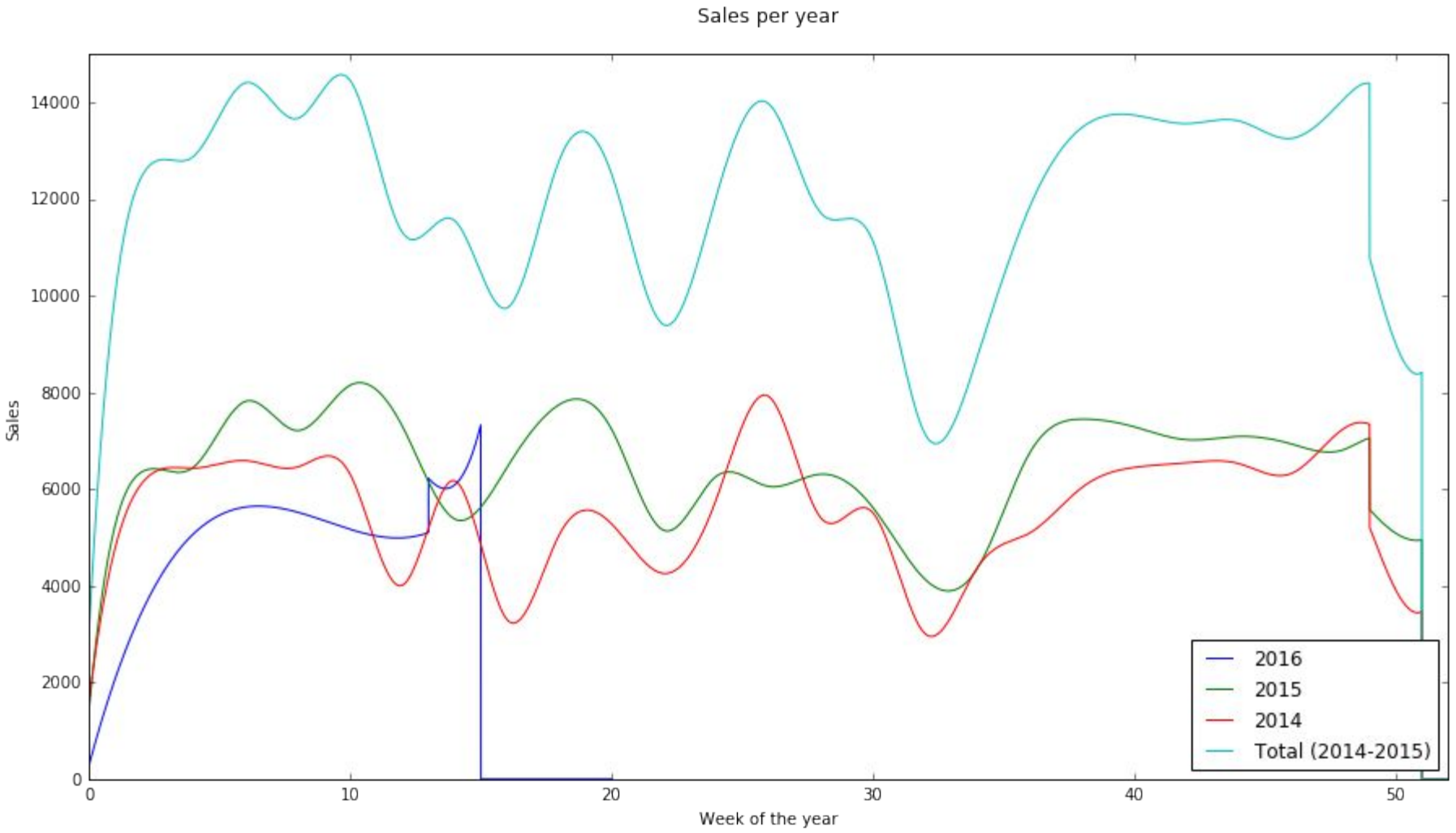
BIP Project

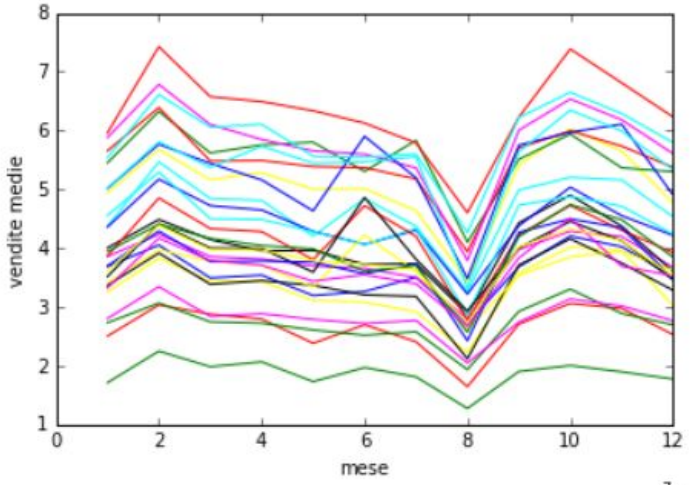
Riccardo Mastellone
Tommaso Carpi
Lorenzo Bisi
Marco Edemanti
Andrea Bellotti

Outline

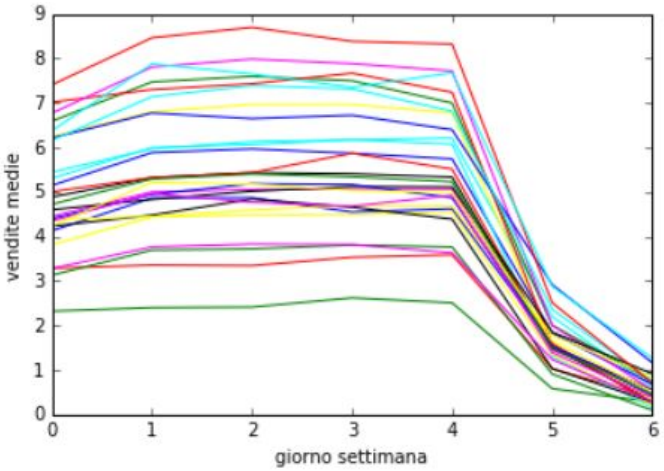
1. Data Exploration
2. Feature Extraction
3. The Model
    a. XGBoost
    b. ARMA
4. Conclusions

