JUNE 13, 2021

SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

ANALYTICS FOR TELCO

ML ARCHITECTURES FOR TODAY'S TELECOM SYSTEMS :

MOBILITY PREDICTION

F. Gerardo Gandara E.

ANALYTICS FOR TELCO

MACHINE LEARNING USE CASE

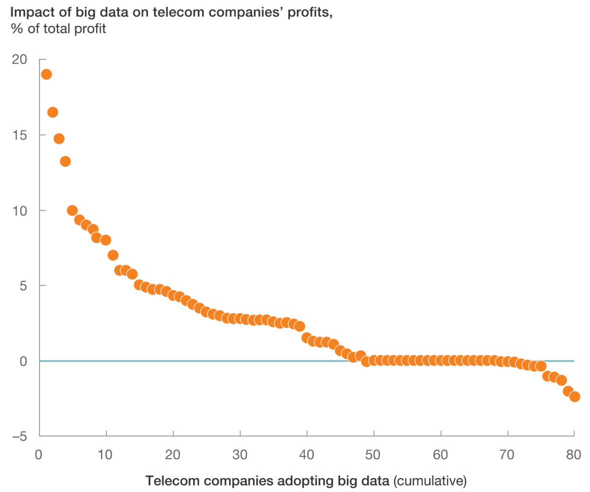
In next generation networks, mobile communication calls for service with higher quality, is bringing new challenge for mobility management. Thereinto, utilization and improvement of mobility prediction helps for preserving resource and providing better performance. So this paper aims to propose a ML improvement model of the mobility prediction in cellular network.

System Monitoring	Managed Services	Intelligent Networks
Anomaly Detection	Ticket Classification	Self-Healing
Root Cause Identification	Churn Prediction	Dynamic Optimization
Predictive Maintenance	SLA Assurance	Automated Network Design

Machine learning use cases in telecom have shown great potential in assisting with anomaly detection, root cause analysis, managed services, and network optimization. But to work effectively, they require specific computational, pipeline and support infrastructure.

Now that subscribers constantly connect to their networks through voice, text, and other smartphone interactions, telecom companies have access to huge quantities of data. Yet relatively few of those that have adopted big data architectures and analytics technologies have pushed aggressively enough to profit from them significantly.

According to an study in 2015, among telecom companies that invested in big data, only few got an incremental profit impact exceeding 10%.



Source: 2015 McKinsey survey of 273 global telecom companies, 80 of which have made big data analytics investments.

The potential for telecom that apply data science effectively is substantial. One of them used analytics models to predict the periods of heaviest network usage arising from video streaming. It subsequently took targeted steps to relieve congestion during those times, reducing its planned capital expenditures by 15 percent. In this case we will studying the implementation of ML algorithms to predict the mobility between .

PREDICTING MOBILITY

Mobility of devices (User Equipment) is one of the essential concerns in 5G core network (CN) design. As a result mobility pattern prediction is very crucial in improving the networks. A 5G network can be seen as a collection of gNB's (gNB is a base station).

How the data is tracked? If a UE is in active mode, the RAN will send a handover message when the UE switches between gNBs, notifying the AMF that user plane tunnels need to be reconfigured to the new gNB.

The UE connects via a DN to the closest application server, then the UE moves to the next edge site. To keep latency low after a gNB handover, the UE's context should be relocated not only on the gNB, but also on the UPF and the application server.

If the network knows in advance that the UE will move to the target edge site, then some parts of the relocation procedure can be done before the actual gNB handover. In this way, the overall procedure is shortened, and the handover can be performed more smoothly.

■ The dataset that we have is a collection of 3.5 days (real data) with 479,154 records of almost 9000 mobiles.

ML MODEL IMPROVEMENT

The current model Random Forest had the following parameters:

- predictionAccuracy': 69.45
- predictionGnodeBAccuracy': 81.11
- predictionTimeSlotAccuracy': 82.74

```
model = RandomForestClassifier(n_estimators=30,
                               min_samples_leaf=2,
                               min_samples_split=2,
                               max_depth=15,
                               oob_score=True,
                               random_state=44)
```

Four more models where evaluated in addition to random forest, but the final selection was Random Forest to do fine-tuning.

DecisionTreeClassifier

```
[ ] model = DecisionTreeClassifier(random_state=42)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_train)
    accuracy = compute_accuracy(y_train, y_pred)
    accuracy

    {'predictionAccuracy': 95.0501508603115,
     'predictionGnodeBAccuracy': 97.08064910707004,
     'predictionTimeSlotAccuracy': 97.325287450053}
```

KNeighborsClassifier

```
[ ] model = KNeighborsClassifier()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_train)
    accuracy = compute_accuracy(y_train, y_pred)
    accuracy

    {'predictionAccuracy': 67.06080621925032,
     'predictionGnodeBAccuracy': 78.59958139661312,
     'predictionTimeSlotAccuracy': 80.45339639566174}
```

