

TEKOÄLYAVUSTEINEN OHJELMISTOTESTAUS

JUHANI TEERINIEMI, TKT, KNOWIT OY

TESTING ASSEMBLY 2017

Esitelmän kulku:

1. Pikaperehdytys koneoppimiseen.
2. Koneoppiminen ohjelmistotestauksessa.
3. Demonstraatio.

AGENDA



KNOWIT HAS OFFICES

in Sweden, Norway, Finland,
Denmark, Germany and Estonia



THE COMPANY IS LISTED

on the Nordic Exchange
in Stockholm



ESTABLISHED

in 1990



240.000.000 €

in net sales 2015

CREATING
DIGITAL
OPPORTUNITIES



MORE THAN 1800

specialists in total

AGE OF EMPLOYEES

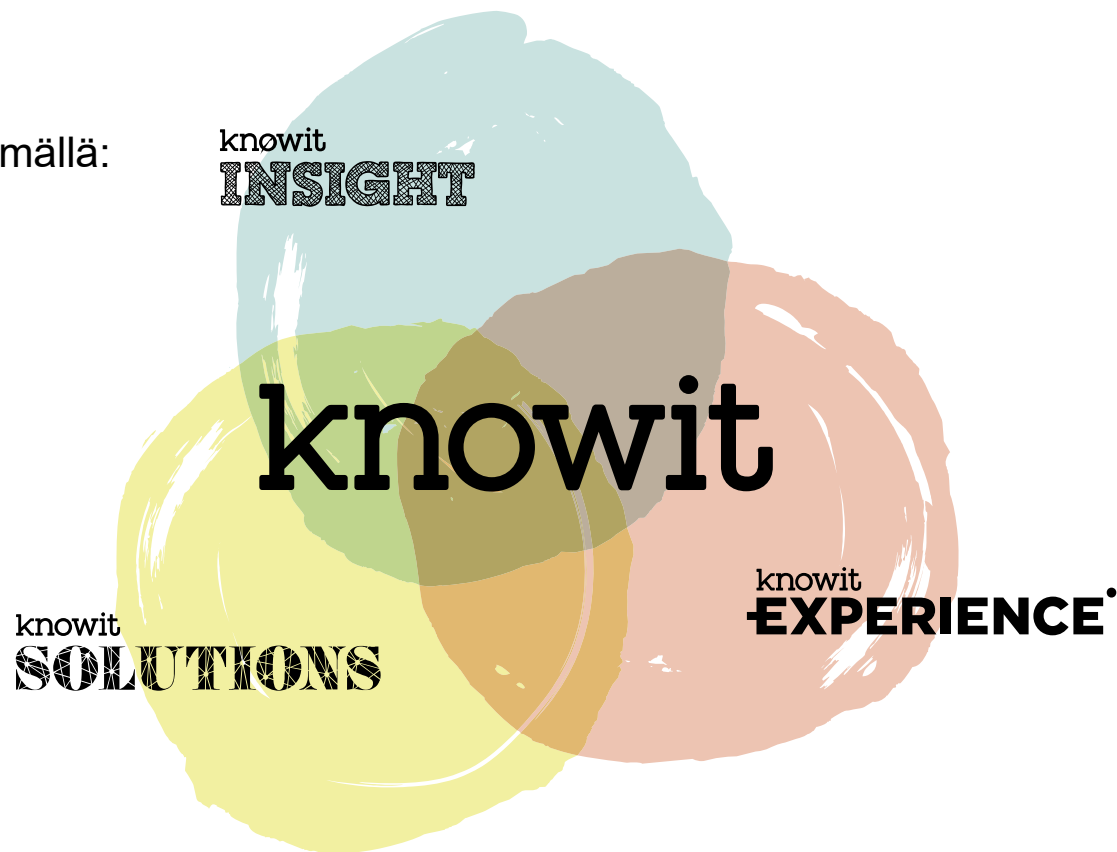


35855

Knowit employee
Twitter followers

Knowit luo digitaalisia mahdollisuuksia hyödyntämällä:

