

Predviđanje održavanja viličara

1. Uvod
2. CRISP - DM metodologija
3. Razumijevanje i definiranje cilja
4. Razumijevanje podataka
5. Priprema podataka
6. Modeliranje
7. Evaluacija
8. Završna riječ

1. Uvod

Tema ovogodišnjeg LUMEN data science natjecanja je predviđanje održavanja strojeva, odnosno korištenje tehnike **Predictive maintenance**. Predictive maintenance je tehnika koja nam pomaže dijagnosticirati stanja u kojima se nalaze naši strojevi kako bi se odredilo vrijeme kada je stroju potrebno održavanje. Strojevi o kojima govorimo su viličari te je naš zadatak stvoriti model kojim ćemo moći predvidjeti kvar i pravovremeno javiti da treba reagirati na održavanju tog viličara. Tijekom projekta služili smo se CRISP - DM metodologijom o čemu ćemo uskoro detaljnije reći te ćemo Vam kroz ovu dokumentaciju približiti naše razmišljanje i put ka konačnom rješenju odnosno modelu predviđanja. Programski jezik kojim smo se služili je Python, neke prikaze podataka i predviđanja radili smo koristeći Octave te R. Dokumentacija je pisana u Jupyter bilježnici kako bi na pregledniji način saželi sve o našem konačnom rješenju. Uz ovu dokumentaciju prilažemo i tehničku dokumentaciju u kojoj možete proučiti kako koristiti napisano rješenje.

2. CRISP - DM metodologija

Jedna od stvari na koju smo se trebali referirati u svom rješenju bila je CRISP - DM metodologija. **CRISP - DM metodologija** (*Cross Industry Standard Process for Data Mining*) jedna je od najpopularnijih metodologija za manipuliranje i rad s podacima pri rješavanju problema iz data sciencea. Ona jasno opisuje pristup rješavanju problema i daje "upute" na što se fokusirati u kojem dijelu. Na taj je način pristupanje samom problemu lakše, a predstavljanje rješenja preglednije i razumljivije. Možemo reći da je ova metodologija zapravo iterativni proces, jer se na neke korake potrebno vraćati više puta ili ih ponavljati kako bi krajnji rezultat bio bolji (Shema 1). Koraci kroz koje prolazimo su:

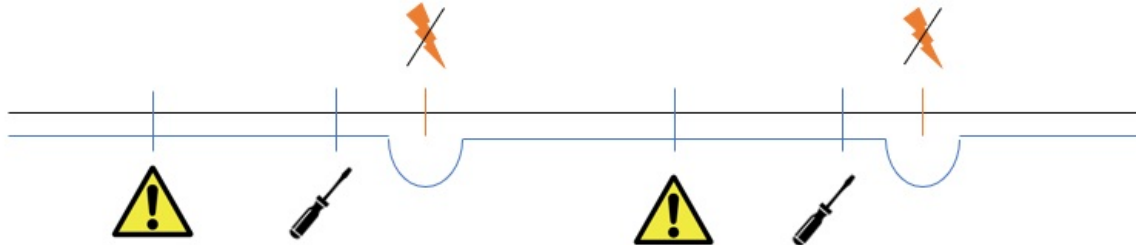
1. Razumijevanje poslovanja/ problema
2. Razumijevanje podataka
3. Priprema podataka
4. Modeliranje
5. Vrednovanje (Evaluacija)
6. Korištenje

Nadalje, dokumentacija našeg rješenja slijedit će korake ove metodologije radi preglednosti, ali i zbog toga što smo upravo koristeći se njome došli do našeg rješenja.



3. Razumijevanje i definiranje cilja

Razumijevanje problema, to jest definiranje i razumijevanje cilja jedan je od najvažnijih koraka u ovoj metodologiji. Pogrešno shvaćanje problema ili cilja može dovesti do neuspjeha u rješavanju, stoga smo ovom koraku pristupili oprezno i detaljno prošli kroz upute i informacije koje smo dobili. Strojevi se vremenom postepeno troše i dolazi do zamora opreme, no na temelju podataka koje imamo o njima možemo predvidjeti mogući kvar i tada pripremljeno reagirati. Pravovremeno reagiranje smanjuje mogućnost dodatnih troškova koji su mogući zbog čekanja dijelova koje je potrebno zamijeniti na stroju ili čekanju stručnog osoblja za održavanje te samo trajanje popravka. Naš je cilj, na osnovi dobivenih podataka, nagovijestiti kvar i spriječiti dodatne troškove.



