Medical Cost Prediction

Genc Gashi

NBI/Handelsakademin

AI - teori och tillämpning, del 1

202509

Abstract

Denna rapport handlar om en kunskapskontroll som utfördes i kursen **AI - teori och tillämpning, del 1.** Kunskapskontrollen gick ut på att i grupp planera och välja ett gemensamt dataset, vilket blev **Medical Cost Personal**. Individuellt skulle man utföra en EDA-analys samt träna, utvärdera och välja den bästa av olika machine learning modeller. Den färdigtränade modellen skulle avslutningsvis produktionssättas i form av en **Streamlit-applikation**. I denna rapport beskrivs inledning, teori, metod, resultat, diskussion av resultaten och en avslutande slutsats där frågeställningarna besvaras.

Innehållsförteckning

1 Inledning

Syftet med denna uppgift var att lära sig utveckla och utvärdera olika machine learning modeller. Utöver detta var målet även att få en övergripande förståelse för hur ett machine learning projekt genomförs och vilket arbetsflöde som följs.

Frågeställningar som besvaras i rapporten är följande:

1. Vilken modell presterade bäst för *Medical Cost Personal*?
2. Vilka faktorer påverkade sjukvårdskostnaderna mest?
3. Går det att förbättra resultaten och i så fall hur?

2 Teori

2.1 Medical Cost Personal

Medical Cost Personal är ett dataset som består av 1 338 rader och 7 kolumner:

- Age – patientens ålder.
- Sex – patientens kön (Male/Female)
- BMI - Body Mass Index.
- Smoker – rökstatus (Yes/No)
- Children – antal barn som patienten har.
- Region – patientens boenderegion.
- Charges – patientens sjukvårdskostnad.

Detta dataset används för att prediktera sjukvårdskostnaden för varje patient baserat på övriga variabler. Datasetet är anpassat för regressionsproblem, eftersom sjukvårdskostnaden som ska predikteras är ett numeriskt värde. (Medical Cost Personal Datasets, 2018)

2.2 Modeller

2.2.1 Linjär Regression

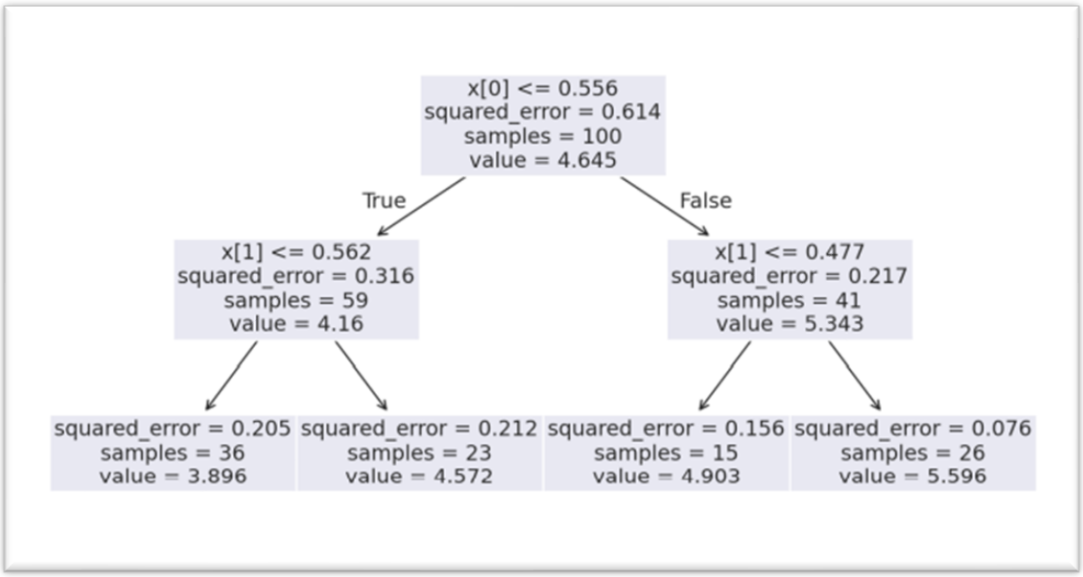
Linjär regression är den enklaste regressionsmodellen och bygger på den räta linjens ekvation.

