



# OPPIMISTEHTÄVÄ

## Koneoppiminen

Hanna Rantanen  
Jenna Rätty

DATA-ANALYYSI JA TEKOÄLYN PERUSTEET  
Joulukuu 2022

Tietotekniikka  
Tietoliikennetekniikka ja tietoverkot

# 1 AIHEEN JA DATAN ESITTELY JA TYÖN TAVOITE

## 1.1 Kuvaus datasta

Oppimistehtävän datasetti löydettiin Kaggle-sivustolta. Kyseinen datasetti on nimeltään '*Suicidal Behaviours Among Adolescents*', ja se tarkastelee tekijöitä, jotka ennustavat nuorten itsetuhoista käyttäytymistä. Dataa on kerätty 27 maasta käyttäen Global School-based Student Health Survey-kyselylomaketta (GSHS).

Datasetti on csv-tiedosto. Rivejä tiedostossa on 107kpl ja sarakkeita 17kpl. Country-sarakkeeseen on tallennettu maa (esim. 'Argentina'), Year-sarakkeessa on vuosiluku väliltä 2010–2018, Age Group-sarakkeessa on ikäryhmä väliltä 13–15 tai 16–17, sekä Sex-sarakkeessa sukupuoli ('Male' tai 'Female'). Lopuissa sarakkeista, jotka kuvaavat itsetuhoisiin yrityksiin vaikuttavia tekijöitä sekä itsetuhoisten yritysten esiintymistä, arvot on esitetty prosenttilukuina kerätyistä aineistosta. Numeroarvot eivät siis esitä ilmiöiden absoluuttista määrää, vaan prosentuaalista osuutta kyseisessä maassa kyseisenä vuonna tietyssä ikäryhmässä ja sukupuolella esiintyneistä arvoista.

Datasetissä on prosenttilukuja, jotka saavat jatkuvia arvoja. Näin ollen koneoppimisen menetelmäksi oli valittava regressiomenetelmä, luokittelumenetelmiä ei tällä datasetillä voida käyttää. Koneoppimisen menetelmiksi valittiin lineaarinen regressio, Decision Tree-regressio sekä Random Forest-regressio.

## 1.2 Kuvaus datan esikäsittelystä

Ensiksi haluttiin tarkistaa, onko datasetissä NaN-arvoja. Tämä katsottiin `isnull()`-funktioilla. Tästä saatiin tieto, että aineistossa on 6 NaN-arvoa. Nämä arvot muutettiin nolllaksi `fillna()`-funktioilla. Tämän jälkeen tarkistettiin jälleen `isnull()`:n avulla, että NaN-arvoja ei enää ole.

`Describe()`-funktioilla aineistoa tarkasteltaessa havaittiin, että `Currently_Drink_Alcohol`-sarakkeessa on virheellinen prosenttiluku 548%. Tässä on varmaankin tarkoitus olla arvo 54.8%. Arvo korjattiin `loc`-toiminnon avulla.

Ennen koneoppimisen menetelmien käyttöä tuli päättää, mitkä sarakkeet datasetistä valitaan opetusvaiheeseen, eli talletettiin X-muuttujaan. Datasetissä oli 17 saraketta, mikä on melko paljon, joten oletuksena oli, ettei niitä kaikkia välttämättä tarvitse käyttää parhaiden oppimistulosten saavuttamiseksi. Tässä vaiheessa kuitenkin jätettiin kaikki sarakkeet X-muuttujaan.

Datasetissä on kategorisia muuttujia, jotka täytyi muuttaa koneoppimisen ymmärtämään muotoon. Tämä tarkoittaa numeerista muotoa, eli luotiin niistä dummy-muuttujat. Datasetin kategoriset muuttujat olivat sarakkeissa 'Country', 'Age Group' sekä 'Sex'. Dummy-muuttujien luomiseen käytettiin `ColumnTransformer`- sekä `OneHotEncoder`-tekniikoita.

## 2 DATAN KÄSITTELYN KUVAUS KONEOPPIMISEN MENETELMÄLLÄ

### 2.1 Datan käsittely lineaarisen regression menetelmällä

X- ja y-muuttujien tallettamisen sekä dummy-muuttujien luomisen jälkeen data jaettiin opetus- ja testidataan. Tässä käytettiin suhdetta 80%-20%, sillä tämä oli kurssilla yleisimmin käytössä ollut tapa jakaa data. Ennen lineaarisen regression mallin opettamista data skaalattiin `StandardScaler()`-funktion avulla. Malli opetettiin `LinearRegression()`-funktion avulla, ja ennustus testidatalla tehtiin `model.predict()`-funktioilla.