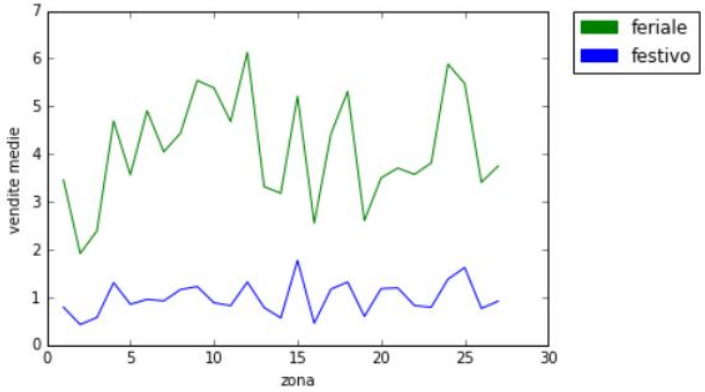
Data Exploration

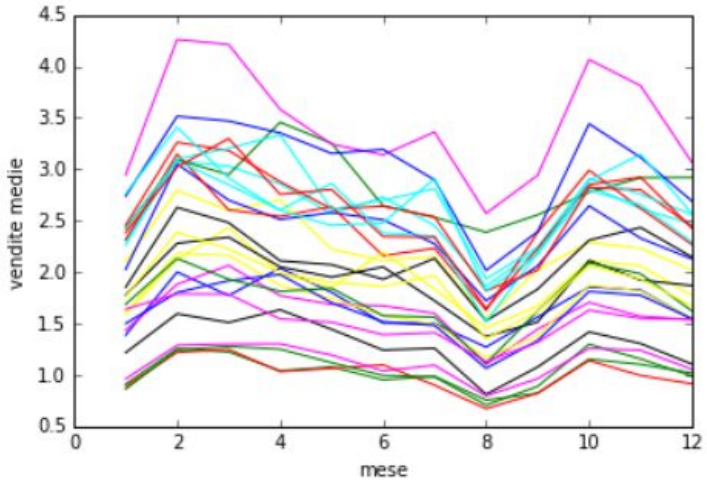




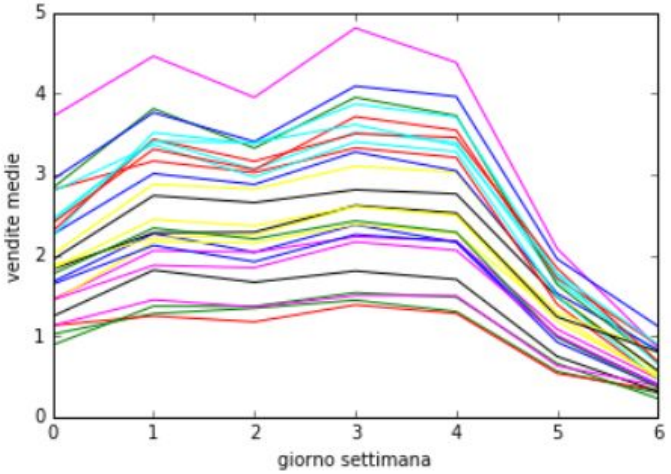
Categoria 1



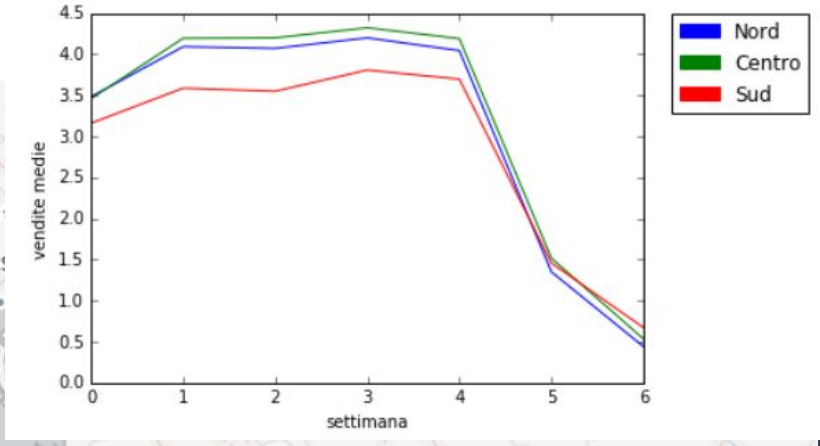