$$\hat{y} = \hat{\theta}_1 + \hat{\theta}_2 x \tag{1}$$

Här är x en oberoende variabel, $\hat{y}$ den beroende variabel som ska predikteras, $\hat{\theta}_1$ är punkten där regressionslinjen skär y-axeln och $\hat{\theta}_2$ är linjens lutning. (Prgomet, Johnson, Solberg, & Rundberg Streuli, 2025-07-29)

2.2.2 Beslutsträd

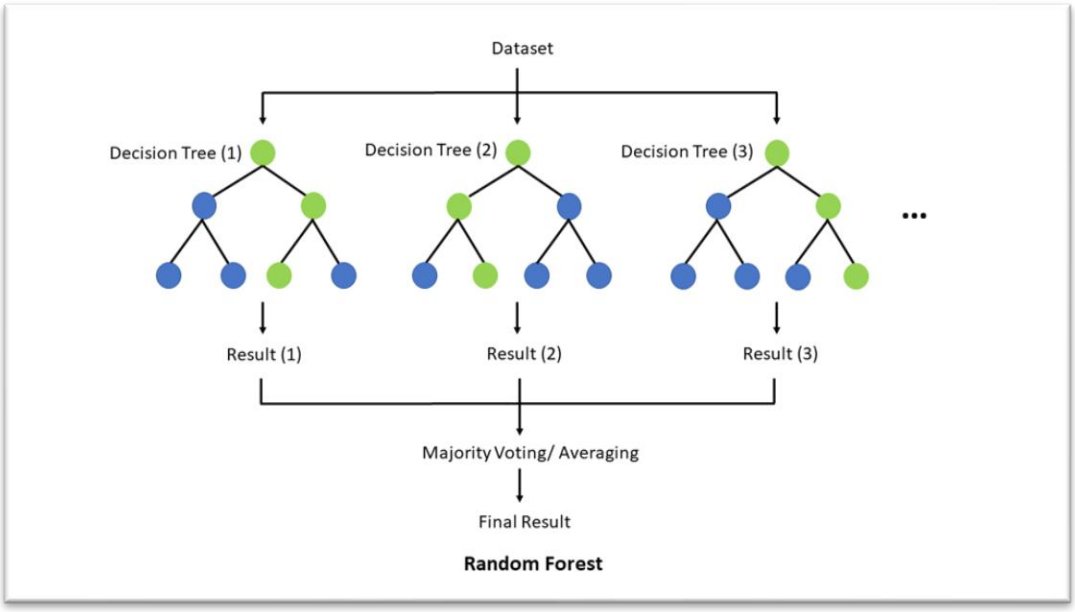
Beslutsträd är en modell som används för att göra prediktioner och fatta beslut. Modellen fungerar som ett flödesschema. Den börjar i rotnoden högst upp och går sedan genom de inre noderna tills den når lövnoderna. (Prgomet, Johnson, Solberg, & Rundberg Streuli, 2025-07-29)



Figur 1: Beslutsträd.

2.2.3 Random Forest

Random forest är en modell som kombinerar resultaten från flera beslutsträd till ett slutgiltigt resultat. Detta gör modellen både mer stabil och mer noggrann jämfört med ett enskilt beslutsträd. (Random Forest Algorithm in Machine Learning, 2025)