Tämän jälkeen mallia voitiin arvioida metriikoiden avulla. Metriikoiksi valittiin R2, MAE sekä RMSE. Opetus- ja ennustusvaiheen jälkeen lineaarisen regression malli sai seuraavat metriikat:

```
Linear Regression:  
R2: 0.40400815310106764  
MAE: 2.883220826427646  
RMSE: 3.9026459653405694
```

R2-scoresta nähdään, että mallin avulla voidaan selittää vain noin 40% havainnoista. Mitä lähempänä R2-score olisi arvoa 1, sen paremmin malli sopii. Tästä päätellen voisi olla hyödyllistä pyrkiä muokkaamaan opetusprosessia R2-scoren parantamiseksi.

Yrityksen ja erehdyksen taktiikalla sekä koodia useaan kertaan ajamalla saatiin optimoitua X-muuttujaan talletetut sarakkeet parhaan R2-scoren tuottaviin valintoihin. Tällöin X-muuttujasta jätettiin pois sarakkeet 'No\_close\_friends' sekä 'Really\_Get\_Drunk'

```
Linear Regression:  
R2: 0.45368968517499664  
MAE: 2.773112323256568  
RMSE: 3.736445880749801
```

R2-score parani hieman, mutta on edelleen melko vaatimaton. Seuraava keino sen parantamiseen oli opetus- ja testidatan jakosuhteen muuttaminen. Internet-lähteiden perusteella yleisimmät datan jakosuhteet ovat 70%-30%, 80%-20% sekä 90%-10%. Jakosuhteella 70%-30% R2-score heikkeni huomattavasti, arvoon 0.057. Jakosuhte 90%-10% taas R2-scoreksi saatiin 0.65, mihin voitiin olla tyytyväisiä. Myös MAE- ja RMSE-metriikoiden arvot paranivat näiden muutosten myötä.

```
Linear Regression:  
R2: 0.6501257557107862  
MAE: 2.5284690323253254  
RMSE: 2.964539430303372
```

Mallin onnistumista voitiin tarkastella myös vertaamalla `y_test`-muuttujan alkuperäisiä arvoja sekä mallin ennustamia `y_pred`-muuttujan arvoja. Alla olevasta kuvasta nähdään, että joidenkin ennustusten kohdalla oli enemmän vaihtelua, mutta suuruusluokaltaan ennustukset osuivat kutakuinkin lähelle totuutta.

y_test - DataFrame		y_pred - NumPy object array	
Index	Attempted_suicide		0
99	16.7	0	21.1003
10	13.4	1	11.875
75	17.8	2	15.8769
2	17.4	3	18.9461
24	11.6	4	13.9911
98	23.1	5	18.1535
105	16.2	6	16.612
7	15.4	7	15.5511
16	4.5	8	7.93407
85	8.2	9	12.6462
68	19.7	10	22.3377

Lineaarisen regression mallia testattiin myös ajamalla sen läpi itse luotua testiaineistoa. Tätä varten luotiin kolme testitapausta: ensimmäisessä tapauksessa selittävien muuttujien arvot saivat keskiluokkaisia arvoja, toisessa tapauksessa selittävät muuttujat saivat suuria arvoja (poislukien 'Have\_Understanding\_Parents'-sarake, jolle annettiin pieni arvo), ja kolmannessa tapauksessa selittävät muuttujat saivat pieniä arvoja (poislukien 'Have\_Understanding\_Parents'-sarake, jolle annettiin suuri arvo). Luodut testitapaukset on esitelty vielä alla.

```
#Tehdään kolme testitapausta
new_cases = [{ 'Country': 'Argentina', 'Year': 2017, 'Age Group': '16-17', 'Sex': 'Female', 'Currently_Drink_Alcohol': 20, 'Overweight': 10,
               'Use_Marijuana': 30, 'Have_Understanding_Parents': 40, 'Missed_classes_without_permission': 25, 'Had_sexual_relation': 55,
               'Smoke_cig_currently': 25, 'Had_fights': 15, 'Bullied': 30, 'Got_Seriously_injured': 43 },
              { 'Country': 'Samoa', 'Year': 2011, 'Age Group': '13-15', 'Sex': 'Male', 'Currently_Drink_Alcohol': 90, 'Overweight': 80,
               'Use_Marijuana': 70, 'Have_Understanding_Parents': 20, 'Missed_classes_without_permission': 60, 'Had_sexual_relation': 50,
               'Smoke_cig_currently': 80, 'Had_fights': 80, 'Bullied': 90, 'Got_Seriously_injured': 80 },
              { 'Country': 'Indonesia', 'Year': 2015, 'Age Group': '16-17', 'Sex': 'Female', 'Currently_Drink_Alcohol': 0, 'Overweight': 0,
               'Use_Marijuana': 0, 'Have_Understanding_Parents': 90, 'Missed_classes_without_permission': 0, 'Had_sexual_relation': 0,
               'Smoke_cig_currently': 0, 'Had_fights': 0, 'Bullied': 0, 'Got_Seriously_injured': 0 }]
```