Shema 2. Grafički prikaz održavanja strojeva prije kvara

4. Razumijevanje podataka

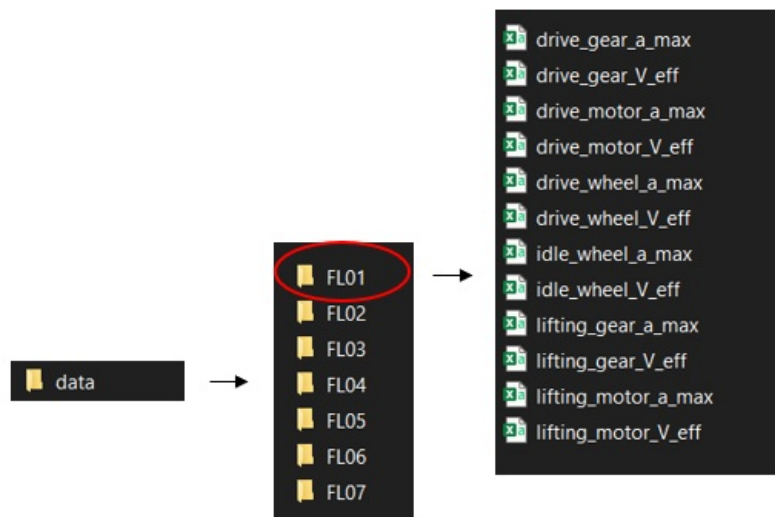
Nakon postavljanja ciljeva i dubljeg shvaćanja problema potrebno je krenuti s konkretnijim poslom. Sljedeći korak u metodologiji kojom se služimo je prikupljanje podataka, njihovo razumijevanje i analiza. Jedan dio ovog koraka je bio odrađen umjesto nas, što smo također morali uzeti u obzir. Naime, u trećem je koraku bitno odrediti koji podaci su nam važni za dobivanje informacija te ih na pravilan način prikupiti, a kasnije onda na njima obavljati analizu. Mi nismo morali mjeriti podatke i tražiti koji su senzori dovoljni za uvid u stanje stroja, već su to organizatori odradili umjesto nas. Posljedica toga je da za dva stroja imamo nešto više podataka i relevantne senzore koji su dovoljni pokazatelji kvarova. U ovom smo koraku odlučili posvetiti malo više vremena proučavanju literature kako bi istražili sve potrebne pojmove vezane uz problem kojim se bavimo i podatke kojima raspolažemo. Naš prvi korak u ovom tijeku projekta bio je učiniti podatke koje imamo pristupačnijim nama.

4.1. Kod za učitavanje podataka

Prvi programerski korak u rješavanju bio je podijeliti veliku datoteku sa svim sensorima i strojevima u manje datoteke za lakši rad s njima. Napisali smo funkciju pomoću koje stvaramo mapu za svaki od stroja, zatim u svakoj mapi datoteku za svaki od senzora. Na kraju postupka imamo podatke posebno za 7 strojeva za svih 6 senzora s vrijednostima brzine i akceleracije.

In [1]:

```
import dataset_loader as dl
dl.split_into_attr_tree("data_full", "data", ["machine_name", "sensor_type"])
#krajnji rezultat izgleda kao na slici
#za više informacija o funkciji pogledati python/dataset_loader.py
```



Slika 1. Prikaz mapa i datoteka nakon poziva funkcije

Nakon što smo na ovaj način podjelili originalnu datoteku lakše je prikazivati podatke i raditi analizu. Na primjeru jednog stroja i senzora prikazat ćemo strukturu podataka. Iz datoteka smo primjetili da se mjerenja ponavljaju tijekom dana, mjerenja ne traju svugdje jednako te da mjerenja na senzorima nisu usklađena. Ovo će nam biti bitno pri filtriranju podataka i korištenju određenih alata za korelacije.

In [2]:

```
import pandas as pd
data = "data/FL01/drive_motor_V_eff.csv"
inf = pd.read_csv(data, sep = ';' , header = 0)
inf
```

Out[2]:

	machine_name	sensor_type	date_measurement	start_timestamp	end_timestamp	realvalue	unit
0	FL01	drive_motor_V_eff	2017-09-02	2017-09-02 15:26:42.12	2017-09-02 15:26:42.12	0.565	mm/s
1	FL01	drive_motor_V_eff	2017-09-02	2017-09-02 15:26:44.933	2017-09-02 15:26:44.933	1.023	mm/s
2	FL01	drive_motor_V_eff	2017-09-02	2017-09-02 15:26:47.763	2017-09-02 15:26:47.763	1.447	mm/s
3	FL01	drive_motor_V_eff	2017-09-02	2017-09-02 15:26:50.59	2017-09-02 15:26:50.59	1.736	mm/s
4	FL01	drive_motor_V_eff	2017-09-02	2017-09-02 15:26:53.403	2017-09-02 15:26:53.403	2.233	mm/s
...
33935	FL01	drive_motor_V_eff	2019-07-24	2019-07-24 22:20:19.287	2019-07-24 22:20:19.287	2.817	mm/s
33936	FL01	drive_motor_V_eff	2019-07-24	2019-07-24 22:20:20.727	2019-07-24 22:20:20.727	2.825	mm/s
33937	FL01	drive_motor_V_eff	2019-07-24	2019-07-24 22:20:22.09	2019-07-24 22:20:22.09	2.835	mm/s
33938	FL01	drive_motor_V_eff	2019-07-24	2019-07-24 22:20:23.5	2019-07-24 22:20:23.5	2.809	mm/s
33939	FL01	drive_motor_V_eff	2019-07-24	2019-07-24 22:20:24.91	2019-07-24 22:20:24.91	2.728	mm/s

33940 rows × 7 columns

4.2. Istraživanje podataka

Kako su podaci koje smo dobili tehničke prirode, odlučili smo malo više istražiti o tehničkim svojstvima koji su nam potrebni. Analiza vibracija je proces koji služi za mjerenje razine vibracija i frekvencije stroja te pomoću tih podataka možemo analizirati stanje stroja. Upoznali smo se s problemima koje analiza vibracija može detektirati, logičkim cjelinama od kojih se sastoji i kategorijama mjerenja vibracija.