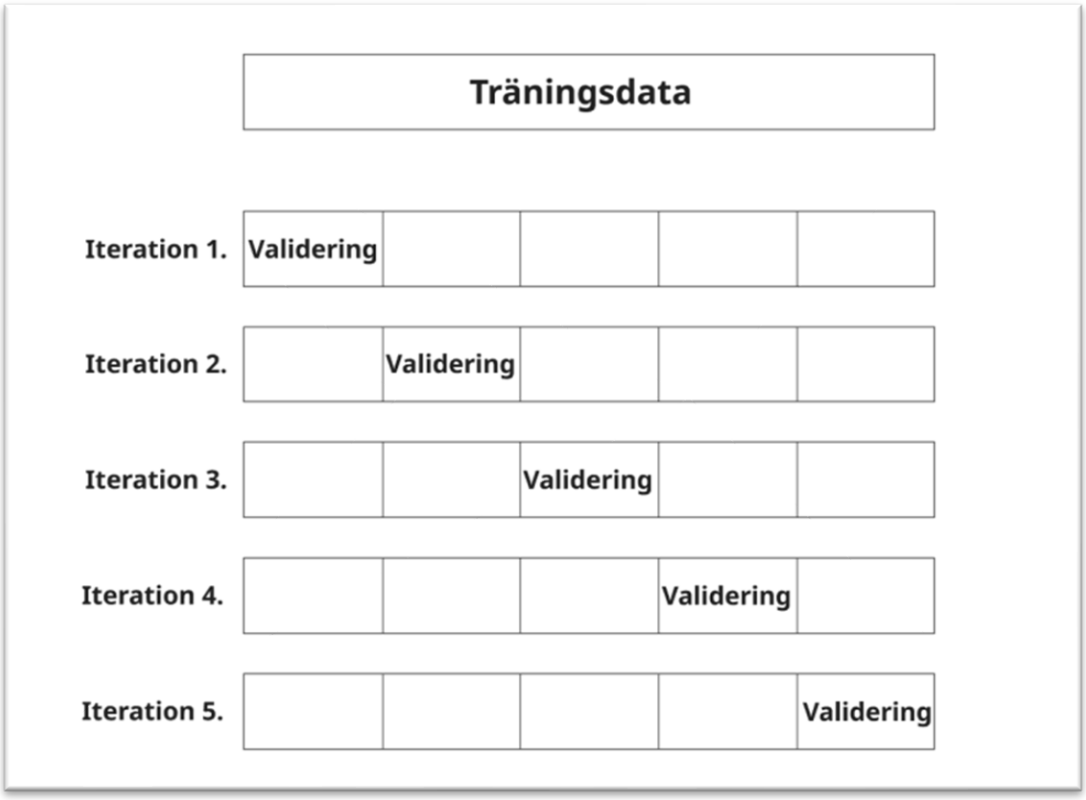


Figur 2: Random Forest.

2.3 Utvärdering av modeller

2.3.1 K-delad korsvalidering

Korsvalidering används för att utvärdera modeller genom att upprepade gånger dela upp data till tränings och testdata. Modellen tränas på en del av datan och testas på de återstående delarna av datan, ett givet antal gånger. Fungerar bra vid små dataset där valideringsdata saknas. (Prgomet, Johnson, Solberg, & Rundberg Streuli, 2025-07-29)



Figur 3: K-delad Korsvalidering

2.3.2 RMSE

RMSE visar hur bra en regressionsmodell predikterar nya värden. Det beräknas genom att ta kvadratroten ur medelvärdet av de kvadrerade skillnaderna mellan de verkliga och predikterade värdena.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_1)^2} \quad (2)$$

Här är y_i de verkliga värdena, $\hat{y}_1$ motsvarar de predikterade värdena och n är antalet observationer. (Prgomet, Johnson, Solberg, & Rundberg Streuli, 2025-07-29)

2.3.3 R^2

R^2 anger hur bra de oberoende variablerna kan förklara variationerna i den beroende variabeln. Värdet av R^2 ligger mellan 0 och 1, ju närmare 1 desto bättre anses modellen prestera.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (3)$$

Här är SS_{res} summan av de kvadrerade residualerna och SS_{tot} är den totala variationen i datan. (Verma, 2023)

2.4 Bästa hyperparametrarna/ Grid Search

Grid search är en metod inom machine learning som används för att hitta de bästa parametrarna för en modell. Olika kombinationer av parametrar testas för att förbättra modellens prestanda. (Grid Searching From Scratch using Python, 2024)

3 Metod

3.1 Problem definition

Det första steget var att välja och läsa in datasetet samt analysera och förstå datan. Efter genomgång av datasetet bestämdes det att målet är att prediktera patienters sjukvårdskostnader baserat på BMI, kön, region, ålder, rökstatus och antal barn.

Detta dataset valdes eftersom den passar ett regressions problem mycket väl. Den kändes lagom utmanande då den inte är extremt stor, vilket gör den hanterbar. Samtidigt finns det vissa svårigheter, till exempel att encoda de kategoriska variablerna. En risk med datasetet är att det är relativt litet och om man rensar extremvärden kan det påverka resultaten så att prediktionerna blir mindre korrekta.

