excel & VBA, FoxPro (SQL) and Nielsen Software (like SCS or VPS).

Third group has a highly-effective link with Microsoft-adjusted programs – SCS and VPS have a powerful and divers tool for importing and exporting many templates of Excel files. If, however, analysis requires more complex form of processing file then queries might be defined through SQL environment (FoxPro).

Here is the most primitive example of what SQL queries in FoxPro environment might look like.

```
SELECT * from Om_kzk_268 WHERE region='CEN'
```

Outlet	Name	Region
010013	Azhar	CEN
010052	Bars	CEN
010067	Ryabinushka	CEN
010069	Kazakhstan	CEN
010076	San	CEN
010083	Apteka TOO Alfiya	CEN
010086	Asan	CEN
010087	Bars	CEN
010092	Impuls	CEN
010094	Gurman	CEN
010098	Magazin	CEN
010100	Tsum	CEN
010101	Gorodskoy rynok	CEN
010108	Zhannet	CEN
010110	Olzha	CEN
010112	Kamila	CEN
010117	Bars	CEN
010118	Vita Farm	CEN
010121	Prigorod	CEN
010127	IMIDZH	CEN
010132	GALANT	CEN
010133	GORODSKOY TSENTRALNY	CEN
010135	AKSORAN	CEN
010138	MAGAZIN	CEN
010140	ARMAN	CEN
090019	Parfyumerny v magazina Teremok	CEN
090031	Diyan Apteka	CEN

Statistical Methods

Due to the disclosure agreement it is not possible to reveal methods of calculus that lie in the basis of BAU or PE processes but it is possible to describe the particular projects ordered by particular clients.

Let's consider a project: Coca-Cola wants to know whether selling 0,33 cl cans (which has been recently implemented in, let's say, 10 biggest cities of Kazakhstan) is profitable. If they are sold well, don't they take some market share of 0,5 cans, and if they do, is it profitable from the point of view of general profit?

The most often tool used in such queries is logistic regression (logit or probit). But sometimes it is more appropriate to use linear regression in order to, for example, make a prediction of a number of cans sold in a shop with a specific number of features.

Let's remind that the general regression model is given by:

$$Y_t = f(\alpha + \sum_{i=1}^{n} \beta_i X_i)$$

In the linear regression case

$$f(x) = x$$

So

$$Y_t = \alpha + \sum_{i=1}^{n} \beta_i X_i$$

In the binary logit case:

$$P(Y_t = 1) = \frac{1}{1 - e^{-(\alpha + \sum_{i=1}^{n} \beta_i X_i)}}$$

So

$$P(Y_t = 0) = 1 - \frac{1}{1 - e^{-(\alpha + \sum_{i=1}^{n} \beta_i X_i)}}$$

In the binary probit case:

$$P(Y_t = 1) = N(\alpha + \sum_{i=1}^{n} \beta_i X_i)$$

So

$$P(Y_t = 0) = 1 - N(\alpha + \sum_{i=1}^{n} \beta_i X_i)$$

As X_i might be chosen population of the city in question, how far it is from the region center, market shares of competitive products. Very often the so-called Herfindahl-Hirschman coefficient is taken as a regressor (not its initial form from economic theory, but adapted version for product analytics). Smaller its value is, more competitive

the market is. So its adaptation for product analytics shows how diversified the product basket of a trading point is, and the hypothesis is that smaller it is, more likely it is that the new product (for example, 0,33 cl can) will be able to grasp its market share.

Goal is to find as many regressors as possible in order to maximize the quality of the model.