1. Teknologista osaamista (Solutions)
2. Luovuutta (Experience)
3. Strategista osaamista (Insight)



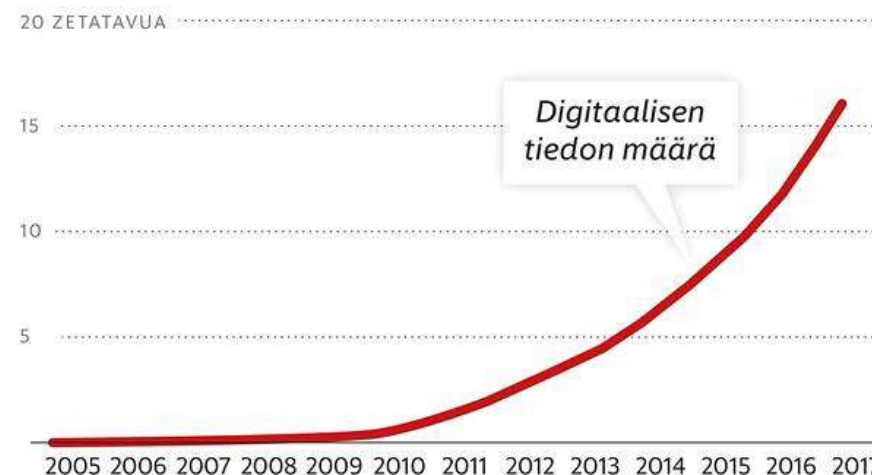


- Malleja kaikkialla.
- Datan määrä kasvaa räjähdysmäisesti.
- Manuaalinen data-analytiikka on **hidasta, kallista ja hankalaa**



- Kiinnostus analytiikan ja mallinnuksen automatisointiin kasvaa.
- Analytiikka-algoritmien kehitys kiihtyy.
- Automaattisesti dataa tiedoksi prosessoivat algoritmit pystyvät tuottamaan dataa ja parempia malleja itsenäisesti.

Datan määrä maailmassa kasvaa.



Kuva: Datan määrä maailmassa
Helsingin sanomat, 2.9.2017.

MIKSI KÄYTTÄÄ KONEOPPIMISTA?

knowit

Alati kasvava sovellusten kirjo

Markkinointi	Asiakkaiden luokittelu
Signaalinkäsittely	Rahoitus, ilmastotutkimus
Kuvankäsittely	Konenäkö / kasvontunnistus
Puheen- ja tekstintunnistus	Konekääntäjät
Kemia	Toksikologia, materiaalitekniikka
Data-analytiikka	Monimutkaisen datan mallintaminen



9/6/2017

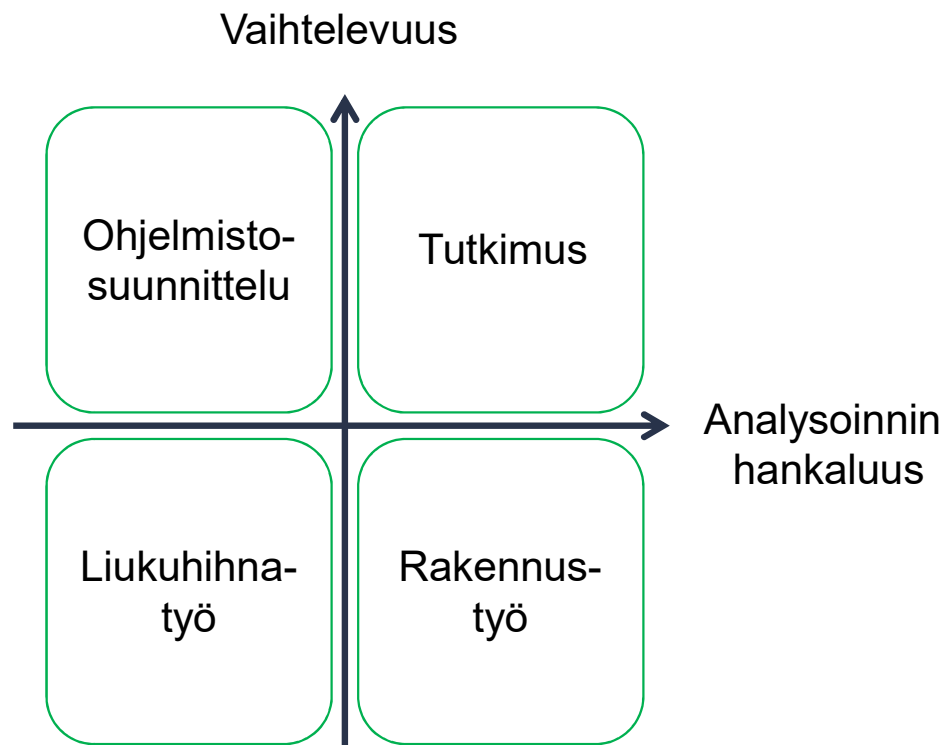
Knowit



MILLOIN KÄYTTÄÄ KONEOPPIMISTA?

knowit

- Koneoppimisen tarve kasvaa, kun työtä on hankala analysoida tai siihen liittyy suurta vaihtelua.