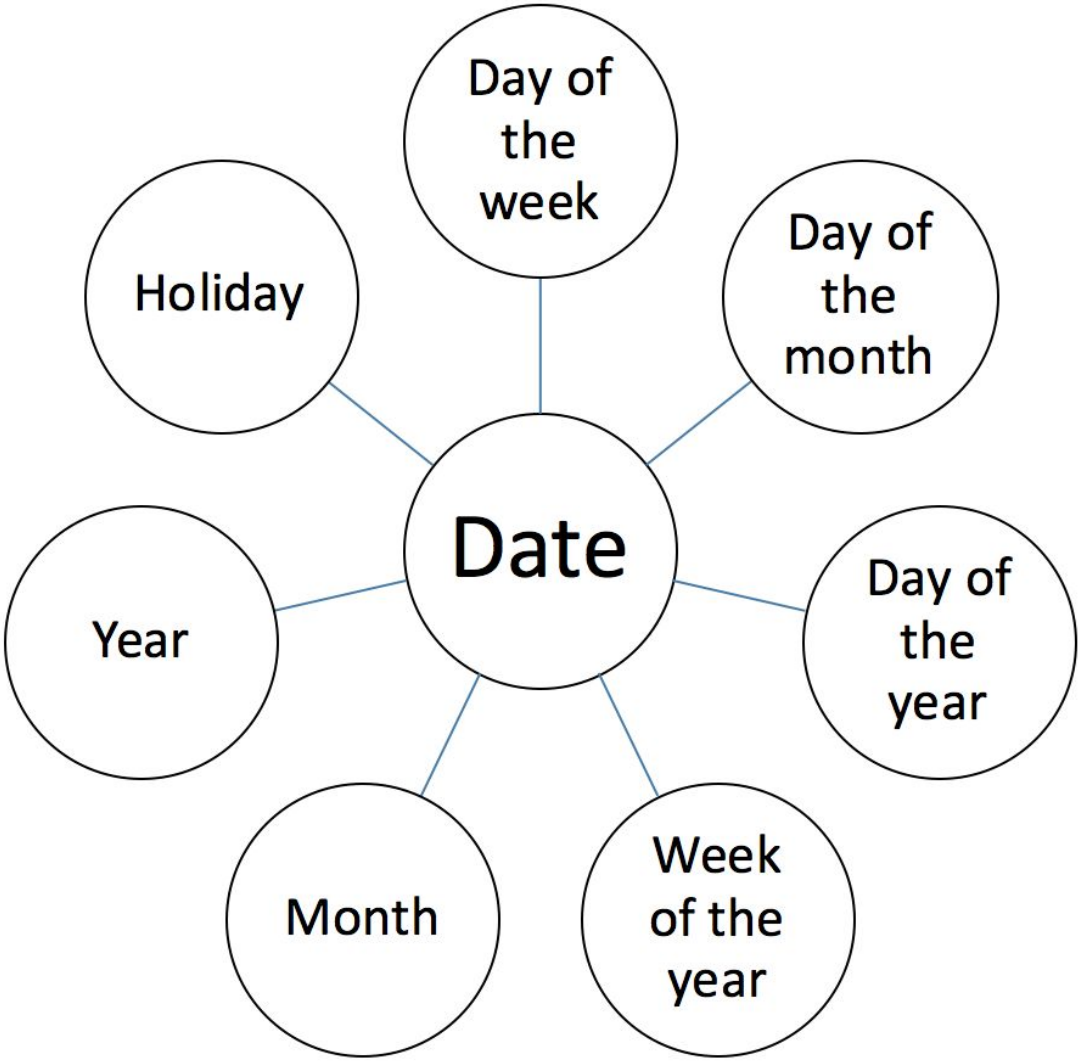
Categoria 2



Data Exploration

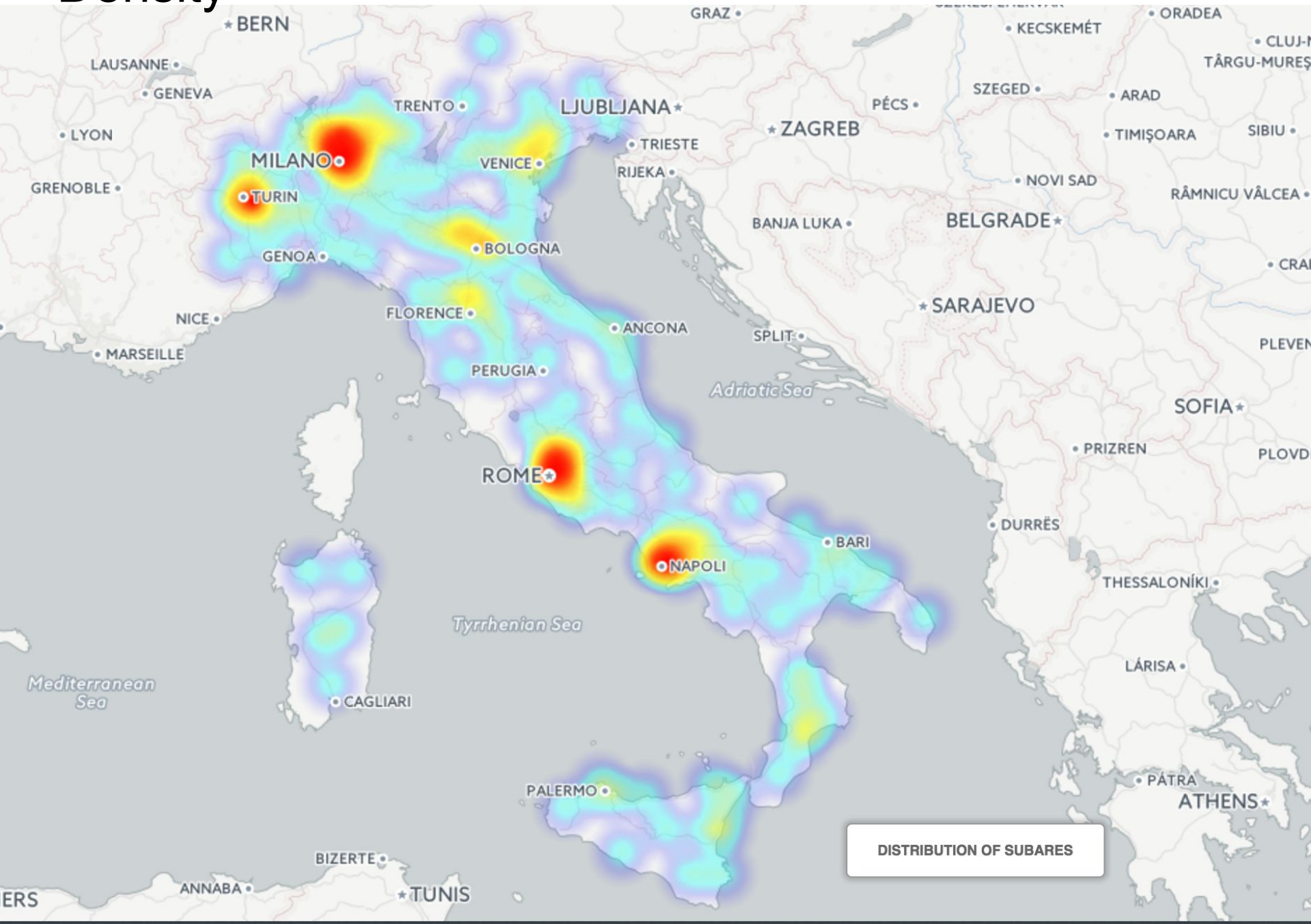


Feature Construction



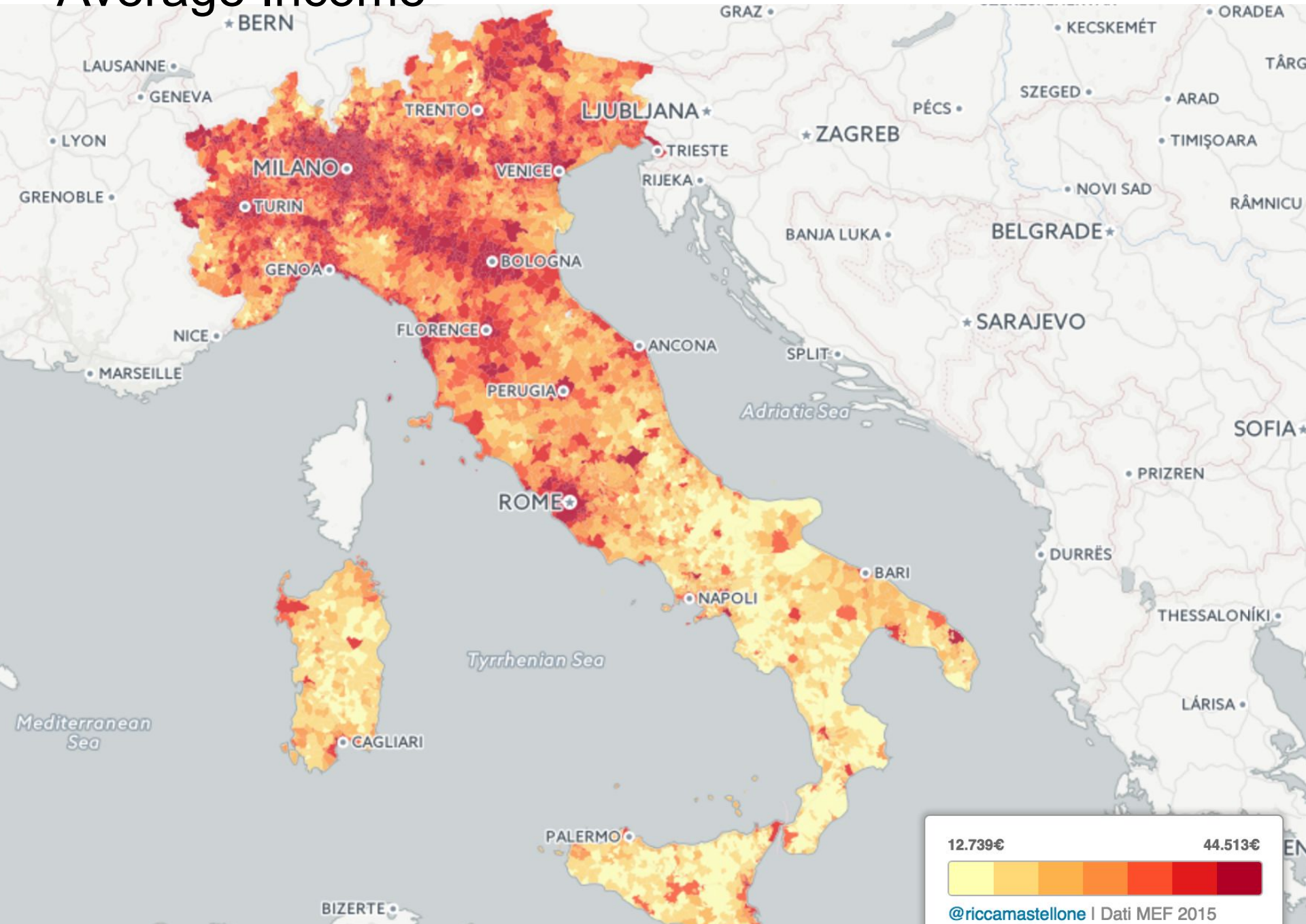
What about the GPS data?