Here is an example of Python implementation of a binary regression.

```
In [27]: # logistic regression for volat

         model=linear_model.LogisticRegression()
         logreg=model.fit(volat.reshape(-1,1),Regressant)

         predict_volat = logreg.predict(volat.reshape(-1,1))
         predict_prob_volat = logreg.predict_proba(volat.reshape(-1,1))
         pred_volat=predict_prob_volat[:,1]

         print(accuracy_score(Regressant, predict_volat))
         #print(predict_risk)
```

```
0.738461538462
```

Python libraries give an opportunity to apply L^1 and L^2 relaxation for ridge and lasso methods in linear regressions.

```
In [218]: # logistic regression for default prediction

          model=linear_model.LogisticRegression(penalty="l2")
          logreg=model.fit(Regressors,Regressant)

          #print(logreg.coef_)
          #print(logreg.intercept_)
```

L^1-relaxation of Lasso regression follows the same logic as OLS (Ordinary Least Square) in ordinary regression but it adds a constraint to the minimization problem. The model has the following form:

$$\begin{cases} min_{\alpha,\beta}\{\frac{1}{N}||Y-\alpha-X\beta||_2^2\} \\ ||\beta||_1 \leq c \end{cases}$$

In this case 1-norm of the vector of coefficients means $\sum_{i=1}^{n}|\beta_i|$

L^2-relaxation poses pretty much the same problem but with a different condition:

$$\begin{cases} min_{\alpha,\beta}\{\frac{1}{N}\left|\left|Y-\alpha-X\beta\right|\right|_2^2\} \\ ||\beta||_2 \leq c \end{cases}$$

In this case 2-norm of the vector of coefficients means $\sum_{i=1}^{n}\beta_i^2$

L^1 /L^2-relaxation techniques are applicable to ridge regressions. Ridge regression model also tends to decrease the sum of coefficients but in a different form (case below for L^1-relaxation) :

$$min_{\alpha,\beta}\{\frac{1}{N}\left|\left|Y-\alpha-X\beta\right|\right|_2^2 + \delta||\beta||_1\}$$

Case of L^1-relaxation:

$$min_{\alpha,\beta}\{\frac{1}{N}\left|\left|Y-\alpha-X\beta\right|\right|_2^2 + \delta||\beta||_2\}$$

These techniques should be used in case of a large number of regressors.

It is important to verify whether all the conditions of Gauss-Markov theorem are verified. Python provides sufficient tools to provide tests for 4 major conditions.

First of all, multicollinearity: the dependence between regressors which inflates the variance and overestimates individual dependence between regressant and regressor.

First procedure implies checking the correlation matrix of regressors (Python). Usually threshold value is chosen 0,85 (or 0,9 if the regressor is known to be important historically). Those rejected after this are checked in the individual regression and in are left in the model in exceptional cases (if R^2 of the individual model is too high or P-value of H_0 is too low).

```
In [6]: a=Regressors.corr()
        print(a[(a>0.9)])
```

Second procedure related to multicollinearity is calculation of Variance Inflation Factor (VIF). VIF is calculated for each regressor.

$$VIF_i = \frac{1}{1 - R_i^2}$$

where R_i^2 is the coefficient of determination of the i-th factor projected on all other factors. If it is sufficiently described by other factors, there is no need to include it in the general model. Usually the threshold value of VIF is chosen 5.

Unfortunately, there is no specified tool in Python to calculate it directly, so here is an implemented script:

```
In [208]: # introducing vector of R-square and vector of VIF
          R=np.zeros(N)
          VIF=np.zeros(N)
          SelectedColumns=[]

          # choosing regressors with small VIF
          for col, i in zip(Regressors.columns, range(Regressors.shape[1])):

              Y = Regressors[col]
              XX = Regressors.drop(col, axis=1)

              model = linear_model.LinearRegression()
              model.fit(XX,Y)
              R[i]=model.score(XX,Y)
              VIF[i]=1./(1-R[i]*R[i])

              if VIF[i]<5:
                  SelectedColumns.append(col)

          #print(SelectedColumns)
          #print(R)
          #print(VIF)

          SelectedRegressors=Regressors.loc[:, SelectedColumns]
          #print(SelectedRegressors)
```

Second condition is that the mean value of residuals is zero. It is easy to verify this statement with confidence interval.

Third condition of Gauss-Markov to verify is homoscedasticity of residuals of the model (which means their constant variance). If this condition is not met than GARCH-type models are applied (but during this job I didn't have an opportunity to apply them).