Tekstintunnistus

Vaihtelevuus: suuri

Analysoitavuus: helppo

Kuva: Calligraphy: the letter J

www.pinterest.com/mccorvallis/calligraphy-the-letter-j/

Koneoppimismenetelmät

Valvottu oppiminen

- Syötteet ja oikeat tulosteet tiedetään.
- Tavoitteena löytää malli, joka ennustaa uusien syötteiden tulosteet.



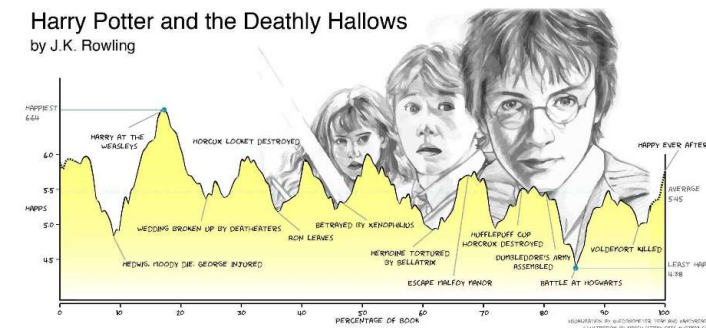
Kannustava oppiminen

- Syötteet tiedetään, mutta oikeita tulosteita ei osata kuvailla tarkasti.
- Hyvältä vaikuttavia ratkaisuja kannustetaan jalostumaan.

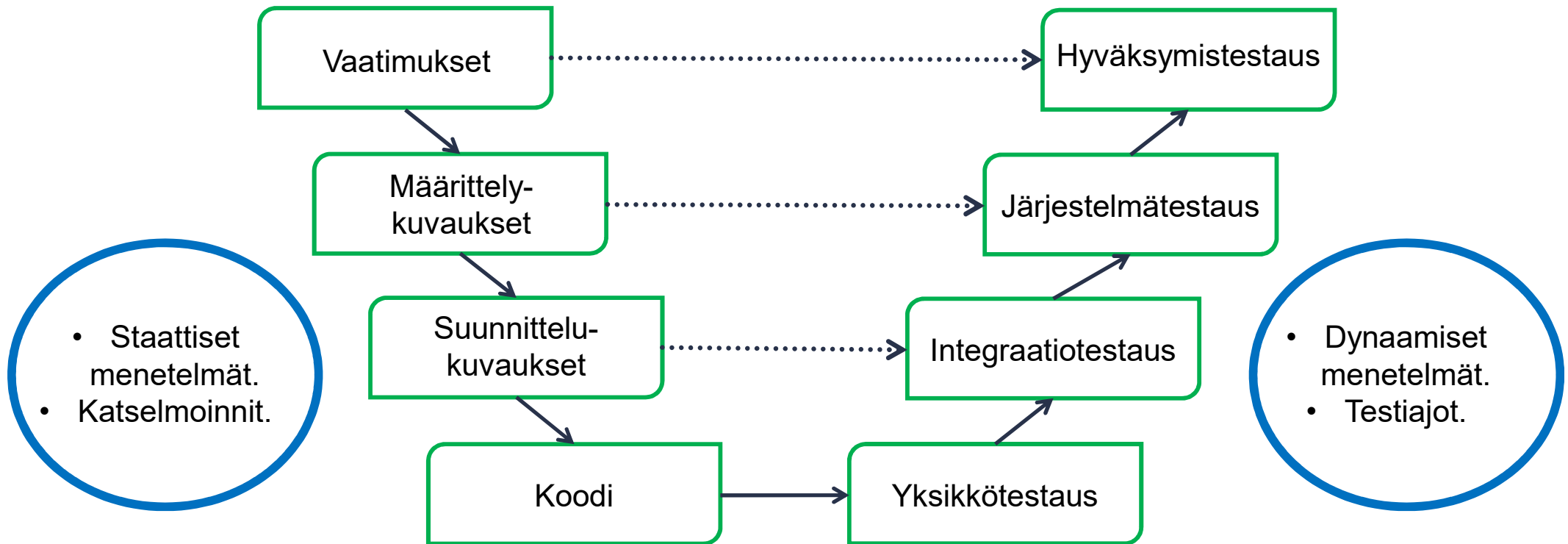


Valvomaton oppiminen

- Syötteet tiedetään, mutta haluttuja tulosteita ei.
- Vastaa tehtävänantoa: "Etsi jotain mielenkiintoista".