Kun näiden testitapausten perusteella tehtiin ennustus luodulla mallilla, tuloksista nähtiin, että keskivertotapauksen todennäköisyys itsetuhoiselle yritykselle olisi noin 34%, suuren riskin tapauksen todennäköisyys olisi noin 66% ja pienen riskin tapauksen noin -18%. Tämä tulos on järkevä ja tukee käsitystä mallin toimivuudesta.

new_y - DataFrame	
Index	0
0	33.8826
1	66.4244
2	-18.4548

## 2.2 Datan käsittely DecisionTree- ja RandomForest-menetelmillä

DecisionTree- ja RandomForest-menetelmien osalta käytettiin regressiota. Aiemmassa vaiheessa oli jo toteutettu X- ja y-muuttujien luominen, opetus- ja testidataan jako sekä dummy-muuttujien luonti. Myös DecisionTree- ja RandomForest-menetelmiä varten data skaalattiin StandardScaler()-funktion avulla. Koska tässä vaiheessa käytettiin kahta menetelmää, päätettiin ne ajaa aliohjelmien avulla. DecisionTree-menetelmässä malli opetettiin DecisionTreeRegressor()-funktiolla ja RandomForest-menetelmässä RandomForestRegressor()-funktiolla. Aliohjelmat suorittamalla ne palauttivat alla esitellyt metriikat.

```
Decision Tree:
R2: 0.05280850865491982
MAE: 3.801002164502164
RMSE: 4.91991829015039

Random Forest:
R2: 0.49639177438497895
MAE: 2.819909090909091
RMSE: 3.5874462555366082
```

Ylläolevasta kuvasta nähdään, että DecisionTree-malli antoi hyvin heikon R2-scoren, 0.05. RandomForest-malli antoi R2-scoreksi 0.496, mikä on kohtalainen mutta ei vielä kovin vahva todiste mallin toimivuudesta. Tämän parantamiseksi päätettiin yrittää muuttaa X-muuttujan arvoja sekä opetus- ja testidatan jakosuhdetta. X-muuttujan arvoihin lisättiin takaisin ne sarakkeet, jotka lineaarisen regression mallia varten poistettiin. Datan jakosuhte vaihdettiin alkuperäiseen 80%-20%. Tämän myötä metriikat olivat seuraavat:

```
Decision Tree:
R2: 0.03748913764323869
MAE: 3.862986013986013
RMSE: 4.959544778208524

Random Forest:
R2: 0.6114412883185596
MAE: 2.4757727272727266
RMSE: 3.151137262236373
```

DecisionTree-mallin R2-score on edelleen heikko, mutta RandomForest-mallin R2-scorea saatiin selkeästi parannettua (0.496 → 0.611).