Density

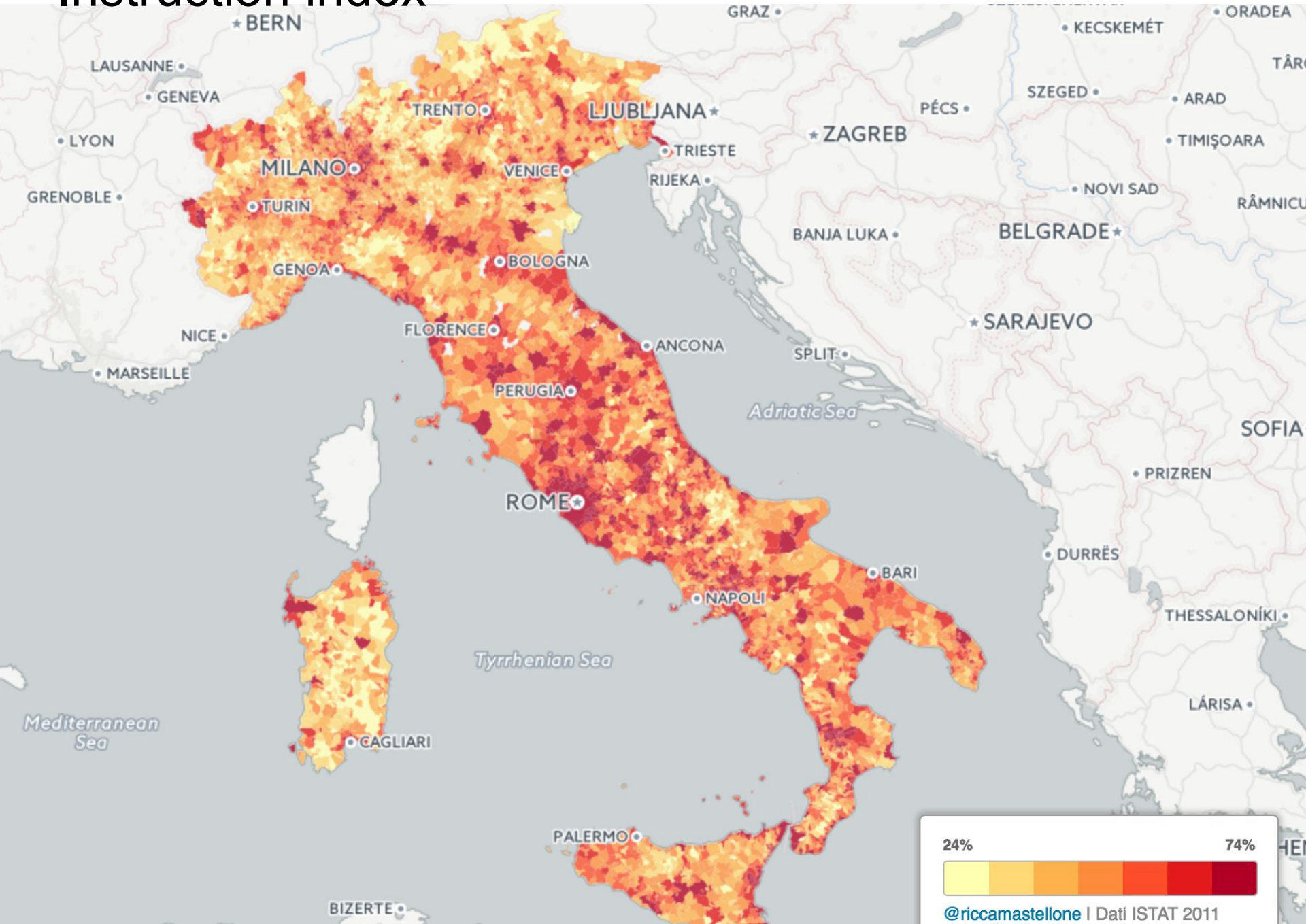

BERN
GRAZ
ORADEA
KECSKEMÉT
LAUSANNE
GENEVA
TRENTO
LJUBLJANA
SZEGED
PÉCS
ARAD
ZAGREB
LYON
TRIESTE
TIMIŞOARA
SIBIU
MILANO
VENICE
RIJEKA
GRENOBLE
TURIN
NOVI SAD
RÂMNICU VÂLCEA
BANJA LUKA
BELGRADE
BOLOGNA
GENOA
SARAJEVO
NICE
FLORENCE
ANCONA
SPLIT
MARSEILLE
PERUGIA
Adriatic Sea
SOFIA
PRIZREN
ROME
DURRËS
BARI
NAPOLI
THESSALONÍKI
Tyrrhenian Sea
Mediterranean Sea
LÁRISA
CAGLIARI
PÁTRA
ATHENS
PALERMO
BIZERTE
ANNABA
TUNIS
DISTRIBUTION OF SUBARES

Some insights after some mining...

