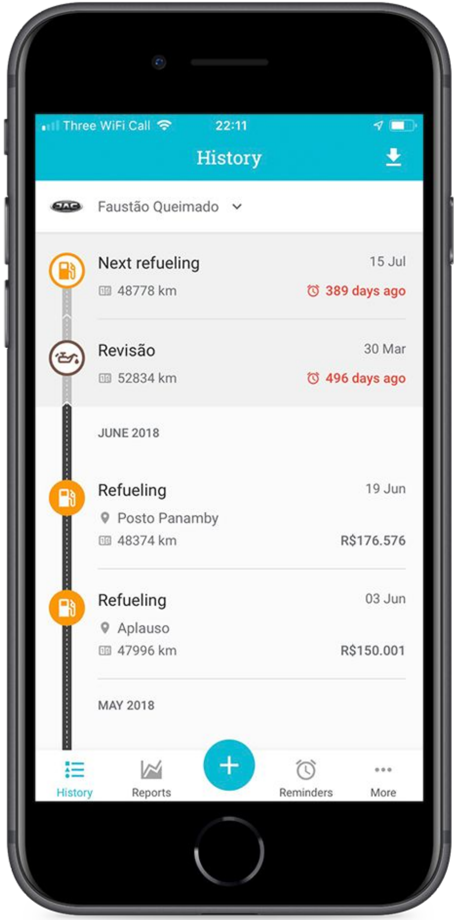
Capstone Project

drivvo

Juliana Maia

drivvo

Drivvo is a mobile app to record and keep track of vehicle expenses and keep on top of regular maintenance.



History

Faustão Queimado

Next refueling 15 Jul
48778 km 389 days ago

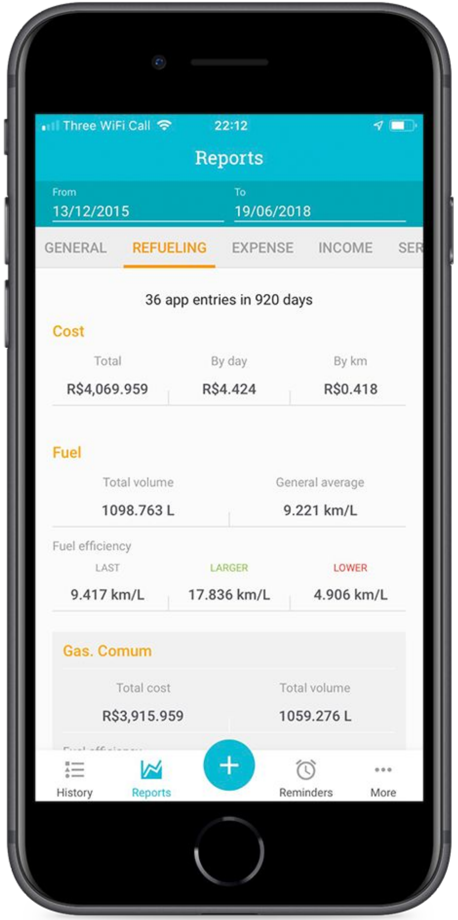
Revisão 30 Mar
52834 km 496 days ago

JUNE 2018

Refueling 19 Jun
Posto Panamby
48374 km R$176.576

Refueling 03 Jun
Aplauso
47996 km R$150.001

MAY 2018



Reports

From 13/12/2015
To 19/06/2018

36 app entries in 920 days

Cost

Total	By day	By km
R$4,069.959	R$4.424	R$0.418

Fuel

Total volume	General average
1098.763 L	9.221 km/L

Fuel efficiency

LAST	LARGER	LOWER
9.417 km/L	17.836 km/L	4.906 km/L

Gas. Comum

Total cost	Total volume
R$3,915.959	1059.276 L

How can I add value to the app?

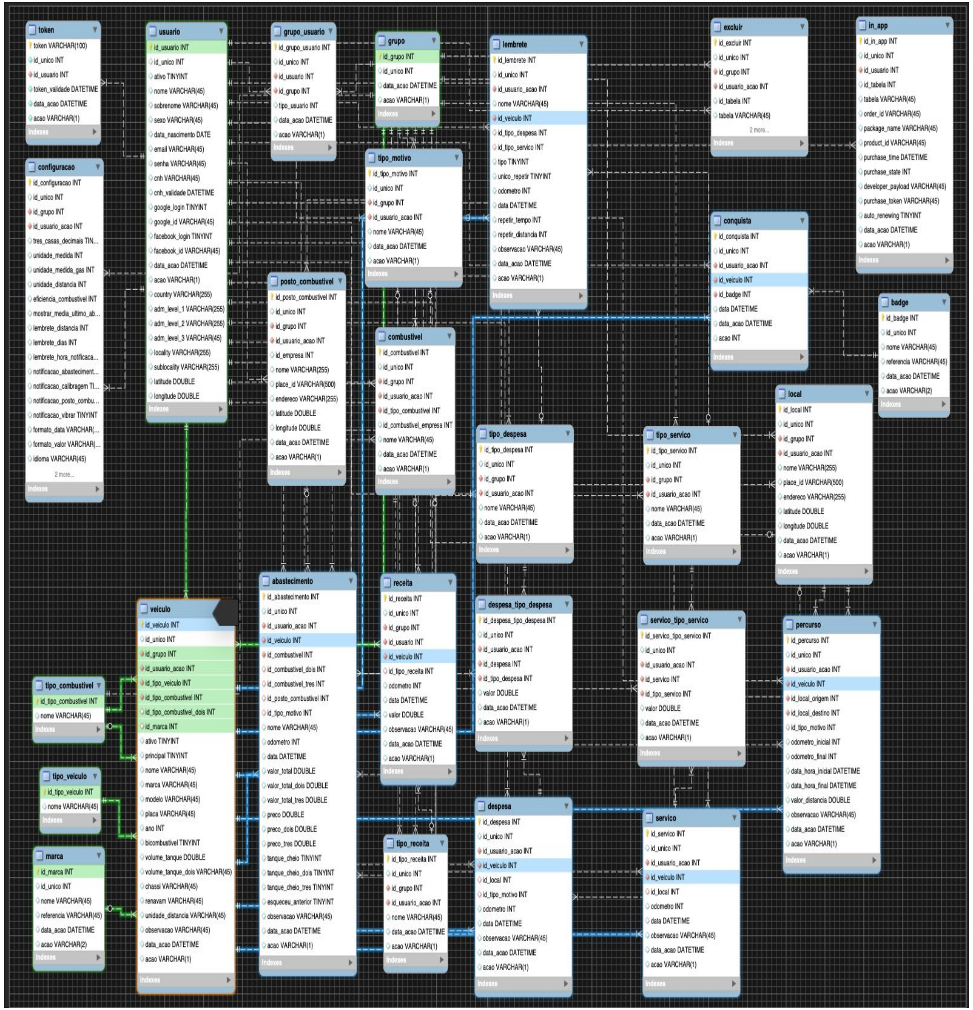
GOAL

Making it more consultative, predicting when the vehicle will need a service.