Prilikom istraživanja naišli smo na standarde koji su vezani uz brzinu. Primjetili smo kojoj vrsti strojeva pripadaju naši viličari te ćemo podatke za tu kategoriju kasnije koristiti. Više o ovom standardu i detalji slijede u djelu prikaza rješenja.

VIBRATION SEVERITY PER ISO 10816						
Machine			Class I small machines	Class II medium machines	Class III large rigid foundation	Class IV large soft foundation
Vibration Velocity Vrms	in/s	mm/s				
	0.01	0.28				
	0.02	0.45				
	0.03	0.71		good		
	0.04	1.12				
	0.07	1.80				
	0.11	2.80		satisfactory		
	0.18	4.50				
	0.28	7.10		unsatisfactory		
	0.44	11.2				
	0.70	18.0				
	0.71	28.0		unacceptable		
	1.10	45.0				

Slika 2. Tablica klasifikacije stanja strojeva prema vrijednostima brzine

4.3. Analiza podataka

U završnom dijelu drugog koraka odlučili smo se grafički prikazati podatke, probati pronaći koeficijent korelacije među sensorima i filtrirati podatke po traženim svojstvima. Slušajući savjete, odlučili smo dosta vremena utrošiti na prikaz naših podataka, tražeći u njima informacije koje bi nam mogle biti od koristi u daljnjem rješavanju problema. U ovom smo se koraku poslužili znanjima iz statistike(karakteristična petorka, koeficijent korelacije, outlieri, histogrami, distribucije), programiranja u R i Octavu te Pythonu. Na ovom ćemo mjestu spomenuti detalj koji se lako uoči u podacima, a bitan je za daljnje algoritme. Naime, nakon održavanja primijetili smo nagli skok u podacima, no o ovoj pojavi reći ćemo još nešto kasnije. Ukratko ćemo prikazati zanimljive grafove, rezultate i dijelove koda koje smatramo najkorisnijima.

Coefficient	machine	sensor1	sensor2
0.851	FL01	drive_wheel_V_eff	idle_wheel_V_eff
0.859	FL07	drive_gear_V_eff	drive_motor_V_eff
0.877	FL01	drive_gear_V_eff	idle_wheel_V_eff
0.893	FL03	drive_motor_V_eff	drive_motor_a_max
0.895	FL01	lifting_gear_a_max	lifting_motor_a_max
0.908	FL01	drive_gear_V_eff	drive_motor_V_eff
0.912	FL05	drive_gear_V_eff	drive_motor_V_eff
0.921	FL05	drive_wheel_V_eff	drive_wheel_a_max
0.977	FL01	lifting_gear_V_eff	lifting_motor_V_eff
0.978	FL02	drive_wheel_V_eff	idle_wheel_V_eff

Tablica 1. Prikaz senzora s najvećom korelacijom

Prikazati graf *rolling meana* za stroj FL03.

In [1]:

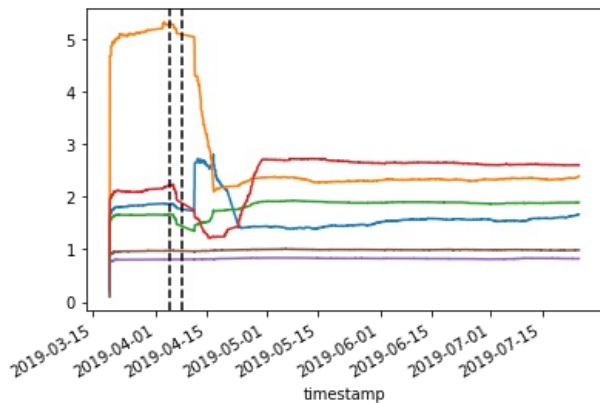
```
import warnings
warnings.simplefilter('ignore')
##Stavljeno samo kako warning outputi nebi smetali pri gledanju grafova
```

In [2]:

```
%matplotlib inline
import visual_test as v
import data_filter as filter
import matplotlib
matplotlib.use('module://ipykernel.pylab.backend_inline')
v.Plot(machine="FL03",sensors=filter.list_V_sensors,feature = "rol-mean" , legend = False)
```

Plotting sensors...

Loading from data/FL03/drive_gear_V_eff.csv
data/FL03/drive_gear_V_eff.csv loaded successfully
FL03 - drive_gear_V_eff
Loading from data/FL03/drive_motor_V_eff.csv
data/FL03/drive_motor_V_eff.csv loaded successfully
FL03 - drive_motor_V_eff
Loading from data/FL03/drive_wheel_V_eff.csv
data/FL03/drive_wheel_V_eff.csv loaded successfully
FL03 - drive_wheel_V_eff
Loading from data/FL03/idle_wheel_V_eff.csv
data/FL03/idle_wheel_V_eff.csv loaded successfully
FL03 - idle_wheel_V_eff
Loading from data/FL03/lifting_gear_V_eff.csv
data/FL03/lifting_gear_V_eff.csv loaded successfully
FL03 - lifting_gear_V_eff
Loading from data/FL03/lifting_motor_V_eff.csv
data/FL03/lifting_motor_V_eff.csv loaded successfully
FL03 - lifting_motor_V_eff



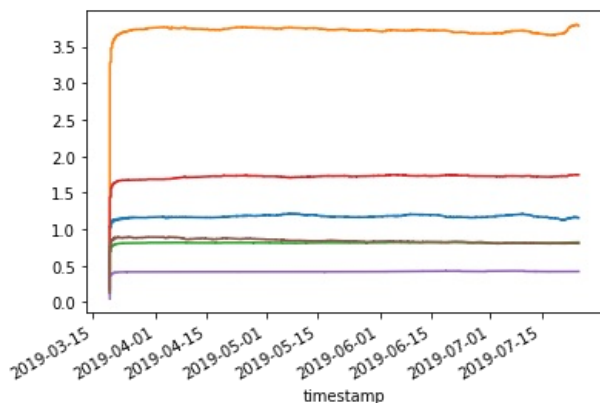
Na ovom primjeru želimo pokazati stabilnost podataka na stroju broj četiri.