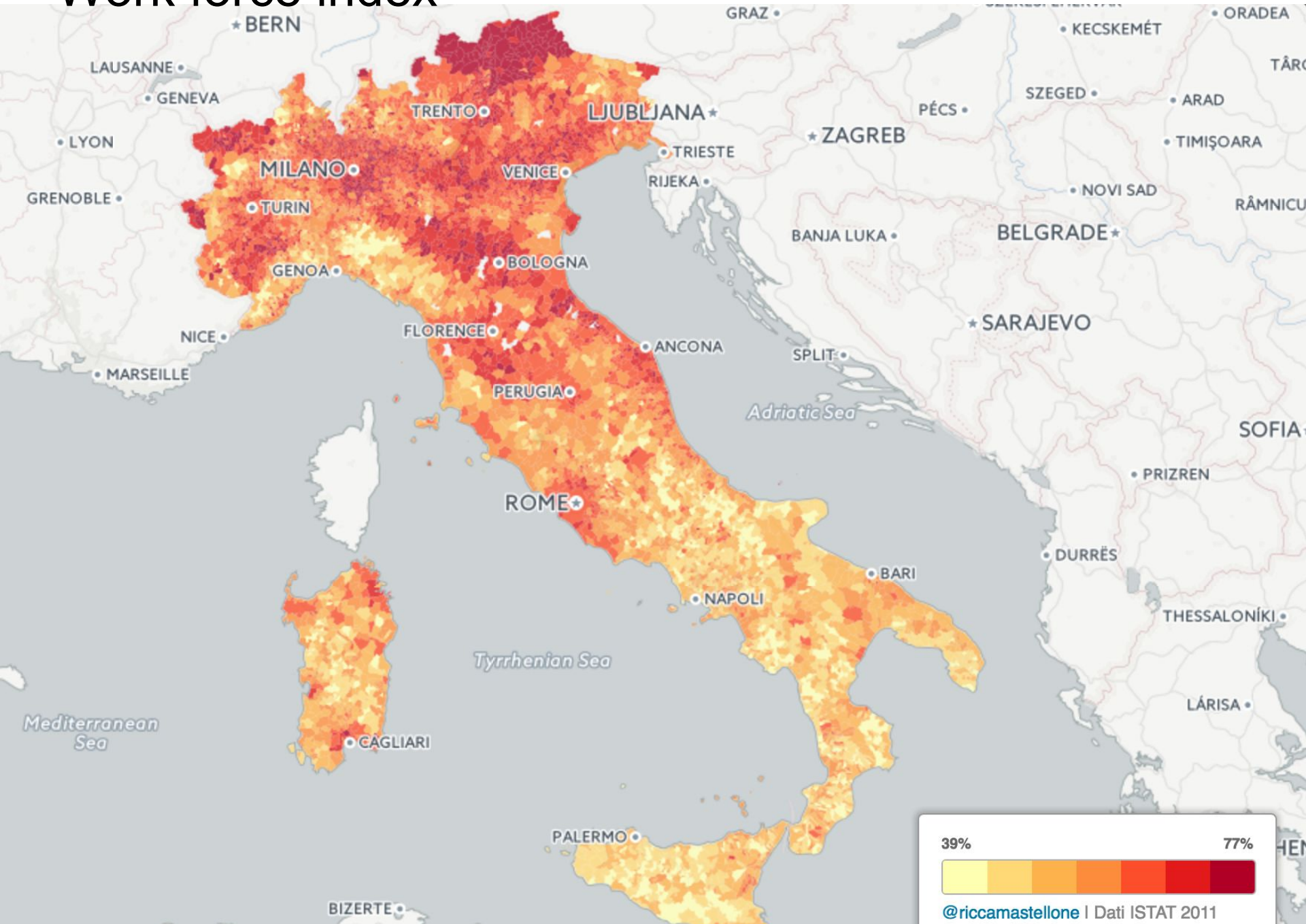
Average Income



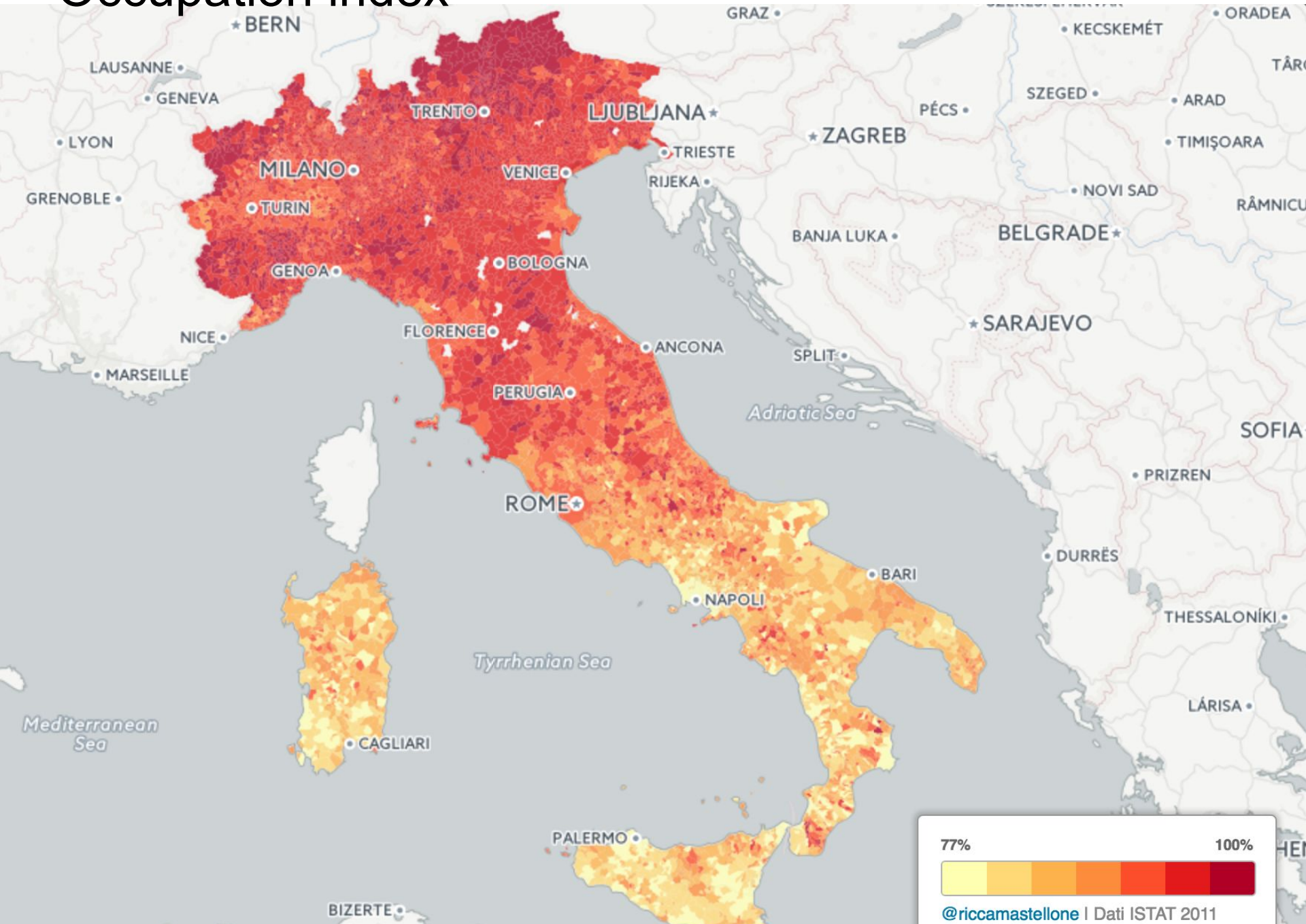
Instruction index



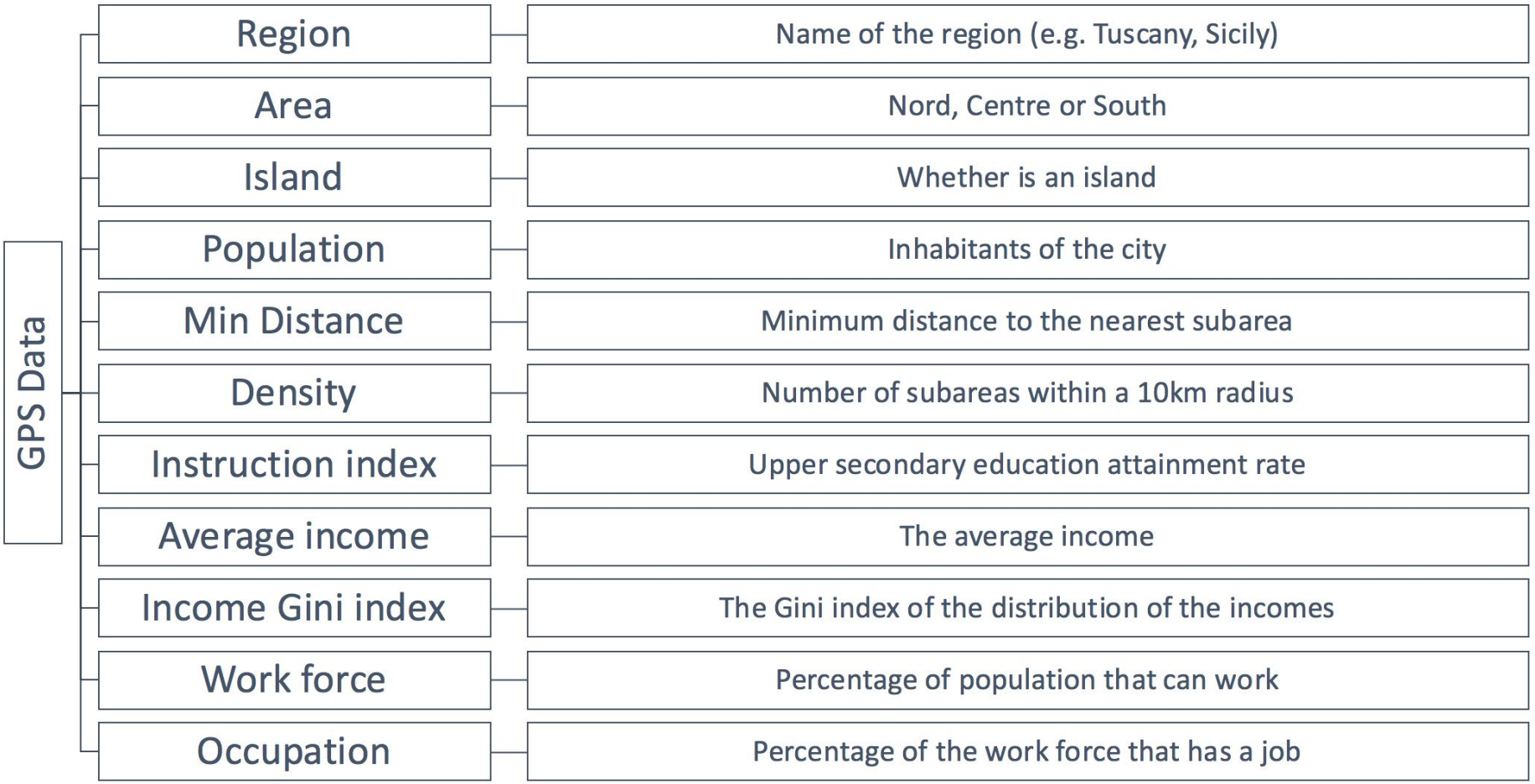
Work force index



Occupation index



Feature Construction



What can we do next?