FIRST STEP: GET THE DATA

Data collection

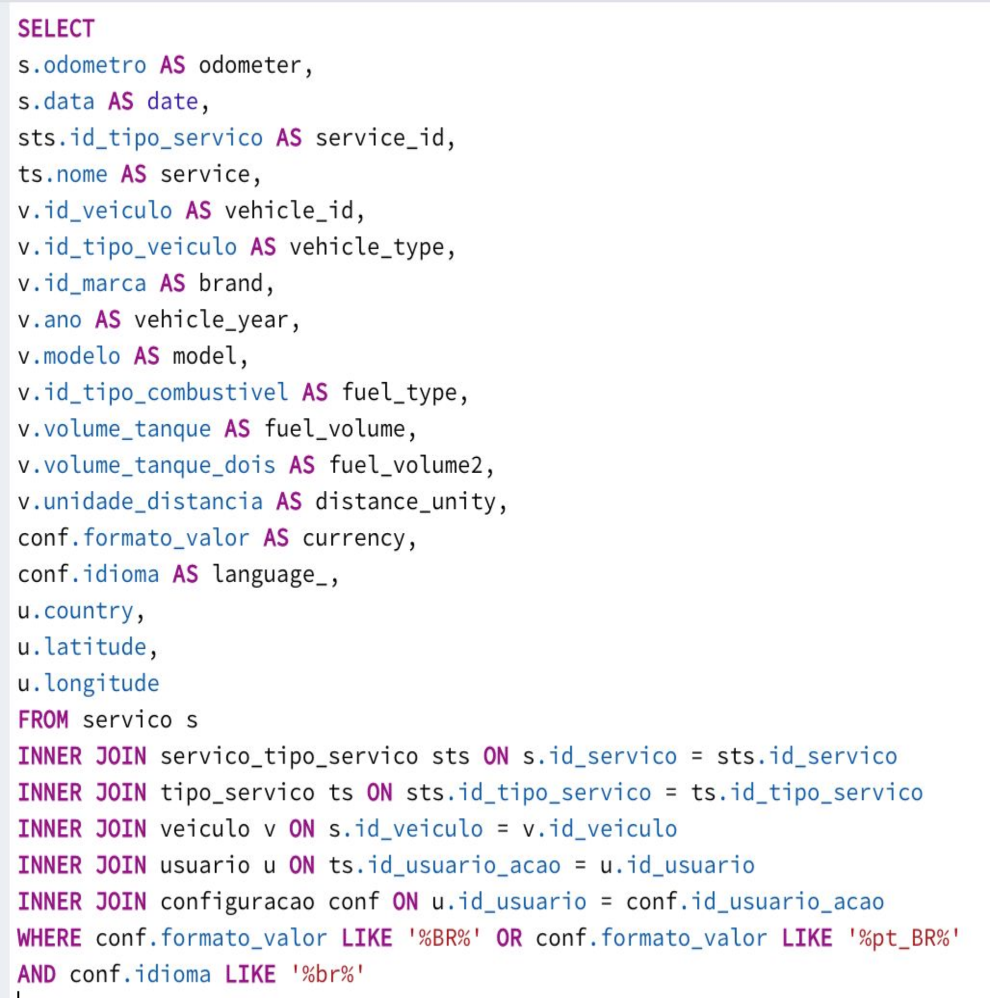
- MySQL Workbench
- PostgreSQL
  - 30+ tables
- Select data to work with
  - What could influence the need for vehicle services?



Data Collection

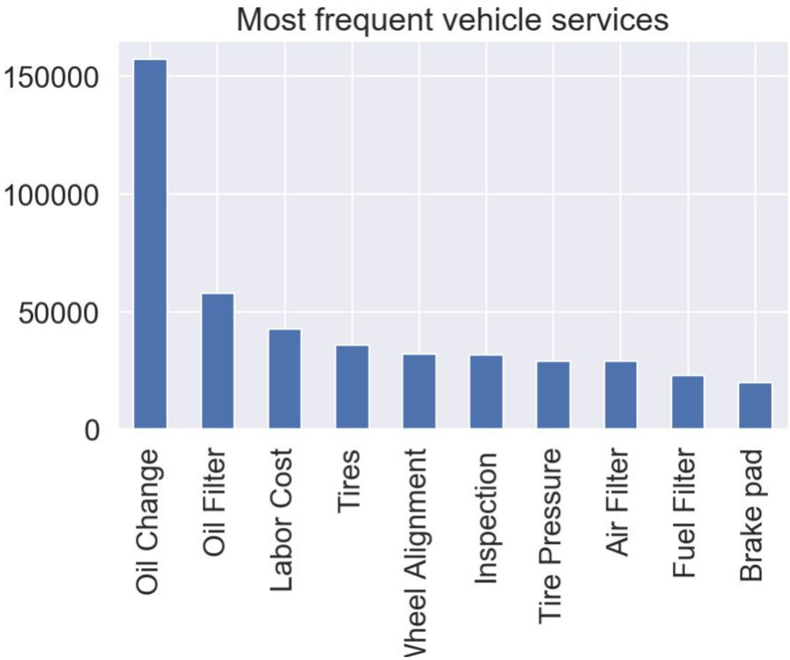
- Brazil
  - largest number of users (23,4%)
- csv
- Jupyter Notebook

