

Loggi SLA Forecasting

Environment setup

```
In [1]: import pandas as pd
import datetime
import numpy as np
import os
from sqlalchemy import create_engine, text
import matplotlib.pyplot as plt
%matplotlib inline
```

SQL Connection setup

```
In [2]: db_host = 'postgresql://team4:team4@ds4a-lbenetton-instance.c6qxfh7ops9d.us-east-2.rds.amazonaws.com/ds4a_team4'
engine=create_engine(db_host, max_overflow=20)

def run_query(sql):
    result = engine.connect().execution_options(isolation_level="AUTOCOMMIT").execute(
text(sql))
    return pd.DataFrame(result.fetchall(), columns=result.keys())
```

Tabla definitions and data

Table Name: task (Tasks registry of each package journey)

```
In [55]: sql = """select * from task limit 5;"""
Task_df = run_query(sql)
Task_df.head()
```

Out[55]:

	taskid	packid	itineraryid	tasktype	ackstatus	waypointrole	completed
0	b4543846-c6fd-d152-887a-ab36c6ebb52a	c958f567-f977-2c85-d3d6-b45c20763ba5	3a782459-c460-5731-1ad9-f55bb3cf5845	Entrega	Realizado com sucesso	Recipient Address	2019-10-30 12:32:14
1	6609ed35-e0dc-bc12-3671-74e768495f53	bd035331-a93a-81c8-fc2e-0e62430d2754	e6ac457e-1637-807f-892a-53c14de44090	Entrega	Realizado com sucesso	Recipient Address	2019-10-30 12:03:36
2	a49dd036-a82b-d515-1cf6-cfb726cdfa99	d9b2f01b-a328-94eb-554f-a55c657642e3	660549ae-3f41-b11f-96b5-4ac295954a69	Retirada no last-mile	Realizado com sucesso	Distribution Center	2019-10-30 12:04:01
3	51ef4835-350a-d51e-a2e5-518374fa1560	f2221aa3-0ee2-8390-ae20-c6b46b8cb025	e5d67a86-2ce3-ca14-6564-d3a1d379a694	Entrega	Realizado com sucesso	Recipient Address	2019-10-30 12:32:40
4	abb1fae4-05b4-3ab0-ce3f-d440c30890fc	f3619b33-1359-f2f2-8e57-3a315d0d39c3	4c65010b-e0cb-b3ee-da07-7ed37275baa5	Retirada no last-mile	Realizado com sucesso	Distribution Center	2019-10-30 12:42:28

Table Name: package (Package registry I)

```
In [56]: sql = """select * from package limit 5;"""
Task_df = run_query(sql)
Task_df.head()
```

Out[56]:

	packid	agreed slo	final city	meso region	company id	status	deadline time	first delivery time
0	8672f758-664e-22e5-1288-648ec23742a1	D1	Osasco	São Paulo	be80bd7a-6e29-c16c-efea-6f3e13b41350	Entregue	2019-01-23 22:00:00	2019-01-23 13:24:53
1	eccb8efd-1bc8-e26e-d5d6-38f04d1f3355	D1	Osasco	São Paulo	be80bd7a-6e29-c16c-efea-6f3e13b41350	Entregue	2019-01-23 22:00:00	2019-01-23 18:58:08
2	aeaae4ce-6c08-f2ec-f815-ce42fbb16c57	D1	São Paulo	São Paulo	be80bd7a-6e29-c16c-efea-6f3e13b41350	Entregue	2019-01-23 22:00:00	2019-01-23 10:42:35
3	c959258f-0afe-24e1-dea5-2ea88ea3ea8a	D1	São Paulo	São Paulo	be80bd7a-6e29-c16c-efea-6f3e13b41350	Entregue	2019-01-23 22:00:00	2019-01-23 10:20:28
4	9b3f0ba6-a77e-d5a0-81b1-617182466b7b	D3	Brasília	Brasília	d840cc5d-906c-3e9c-8437-4c8919d2074e	Entregue	2019-01-28 22:00:00	2019-01-24 16:30:43

Table Name: packagedenorm (Package registry II)

```
In [57]: sql = """select * from packagedenorm limit 5;"""
Task_df = run_query(sql)
Task_df.head()
```