- Both **categorical** and **numerical** features are present
- **High number** of features in the extended dataset
- Data are a **temporal series**

Initial attempts

- Regression Trees, Time Series, Neural Networks

Our plan:

- XGBoost
- ARMA/ARMAX
- Hybrid

Evaluation Metrics

Mean Average Prediction Error

Represents the overall evaluation of the error

$$MAPE1 = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Maximum Absolute Prediction Error

Provides an estimate of the worst error

$$MAPE2 = max\left(|A_t - F_t|\right)$$

XGBoost

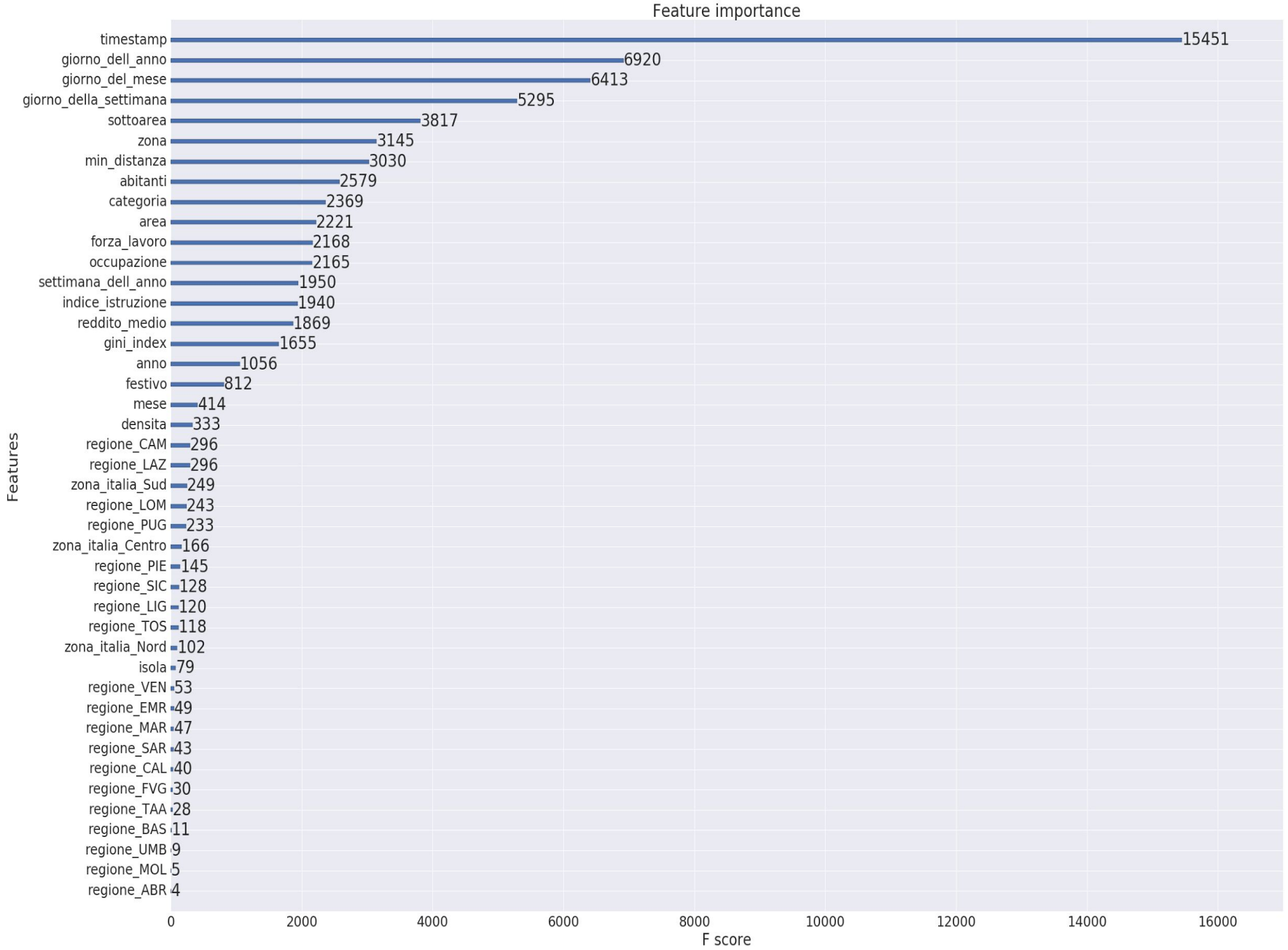
- Grid Search in order to find the set of parameters
  - max_depth : 10 (maximum depth of a tree)
  - min_child_weight: 10 (minimum number of instances needed to be in each node)

- Train the model for the **offline** evaluation
  - Last 10 days of the training set
  - Evaluate **MAPE 1** and **MAPE 2**

XGBoost - Results

MAPE1	Category 1	Category 2
2016 only	0.260363	0.410086
All years	0.228507	0.363510
Without 2014	0.234723	0.386823

MAPE2	Category 1	Category 2
2016 only	2.979166	2.062500
All years	2.854167	2.104167
Without 2014	3.006944	2.131944

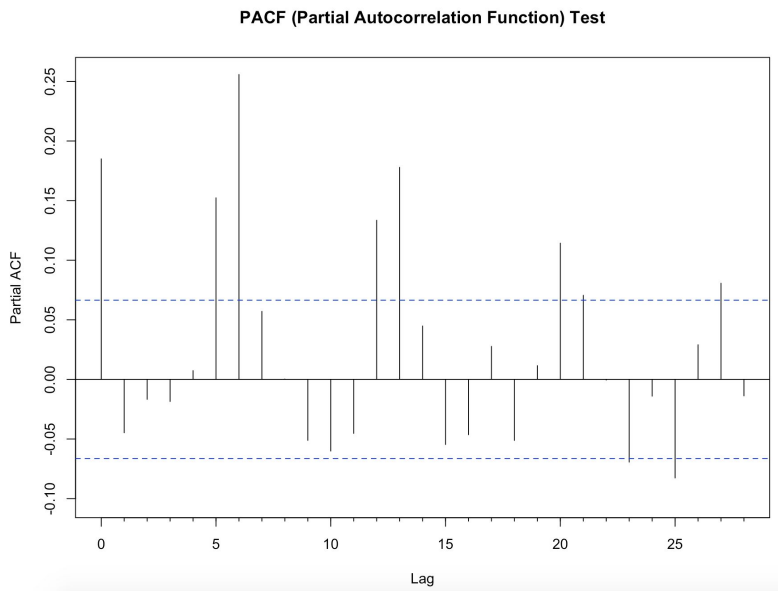

Feature importance
timestamp 15451
giorno_dell_anno 6920
giorno_del_mese 6413
giorno_della_settimana 5295
sottoarea 3817
zona 3145
min_distanza 3030
abitanti 2579
categoria 2369
area 2221
forza_lavoro 2168
occupazione 2165
settimana_dell_anno 1950
indice_istruzione 1940
reddito_medio 1869
gini_index 1655
anno 1056
festivo 812
mese 414
densita 333
regione_CAM 296
regione_LAZ 296
zona_italia_Sud 249
regione_LOM 243
regione_PUG 233
zona_italia_Centro 166
regione_PIE 145
regione_SIC 128
regione_LIG 120
regione_TOS 118
zona_italia_Nord 102
isola 79
regione_VEN 53
regione_EMR 49
regione_MAR 47
regione_SAR 43
regione_CAL 40
regione_FVG 30
regione_TAA 28
regione_BAS 11
regione_UMB 9
regione_MOL 5
regione_ABR 4
Features
0
2000
4000
6000
8000
10000
12000
14000
16000
F score