800K+ entries

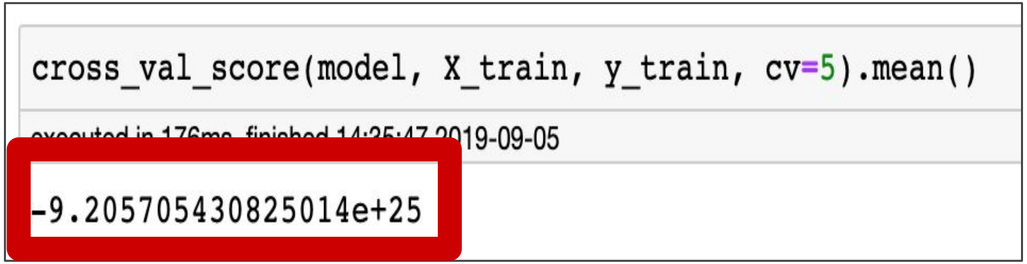


```
SELECT
s.odometro AS odometer,
s.data AS date,
sts.id_tipo_servico AS service_id,
ts.nome AS service,
v.id_veiculo AS vehicle_id,
v.id_tipo_veiculo AS vehicle_type,
v.id_marca AS brand,
v.ano AS vehicle_year,
v.modelo AS model,
v.id_tipo_combustivel AS fuel_type,
v.volume_tanque AS fuel_volume,
v.volume_tanque_dois AS fuel_volume2,
v.unidade_distancia AS distance_unity,
conf.formato_valor AS currency,
conf.idioma AS language_,
u.country,
u.latitude,
u.longitude
FROM servico s
INNER JOIN servico_tipo_servico sts ON s.id_servico = sts.id_servico
INNER JOIN tipo_servico ts ON sts.id_tipo_servico = ts.id_tipo_servico
INNER JOIN veiculo v ON s.id_veiculo = v.id_veiculo
INNER JOIN usuario u ON ts.id_usuario_acao = u.id_usuario
INNER JOIN configuracao conf ON u.id_usuario = conf.id_usuario_acao
WHERE conf.formato_valor LIKE '%BR%' OR conf.formato_valor LIKE '%pt_BR%'
AND conf.idioma LIKE '%br%'
```

Initial Idea



- Predict the next Oil Change



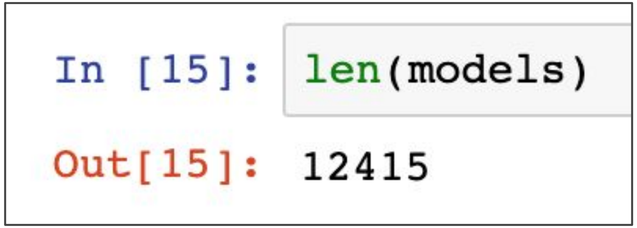
```
cross_val_score(model, X_train, y_train, cv=5).mean()
```