ExtraTreeClassifier

```
[ ] model = ExtraTreeClassifier(random_state=42)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_train)
    accuracy = compute_accuracy(y_train, y_pred)
    accuracy

    {'predictionAccuracy': 95.0501508603115,
     'predictionGnodeBAccuracy': 97.08064910707004,
     'predictionTimeSlotAccuracy': 97.325287450053}
```

RadiusNeighborsClassifier

```
[ ] model = RadiusNeighborsClassifier()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_train)
    accuracy = compute_accuracy(y_train, y_pred)
    accuracy

    {'predictionAccuracy': 87.94204789475116,
     'predictionGnodeBAccuracy': 92.3931066351355,
     'predictionTimeSlotAccuracy': 93.64619859197043}
```

After applying in Random Forest (original algorithm) : TPOT classifiers, GridsearchCV hyperparameter selection, the improvement raised 39% reaching accuracy of 95% in training dataset.

So far the best model is :

> RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, max_samples=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None, oob_score=True, random_state=42, verbose=0, warm_start=False)

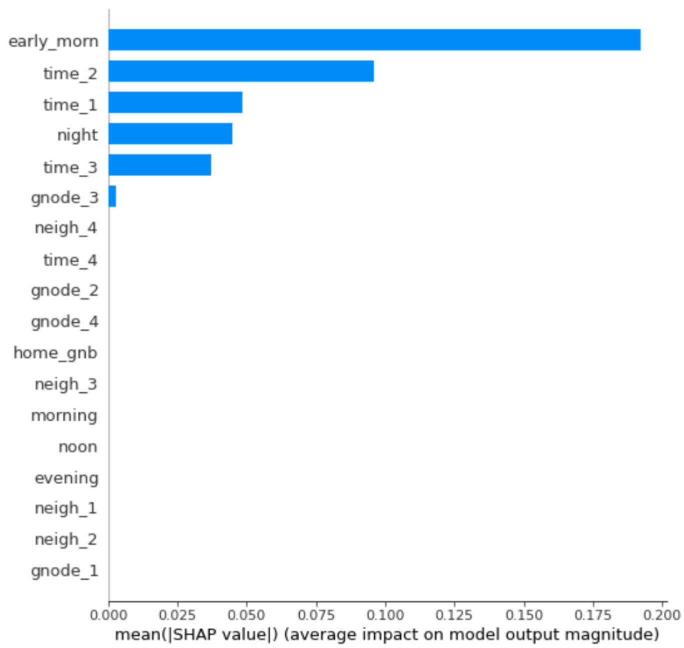
1. 'predictionAccuracy': 95.99911657775814
2. 'predictionGnodeBAccuracy': 97.49413883320308
3. 'predictionTimeSlotAccuracy': 97.73538106078624

Additional models evaluated: MultiOutputClassifier - KNeighborsClassifier and GridSearchCV with MultiOutputClassifier but the accuracy didn't increase.

In TEST, using the second day data of users, with all that new dataset, the model was applied and the increase was from 66% to 82%.

Another connected that we applied in Telecom datasets that happens often, is that the data collected might be drifted and that should be considered in the improvement. Dataset was evaluated and we found drift.

```
data_drift_checker_ok.ml_model_can_discriminate();
```




```
Drift found in discriminative model step, take a look on the most discriminative features (plots when minimal is set to False)
```

We proceeded to drop the 2most important columns of drifting and the model was OK. Proceeded to evaluate and the improve raised to 28%. Finally accuracy of 85%

```
{'predictionAccuracy': 85.02839600920952,
 'predictionGnodeBAccuracy': 90.18726016884114,
 'predictionTimeSlotAccuracy': 91.49654643131235}
```

CONCLUSIONS

Model and data drift need to be constantly validated in order to carry out model retraining. That is a crucial part of the cycle, in production we have to keep monitoring the datasets.

Since ML models strongly depend on data, the data needs to be carefully tracked and managed. Minor data schema or quality changes can significantly affect ML model accuracy. Therefore, data provenance and lineage should be versioned and tracked, and data quality should be assessed and validated.

There is a lot of potential of the use of genetic algorithm (GA) in telecom. Feature selection using GA would improve the accuracy classification of the gNBs at least in 5%

It is recommended also to evaluate a hybrid approach of a (GA) and local search algorithms for the backbone design of communication networks. The backbone network design problem is defined as finding the network topology minimizing the design/operating cost of a network under performance and survivability considerations.

ML solutions are complex systems comprised of several components that may differ from the existing infrastructure organizations have in place. Depending on the particular use case, each of these sub-components may be implemented in a different manner, but one important take in ML implementations in Telecom, the Inference pipeline the most important outcome is the drift detection, to re-evaluate our model.

Finally, keep in mind that In 2026, 3.5 billion 5G subscriptions are forecast, accounting for 40 percent of all mobile subscriptions and time-critical services for consumers, as well as enterprises, will be enabled when Critical IoT is introduced with 5G network, the potential can be limitless.

Figure 1: Mobile subscriptions by technology (billion)