Homoscedasticity of residuals can be verified with a vast number of tests. The most often used is Glejser test.

It doesn't have a direct implementation in Python libraries but it is quiet easy to write a script. This test implies modelling three regressions:

$$|e| = \alpha + \sum_{i=1}^{n} \beta_i X_i + \varepsilon$$

$$|e| = \alpha + \sum_{i=1}^{n} \beta_i \sqrt{X_i} + \varepsilon$$

$$|e| = \alpha + \sum_{i=1}^{n} \beta_i \frac{1}{X_i} + \varepsilon$$

$|e|$ are the absolute values of the residuals. The model with the highest R^2 is chosen and if there are highly significant β_i for a number of factors, the hypothesis of homoscedasticity is rejected.

A different variation of Glejser test is Breusch-Pagan test.

The test is following:

$$e^2 = \alpha + \sum_{i=1}^{n} \beta_i X_i + \varepsilon$$

Breusch-Pagan statistics is:

$$BP = N\, R^2$$

It is distributed as Chi^2_{n-1} and *N* is the number of observations, *n* is the number of regressors in the model. So if *BP* is sufficiently big, the hypothesis of homoscedasticity is rejected.

In Python there is a method for White test which can be called directly.

It is quite possible to combine the two tests (thus, *White test*). It means to model a regression where regressant is e^2 and regressors are $\sqrt{X_i}$ or $\frac{1}{X_i}$ (or any other functional form of X_i). The highest R^2 undergoes BP verification with Chi^2_{n-1}.

Final condition to verify is the absence of autocorrelation of residuals.

Absence of autocorrelation of residuals means:

$$cov(\varepsilon_i, \varepsilon_j) = 0$$

If this condition is not met, the consequences for the parameters' estimates are the same as in case of heteroscedasticity of residuals: they will remain unbiased and consistent but the property of efficiency (has the smallest variance in the class of unbiased linear estimates) is lost.

The most common tool to check for residuals' autocorrelation is Durbin-Watson test. It calculates as follows:

$$DW = \frac{\sum_{i=2}^{N}(e_i - e_{i-1})^2}{\sum_{i=2}^{N} e_i^2}$$

This quantity is has a linear dependence of sample autocorrelation of the residuals. Both are easily calculated with Python.

In the best case scenario *DW* equals 2, in worst case 0 or 4. This statistics is *DW*-distributed and P-value quantiles are defined with the help of *DW*-distribution.

If all conditions of Gauss-Markov are met then the estimates obtained by the linear regression are unbiased, consistent and efficient. Then they can be used for prediction.

The quality of the linear regression can be measured in different ways but the most often used in Nielsen is *MSE* which is $||e||_2^2$.

Threshold values might be chosen differently depending on the regressors and regressant.

It is more complicated for binary regressions – there is a number of metrics and curves that estimate the quality of prediction of

$P(Y_t = 1)$ in the model.

Most of them are used to compare two binary ranges (one – the real values and the other one – the predicted values).

All the results presented below are not derived from real data but they do give an idea about how these estimations look like.

The most often tool is ROC-curve which represents a curve above the diagonal in the axis *tpr – fpr.*

$$tpr\ (\text{True Positive Ratio}) = \frac{TP}{TP + FP}$$

$$fpr\ (\text{False Positive Ratio}) = \frac{FP}{TP + FP}$$

TP is the number of ones guessed right by the model, *FP* is the number of real 0 that were classified as ones by the model.