executed in 176ms, finished 14:35:47 2019-09-05

```
-9.205705430825014e+25
```

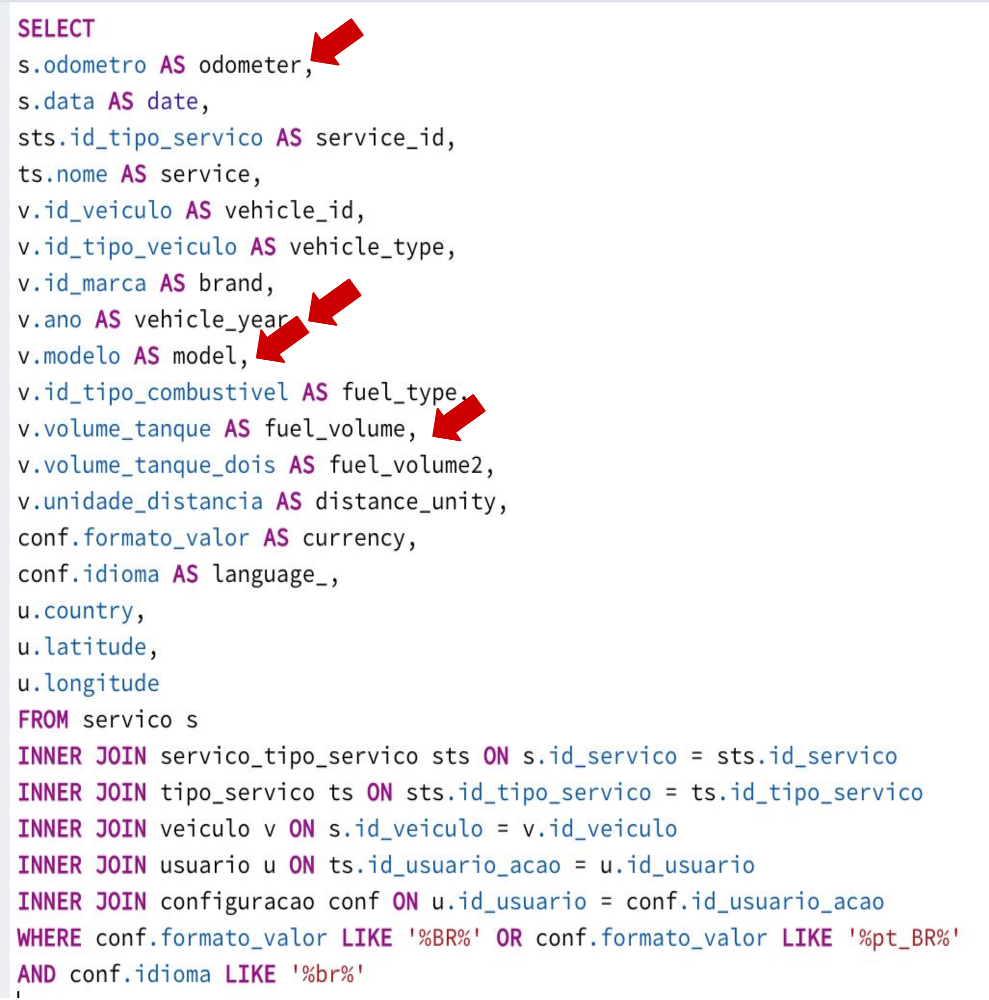
First Lesson

Unique models



```
In [15]: len(models)

Out[15]: 12415
```

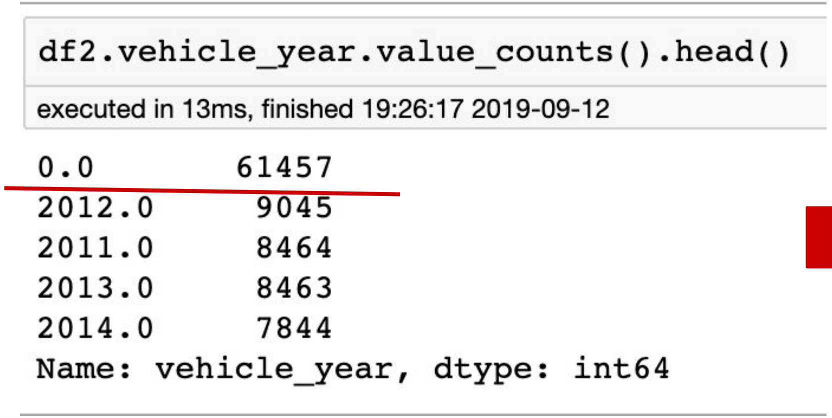


```
SELECT
s.odometro AS odometer,
s.data AS date,
sts.id_tipo_servico AS service_id,
ts.nome AS service,
v.id_veiculo AS vehicle_id,
v.id_tipo_veiculo AS vehicle_type,
v.id_marca AS brand,
v.ano AS vehicle_year,
v.modelo AS model,
v.id_tipo_combustivel AS fuel_type,
v.volume_tanque AS fuel_volume,
v.volume_tanque_dois AS fuel_volume2,
v.unidade_distancia AS distance_unity,
conf.formato_valor AS currency,
conf.idioma AS language_,
u.country,
u.latitude,
u.longitude
FROM servico s
INNER JOIN servico_tipo_servico sts ON s.id_servico = sts.id_servico
INNER JOIN tipo_servico ts ON sts.id_tipo_servico = ts.id_tipo_servico
INNER JOIN veiculo v ON s.id_veiculo = v.id_veiculo
INNER JOIN usuario u ON ts.id_usuario_acao = u.id_usuario
INNER JOIN configuracao conf ON u.id_usuario = conf.id_usuario_acao
WHERE conf.formato_valor LIKE '%BR%' OR conf.formato_valor LIKE '%pt_BR%'
AND conf.idioma LIKE '%br%'
```

SECOND STEP: DATA CLEANING (and extraction)

Why Extraction?

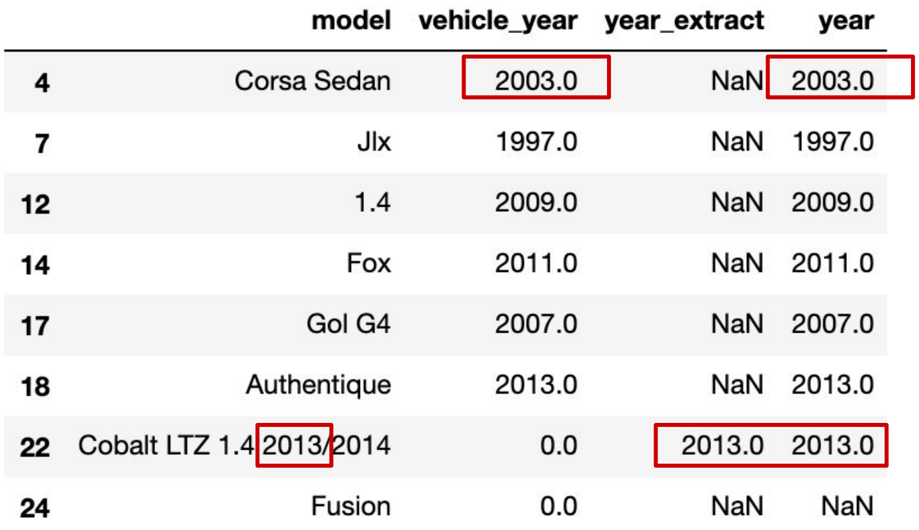
- Get as much information as possible from the users inputs



```
df2.vehicle_year.value_counts().head()
```

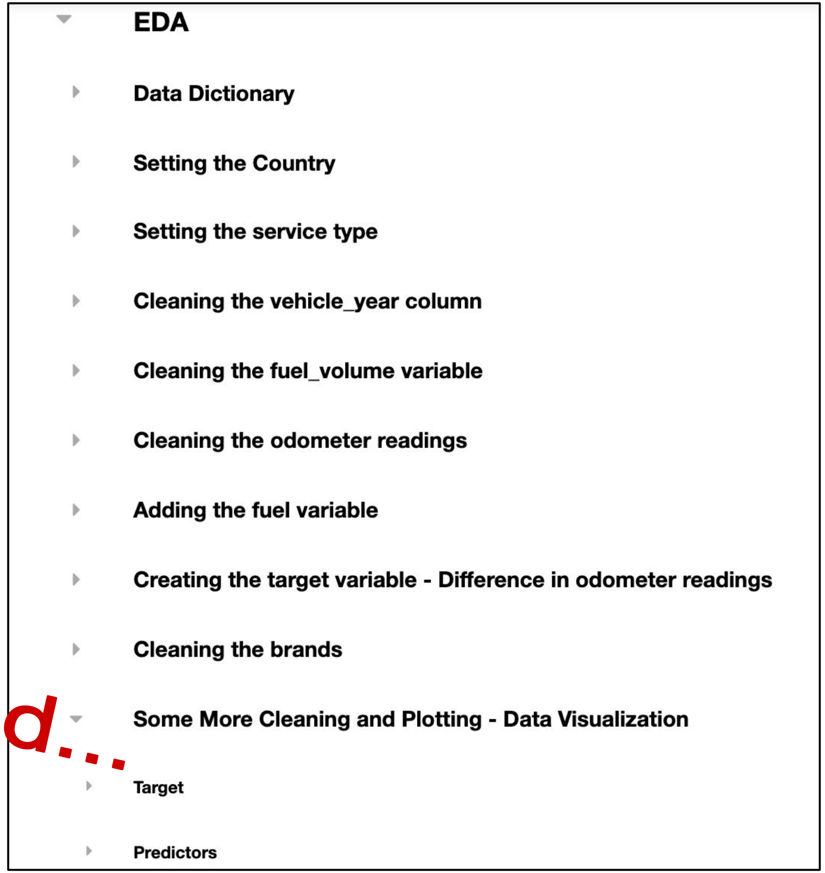
executed in 13ms, finished 19:26:17 2019-09-12

```
0.0          61457
2012.0        9045
2011.0        8464
2013.0        8463
2014.0        7844
Name: vehicle_year, dtype: int64
```