Al-kirjallisuusanalyysi[1]



- Staattisen testaamisen puolella lukuisia koneoppimismenetelmiä voidaan hyödyntää koodin laadun mittaamiseen.[1] NASA:n koodikatselmointi.
- Dynaamisen testaamisen puolella mallipohjaisia regressiotestejä voidaan korvata algoritmeilla, jotka generoivat testikulkuja 'on-the-fly'-tyyppisesti.[2] Netflixin Kaaosapina.
- Ylläpitotestauksessa koneoppimisella voidaan suunnitella realistisempia kuormitustestejä muodostamalla tilastollisia malleja sovelluksen todellisista käyttötavoista.



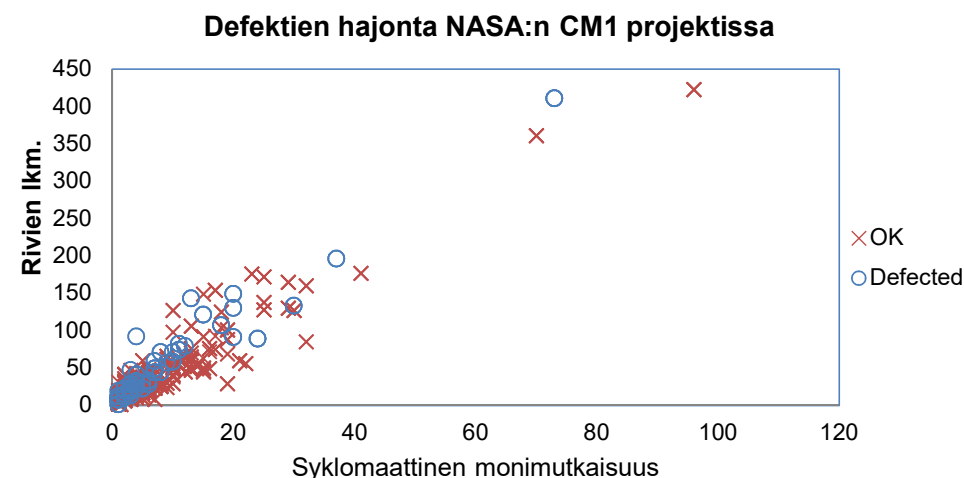
Kuva: Kunniamitalilla palkittu Margaret Hamilton Apollo sukkulan ohjausjärjestelmän lähdekoodin vieressä.

www.nasa.gov/

- Koodin laadun mittaamiseen on olemassa lukuisia metriikoita, joiden mittaaminen voidaan automatisoida. Suosituimmat kehitetty 1970-luvulla.[1,2]
- Ajatuksena on, että defektien todennäköisyys olisi verrannollinen johonkin helposti mitattavaan suureeseen (rivien lukumäärä, kommenttien lukumäärä, operaattoreiden lukumäärä jne.).
- Tavoitteena:
 - Antaa **kehittäjille** palautetta koodin laadusta.
 - Antaa **testaajille** palautetta tarvittavasta testauksen kattavuudesta.
 - Priorisoida testien ajojärjestystä, jotta virheet huomataan nopeammin.



- Vaikka perinteisiä metriikoita käytetään aktiivisesti, on niiden hyödyllisyyttä myös kyseenalaistettu.[1]
- Oikeassa maailmassa esimerkiksi koodirivien lukumäärä ei ole kovin hyvä indikaattori moduulin hyvyydestä.



