

Presentation script

SLIDE #1 - LUKA

Dame i gospodo, dobar dan i dobrodošli na prezentaciju rješenja predictive maintenancea. Mi smo Data Mastersi, uživajte!

Na početku, prije nego vam objasnimo srž svojeg rješenja, objasniti ćemo Vam što se to krije iza pojma Predictive Maintenancea koji se u zadnje vrijeme sve više i više spominje. Svašta se govori o njemu, neki govore da je rješenje svih problema, da je odgovor na sve. S druge strane, neki su oprezniji te ukazuju na njegove mane. No što je zapravo to? Kako bismo Vam najjednostavnije objasnili ideju, moramo Vas upoznati s našim dragim prijateljem Matom.

SLIDE #2 - DOMINIK

Mate je upravitelj skladišta u Svetoj Nedjelji. Za premiještanje paleta po skladištu koristi nekoliko automatiziranih viljuškara koji rade od jutra do sutra. Svaki prekid rada viljuškara za Matu i njegovu tvrtku predstavlja značajan trošak. Viljuškari, kao i svi ostali strojevi, zahtijevaju povremene servise. Upravo se tu javlja ideja Predictive Maintenancea koja želi riješiti problem servisnih intervala viljuškara. Neki će se možda pitati: "pa koji je to problem kod servisiranja, samo ga pošalji na servis svakih 3 mjeseca." Eh da je rješenje tako lako, nebi nikad došli do ovakve nauke o kakvoj danas govorimo.

SLIDE #3 - DOMINIK

Naime, kada pošaljemo viljuškar na servis, koji moramo zapamtiti sam za sebe predstavlja trošak vremena i novca, imamo jedan od dva slučaja. Ili smo viljuškar poslali na servis prerano ili prekasno. Ukoliko smo viljuškar poslali na servis prekasno, viljuškar se već pokvario toliko da nije popravljiv, a nedaj bože se možda već pokvario za vrijeme rada. U tom slučaju, Matina tvrtka treba kupiti novi viljuškar što je ogroman trošak. Drugi, malo vedriji, slučaj je kada viljuškar pošaljemo na servis prerano. Tada je sa viljuškarom sve u savršenom redu te zapravo servis uopće nije potreban. Iako će neki možda pomisliti da tu nema problema, te da bi tako trebali raditi, u ovom slučaju također imamo troškove. Ti troškovi su vremena učestalih servisa te njihovi povezani novčani troškovi.

SLIDE #4 - DOMINIK

Možemo zaključiti da u oba slučaja imamo nepotrebne troškove vremena i novca. Kažemo nepotrebne, jer smo zapravo mogli proći i bez njih. Kako? Odgovor na to nam daje predictive maintenance! Sad smo napokon spremni doći do srži ideje koja se krije iza tog pojma.

SLIDE #5 - DOMINIK

Predictive Maintenance želi riješiti upravo taj problem minimizirajući vremenske i novčane troškove održavanja, u ovom slučaju viljuškara, a u općenitom strojeva. Ideja se krije u senzorima koji se postavljaju na razne dijelove stroja. Na temelju podataka o vibracijama koje nam pružaju senzori možemo izraditi model koji će nam reći kada trebamo servisirati stroj u najoptimalnijim intervalima.

Sada kada je svima jasna osnovna ideja Predictive Maintenancea, ispričat ćemo vam u najvažnijim crtama kako je izgledao naš put do rješenja.

SLIDE #6 - LUKA

Većina nas smo tu studenti matematike te volimo strukturirati ono na čemu radimo na najbolji mogući način. Upravo zato smo naš put do rješenja podijelili na tri dijela: na analizu podataka, modeliranje i evaluaciju, tj. ocjenu. Svaki od tri dijela su na svoj način opsežni i važni u cijeloj priči koju možete pronaći u dokumentaciji. Tijekom ove prezentacije ćemo se držati samo najvažnijih dijelova priče.

SLIDE #7 - PETAR

Krećemo s analizom podataka. Kada smo dobili dataset zadatka i vidjeli da ima više od 400 MB (čitaj: megabajta) pao nam je mrak na oči. Tu reakciju je trebalo snimiti, nismo znali od kud da krenemo. No ne brinite, taj pri dojam nije dugo trajao, već smo se brzo primili posla. Malo po malo, dolazili smo do sve više korisnih informacija. Između ostalog, saznali smo da imamo podatke od 6 senzora sa 7 viljuškara. To su: podizni motor i zupčanik, pogonski motor, zupčanik i kotač te nepogonski kotač.

SLIDE #8 - PETAR

Jedna od interesantnih informacija koju smatramo da je zanimljivo istaknuti je raspodijela podataka koje imamo prema viljuškaru. Kao što možete vidjeti na grafu, većinu podataka smo dobili sa viljuškara 1 i 7, dok su ostali podaci relativno ravnomjerno raspoređeni po strojevima 2, 3, 4, 5 i 6. Kao što se i vi vjerojatno sami pitate, kako je moguće da imamo neproporcionalno više podataka o dva viljuškara u odnosu na ostale. Isto pitanje smo i mi imali... pa smo odlučili istraživati. Saznali smo da su viljuškari 1 i 7 prikupljali podatke skoro 2 godine, dok su ostali prikupljali podatke tek oko 4 mjeseca. Ovo zapažanje se pokazalo veoma korisnim u kasnijoj izradi modela.