Alla olevissa kuvissa on esitelty molempien menetelmien ennustusten ja todellisten y:n arvojen suhdetta (vasemmalla DecisionTree, oikealla RandomForest).

y_test - DataFrame		y_pred_dt - NumPy object array		y_test - DataFrame		y_pred_rf - NumPy object array	
Index	Attempted_suicide		0	Index	Attempted_suicide		0
99	16.7	0	19.632	99	16.7	0	18.615
10	13.4	1	19.632	10	13.4	1	17.44
75	17.8	2	13.159	75	17.8	2	17.285
2	17.4	3	13.159	2	17.4	3	16.501
24	11.6	4	19.632	24	11.6	4	17.081
98	23.1	5	19.632	98	23.1	5	22.83
105	16.2	6	13.159	105	16.2	6	16.836
7	15.4	7	13.159	7	15.4	7	15.675
16	4.5	8	6.46111	16	4.5	8	7.714
85	8.2	9	6.46111	85	8.2	9	12.224
68	19.7	10	19.632	68	19.7	10	20.907
22	12.1	11	13.159	22	12.1	11	14.417
45	6.7	12	6.46111	45	6.7	12	6.575
60	9.4	13	19.632	60	9.4	13	13.729
76	7.9	14	13.159	76	7.9	14	12.809
52	9.2	15	13.159	52	9.2	15	11.412
13	10.3	16	6.46111	13	10.3	16	11.444
73	20.5	17	19.632	73	20.5	17	20.019
84	9.6	18	6.46111	84	9.6	18	11.973
54	7.2	19	19.632	54	7.2	19	14.548
101	19.5	20	19.632	101	19.5	20	23.808
8	14.4	21	19.632	8	14.4	21	16.845

Silmämääräisesti vertailemalla voidaan todeta, että RandomForest-mallin ennustuksien virheet ovat pienempiä kuin DecisionTree-mallilla.

## 2.3 Saatujen tulosten pohdinta ja vertailu

Työssä käytettiin kolmea koneoppimisen menetelmää: lineaarinen regressio, DecisionTree-regressio sekä RandomForest-regressio.

```
Linear Regression:
R2: 0.6501257557107862
MAE: 2.5284690323253254
RMSE: 2.964539430303372
```

```
Decision Tree:
R2: 0.03748913764323869
MAE: 3.862986013986013
RMSE: 4.959544778208524
```

```
Random Forest:
R2: 0.6114412883185596
MAE: 2.4757727272727266
RMSE: 3.151137262236373
```

Yllä olevissa kuvissa on esitelty eri koneoppimisen mallien metriikat. Ensimmäiseksi tarkasteltiin  $R^2$ -scorea. Se kertoo, kuinka hyvin malli sopii dataan, ja sen arvo tulisi olla mahdollisimman lähellä arvoa 1.  $R^2$ -scoren arvoja tarkastelemalla voidaan todeta, että lineaarisen regression menetelmä toimii parhaiten ( $R^2=0.65$ ). Seuraavaksi parhaiten toimiva malli saatiin RandomForest-menetelmällä ( $R^2=0.61$ ). DecisionTree-menetelmä toimi tässä yhteydessä heikosti ( $R^2=0.04$ ).

Seuraavaksi voitiin tarkastella MAE- ja RMSE-arvojen suhteita. Mitä suurempi on näiden ero, sitä suurempaa vaihtelua on virheissä. Näin ollen RMSE- ja MAE-arvon erotus tulisi olla mahdollisimman pieni. Lineaarisen regression menetelmässä arvojen erotus on 0.436, DecisionTree-menetelmällä 1.0965 ja RandomForest-menetelmässä 0.6753. Näin ollen voidaan todeta, että lineaarisen regression menetelmässä virheiden vaihtelu on pienempää kuin muilla menetelmillä.

Tulosten perusteella voitiin vetää yhteen, että tässä oppimistehtävässä parhaiten koneoppimisen menetelmistä suoriutui lineaarinen regressio. RandomForest-menetelmä taas toimi paremmin kuin DecisionTree, mikä on järkevää, sillä se hyödyntää toiminnassaan useita DecisionTree-menetelmän luomia päätöspuita. Lineaarinen regressio saattoi tässä tilanteessa toimia RandomForestia paremmin siksi, että RandomForest hyödyntää toiminnassaan myös luokittelun menetelmiä, jotka eivät välttämättä sopeet valittuun datasettiin ja sen avulla opettamiseen. Tämän lisäksi RandomForest saattaa olla yksinkertaisesti liian monimutkainen ja raskas menetelmä koneoppimiseen tämän aineiston kohdalla, jolloin ”kevyempi” lineaarinen regressio tuo paremmat tulokset.

### 3 TYÖN ARVIOINTI

Tekijöiden mielestä tässä oppimistehtävässä on käytetty laajasti kurssin koneoppimisosuudessa opittuja regressioon soveltuvia tekniikoita ja menetelmiä. Kurssin harjoitustehtävien lisäksi työssä käytettiin lähteenä internetiä.