	model	vehicle_year	year_extract	year
4	Corsa Sedan	2003.0	NaN	2003.0
7	Jlx	1997.0	NaN	1997.0
12	1.4	2009.0	NaN	2009.0
14	Fox	2011.0	NaN	2011.0
17	Gol G4	2007.0	NaN	2007.0
18	Authentique	2013.0	NaN	2013.0
22	Cobalt LTZ 1.4 2013/2014	0.0	2013.0	2013.0
24	Fusion	0.0	NaN	NaN

Data Cleaning



EDA

- Data Dictionary
- Setting the Country
- Setting the service type
- Cleaning the vehicle_year column
- Cleaning the fuel_volume variable
- Cleaning the odometer readings
- Adding the fuel variable
- Creating the target variable - Difference in odometer readings
- Cleaning the brands
- Some More Cleaning and Plotting - Data Visualization
  - Target
  - Predictors

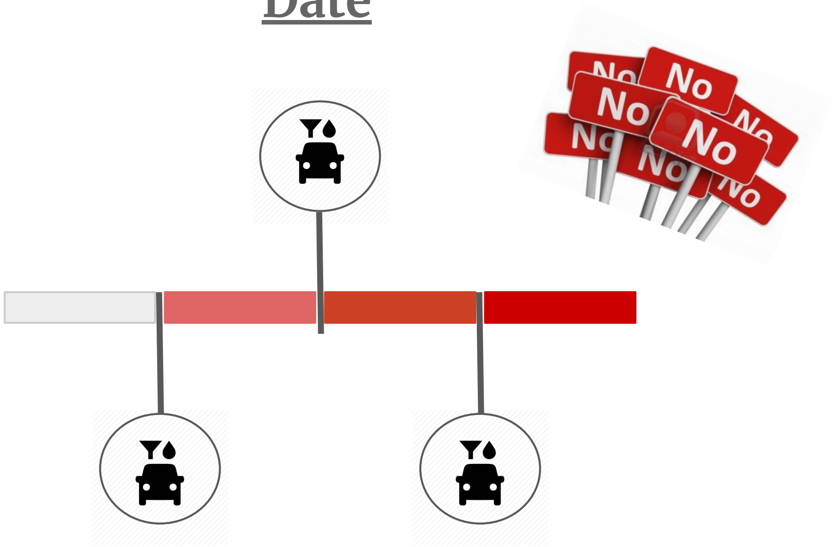
and...

