Bridgestone Challenge

Rome, May 2019

BRIDGESTONE
Your Journey, Our Passion

Introduction:

Bridgestone is developing a new vision for which becoming a brilliant mobility solution provider is a key point

For this purpose a system of connected vehicles was developed in the R&D Technical Center (here in Rome)

Based on the expanding dataset that Bridgestone is creating the Bridgestone Digital Garage is lunching a Bridgestone Challenge

This “Mobility Hackathon” is proposed to students that wants to gain experience, visibility and to promote digital skills in Big Data and Mobility fields

Mobility Hackathon:

Promoted by Bridgestone, “Mobility Hackathon” is a programming marathon that offers to young students the opportunity to gain experience, visibility and to promote digital skills in Big Data and Mobility fields.

What is an Hackathon?
The hackathon is a development marathon in which developers, hackers, students, engineers and technology enthusiasts join forces to develop prototypes of software projects, sharing ideas and increasing skills in a fun way

Why?
Share ideas, work and projects in a context designed to facilitate collaboration, develop creativity and ingenuity in order to achieve a goal in such a short time, promote their skills and create value for themselves and others.

Challenge and Data Description

Details:

More than 500GB of Telemetry stored
More than 30 vehicles connected
More than 100km per vehicle per day
325 single messages per second per vehicle

Vehicle Telemetry Channels :

WheelSpeeds x4 (Km/h)
(RR – Rear Right; FL – Front Left)
Throttle Pedal Position (Percentage)
Longitudinal Acceleration (mG)
Lateral Acceleration (mG)
Odometer (Km)
Engine Speed (RPM)

Challenges:

1) Build an algorithm to identify Driver Fingerprint
2) Estimate tyre consumption based on Driving Style
3) Clusterize driving behaviour without looking at tyre consumption

Time:

1) 6 weeks (Mid-July) to analyze the training dataset
2) After some weeks will receive the rest of dataset and provide the output as requested per model

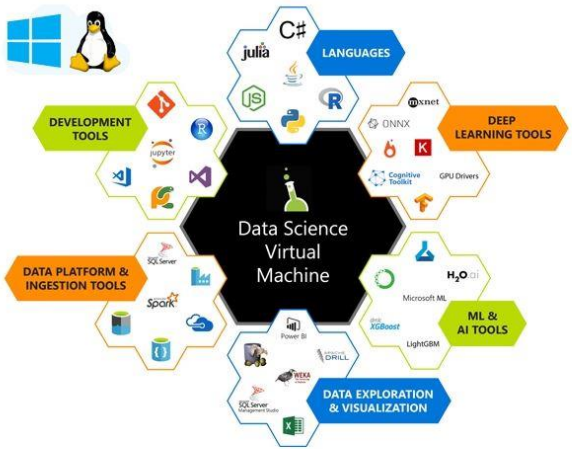
Example of Data
(format is an HIVE TABLE)

Features are extracted at different frequencies so NA are expected for low frequency signals

row_num	totmsec	car	gpsspeed	odometer	throttle	accelerometer	enginespeed	rrws	rlws	frws	flws
1	0	E177000985	34	173108	22	111	1229	37.4	37.4	37.8	38
2	100	E177000985				95	1239	37.7	37.6	37.96363	37.88182
3	200	E177000985				80	1244	38	37.8	38.01818	37.94545
4	300	E177000985				59	1255	38	38	38.27273	38.18182
5	400	E177000985				81	1261	38.1	38	38.21818	38.14545
6	500	E177000985			18	79	1257	38.2	38	38.36364	38.2
7	600	E177000985				81	1260	38.3	38.4	38.50909	38.47273
8	700	E177000985				81	1269	38.9	38.5	38.68182	38.59091
9	800	E177000985				49	1281	38.6	38.7	39.00909	38.93636
10	900	E177000985				62	1281	38.7	38.7	39.01818	38.84546
11	1000	E177000985	37	173108	20	72	1291	39	39.1	39.2	39
12	1100	E177000985				59	1301	39.2	39	39.56364	39.41818
13	1200	E177000985				84	1295	39.3	39.3	39.53636	39.4
14	1300	E177000985				95	1270	39.3	39.2	39.39091	39.24546
15	1400	E177000985				49	1283	39.3	39.4	39.42727	39.34546
16	1500	E177000985			0	23	1284	39.4	39.5	39.36364	39.31818
17	1600	E177000985				29	1289	39.2	39.1	39.25455	39.24546
18	1700	E177000985				29	1284	39.1	39.1	39.22727	39.22727
19	1800	E177000985				14	1273	39.3	39.1	39.3	39.21818
20	1900	E177000985				-8	1260	39.1	39	39.13636	39.13636
21	2000	E177000985	38	173108	0	5	1243	38.9	39	39	38.9
22	2100	E177000985				-24	1243	39	38.7	38.60909	38.56364
23	2200	E177000985				-53	1273	38.3	38.8	38.51818	38.47273
24	2300	E177000985				-58	1258	37.9	38	38.20909	38.14545
25	2400	E177000985				-54	1238	37.9	37.9	37.87273	37.89091
26	2500	E177000985			0	-53	1236	37.5	37.5	37.57273	37.56364
27	2600	E177000985				-74	1219	37.4	37.3	37.33637	37.35455

Development environment:

Data are accessible via VM that will be provided by Bridgestone Digital Garage per each Team



VM – Technology Stack

We will provide a full Virtual Machine environment that includes popular tools for data exploration, analysis, modeling & development.