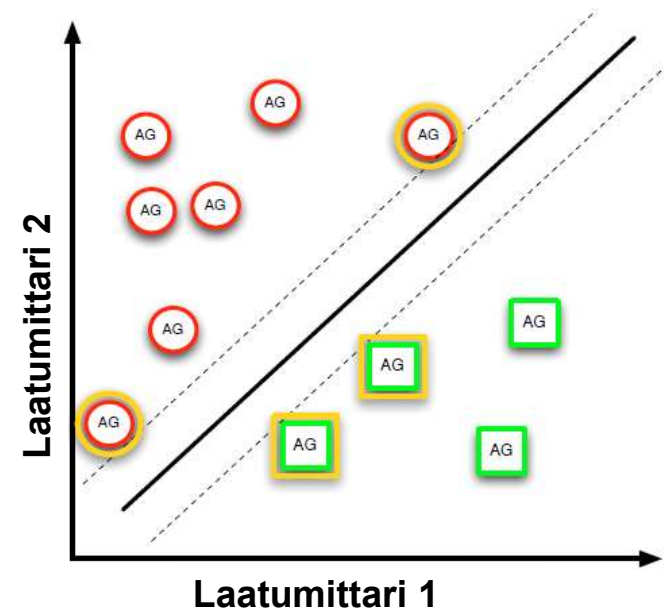
Kuva: NASA Metrics Data Program, CM1 Project[2]

Virheettömät ja virheelliset moduulit eivät erotu rivien lukumäärän tai syklomaattisen monimutkaisuuden perusteella.

- Tästä demonstraatiota varten kehitettiin tukivektorikonetyyppinen algoritmi.
- Ongelmaa voidaan lähestyä valvotulla koneoppimismenetelmällä, koska syötteet (laatumetriikat) ja oikeat tulosteet (defektit) tiedetään.
- Tavoitteena on löytää sellainen hypertaso moniulotteisessa data-avaruudessa, joka erottaa virheelliset moduulit virheettömistä.
- Matemaattisesti ongelma voidaan kuvata algoritmina:

$$\text{Minimoi} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$

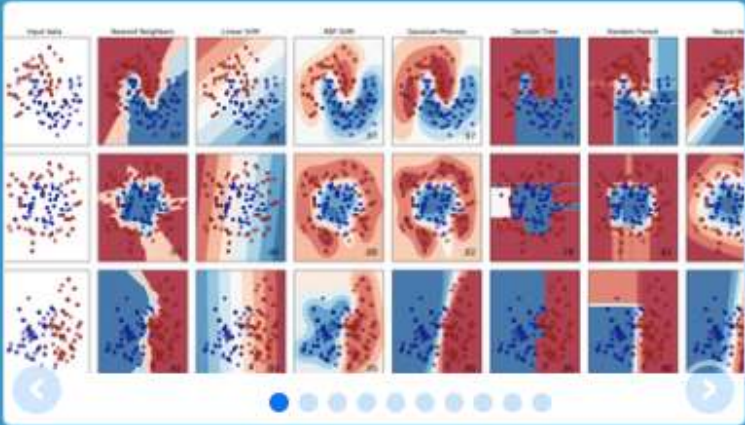
$$\text{Ehdolla} \quad 1 - y_i (\langle w, f(x_i) \rangle + b) \leq \xi_i \quad \text{kaikilla } i\text{:n arvoilla.}$$



TEKOÄLYAVUSTEINEN OHJELMISTOTESTAUS, DEMONSTRAATIO

knowit

- Jos olet nörtti, tee se itse! Tarvitset vain perustiedot kernelimenetelmistä, konvekssioptimoinnista, radiaalikantafunktioverkoista...
- Tai käytä esimerkiksi Pythonin scikit-learn -kirjastoa:



scikit-learn
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

TEKOÄLYAVUSTEINEN OHJELMISTOTESTAUS, DEMONSTRAATIO

knowit

- Valitettavasti tekoälylle täytyy antaa myös muita ohjeistuksia. Funktiota $AI(data)$ ei vielä ole olemassa.
- Menetelmien käyttö ilman ymmärrystä johtaa hyvin todennäköisesti harhaan.
- Esimerkki I: “Merkitse kaikki defektit” voi johtaa siihen, että tekoäly merkitsee kaikki moduulit viallisiksi. Tällöin kaikki defektit on merkitty.
- Esimerkki II: “Älä tee virheitä” voi johtaa siihen ettei tekoäly tee mitään. Helpoin tapa välttää virheitä on olla yrittämättä.



Kuva: Linnunradan käsikirja liftareille (2005).

TEKOÄLYAVUSTEINEN OHJELMISTOTESTAUS, DEMONSTRAATIO

knowit

- Tässä demonstraatioissa algoritmia pyydettiin maksimoimaan mallin F-pisteet, joka ottaa huomioon mallin tarkkuuden (sisäisen) ja saannon.