THIRD STEP: TARGET AND PREDICTORS

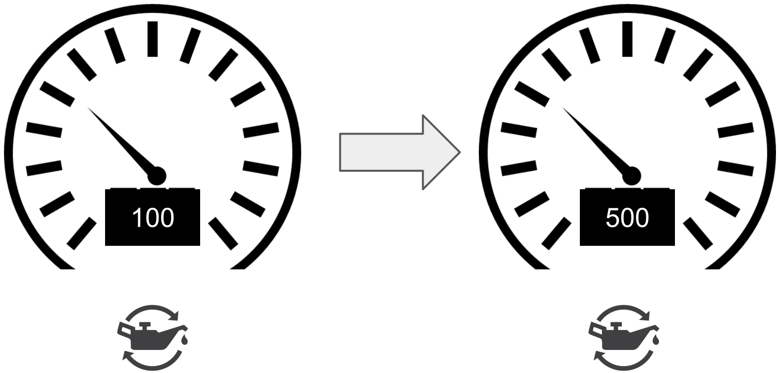
Defining the Target

- Main goal:
  - Predict when a vehicle will need the next oil change

Date

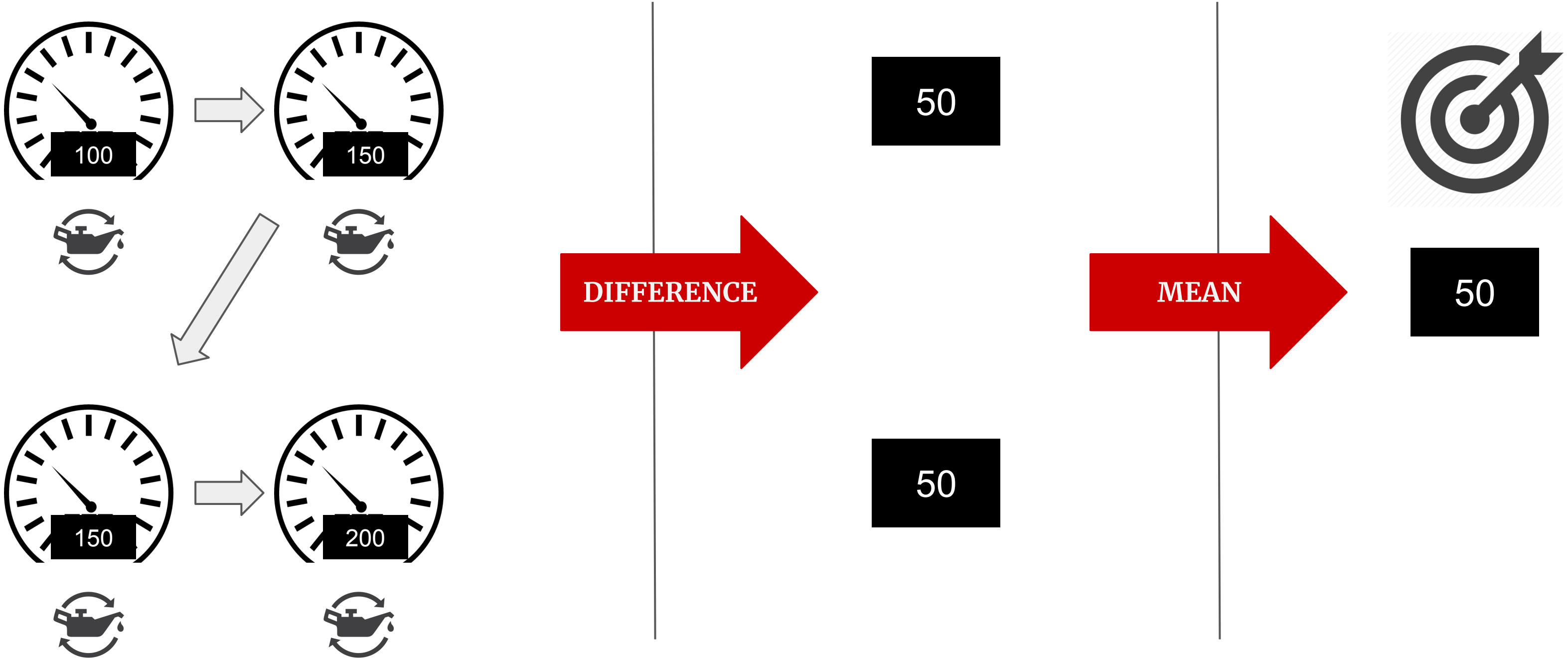


Odometer Reading



Defining the Target

- Attention to autocorrelation



100

150

150

200

DIFFERENCE

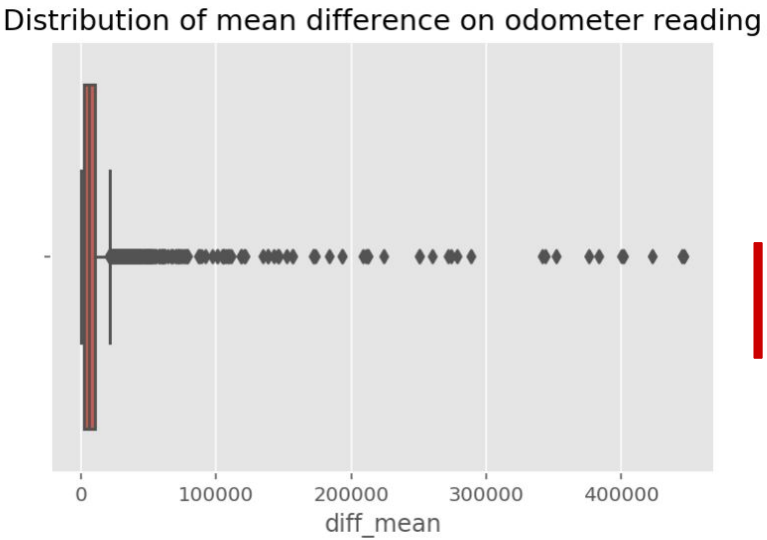
50

50

MEAN

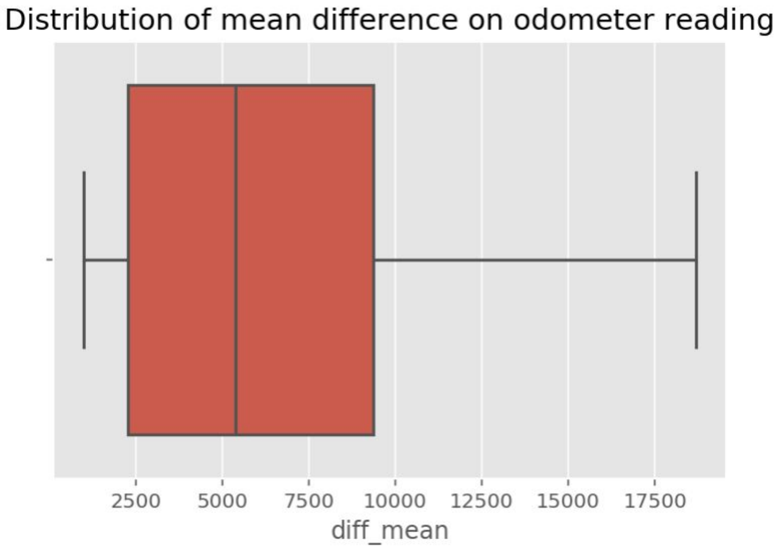
50

Defining the Target



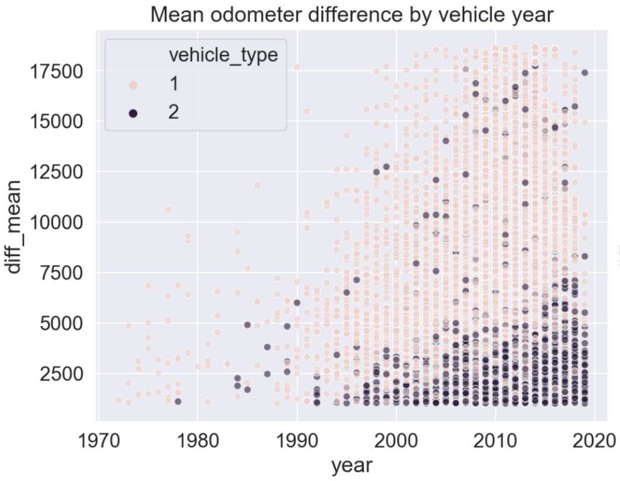
Distribution of mean difference on odometer reading