Out[57]:

	packid	agreed slo	final city	meso region	company id	status	height	length	width	real weight
0	95ebd2b7-2804-4c2c-a638-f4de0c2f8600	D0	São Paulo	São Paulo	afebc5be-1aea-43a1-8e1b-1bb5610f11cd	Não conferido	7.0	26.0	20.0	0
1	4cf19b35-1565-4317-bebd-268134b585ce	D1	Barueri	São Paulo	8d229e4d-6045-4052-b3a4-edebb47080a8	Entregue	10.0	30.0	30.0	3
2	59335aaa-1445-4aad-853b-93a204bdf3b	D1	São Paulo	São Paulo	722d1955-b5f9-4fd7-8d4c-5a226459365b	Entregue	8.0	23.0	25.0	1
3	c18a208b-f0cf-4fd0-a3ea-64da01364fa5	D1	Guarulhos	São Paulo	fab9f8f4-a619-4102-a267-b7baa2171816	Entregue	8.0	23.0	25.0	1
4	9097480e-0e4b-404c-b31a-b4dc1892adef	D1	São Bernardo do Campo	São Paulo	ff919694-466f-4f19-adfd-4f09a08b26f2	Entregue	8.0	23.0	25.0	1

Table Name: itinerary (Itinerary registry for packages)

```
In [58]: sql = """select * from itinerary limit 5;"""
Task_df = run_query(sql)
Task_df.head()
```

Out[58]:

	itineraryid	created	packages	accepted	checkedin	pickupcheckout	city	status	transi
0	0ea5531d-2abc-3fde-6a6f-5da703894b9d	2019-06-12 10:08:23.247	1.0	2019-06-12 10:08:59.313	2019-06-12 10:14:11	2019-06-12 10:23:51	Curitiba	finished	
1	47ed38c1-01f1-0392-f934-40c2fa897e99	2019-06-12 10:08:16.069	1.0	2019-06-12 10:12:53.786	2019-06-12 10:23:50	2019-06-12 10:25:11	São Paulo	finished	
2	8b05ca63-dd74-e21d-40ce-e8ec85579bf0	2019-06-12 10:07:19.902	6.0	2019-06-12 10:10:24.756	2019-06-12 10:27:23	2019-06-12 10:39:18	Santos	finished	
3	153bda12-90cd-6682-3ff5-5271d77f265a	2019-06-12 10:06:47.943	7.0	2019-06-12 10:08:28.393	2019-06-12 10:12:05	2019-06-12 10:20:16	Rio de Janeiro	finished	
4	28f99134-5364-3973-9239-5899b4289190	2019-06-12 10:06:46.266	8.0	2019-06-12 10:07:17.074	2019-06-12 10:11:37	2019-06-12 10:15:11	Rio de Janeiro	finished	

Table Name: transfer (Transfer ID registry and location)

```
In [59]: sql = """select * from transfer limit 5;"""
Task_df = run_query(sql)
Task_df.head()
```

Out[59]:

	transfer	status	loadstartedtime	transferstartedtime	transferendedtime	unloadendedtime	origina
0	cc237aa1-4fd2-43af-aaf8-746e534c4bc2	finished	None	None	None	None	São Pa - \ Leopold
1	6a7bbdb7-df44-4165-81d4-32c1246467ec	finished	None	None	None	None	São Pa - \ Leopold
2	e1e0b311-8ae1-4d02-926c-309936ca3689	finished	None	None	None	None	São Pa - \ Leopold
3	d672286d-5a8b-47e4-8333-27cf24894b80	finished	None	None	None	None	Cajar
4	44bb0ea8-13d9-4849-a79f-469bbda4694c	finished	None	None	None	None	São Pa - \ Leopold

Table Name: weather (Weather by time for Brazilian Locations)

```
In [61]: sql = """select * from weather limit 5;"""
Task_df = run_query(sql)
Task_df.head()
```

Out[61]:

	estacao	data	hora	precipitacao	tempbulboseco	tempbulbomido	tempmaxima	tempminima	umidac
0	82900	2018-01-01	0	NaN	26.6	23.1	32.4	NaN	
1	82900	2018-01-01	1200	0.0	29.7	23.3	NaN	23.2	
2	82900	2018-01-01	1800	NaN	30.9	25.4	NaN	NaN	
3	82900	2018-01-02	0	NaN	27.2	24.9	31.9	NaN	
4	82900	2018-01-02	1200	0.0	29.5	24.7	NaN	21.8	

5 rows × 21 columns