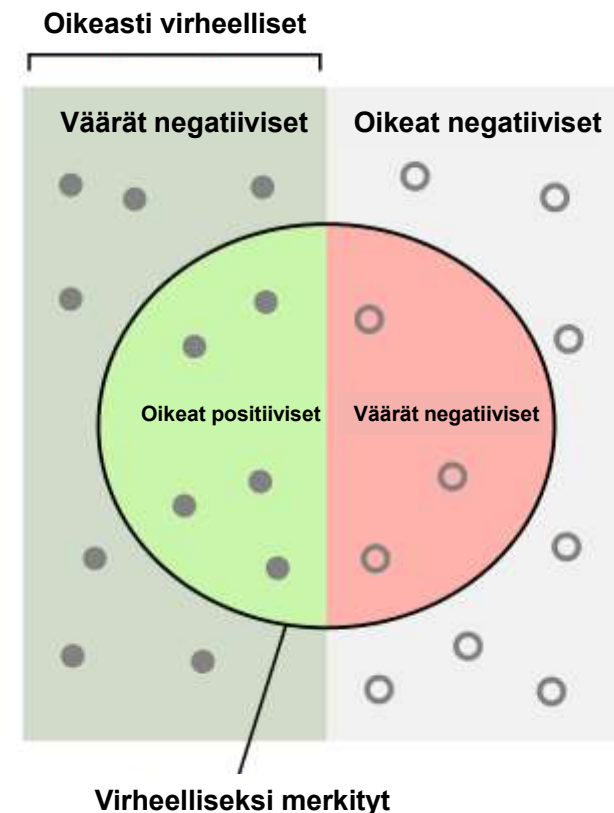
$$tarkkuus = \frac{\text{oikein virheelliseksi merkityt}}{\text{kaikki virheelliseksi merkityt}}$$

$$saanto = \frac{\text{oikein virheelliseksi merkityt}}{\text{kaikki oikeasti virheelliset}}$$

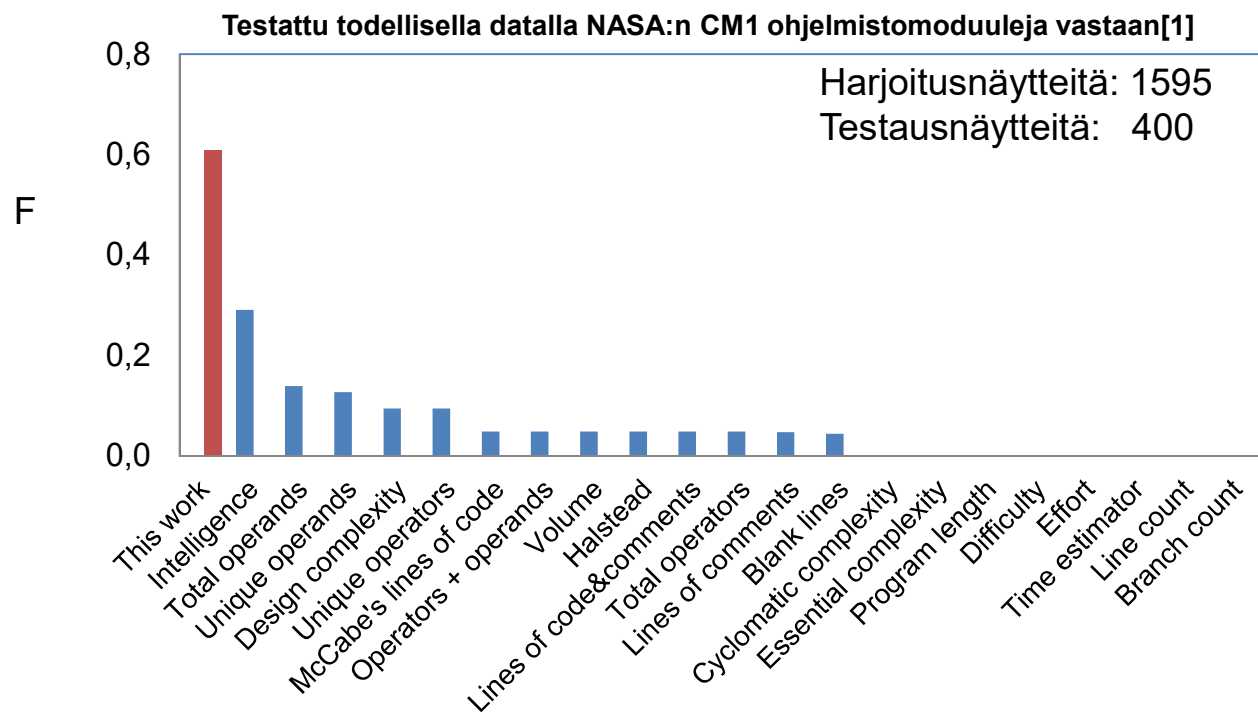
$$F = \frac{2 * tarkkuus * saanto}{tarkkuus + saanto}$$

Minimoi väärät hälytykset

Merkitse niin monta defektiä kuin mahdollista.