Highlights :

- Microsoft ML Server - Dev Edition (Scalable R & Python)
- Azure Machine Learning Workbench
- Anaconda Python
- SQL Server 2017 Dev. Edition - With In-Database R and Python analytics
- Microsoft Office 365 ProPlus BYOL - Shared Computer Activation
- Julia Pro + Juno Editor
- Jupyter notebooks
- Visual Studio Community Ed. + Python, R & node.js tools
- Power BI Desktop
- Deep learning tools e.g. Microsoft Cognitive Toolkit (CNTK, TensorFlow, Chainer, & mxnet
- ML algorithm libraries e.g. xgboost, Vowpal Wabbit
- Azure SDKs + libraries for various Azure Cloud offerings. Integration tools are included for:
    - Hadoop
    - Hive
    - Blob storage
    - ML & Data Science tutorials as Jupyter notebooks

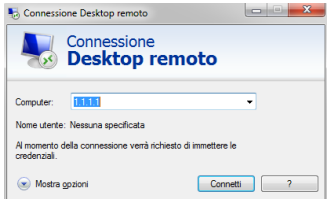
VM in this phase are using only CPUs

How to access the data:

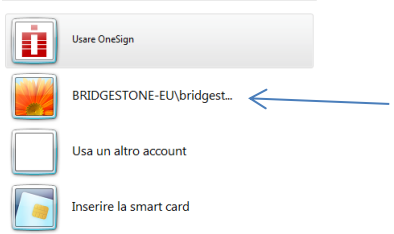
Data are accessible via VM that will be provided by Bridgestone Digital Garage per each Team

Access to the VM

1) Open a connection to the IP (provided per team)



2) Insert User Name and Password (provided per team)



User Name and Password

Access to the data from VM

1) User name and password per team to access HIVE ; Server, Database and table name will be the same for everybody

2) OBDC connection via Jupyter Notebook

```
#Create connection to an Hive SCHEMA using ODBC
SERVER_NAME='DummyName'
DATABASE_NAME='DummyName'   #Hive database is the schema
USER='DummyName'
PASSWORD='DummyName'
DB_DRIVER='Microsoft Hive ODBC Driver'
driver='DRIVER={'+DB_DRIVER+'}'
server='Host=%s;Port=10001' %(SERVER_NAME)
database='Schema=%s' %(DATABASE_NAME)
hiveserv='HiveServerType=2'
auth='AuthMech=7'
uid='UID=%s' %(USER)
pwd='PWD=%s' %(PASSWORD)

CONNECTION_STRING= ';'.join([driver,server,database,hiveserv,auth,uid,pwd])

connection = pyodbc.connect(CONNECTION_STRING, autocommit=True)
```

3) After connection simply SQL/HIVE query to retrieve data are possible

```
query_dummy = """Select * from TABLENAME
where car = "T1" and telemetry_channel = "Ax"
Limit 10 """
mypd = pd.read_sql(query_dummy, connection)
mypd.to_csv('C:\storing_data.csv')
```

The Challenge: Task 1

Challenges:

Build an algorithm to identify Driver Fingerprint

The goal is to be able to discriminate the different drivers.

Bridgestone will provide more than 30 vehicles/drivers in the training dataset with an ID
T1,T2,T3,T4,…
The student could access to this information using the variable “car”.

In the validation sample the same vehicles/drivers will be considered (*for a different time period*) with different ID
V1,V2,V3,V4,…

Student have to match the two vectors of IDs (finding the correct correspondences). More correspondences are taken correctly higher will be the score.

The Challenge: Task 2

Challenges:

Estimate tyre consumption based on Driving Style

The goal is to be able to estimate the tyre consumption using Driving Style information and consumption measurement.

Consumption measurements will be provided in a separate table with information of car ID per measurement. Consumption measurements are expressed in terms of residual tread (how much rubber to be consumed before change is left to the tire).

Bridgestone will provide the wear for more than 20 vehicles of the training dataset.

Student will provide the wear of the remaining vehicles. The result considered to evaluate the model goodness will be the RMSE.

The Challenge: Task 3

Challenges:

Clusterize driving behaviour without looking at tyre consumption

Data are the same of Finger Printing.

Student will provide clusters/score for driving behaviour (5 or more clusters). The result considered to check the model goodness will be the Silhouette (or DB Index). If necessary the cluster will be also tested against additional Bridgestone information (mianly vehicle maintenance)