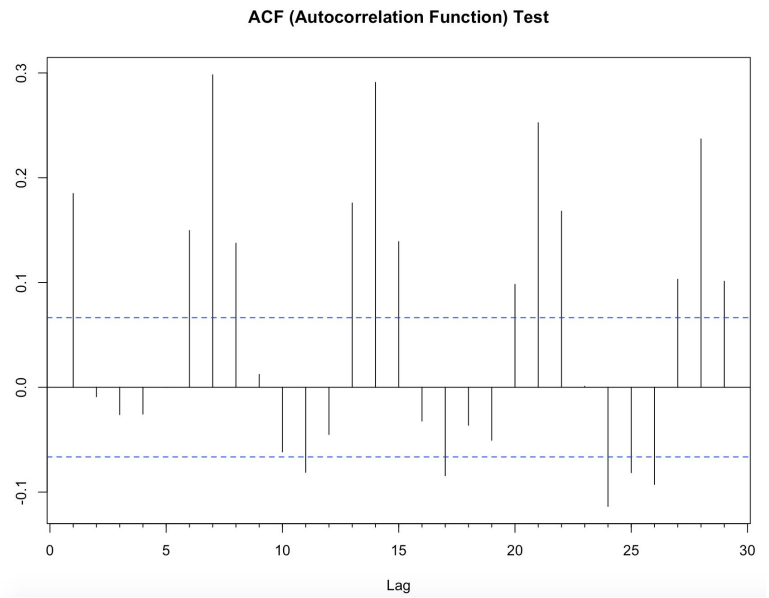
ARMA - Autoregressive Moving Average

$$y(t) = -\alpha_1 y(t-1) - \ldots - \alpha_n y(t-n) + \beta_0 u(t) + \beta_1 u(t-1) + \ldots + \beta_n u(t-n)$$

- Data Are Time Series
- Stationarity Tests
- Outlier Analysis
- ARMAX

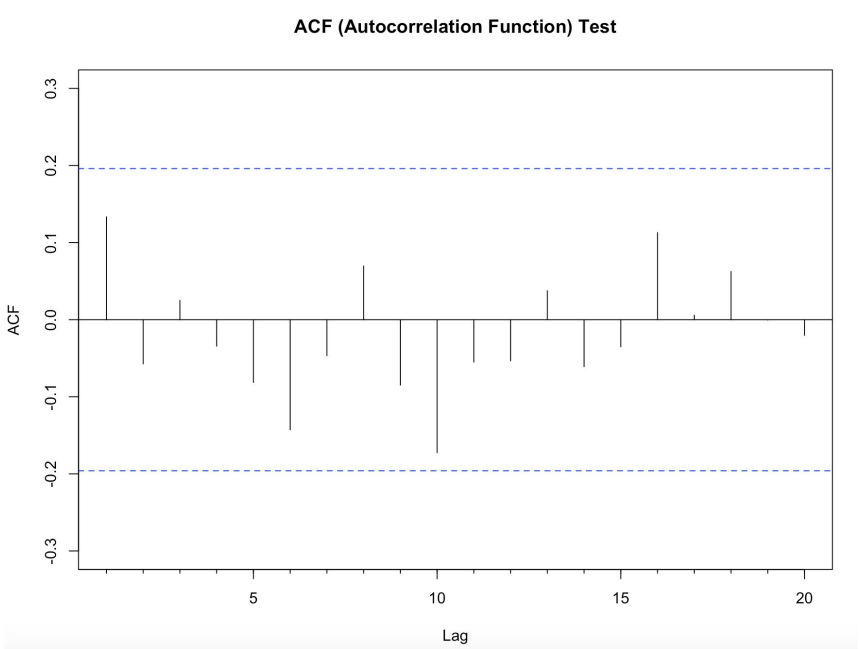
ARMA - Stationarity (before outlier adjustment)

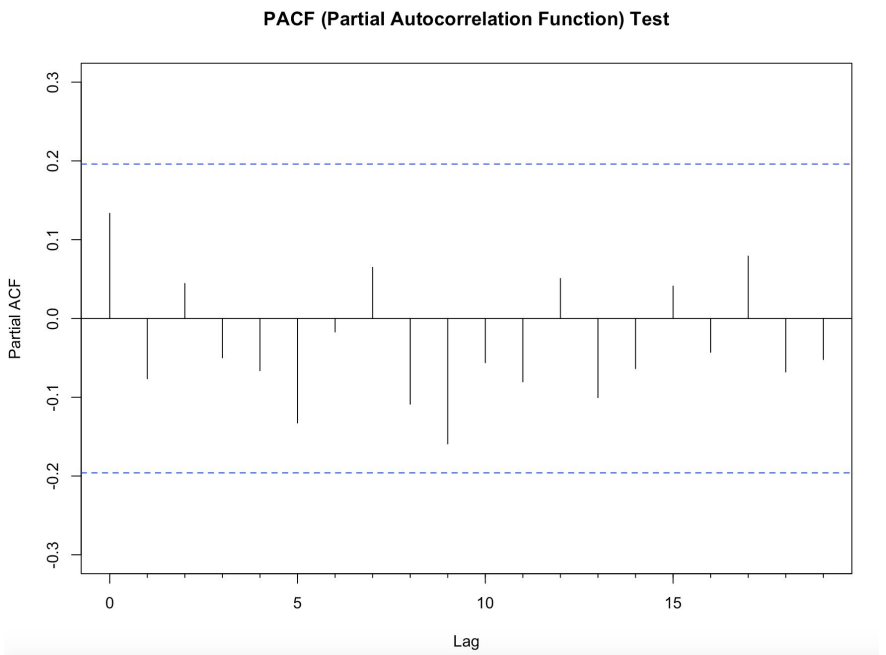





```
> kpss.test(datats)

        KPSS Test for Level Stationarity

data:  datats
KPSS Level = 0.76127, Truncation lag parameter = 6, p-value = 0.01

Warning message:
In kpss.test(datats) : p-value smaller than printed p-value
```

ARMA - Stationarity (after outlier adjustment)