This tool is implemented in Python:

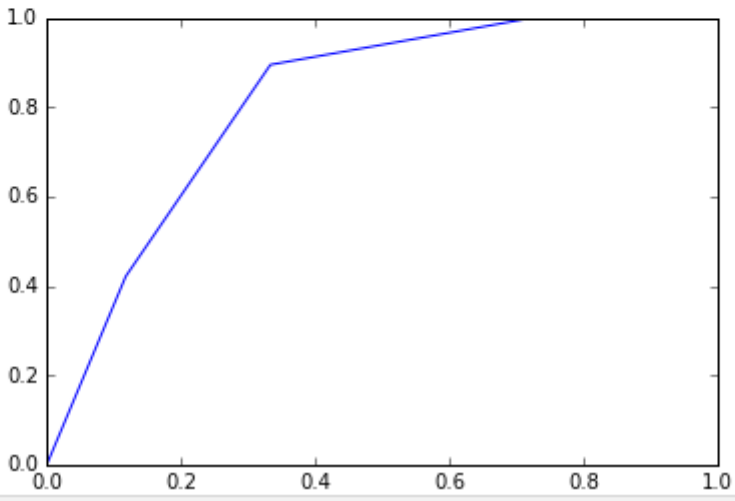
```
fpr_risk,tpr_risk,thresh_risk=roc_curve(Regressant,pred_risk)
fpr_risk

fpr_transpar,tpr_transpar,thresh_transpar=roc_curve(Regressant,pred_transpar)
fpr_transpar

fpr_market,tpr_market,thresh_market=roc_curve(Regressant,pred_market)
fpr_market

fpr_geogr,tpr_geogr,thresh_geogr=roc_curve(Regressant,pred_geogr)
fpr_geogr
```

```
In [311]: plt.plot(fpr_eqass,tpr_eqass)
          plt.plot(fpr_fcf,tpr_fcf)
          plt.plot(fpr_manag,tpr_manag)
          plt.plot(fpr_risk,tpr_risk)
```

Here's an example of a ROC-curve built on a small partition of real data (no smoothing).

```
#plt.plot(fpr_N3,tpr_N3)
#plt.plot(fpr_DIVERS,tpr_DIVERS)
#plt.plot(fpr_ROE,tpr_ROE)
#plt.plot(fpr_ROA,tpr_ROA)
#plt.plot(fpr_NIM,tpr_NIM)
#plt.plot(fpr_GOVSTRUCT,tpr_GOVSTRUCT)
#plt.plot(fpr_GOVQUAL,tpr_GOVQUAL)
#plt.plot(fpr_RISK,tpr_DEVELOP)
plt.plot(fpr_MARKET,tpr_MARKET)
#plt.plot(fpr_GEOGR,tpr_GEOGR)
#plt.plot(fpr_BUSINDIVERS,tpr_BUSINDIVERS)
```

```
Out[29]: [<matplotlib.lines.Line2D at 0xa848940>]
```



In this code ROC-curve is built for every regressor of one of the binary models designed to predict the probability of a good to be successfully implemented in market.

Here is a code sample that calculates the AUROC-score which is the area under ROC-curve.

```
In [30]: ################################################################
         #roc_auc_score(Regressant,pred_rating)
         ################################################################

         #roc_auc_score(Regressant,pred_n1)
         #roc_auc_score(Regressant,pred_POSTDUE)
         #roc_auc_score(Regressant,pred_COVERAGE)
         #roc_auc_score(Regressant,pred_N3)
         #roc_auc_score(Regressant,pred_DIVERS)
         #roc_auc_score(Regressant,pred_ROE)
         #roc_auc_score(Regressant,pred_ROA)
         #roc_auc_score(Regressant,pred_NIM)
         #roc_auc_score(Regressant,pred_GOVSTRUCT)
         #roc_auc_score(Regressant,pred_GOVQUAL)
         #roc_auc_score(Regressant,pred_RISK)
         #roc_auc_score(Regressant,pred_DEVELOP)
         roc_auc_score(Regressant,pred_MARKET)
         #roc_auc_score(Regressant,pred_GEOGR)
         #roc_auc_score(Regressant,pred_BUSINDIVERS)

Out[30]: 0.81269349845201244
```

Normally, ROC-score threshold is 0,7 which implies that the individual regression for regressor ‘MARKET’ (code above) is good enough.