5 - 95%

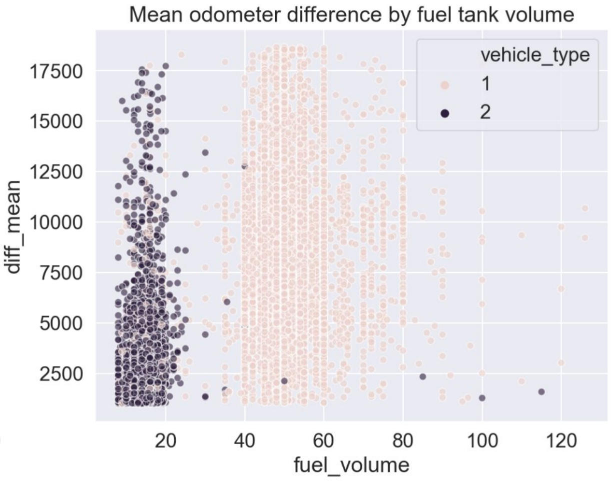


Distribution of mean difference on odometer reading

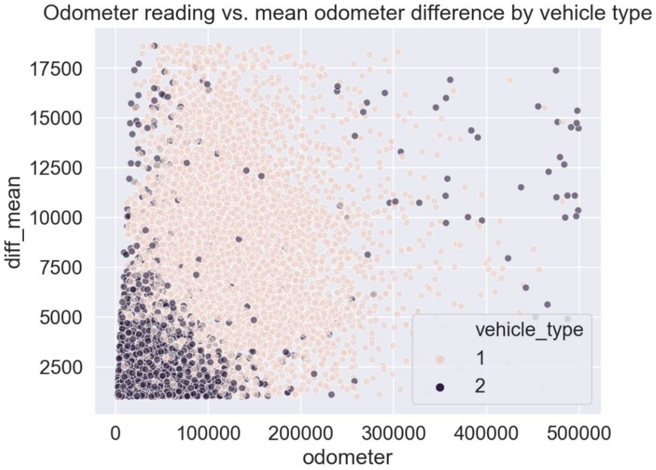
Predictors vs. Target



Mean odometer difference by vehicle year



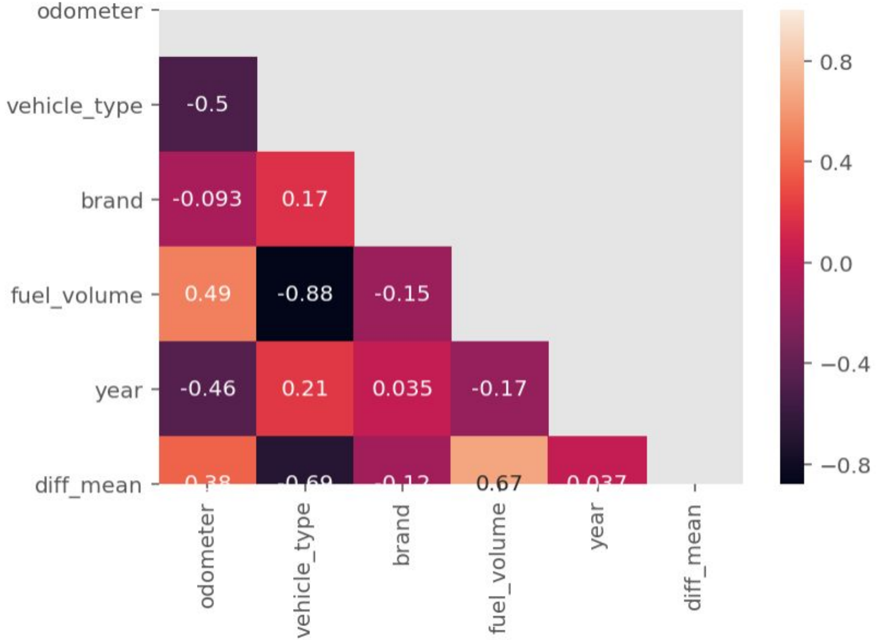
Mean odometer difference by fuel tank volume



Odometer reading vs. mean odometer difference by vehicle type

Predictors vs. Target

Correlation Heatmap

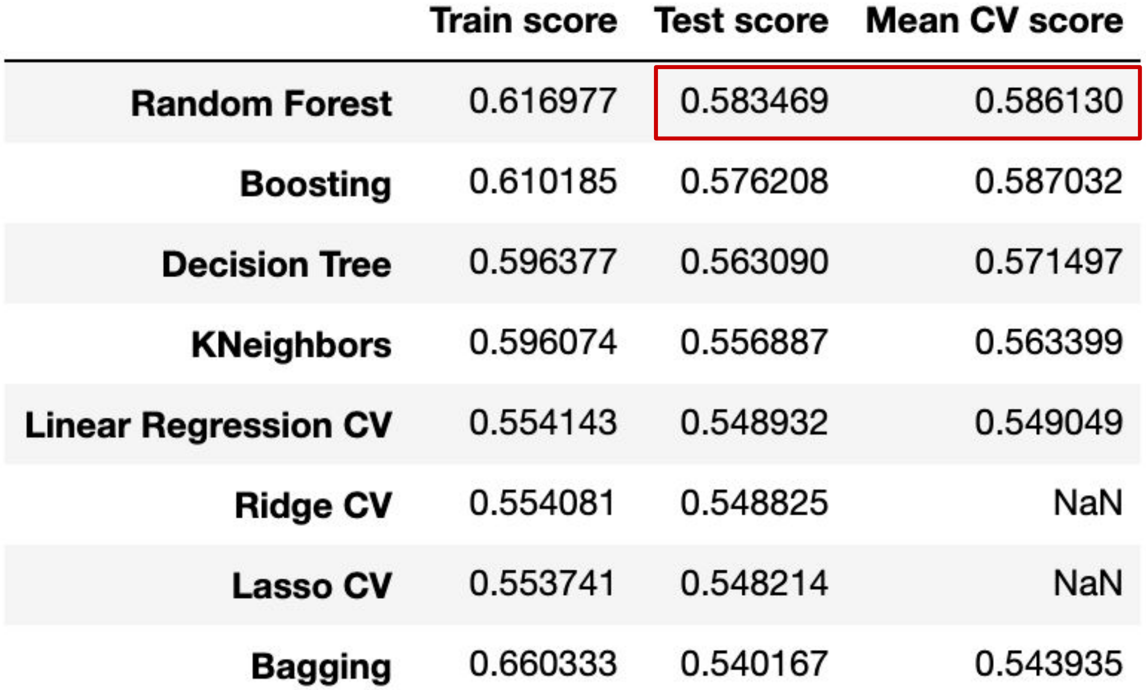


FOURTH STEP: MODELING

Regression Models

- Linear Regression
- Ridge
- Lasso
- Gradient Boosting
- K Neighbors
- Decision Tree
- Bagging
- Random Forest