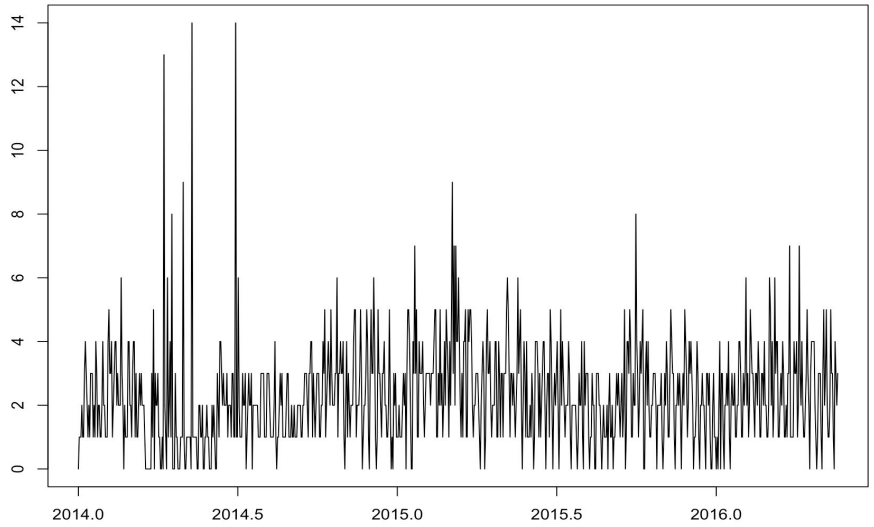


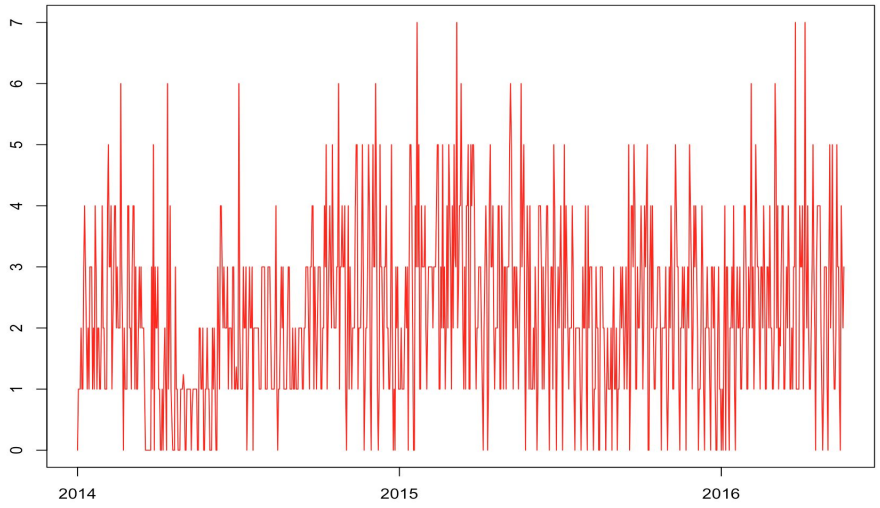


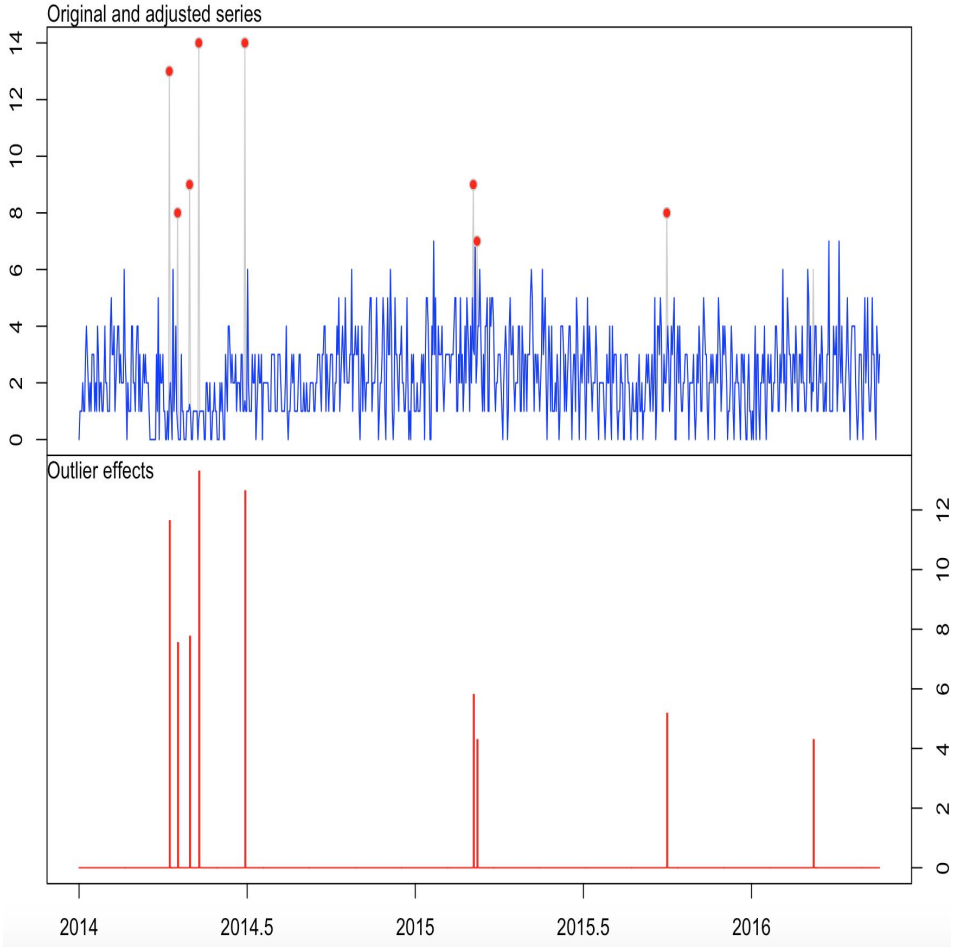

```
> res <- tso(datats, tsmethod = "auto.arima")
> res
Series: datats
ARIMA(2,1,2)

Coefficients:
         ar1      ar2      ma1     ma2     AO99   AO108   AO121    AO131    AO181   TC429   AO639
      0.7787  -0.3013  -1.5047  0.5443  11.6434  7.5141  7.7488  13.2642  12.6196  4.9900  5.2241
s.e.  0.0798   0.0362   0.0780  0.0751   1.2264  1.2285  1.2272   1.2273   1.2267  0.9783  1.2288

sigma^2 estimated as 1.708:  log likelihood=-1461.25
AIC=2946.5   AICc=2946.87   BIC=3003.71

Outliers:
  type ind     time coefhat  tstat
1   AO  99  2014:99  11.643  9.494
2   AO 108 2014:108   7.514  6.117
3   AO 121 2014:121   7.749  6.314
4   AO 131 2014:131  13.264 10.807
5   AO 181 2014:181  12.620 10.288
6   TC 429  2015:64   4.990  5.100
7   AO 639 2015:274   5.224  4.251
```

```
> kpss.test(res$yadj)

        KPSS Test for Level Stationarity

data:  res$yadj
KPSS Level = 1.5499, Truncation lag parameter = 6, p-value = 0.01

Warning message:
In kpss.test(res$yadj) : p-value smaller than printed p-value
```

ARMA - Outliers







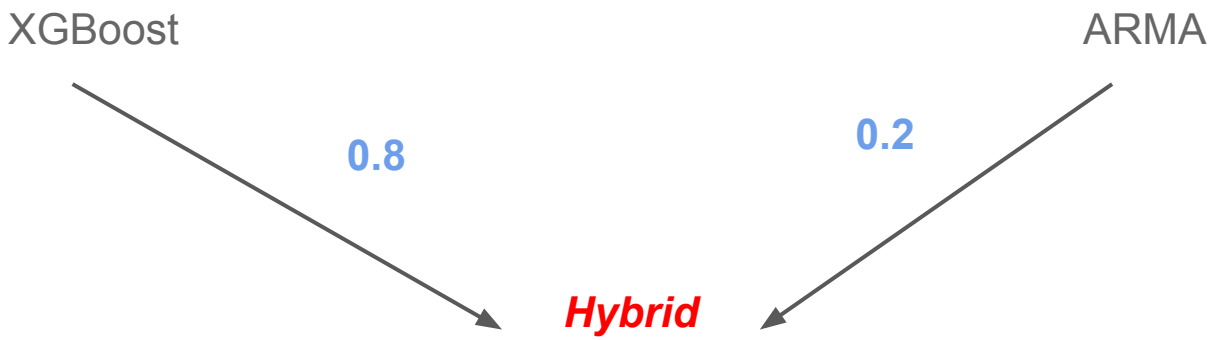
ARMAX

- KPIs as exogenous variables
  - DayOfTheWeek
  - Month
  - IsHoliday
- Optimal Model Not Found
- ARMAX(2,0,7)

ARMA - Results

Model	Category	MAPE1	MAPE2
ARMA	Category1	0.3130079	4.260563
ARMA	Category2	0.458818	2.540146
ARMA(outliers adjusted)	Category1	0.3436393	4.10951
ARMA(outliers adjusted)	Category2	0.4413407	2.317809
ARMAX(2,0,7)	Category1	0.8150487	

Model Hybrid



Hybrid	MAPE1	MAPE2
Category1	0.219272	2.782712
Category2	0.358711	1.989423

Summary of the results

Model	Category	MAPE1	MAPE2
ARMA	Category1	0.313008	4.260563
ARMA	Category2	0.458818	2.540146
XGBoost	Category1	0.228507	2.854167
XGBoost	Category2	0.363510	2.104167
Hybrid	Category1	0.219272	2.782712
Hybrid	Category2	0.358711	1.989423

Conclusions

- Extraction and addition of features improved a lot the quality of the prediction
- Two completely different approaches that perform well
- Time Series approach performs well even if processes are not stationary
- Hybrid of the two models performs even better
- All the models perform better with Category1