Kehityskohteena tämän työn myötä voisi olla sellaisen datasetin valitseminen, joka soveltuisi luokittelevien koneoppimismenetelmien käyttöön. Tällöin tulisi harjoiteltua myös luokittelevien menetelmien käyttöä sekä niihin liittyviä visualisointimenetelmiä, kuten confusion matrixia. Tämän lisäksi valittu datasetti olisi voinut olla hieman laajempi, jotta metriikka-arvot olisivat mahdollisesti olleet paremmat ja luodut mallit siten luotettavampia. Suuremman datamäärän avulla olisi mahdollisesti voitu hyödyntää myös neuroverkkojen menetelmiä.

Onnistumisena tämän työn osalta oli kolmen tarkoituksenmukaisen koneoppimis-menetelmän käyttö. Menetelmiä käyttämällä saatiin selkeästi vertailtua niiden toimivuutta hyödyntäen muun muassa erilaisia metriikoita. Myös koneoppimista edeltävä datan ja sen esikäsittelyn kuvaus oli kattava ja perusteellinen. Oppimistehtävässä pyrittiin hiomaan lopputulos mahdollisimman hyväksi muokkaamalla mallille syötettäviä arvoja, kuten X-muuttujaa sekä datan jakosuhdetta. Oppimistehtävän raportti on huolellisesti ja johdonmukaisesti laadittu, ja se etenee selkeästi havainnollistavien kuvien ja sanallisen kuvauksen myötä. Oppimistehtävä koettiin tekijöiden toimesta onnistuneeksi ja se sisältää 2 pisteen työlle vaadittavat ominaisuudet.