- Tekoälyavusteinen lähestymistapa antaa paljon parempia arvioita koodin laadusta kuin yksittäiset laatumetriikat.

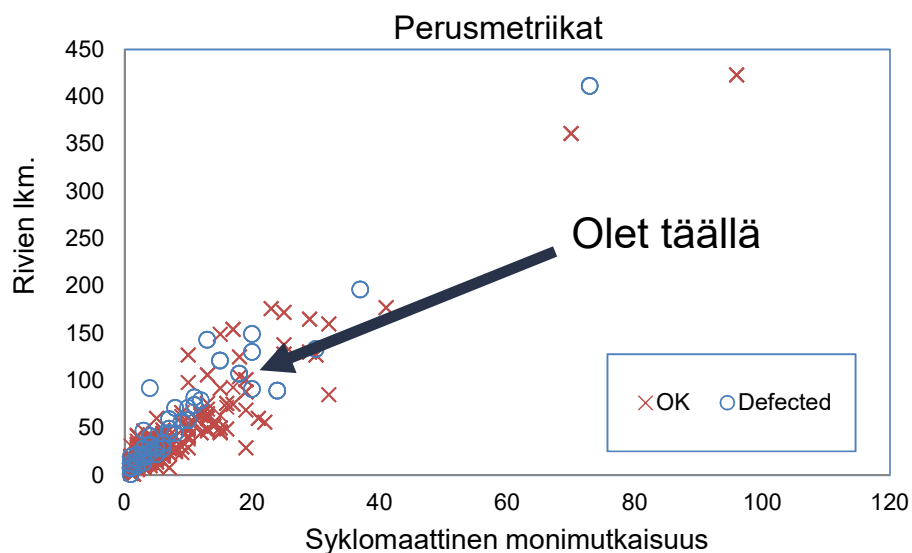


$$F = \frac{2 * tarkkuus * saanto}{tarkkuus + saanto}$$

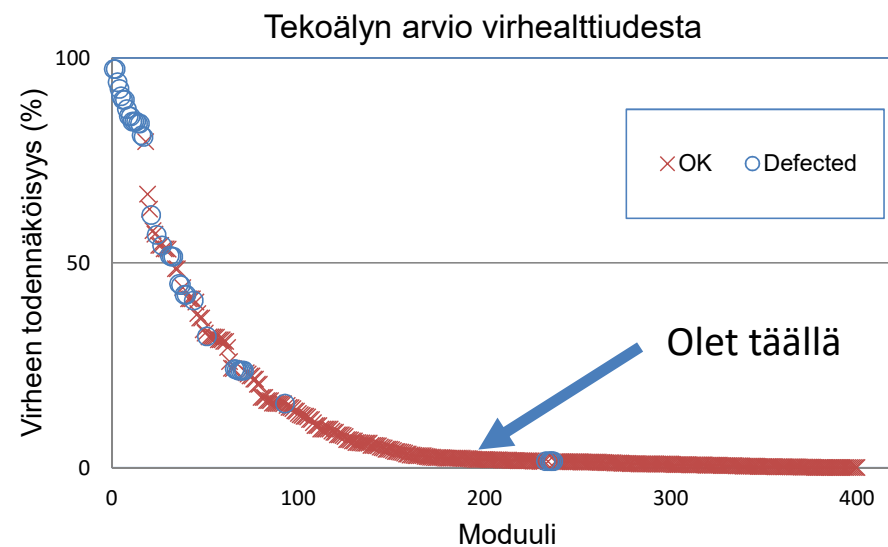
TEKOÄLYAVUSTEINEN OHJELMISTOTESTAUS, DEMONSTRAATIO

knowit

- Paremman suorituskyvyn lisäksi algoritmi antaa **intuiitiivisempaa palautetta** kuin tavalliset metriikat.
- Alla on verrattu uuden moduulin katselmointia perinteisellä menetelmällä (vasen) ja tekoälyavusteisesti (oikea).