Another tool to estimate the quality of logistic regression is Kolmogorov-Smirnov test. KS statistics is designed in more broad sense to verify if two series have the same distribution but in this case it is used to estimate proximity between two binary series.

Here is an example of code:

```
# Kolmogorov-Smirnov test for 2 distributions

#ks_2samp(Regressant,predict_n1)
#ks_2samp(Regressant,predict_POSTDUE)
#ks_2samp(Regressant,predict_COVERAGE)
#ks_2samp(Regressant,predict_N3)
#ks_2samp(Regressant,predict_DIVERS)
#ks_2samp(Regressant,predict_ROE)
#ks_2samp(Regressant,predict_ROA)
#ks_2samp(Regressant,predict_NIM)
#ks_2samp(Regressant,predict_GOVSTRUCT)
#ks_2samp(Regressant,predict_GOVQUAL)
#ks_2samp(Regressant,predict_RISK)
#ks_2samp(Regressant,predict_DEVELOP)
ks_2samp(Regressant,predict_MARKET)
#ks_2samp(Regressant,predict_GEOGR)
#ks_2samp(Regressant,predict_BUSINDIVERS)
```

```
Ks_2sampResult(statistic=0.071428571428571508, pvalue=0.99208417867962795)
```

P-Value is calculated for H_0 that two series have the same distribution. Which makes the individual model shown in the code of high quality.

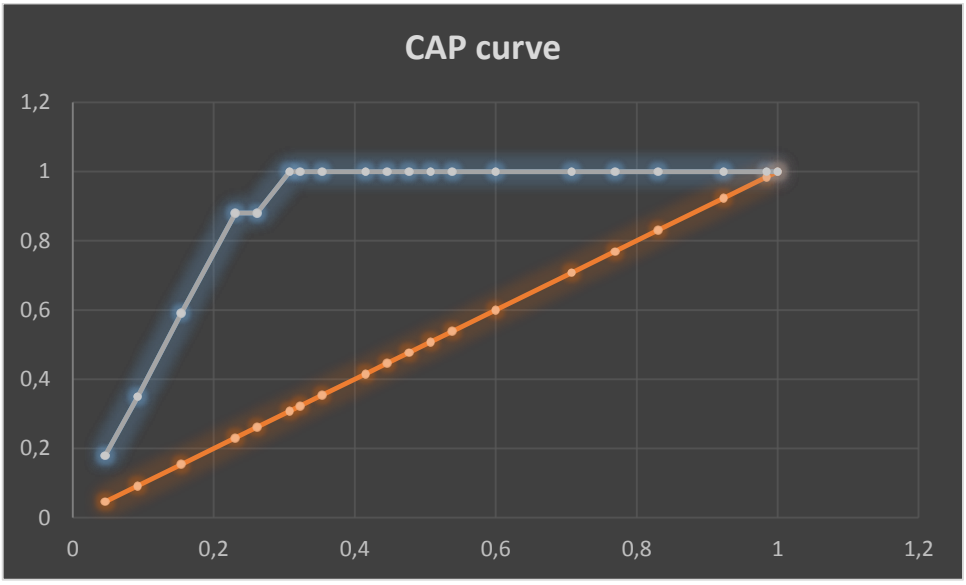
Python package allows to have direct access to the predictions of the model and AR (Accuracy Ratio):

In [33]:
```
model=linear_model.LogisticRegression()
logreg=model.fit(BUSINDIVERS.reshape(-1,1),Regressant)

predict_BUSINDIVERS = logreg.predict(BUSINDIVERS.reshape(-1,1))
predict_prob_BUSINDIVERS = logreg.predict_proba(BUSINDIVERS.reshape(-1,1))
pred_BUSINDIVERS=predict_prob_BUSINDIVERS[:,1]

print(accuracy_score(Regressant, predict_BUSINDIVERS))
#print(pred_eqassets)
print(predict_BUSINDIVERS)
#print(logreg.coef_)
#print(logreg.intercept_)
```

```
0.771428571429
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0]
```

Another tool is CAP-curve. See Image …: CAP-curve also shows the quality of logistic approximation. The axis are: *x* – total percentage of shops, *y* – percentage of “positive” shops. The ideal model would look like a right angle that maximizes the area between the CAP-curve and

diagonal. CAP-curve that coincides with the diagonal corresponds to absolutely random definition of 0 and 1.