## 4 LIITTEET

### Liite 1. Oros datasta

Index	Country	Year	Age Group	Sex	Currently_Drink_Alcohol	Really_Get_Drunk	Overweight	Use_Marijuana	Have_Understanding_Parents	Missed_classes_without_permission	Had_sexual_relation	Smoke_cig_current	Had_fights	Bullied	Got_Seriously_injured	No_close_friends	Attempted_suicide
0	Argentina	2018	13-15	Female	50.3	30.7	27.8	7.9	41.5	24.7	25.7	16.8	17.2	0	27.5	4.8	19.9
1	Argentina	2018	13-15	Male	44.9	26.1	39.1	8.4	44.5	27.9	38.4	12.1	33.2	0	37.4	5.5	18.4
2	Argentina	2018	16-17	Female	67.2	56.3	22.5	21.9	37.1	34	59.1	28.5	15.1	0	38.1	6.3	17.4
3	Argentina	2018	16-17	Male	68.1	55.8	27.9	27	39.8	39.4	68.6	28	33.6	0	40.3	6.6	11.2
4	Argentina	2012	13-15	Male	49.3	28.9	35.9	10.6	46.2	32	43.5	17	44.2	42.1	24.8	6.1	12.9
5	Argentina	2012	13-15	Female	50.7	26.8	21.8	6.5	49.9	29.4	30.7	20.5	24.7	25.2	24.2	4.3	18.9
6	Barabados	2011	13-15	Male	29.4	80.2	32.3	15.6	48.1	14.6	32.9	0	42.7	30.3	49.1	6.8	10.4
7	Barabados	2011	13-15	Female	26.6	78	39.1	7.4	44.8	11.7	13.5	0	29.7	31.1	41.1	8.3	15.4
8	Benin	2016	13-15	Male	38	19.3	12.7	0.8	35.7	18.8	31.8	5.1	32.1	47.4	48.4	10.8	14.4
9	Benin	2016	13-15	Female	42.4	16.8	18.3	0.2	39.7	6.5	14.2	1.3	27.3	51.5	40.4	14.2	9.6
10	Benin	2016	16-17	Male	46.5	29.6	3.3	2.1	33.3	17.3	35.7	6.9	28.8	43.3	51	9.7	13.4
11	Benin	2016	16-17	Female	36.1	20	14.7	0.6	34.1	20.6	40.1	2	17.8	39.8	39.5	15.3	14.1
12	Bhutan	2016	13-15	Male	26	22.8	9.7	21.7	38.7	28.4	18.4	32.5	51.7	31.2	50.8	7.8	18.9
13	Bhutan	2016	13-15	Female	11.7	9.3	15.7	3.6	45.9	19.3	7.8	9.6	34.3	28.9	38.9	10.3	10.3
14	Bhutan	2016	16-17	Male	41.2	44.6	5.9	35.5	41	31.6	27	43.8	46.6	21.2	46.7	6.9	9.7
15	Bhutan	2016	16-17	Female	21.4	20.9	13.2	7.4	48.1	20.6	10	17.1	29.6	23.2	39.1	9.1	14
16	Brunei Darussalan	2014	13-15	Male	4.4	4.5	37.4	0.6	31.4	38.5	11.5	13.9	31.9	25.3	37.1	2.7	4.5
17	Brunei Darussalan	2014	13-15	Female	3.2	2.3	34.8	0	25.7	35.5	8.5	4.3	17.1	21.7	22.8	2.5	5.9
18	Brunei Darussalan	2014	16-17	Male	9.4	7.5	32.5	0.5	30.8	39.6	16.7	26.5	27.2	15.3	38.9	4.9	4.5
19	Brunei Darussalan	2014	16-17	Female	2.1	2.4	33.6	1	25.4	36.6	11.6	6.3	13.1	14.8	21.1	2.9	9.6
20	Dominican Republic	2016	13-15	Male	35.8	26.3	31.7	4.5	48.8	25.6	47	8.1	32.4	26.3	38.5	6.9	12.3
21	Dominican Republic	2016	13-15	Female	40.4	21.9	33	4.9	43.8	23	16.9	6.7	19.2	22.3	30.3	7.1	20.1
22	Dominican Republic	2016	16-17	Male	49.3	34	17.1	4	43.6	28.3	59.6	9.7	28.7	26.4	35.3	5.7	12.1
23	Dominican Republic	2016	16-17	Female	50.2	29.8	27.8	3.1	42.8	24.6	26.6	9	14.6	28.1	24.6	11.8	24.3
24	Fiji Islands	2016	13-15	Male	15.5	12.7	27.6	7.2	43.7	25.1	22	11.9	43.1	33.1	55.7	9.3	11.6
25	Fiji Islands	2016	13-15	Female	9.3	4.5	27.4	2.5	53.7	20.6	9.9	4.3	24.2	25.7	38.7	6.6	8
26	Fiji Islands	2016	16-17	Male	27.5	26.5	22.1	13.7	40.4	27.4	32.4	22.3	39.3	25	54.2	9.6	12.8
27	Fiji Islands	2016	16-17	Female	14.8	11.8	36.9	4.2	50.6	20.4	12.4	10	24.7	23	37.2	7.6	11.2
28	Indonesia	2015	13-15	Male	6.1	5.1	15.7	2.1	31.8	23.5	6.9	21.5	35.9	23.7	39.6	3.8	4.4
29	Indonesia	2015	13-15	Female	1.4	0.8	14.9	0.7	36.2	16.9	4	1.5	13.1	19	21.2	2.4	3.6
30	Indonesia	2015	16-17	Male	12.1	12.7	12.6	2.9	30.2	26.1	7.4	33	23.5	24	34.8	4	3.7
31	Indonesia	2015	16-17	Female	2.2	1.1	13.1	0.2	38.8	15.8	2.5	1.5	7	13	18.4	1.6	2.7
32	Jamaica	2017	13-15	Male	54.8	33.2	22.8	20.9	34.1	30.2	60.6	17.9	44.3	26.3	45.4	9.6	16.1
33	Jamaica	2017	13-15	Female	36.1	17.8	28.2	13.3	31.6	19.7	15.5	8.7	25.4	24.8	34	8	20.6
34	Jamaica	2017	16-17	Male	63.3	41.7	16.6	37.8	32.4	32.8	73.9	20.7	34.6	20.1	45	9.9	13.2
35	Jamaica	2017	16-17	Female	44.9	26.5	24.1	19.4	26.2	32.1	45	14	19.7	24.2	32.5	7	21.8
36	Kiribati	2011	13-15	Male	43.7	35.4	31.9	6.8	14.8	38.1	37.2	34.3	43.3	42.1	64.2	3.1	29.8
37	Kiribati	2011	13-15	Female	19.3	10.4	46.4	1.6	15.4	30.1	9.8	19.5	28.5	32.2	53.2	2.1	31.5
38	Laos	2015	13-15	Male	18.9	14.8	11.3	0.6	18.5	39.9	12.5	5.7	12	15.2	20	7	4.9
39	Laos	2015	13-15	Female	20.8	10.9	11.8	0.1	16.5	36	6.9	1.6	8.7	11.3	16.9	3.5	6.9
40	Laos	2015	16-17	Male	36.1	34	10.3	1.2	15.5	45.4	24.7	12.1	6.4	9.9	16.1	5.5	3.8
41	Laos	2015	16-17	Female	35.7	23.2	11	0.3	16.7	39.1	10.7	1.2	3.5	12	15.2	3.5	5.5
42	Malaysia	2012	13-15	Male	9.3	6.8	25.3	1.4	32.6	30	9.5	17.1	38.5	24	43.7	3.6	6.6
43	Malaysia	2012	13-15	Female	5.7	3	22.2	0.4	32.3	25.4	6.9	1.9	21.9	17.8	28.4	2.8	7.4
44	Malaysia	2012	16-17	Male	13.6	11	11.9	1.5	31.4	36.9	9.7	27	29.1	12.9	39.7	3.2	5.7
45	Malaysia	2012	16-17	Female	7.4	5.8	11.2	0.3	29.6	34.1	7.5	2.5	16.4	12	26.4	2.7	6.7
46	Mauritius	2017	13-15	Male	20.2	18.2	28.2	8.5	39.7	25.8	22.5	19.6	42.7	29	46.9	11.9	11.5
47	Mauritius	2017	13-15	Female	21.2	17	21.8	2.1	39.3	17.2	10.2	11.5	22.6	22	31.6	7.3	16.5