SLIDE #9 - PETAR

Puno pričamo o podacima, već provodimo svakakve analize, a još ih nismo vidjeli kako izgledaju. Pa evo jednog grafa senzora viljuškara broj 1. Kao što i svi pretpostavljam sami zamjećujete, podaci su veoma šumoviti. Oscilacije se vide iz aviona. To predstavlja veliki problem za buduće modeliranje. Kako bi se riješili neželjenih oscilacija, primjenit ćemo postupak rolling mean transformacije.

SLIDE #10 - PETAR

Podaci sa primjenjenom transformacijom su prikazani plavom bojom. Vidimo da su takvi podaci puno ljepši, te samim time pogodniji za modeliranje. No kako smo točno to postigli? Rolling mean transformacijom smo na svaku vrijednost senzora postavili prosjek zadnjih nekoliko vrijednosti toga senzora.

SLIDE #11 - LUKA

Sada kada imamo podatke pogodne za modeliranje, prelazimo na nama najzanimljiviji dio: duboko učenje ili na engleskom: deep learning.

SLIDE #12 - PATRIK

Za riješiti problem Predictive Maintenancea smo odlučili izraditi umjetnu inteligenciju koja će predviđati buduće vrijednosti senzora. Kako će nam to pomoći ćemo objasniti kasnije, a sada ćemo objasniti kako smo to postigli.

Odlučili smo se za takozvan Recurrent Neural Network kojem dajemo zadnjih 30 dana vrijednosti senzora, a on nam zauzvrat predviđa koja će biti vrijednost senzora sutradan. Za one tehnički potkovanije ćemo na brzinu objasniti strukturu naše neuralne mreže. Ona se sastoji od 3 LSTM sloja između kojih se nalazi dropout slojevi te na kraju 2 gusta sloja. Za funkciju gubitka koristimo kvadrat srednje pogreške, a za optimizaciju Adam optimizer.

Dobro pitanje koje se zna postaviti je zašto smo koristili dropout slojeve između LSTM slojeva. Odgovor na to je da spriječimo umjetnu inteligenciju da napamet nauči trening podatke te da na taj način dobijemo umjetnu inteligenciju koja je odlična u generalizaciji.

SLIDE #13 - PATRIK

Za učenje smo se dvoumili između dva načina. Kako nismo mogli odlučiti koji je bolji, isprobali smo oba pa smo na temelju rezultata odabrali onaj bolji.

Prvi način je takozvani per machine training. Izradili smo posebnu umjetnu inteligenciju za svaki senzor svakog stroja te smo svaku učili na odgovarajućim podacima.

Drugi način je zanimljiviji te se sastoji od izrade jedne umjetne inteligencije za svaki senzor. Te umjetne inteligencije smo tada istrenirali na podacima od svih viljuškara. Na taj način smo dobili umjetne inteligencije koje su odlične u generaliziranju, ali nisu toliko dobre u predviđanju za određen stroj. Zato smo tada te umjetne inteligencije multiplicirali za svaki stroj te ih daljnje trenirali na podacima pojedinih strojeva.

Dva veoma različita postupka koji na kraju rješavaju isti problem, pa ajmo vidjeti rezultate.

SLIDE #14

-

SLIDE #15 - PATRIK

Kao što možemo vidjeti iz grafova, umjetne inteligencije koje smo izradili na drugi način su za red veličina preciznije u predviđanju vrijednosti senzora u odnosu na umjetne inteligencije izrađene na prvi način. Mnogi će se pitati koji je razlog za toliku različitost u rezultatima?

Odgovor se krije u tome da su umjetne inteligencije izrađene na drugi način prvo upoznale stanje senzora na svim strojevima pa su se onda tek na kraju specijalizirale za određen stroj. Također su vidjele su puno više podataka. Upravo zato su te umjetne inteligencije puno bolje u generaliziranju, pa samim time predviđanju prije neviđenih podataka, od umjetnih inteligencija istreniranim na prvi način.

SLIDE #16 - PATRIK

Vidjevši takve rezultate, nije nam bilo teško opredijeliti se za drugi način treniranja umjetnih inteligencija.

No sad se postavlja pitanje kako će nam sve to pomoći u rješavanju prvotno postavljenog problema.

SLIDE #17 - PATRIK

Nakon dugog razmišljanja smo došli do ideje da umjetnom inteligencijom predviđamo vrijednosti senzora te uspoređujemo sa mjerenim vrijednostima. Ukoliko je pogreška predviđanja veća od nekog praga nekoliko dana zaredom, to znači da nešto nije dobro na tom viljuškaru te da ga treba poslati na servis. Postavlja se pitanje koji će to biti prag, mi predlažemo da prag bude dupla vrijednost greške pri treniranju te umjetne inteligencije.

U ovom pristupu se skriva problem toga da će stroj s vremenom mijenjati svoje vrijednosti, a i dalje raditi stabilno. Taj problem bi se riješio redovitim treniranjem umjetnih inteligencija novim podacima nekoliko puta godišnje.

SLIDE #18 - PATRIK

Kako bi što više pojednostavili implementaciju našeg rješenja, izradili smo TensorFlow Lite modele naših umjetnih inteligencija koje je veoma jednostavno postaviti na raznorazne IoT i mobilne uređaje kakvi se, pretpostavljamo, nalaze na viljuškarima. Ukoliko takvi uređaji na viljuškarima i ne postoje, vrlo ih je jednostavno postaviti zbog njihove male veličine i zanemarive potrošnje struje.

SLIDE #19

-

SLIDE #20 - LUKA

Za kraj ćemo vas ostaviti sa jednim citatom iz, pretpostavljam svima nam drage serije Teorija velikog praska: “The alternative [to thinking ahead] would be to think backwards . . . and that’s just remembering.”