In [12]:

```
v.Plot(machine = "FL04", sensors = filter.list_V_sensors, feature = "rol-mean", legend = False)
```

Plotting sensors...

FL04 - drive_gear_V_eff
FL04 - drive_motor_V_eff
FL04 - drive_wheel_V_eff
FL04 - idle_wheel_V_eff
FL04 - lifting_gear_V_eff
FL04 - lifting_motor_V_eff



<Figure size 432x288 with 0 Axes>

<Figure size 432x288 with 0 Axes>

Na stroju FL01 primjećujemo povećanje vrijednosti samo kod jednog senzora

In [15]:

```
v.Plot(machine = "FL01", sensors = filter.list_V_sensors, feature = "rol-mean", legend = False)
```

Plotting sensors...

FL01 - drive_gear_V_eff

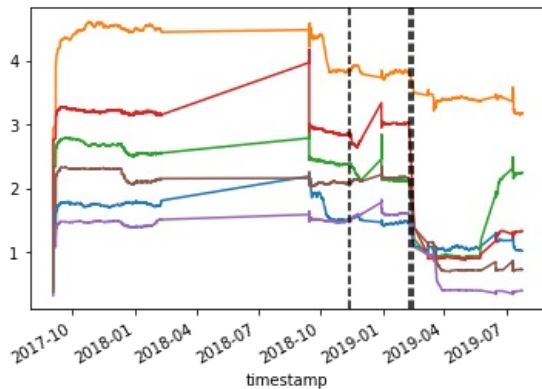
FL01 - drive_motor_V_eff

FL01 - drive_wheel_V_eff

FL01 - idle_wheel_V_eff

FL01 - lifting_gear_V_eff

FL01 - lifting_motor_V_eff



<Figure size 432x288 with 0 Axes>

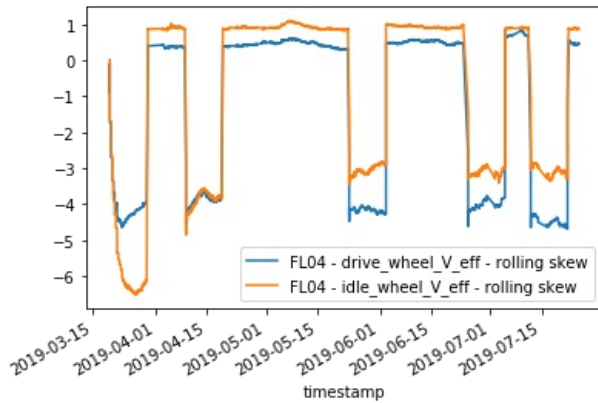
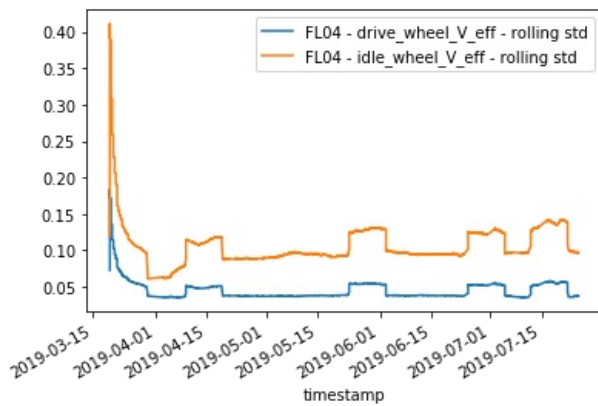
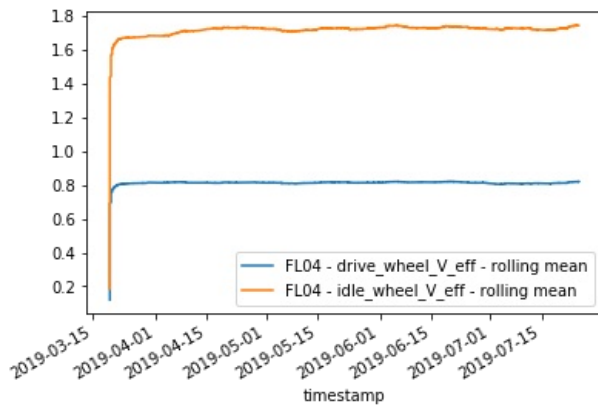
<Figure size 432x288 with 0 Axes>

Na "lijepim" podacima stroja FL04 analizirajmo jos neke *feature*, a ovdje prikazujemo *rolling standard deviation* i *rolling skewness*.