## Liite 2. Python-koodi

```

"""
Data-analyysi ja tekoälyn perusteet
Oppimistehtävä 2
Hanna Rantanen & Jenna Rätty
201227
"""

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
from scipy.stats.stats import pearsonr
from scipy.stats import chi2_contingency
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

#Luetaan aineisto dataframeen
df = pd.read_csv('GSHH_Pooled_Data1.csv')

##### DATAN ESIKÄSITTELY #####

#Tarkistetaan onko NaN-arvoja
print (df.isnull().values.any())

#Katsotaan NaN-arvojen lukumäärä
print (df.isnull().sum().sum())

#Korvataan NaN-arvot nolllalla
df = df.fillna(0)

#Tarkistetaan onko NaN-arvoja
print (df.isnull().values.any())

#Katsotaan NaN-arvojen lukumäärä
print (df.isnull().sum().sum())

```

```

#Korvataan virheellinen prosenttilukema
df.loc[df['Currently_Drink_Alcohol'] == 548, 'Currently_Drink_Alcohol'] = '54.8'

##### LINEAARINEN REGRESSIO #####

#Luodaan X- ja y-muuttujat
X = df.loc[:, ['Country', 'Year', 'Age Group', 'Sex', 'Currently_Drink_Alcohol', 'Overweight',
               'Use_Marijuana', 'Have_Understanding_Parents', 'Missed_classes_without_permission',
               'Had_sexual_relation', 'Smoke_cig_currently', 'Had_fights', 'Bullied', 'Got_Seriously_injured']]
y = df.loc[:, ['Attempted_suicide']]

# Poistettut sarakkeet: 'No_close_friends' 'Really_Get_Drunk' --> saatu R2-score optimoitua

#Dummy-muuttujien luominen Country-, Age Group- ja Sex-sarakkeista
X_orig = X
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(drop='first'), ['Country', 'Age
                                     Group', 'Sex'])], remainder='passthrough')
X = ct.fit_transform(X)

#Jaetaan aineisto opetus- ja testidataan (90% & 10%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=0)

#Skaalaus
scaler_x = StandardScaler(with_mean=False)
X_train = scaler_x.fit_transform(X_train)
X_test = scaler_x.transform(X_test)

#Opetetaan lineaarisen regression malli
model = LinearRegression()
model.fit(X_train, y_train)

#Ennustetaan testidatalla
y_pred = model.predict(X_test)

#Arvioidaan malli käyttäen metriikoita
R2 = r2_score(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print(f'\nLinear Regression:\nR2: {R2}')
print(f'MAE: {mae}')
print(f'RMSE: {rmse}')