Regression Models

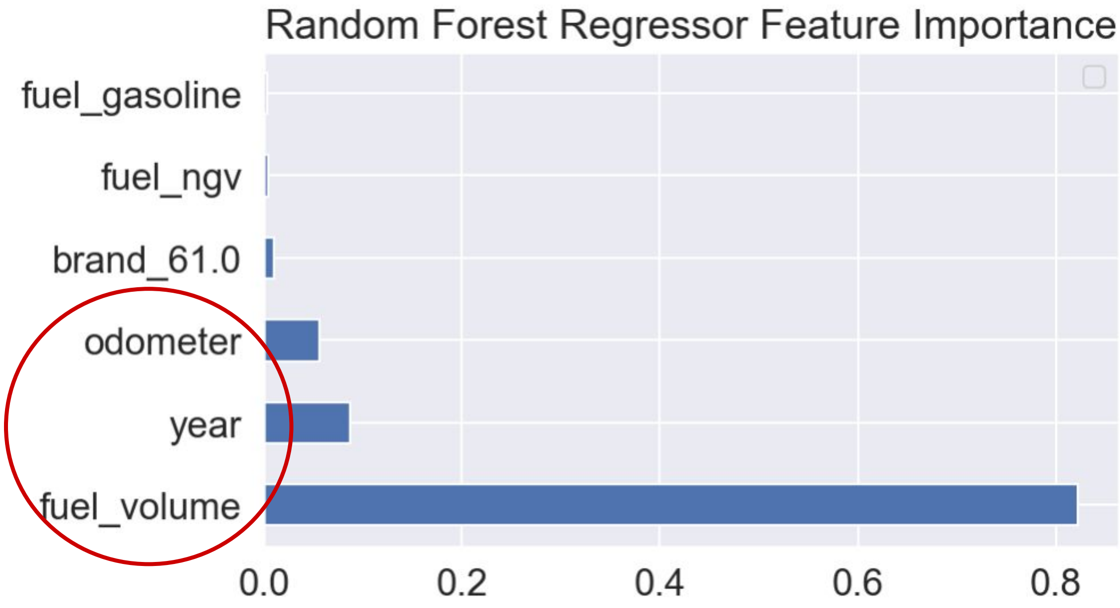


	Train score	Test score	Mean CV score
Random Forest	0.616977	0.583469	0.586130
Boosting	0.610185	0.576208	0.587032
Decision Tree	0.596377	0.563090	0.571497
KNeighbors	0.596074	0.556887	0.563399
Linear Regression CV	0.554143	0.548932	0.549049
Ridge CV	0.554081	0.548825	NaN
Lasso CV	0.553741	0.548214	NaN
Bagging	0.660333	0.540167	0.543935

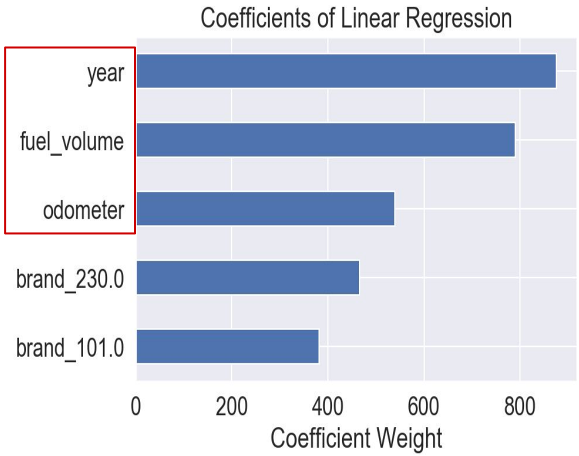
around 58% of the variation on target is explained by the independent variables

Good at generalizing

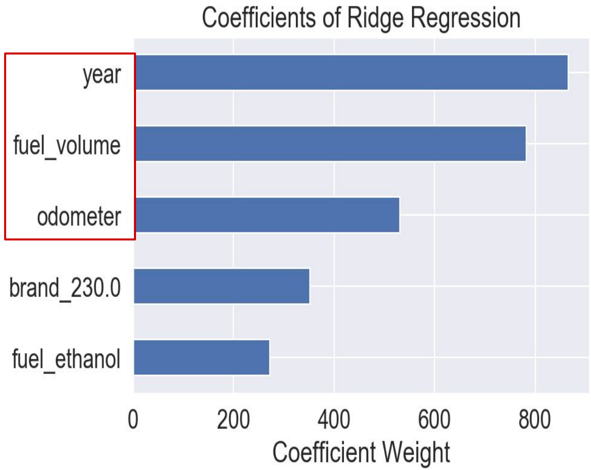
Regression Models



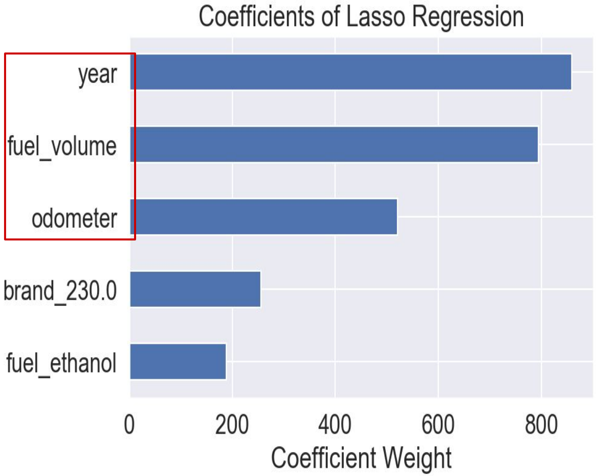
Regression Models



Test score:
0.548932



Test score:
0.548825

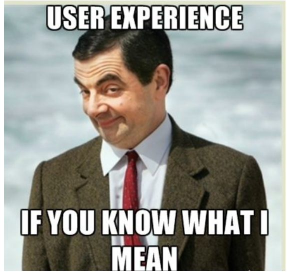


Test score:
0.548214

NEXT STEPS

Next Steps

- Investigate further the role of the features on vehicle performance
- Expand the model for other countries
- Create model for different services
- Improve the user interaction inside the app (and hopefully get better data)



Thank you