In [8]:

```
v.Plot(machine="FL04",sensors=["drive_wheel_V_eff","idle_wheel_V_eff"], feature="rol-mean")
v.Plot(machine="FL04",sensors=["drive_wheel_V_eff","idle_wheel_V_eff"],feature = "rol-std")
v.Plot(machine="FL04",sensors=["drive_wheel_V_eff","idle_wheel_V_eff"],feature = "rol-skew")
```

```
Plotting sensors...
FL04 - drive_wheel_V_eff
FL04 - idle_wheel_V_eff
Plotting sensors...
FL04 - drive_wheel_V_eff
FL04 - idle_wheel_V_eff
Plotting sensors...
FL04 - drive_wheel_V_eff
FL04 - idle_wheel_V_eff
```



5. Priprema podataka

Sljedeći korak koji prethodi "glavnom djelu" rješavanja je priprema podataka. Naši podaci neće uvijek biti u obliku u kojem nam je pogodno koristiti algoritme na njima, primjerice neki će algoritmi bolje raditi na standardiziranim podacima, stoga je veoma važno pripremiti odgovarajući oblik podataka kako bi se pri treniranju fokusirali samo na algoritam. U ovom dijelu također bi se trebalo fokusirati na nadoknađivanje vrijednosti ukoliko izostaju.

Jedan od prvih izazova s kojima smo se susreli je bio usko vezan uz vremenske intervale i vremenom mjerenja podataka koji nije usklađen kod svih strojeva niti među senzorima. Tom smo problemu pristupili na razne načine. Jedan isprobani način usklađivanja bio je svrstavanje mjerenja u blokove, na način da dva mjerenja spadaju u isti vremenski blok ako je razlika među njima dovoljno mala. Kako bi uskladili vremena mjerenja na svim senzorima krenemo od zadnjeg bloka i trebamo odbacivati sve one blokove koji imaju prazan presjek s ostalim blokovima po senzorima. Broj odbačenih intervala bio je prevelik pa smo zaključili da je ovaj algoritam neučinkovit pa da se usklađivanje mjerenja ne može obaviti na svim senzorima odjednom. Ipak, ovakav način usklađivanja funkcionira za parove senzora te smo na temelju ovog algoritma tražili korelaciju među dva senzora i na njemu radili *Linear regression*.

6. Modeliranje

Nakon nekog vremena provedenog u pripremi podataka, traženju bitnih svojstava i analizi danih podataka, došlo je vrijeme za iskoristiti sve što smo pripremili. U ovom koraku htjeli smo upotrijebiti neki od algoritama strojnog učenja koji bi nam pomogao pomaknuti se bliže cilju - predviđanju kvara, te davanja određene ocjene kvalitete rada stroja. Na taj bismo način omogućili pravovremeno održavanje. Razmijenili smo razne ideje, neke od njih pretvorili smo u pokušaje radeći prototipove, no neke od ideja morali smo u konačnici odbaciti, a one za koje smo procijenili da su najbitnije za ovaj zadatak, implementirali smo u konačnom rješenju. U dokumentaciji nećemo detaljno ulaziti u svaki od tih pokušaja koje smo imali, ali su svakako vrijedni spomena. Također, niti jedan od tih pokušaja nismo naglo odbacili te su nam oni poslužili kao povratna informacija koja nam je dala neke smjernice za daljnje oblikovanje konačnog rješenja.

Jedan od prvih algoritama koji smo odlučili isprobati jest model Linearne regresije za predviđanje jačine mjerenja na jednom senzoru, pomoću poznatih mjerenja na drugom senzoru. Iako nam je dao određenu slutnju o povezanosti cijeloga stroja, ovaj model u konačnici se ne nalazi u našem rješenju jer nije bio od većeg značaja za krajnjeg korisnika.

1. pokušaj: Linearni regresijski model

U ovome koraku pokušali smo odrediti postoji li korelacija među mjerenjima obavljenima na jednom stroju, ali na različitim senzorima. Konkretno, recimo da promatramo stroj FL01, te dva senzora na njemu (neka su to A i B radi jednostavnijih imena). Ako znamo da je senzor A pokazivao u 12:00:00 mjerenje jačine 2mm/s, htjeli bismo odrediti koliko je u tom trenutku bilo mjerenje na senzoru B. Prvi problem oko ovog algoritma bio je taj što za senzore A i B nismo gotovo nikad dobili mjerenja u točno istom trenutku, pa smo morali napraviti određeno vremensko usklađivanje. Nakon što smo uskladili mjerenja koja su vremenski relativno blizu na senzoru A i B pokrenuli smo linearnu regresiju. Rezultat tog algoritma dao je određeni model po kojem su mjerenja na nekim parovima senzora zaista međusobno povezana. Glavni razlog zašto se ovaj algoritam ne nalazi u konačnom rješenju jest taj što on ne rješava problem treba li se na stroju obaviti popravak ili ne. Također, u većini vremenskih intervala ipak postoje mjerenja na svim senzorima pa onda nije potrebno predviđati nešto što već znamo.

2. pokušaj: Detekcija anomalija

I dalje je ostalo ključno pitanje - Je li stroj zdrav ili ne? Hoće li uskoro biti potrebno odraditi popravak? Odgovore na ova pitanja pokušali smo odrediti primjenom detekcije anomalija.