This graph was made in Excel due to the fact that Python libraries do not have direct access to CAP-curve tools.

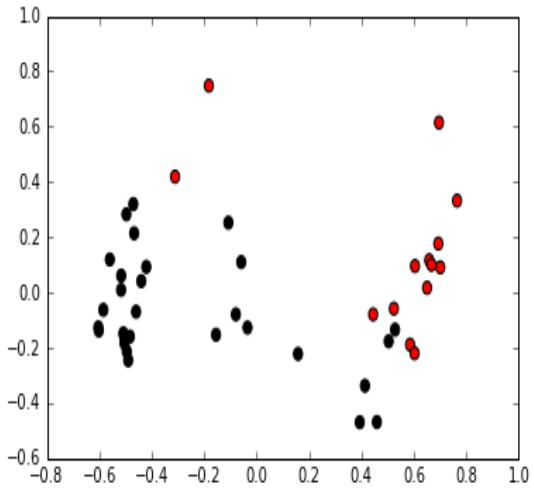
Other metrics are similar, they also compare model results with real data. If their values are considered acceptable (for example, AUROC which is calculated as the area under ROC curve should not be less than 0,7), the model is considered good and the results (saying whether sell of 0,33 can should be launched in a shop newly added to the panel basing on its characteristics) are sold to the client.

If there are too many regressors it is sometimes useful to lower dimension for better visualization of data. It is really useful in classification tasks to be able to draw an optimal line (or a plane in 3D) that suits best for dividing binary results. This method diminishes the dimension of data thus bringing it ti 2D and 3D form.

Here is Python code (for bringing SelectedRegressors of dimension n to 2D graph) and results:

In [225]:
```
#principal component analysis to represent 28-dimensional data in two dimensions

pca = PCA(n_components=2)
new_data = pca.fit_transform(SelectedRegressors)

plt.scatter(new_data[:, 0],new_data[:, 1], c=['black' if x==0 else 'red' for x in Regressant], s=40)
```

Out[225]:
```
<matplotlib.collections.PathCollection at 0xd5666a0>
```



Conclusion

My work experience in AC Nielsen (which lasted 8 months – from 02.17 to 10.17) has attributed to my professional portfolio in different ways.

First of all, the job was interesting from the point of view of theoretical material and as a suite to engineering education. Different methods from diverse fields of statistics were often implied and often needed alterations and upgrading. I was able to revisit the course of statistics in its practical application.

Secondly, I got an opportunity to work with really big data. Sometimes exports from, for example, VPS were so overloaded that it was impossible to process it with Excel. So procedures were to be recoded

from VBA form to, for example, SQL form. These volumes often required optimization methods which were also implemented by either Data Science or StatOps Department.

Thirdly, for the first time I had a chance to work in a really big company with such an intricate system of inner communications. We worked with colleagues from all over the world (accented on StatOps Bulgaria, StatOps Russia, StatOps Romania, DS Russia, DS Belarus, StatOps Ukraine, DS Greece and many others) which gave an opportunity to communicate in English. This work-time has definitely been fruitful for my general corporate experience and situational knowledge of business.

Finally, I could see myself how the company ranked as “high-quality” in its sector functions. AC Nielsen is one of the most expensive companies to address to if you need audit and analysis of data but for several decades Nielsen has been showing unsullied quality and unbound aspiration to being the largest and the most efficient player in the market. Among Nielsen clients there are all well-known brands from Yahoo and Facebook (online analytics) to Coca-Cola and Pepsi-Cola, from Procter& Gamble and L’Oreal to Red Bull and Heinz. So, in spite of having changed my career path to the banking sphere, I really hope that I will have an opportunity to somehow collaborate with AC Nielsen.