```

```
#Tehdään kolme testitapausta: keskiverto, suuri riski, pieni riski
new_cases = [{'Country':'Argentina', 'Year':2017, 'Age Group':'16-17', 'Sex':'Female',
              'Currently_Drink_Alcohol':20, 'Overweight':10, 'Use_Marijuana':30, 'Have_Understanding_Parents':40, 'Missed_classes_without_permission':25, 'Had_sexual_relation':55, 'Smoke_cig_currently':25, 'Had_fights':15, 'Bullied':30, 'Got_Seriously_injured':43},
              {'Country':'Samoa', 'Year':2011, 'Age Group':'13-15', 'Sex':'Male', 'Currently_Drink_Alcohol':90, 'Overweight':80, 'Use_Marijuana':70, 'Have_Understanding_Parents':20, 'Missed_classes_without_permission':60, 'Had_sexual_relation':50, 'Smoke_cig_currently':80, 'Had_fights':80, 'Bullied':90, 'Got_Seriously_injured':80},
              {'Country':'Indonesia', 'Year':2015, 'Age Group':'16-17', 'Sex':'Female', 'Currently_Drink_Alcohol': 0, 'Overweight':0, 'Use_Marijuana':0, 'Have_Understanding_Parents':90, 'Missed_classes_without_permission':0, 'Had_sexual_relation':0, 'Smoke_cig_currently':0, 'Had_fights':0, 'Bullied':0, 'Got_Seriously_injured':0}]
```

```
#Talletetaan testitapauksen data, tehdään dummyt sekä skaalaus
```

```
new_data = pd.DataFrame(new_cases)
new_data = ct.transform(new_data)
new_data = scaler_x.transform(new_data)
```

```
#Ennustetaan kolmen testitapauksen perusteella
```

```
new_y = pd.DataFrame(model.predict(new_data))
```

```
##### DECISIONTREE- & RANDOMFOREST-REGRESSIO (TOTEUTUS ALIOHJELMILLA) #####
```

```
#Datan skaalaus
```

```
scaler_x = StandardScaler(with_mean=False)
X_train = scaler_x.fit_transform(X_train)
X_test = scaler_x.transform(X_test)
scaler_y = StandardScaler()
y_train = scaler_y.fit_transform(y_train)
```

```
#DecisionTree regressio-mallin aliohjelma
```

```
def decisionTree(max_depth, min_samples_split):
```

```
#Opetetaan DecisionTree regressio-malli opetusdatalla
```

```
model_dt = DecisionTreeRegressor(max_depth=max_depth, min_samples_split=min_samples_split)
model_dt.fit(X_train, y_train)
```

```
#Ennustetaan testidatalla
```

```
y_pred_dt = scaler_y.inverse_transform(model_dt.predict(X_test).reshape(-1,1))
```

```
#Arvioidaan malli käyttäen metriikoita
```

```
R2 = r2_score(y_test, y_pred_dt)
mae = mean_absolute_error(y_test, y_pred_dt)
mse = mean_squared_error(y_test, y_pred_dt)
rmse = np.sqrt(mse)
```

```
return R2, mae, rmse, y_pred_dt
```

```

#RandomForest regressio-mallin aliohjelma
def randomForest(n_estimators):
#Opetetaan RandomForest regressio-malli opetusdatalla
model_rf = RandomForestRegressor(n_estimators=n_estimators)
model_rf.fit(X_train, y_train.ravel())

#Ennustetaan testidatalla
y_pred_rf = scaler_y.inverse_transform(model_rf.predict(X_test).reshape(-1,1))

#Arvioidaan malli käyttäen metriikoita
R2 = r2_score(y_test, y_pred_rf)
mae = mean_absolute_error(y_test, y_pred_rf)
mse = mean_squared_error(y_test, y_pred_rf)
rmse = np.sqrt(mse)

return R2, mae, rmse, y_pred_rf

#Palautetaan metriikat DecisionTreelle
R2, mae, rmse, y_pred_dt = decisionTree(10,50)
print (f'\nDecision Tree:\nR2: {R2}')
print (f'MAE: {mae}')
print (f'RMSE: {rmse}')

#Palautetaan metriikat RandomForestille
R2, mae, rmse, y_pred_rf = randomForest(100)
print (f'\nRandom Forest:\nR2: {R2}')
print (f'MAE: {mae}')
print (f'RMSE: {rmse}')

```