Motivacija: analiziranjem podataka odredili bismo neke vremenske intervale za koje smatramo da je stroj u njima dobro radio. U tom povoljnom vremenskom razdoblju izmjerena su mjerenja koja ćemo zvati referentni podaci. Ako nastavimo promatrati mjerenja nakon tog povoljnog razdoblja, dobit ćemo neke nove podatke. Postoje dva scenarija:

- a) novi podaci veoma su slični referentnim podacima
- b) novi podaci odskaču i razlikuju se od referentnih

U prvom scenariju možemo s opravdanjem reći da stroj i dalje radi dosta dobro. U drugom scenariju potrebna je detaljnija analiza zašto su se podaci počeli mijenjati i kakav je trend promjene. Naime, ako su se mjerenja iznenada počela povećavati to može biti znak da će stroju uskoro biti potreban popravak.

Konačno rješenje

Nakon pokušaja i "pogreški" dolazimo do algoritma koji smo spremni predati Vam kao naše rješenje. Spomenuli smo već kako smo istraživanjem teme naišli na klasifikaciju strojeva u odnosu na brzine.(Slika 2.) To nam je dalo ideju da se probamo fokusirati na taj dio i vidjeti što s time dobivamo.

Naš program se bazira na mogućnosti tvrtke da unese podatke mjerenja, te uz pomoć jednostavnog korisničkog sučelja implementiranog u Pythonu navodi korisnika kako da pokrene dijagnozu podataka koje je sakupio.

Dijagnoza se sastoji od dva dijela:

1. Klasifikacija mjerenja

Motivacija: Za vrijeme rada stroja njegovi dijelovi vibriraju, a jačinu tih vibracija mjere senzori. Pod jačinom vibracija spadaju dvije fizikalne veličine: brzina kojom senzor titra izražena u mm/s te akceleracija izražena u mg. Ako su te vrijednosti suviše visoke, stroj je u potencijalnoj opasnosti od kvara, te je najvjerojatnije potrebno odraditi popravak na stroju.

Kako bismo odredili što za stroj znači "suviše jaka vibracija" potražili smo u literaturi vezanoj uz analizu vibracija. Tako smo pronašli ISO 10816 standard (Slika 2.) te pomoću njega odredili intervale brzina u kojima senzori mogu vibrirati. Na taj način svako mjerenje možemo ocijeniti nekom od kategorija:

- Good
- Satisfactory
- Unsatisfactory
- Unacceptable

Koristili smo podatke za Strojve Klase III - velikih strojeva budući da naši viličari spadaju u tu kategoriju. Identičnu stvar napravili smo i za mjerenja akceleracije, međutim kategorizacija akceleracija naša je procjena na temelju istraživanja i praćenja mjerenja na senzorima iz podataka koje smo dobili. Nismo uspjeli pronaći neke službene referentne podatke za akceleraciju, pa je za točnost i kvalitetu ove kategorizacije potrebno konzultirati se sa stručnjakom. Upravo iz tog razloga, svaku od tih kategorija mjerenja korisnik tj. stručnjak može mijenjati uređivanjem .config datoteke u config folderu te tako odabrati kategorizaciju koja je najpovoljnija za određeni stroj.

2. Provjera kompatibilnosti

Eksperimentalno je utvrđeno da se gotovo bilo koji vremenski interval iz kojeg se uzmu mjerenja ustvari ponaša više ili manje kao Gaussova normalna raspodjela. To je direktna potvrda smislenosti algoritma o detekciji anomalija. U ovom dijelu programa tako korisnik može odrediti vremenski interval u kojem se nalaze najnovije izmjerene podatke te pronaći sličnosti između tih podataka sa onima koja su prethodila tom vremenskom intervalu. Taj "prethodni" vremenski interval ovdje nazivamo "referentni podaci", a nova mjerenja nazivamo "novi podaci". Ako novi podaci ne upadaju u referentne podatke, stroj se u novije vrijeme počeo ponašati drugačije. Takvo ponašanje potencijalno može biti dobro, na primjer ako se to počelo događati nakon popravka. Međutim, može voditi i k pogubnom scenariju za stroj ako su se mjerenja počela iznenada povećavati. Zato, ako je došlo do nekakve promjene, odnosno novi podaci ne sliče onim referentnima, potrebno je konzultirati se sa stručnjakom u analizi vibracija kako bi on donio stručno mišljenje je li stroju potrebno održavanje ili ne.

Rezultat našeg programa jest dijagnoza rada stroja koja je spremljena u .pdf datoteci. Ona se sastoji od različitih komponenti koje uključuju ispis te grafove prikaza mjerenja sa njihovim klasifikacijama, grafove distribucije te sličnost novih podataka sa referentnima.

Estimatori

Estimator je glavni dio ovog softverskog rješenja te su oni odgovorni za sve postavke pod kojima se dijagnoza odvija. Tako se u Estimatoru mogu mijenjati sve postavke koje korisnik želi da software dijagnosticira:

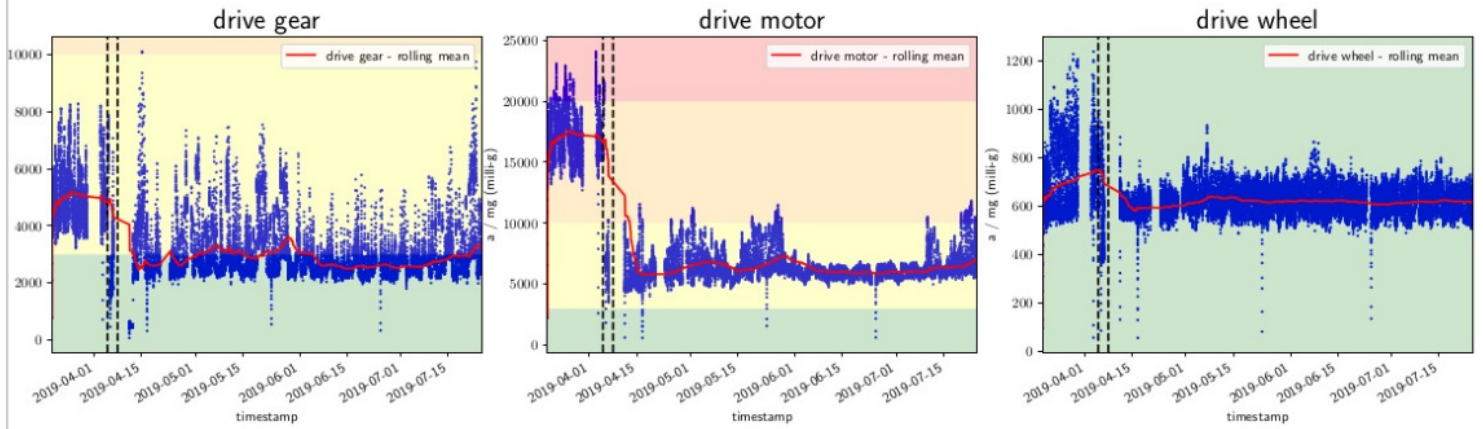
- Koje podatke software treba klasificirati:
 - Senzore koji mjere brzinu
 - Senzore koji mjere akceleraciju
 - Svi podaci
 - Samo novi podaci
- Usporedba sličnosti novih podataka sa:
 - Zadnjih tjedan dana
 - Zadnjih n dana
 - Preporučenim referentnim vrijednostima

Detaljnije o podešavanju rada Estimatora nalazi se u tehničkoj dokumentaciji.

Nakon pokretanja software-a, dijagnostika i analiza podataka sprema se kao rezultat u ".pdf" file.

Primjeri dijagnostike

Acceleration sensors all data



Good: 20027/30390 = 66%
Satisfactory: 10361/30390 = 34%
Unsatisfactory: 2/30390 = 0%
Unacceptable: 0/30390 = 0%

Unsatisfactory

Good: 22/30395 = 0%
Satisfactory: 26355/30395 = 87%
Unsatisfactory: 3637/30395 = 12%
Unacceptable: 381/30395 = 1%

Unacceptable

Good: 30588/30588 = 100%
Satisfactory: 0/30588 = 0%
Unsatisfactory: 0/30588 = 0%
Unacceptable: 0/30588 = 0%

Good

drive gear

$$\mu_{ref} = 1.56$$

$$\sigma_{ref} = 0.17$$

$$\sigma_{ref}^2 = 0.03$$

$$\mu_{new} = 1.6$$

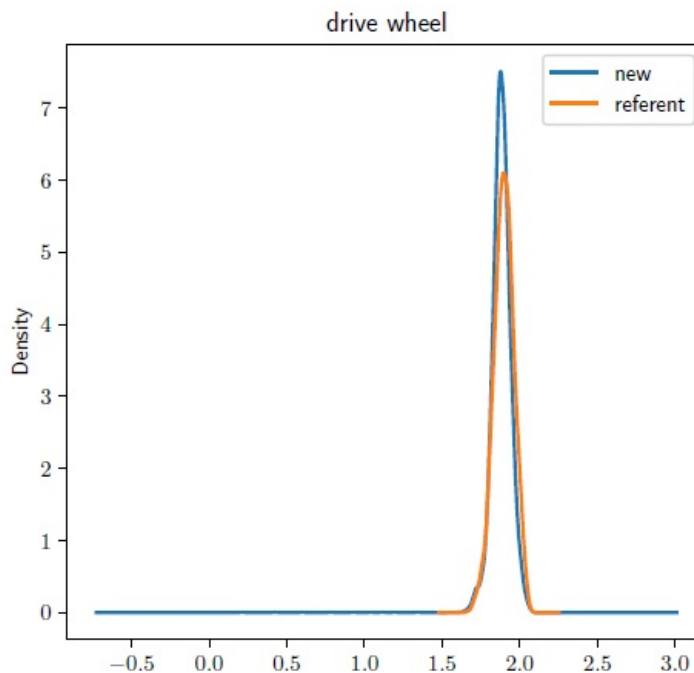
$$\sigma_{new} = 0.18$$

$$\sigma_{new}^2 = 0.03$$

$$good_{cnt}/all_{cnt}$$

$$5886 / 5947 = 99\%$$

GOOD FIT



Objašnjenje sadržaja dijagnostike:

Kategorije	Opis
Diagnosis for <i>machine name</i>	naslovna stranica, dijagnoza je obavljena za stroj pod nazivom <i>machine name</i>
Categorization of measurements [sensor type] - Time interval: [time interval]	U ovom odjeljku nalaze se klasifikacije mjerenja po sensorima. Za svaki od senzora koji pripada sensorima iz [sensor type] prikazuju se podaci iz vremenskog intervala [time interval]. Podaci obuhvaćaju graf svih mjerenja iz tog vremenskog intervala, te njihov rolling mean sa vremenskim "window-om" od 10 dana. Ispod grafa nalazi ispis raspodjele mjerenja po kategorijama. Za svaku od kategorija piše koliko mjerenja je klasificirano u tu kategoriju kroz ukupan broj mjerenja te postotak koliko to iznosi. Konačna ocjena senzora je onakva kako je najgore moguće mjerenje klasificirano.
Compatibility check for [sensor type] - new data: [new data] - referent data: [ref data]	U ovom odjeljku nalazi se provjera kompatibilnosti novih podataka sa onim referentnima. Novi podaci (new data) su oni podaci koje korisnik bira u User Interface-u kada ga se pita "Select time interval for analysis (new data)". Referentni podaci (Referent data) ovise o tome kakav se tip usporedbe kompatibilnosti analizira. Te postavke su specificirane u Estimator-u, a mogu se uređivati u .config fileu, što je objašnjeno u tehničkoj dokumentaciji. Jedan od primjera je da su referentni podaci svi oni koji su se događali u roku tjedan dana prije početka intervala novih podataka. Drugi je primjer referentni podaci koji su ručno unešeni, a trebali bi predstavljati optimalan način rada nekog stroja. Ako je mjerenje iz novih podataka unutar raspona u kojem se nalazi 99.7% mjerenja iz referentnih podataka, onda je to mjerenje dobro. U konačnici se prikazuje broj dobrih mjerenja kroz ukupan broj mjerenja, a kompatibilnost se ocjenjuje sa: "Good fit" ako je više od 90% dobrih mjerenja, "Partial fit" ako je više od 50% dobrih mjerenja, "Bad fit" ako je manje od 50% dobrih mjerenja.

- [machine name] je iz skupa {"FL01", "FL02", "FL03", "FL04", "FL05", "FL06", "FL07"}
- [sensor type] je iz skupa {"acceleration sensors", "velocity sensors"}
- [time interval] označava u kojem vremenskom intervalu su analizirani trenutni podaci.
 - Primjer 1: "all data" označava da je analiziran cijeli skup mjerenja koji postoji za taj senzor.
 - Primjer 2: "from 2019-06-20 to 2019-07-25" označava da su analizirana sva mjerenja od 20. lipnja sve do 25. srpnja.
- [new data] označava podatke za koje se pokušava koliko dobro pripadaju distribuciji referentnih podataka.
 - Primjer: "from 2019-06-20 to 2019-07-25" označava da su analizirana sva mjerenja od 20. lipnja sve do 25. srpnja.
- [ref data] označava podatke čija se distribucija smatra referentnom, tj. onom s kojom se new_data uspoređuje. Ako je [ref data] vremenski interval, kao u primjeru gore, onda se za taj vremenski interval odredi Gaussova distribucija (srednja vrijednost, standardna devijacija i varijanca) te se new_data smatra dobrim ako spada u tu distribuciju, odnosno u interval koji je najviše za $3 \times (\text{std. devijacija})$ udaljen od srednje vrijednosti -u taj interval spada 99.7% referentnih podataka. Ako je [ref data] = "recommended distribution (from .config)", onda postoji ručno određena Gaussova distribucija zapisana u .config fileu. Konkretno, ovo bi trebala biti pažljivo odabrana distribucija za svaki od strojeva i svaki od senzora posebno, a odabrana tako da pokriva povoljno stanje u kojem bi neki senzor trebao raditi. Ako u .config fileu ovi podaci nisu specificirani, onda se za njih pokazuje poruka "NA" (Not Available), ali je i dalje omogućen prikaz distribucije za [new data].

7. Evaluacija

Došli smo do kraja našeg rješavanja i bavljenja zadanom problematikom. Algoritam smo napisali i sad je vrijeme da ga i Vi testirate. Kako se u cijelom ovom radu držimo CRISP - DM metodologije, upravo je evaluacija njen završni korak. Evaluaciju rješenja možete provesti koristeći tehničku dokumentaciju koju Vam prilažemo uz rješenje i ovu dokumentaciju. Htjeli smo fokus staviti na algoritam i proučavanje problema, stoga je sustav za evaluaciju odnosno User Interface jednostavan, ali precizno definira rezultate i obrađuje zahtjeve korisnika. Produkt algoritma je PDF file u kojem se nalazi dijagnostika koju smo pokrenuli u programu. Na njemu se mogu isčitati dodatni podaci poput postotka poklapanja, grafički prikaz i statistička obilježja koja će pomoći stručnjaku donijeti zaključak. Gore smo prikazali i neke prikaze iz tog dokumenta.

8. Završna riječ - zaključak

Na kraju našeg rada, sumirali smo novostečeno znanje i prezentirali Vam ga u ovom obliku. Iako sam algoritam ne izgleda previše zahtjevno, vjerujemo da je analiziranje podataka i priprema podataka bila jednako važan dio posla, kao i sam algoritam. Bez shvaćanja podataka, te bez grafičkog prikazivanja istih, rezultati bi se doimali apstraktni i možda neshvaćeni. Vjerujemo da slika govori više od 1000 riječi, te da je priprema pola posla, što smo se doista uvjerali pri pisanju programa. Literatura koju smo pročitali o vibracijama strojeva iako veoma teorijski zahtjevnija, zbog stalne potrebe za znanjima iz tog područja koje je veoma popularno i traženo danas, navela nas je na neke činjenice koje smo mogli primijeniti na naše rješenje. Također, CRISP - DM metodologija bila je ključan dio rješavanja problema kako bi ono ostalo tijekom cijelog vremena razumljivo za izlaganje te na kraju i korištenje istog.