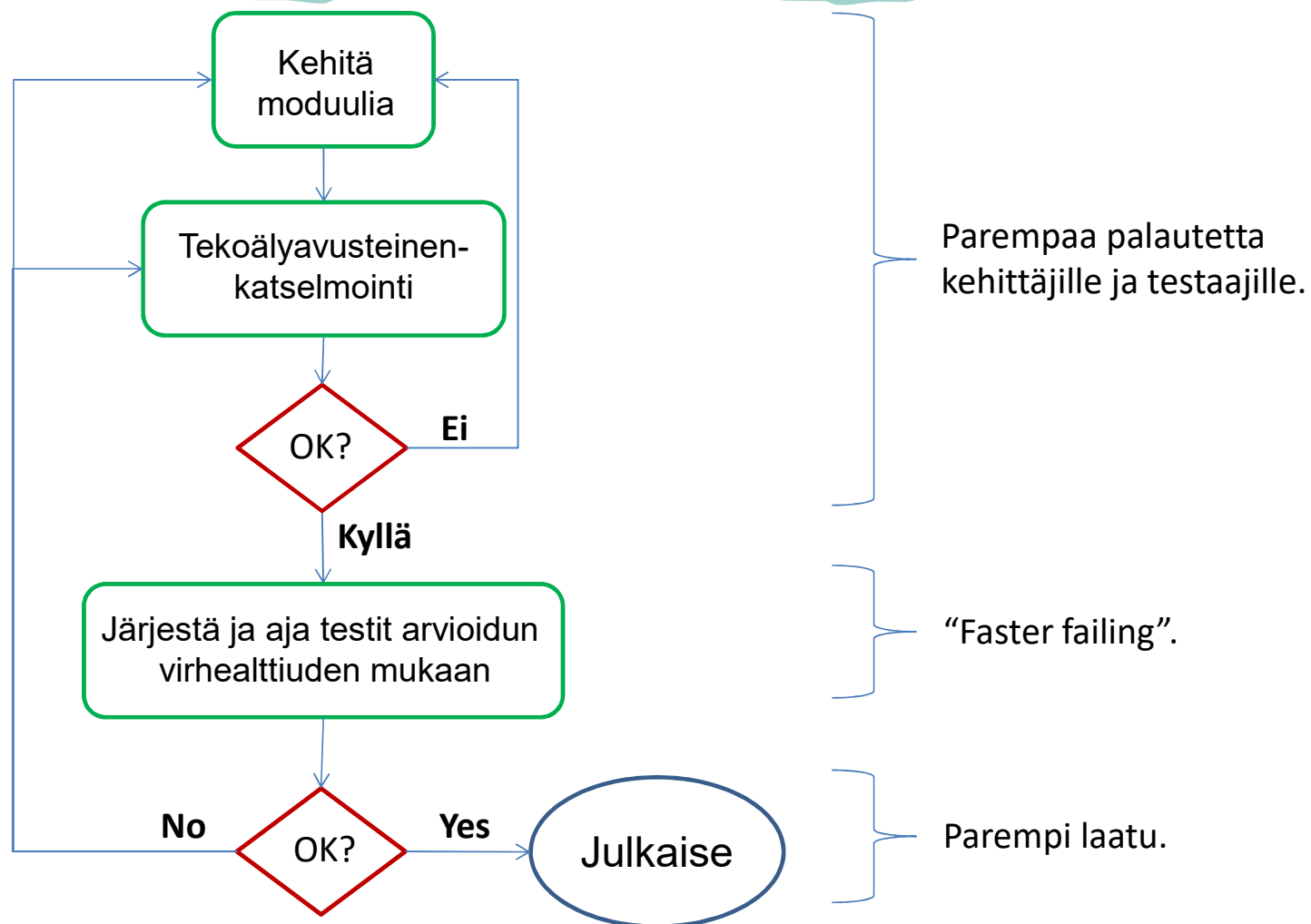


vastaan



TEKOÄLYAVUSTEINEN OHJELMISTOTESTAUS, DEMONSTRAATIO

knowit



- Koneoppiminen ohjelmistotestauksessa on (melko) uusi ja lupaava alue.
- Koneoppimisella testausta voidaan kohdentaa korkean riskin alueille.
- Koneoppimisella voidaan ohjata dynaamista testausta, jolloin testien kattavuutta ja automaatiota voidaan parantaa.

KiiTOS!