3.2 Tillgång till data

Nästa steg var att läsa in CSV-filen i Jupyter Notebook och därefter hämta information om datasetet, dess beskrivning samt de första raderna. Därefter delades datan upp i tränings och testdata enligt figur 4.

```
train, test = train_test_split(df, test_size=0.2, random_state=40)
```

Figur 4: Uppdelning av data.

Träningsdatan användes i de kommande stegen, medan testdatan sparades för att senare utvärdera den bästa modellen.

3.3 EDA analys & Databearbetning

En EDA-analys utfördes på träningsdatan där visualiseringar skapades för att undersöka fördelningen av både kategoriska och numeriska variablerna. Syftet var att få en bättre förståelse för datan samt att upptäcka mönster eller avvikelser.

```
# Stapeldiagram på rökstsatus.
plt.bar(train['smoker'].value_counts().index,train['smoker'].value_counts().values, color=['grey', 'red'])
plt.title("Distrubution of smokers")
plt.ylabel("count")
plt.show()
```

Figur 5: Visualisering på kategorisk variabel.

```
# Histogram på bmi.
plt.hist(train['bmi'], edgecolor='black', color='green')
plt.title('Distrubution of bmi')
plt.xlabel("bmi")
plt.ylabel("count")
plt.show()
```

Figur 6: Visualisering av numerisk variabel.

I nästa steg deklarerades vilka variabler som skulle vara y-värdet och vilka som skulle vara x-värden, både för tränings och testdatan. I detta fall blev y-värdet sjukvårdskostnaden(charges) och x-värdena de övriga variablerna. Deklarationen gjordes enligt figur 7 och 8 för både tränings och testdatan.

```
X_train = train.drop('charges', axis=1)
```

Figur 7: Deklarerar X_train (Samma görs för X_test).

```
y_train = train['charges']
```

Figur 8: Deklarerar y_train (Samma görs för y_test).

Eftersom X-variablerna innehåller kategoriska värden användes **Sklearns OneHotEncoder** för att omvandla dessa till numeriska variabler. **fit_transform** tillämpades på träningsdatan och därefter **transform** på testdatan, enligt nedanstående figur.

```
X_train_encoded = encode.fit_transform(X_train[['sex','smoker','region']])
X_test_encoded = encode.transform(X_test[['sex','smoker','region']])
```

Figur 9: One Hot Encoding.

Efter omvandlingen av de kategoriska variablerna slogs dessa samman med de numeriska variablerna. De ursprungliga kolumnerna togs bort och ersattes med de nya kolumnerna, vilket skapade **X_train_final** och **X_test_final** som används i kommande steg.

```
X_train_final = np.hstack([X_train.drop(['sex','smoker','region'], axis=1).values, X_train_encoded])
X_test_final = np.hstack([X_test.drop(['sex','smoker','region'], axis=1).values, X_test_encoded])
```

Figur 10: Slutgiltig tränings- och testdata.

3.4 ML-Modellering

3.4.1 Modeller: Träning och utvärdering

I detta projekt tränades och utvärderades tre modeller:

- Linjär regression.
- Random Forest.
- Decision Tree.

Dessa modeller valdes eftersom linjär regression är enkel och lätt att förstå, beslutsträd kan fånga icke-linjära samband och Random Forest kan ge stabila resultat genom att kombinera flera beslutsträd.

Modellerna tränades och utvärderades med korsvalidering. För varje modell beräknades R^2 och $RMSE$ för att se vilken modell som presterar bäst. Korsvalideringen genomfördes enligt figur 11.

```
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)


cv_rf = cross_validate(rf_model, X_train_final, y_train, cv=5, scoring = ['r2','neg_root_mean_squared_error'])

r2_mean = cv_rf['test_r2'].mean()
rmse_mean = cv_rf['test_neg_root_mean_squared_error'].mean()

print('R2: ', cv_rf['test_r2'])
print('R2_mean: ',r2_mean)
print('RMSE: ' , cv_rf['test_neg_root_mean_squared_error'])
print('RMSE_mean: ',rmse_mean)
```

Figur 11: Krossvalidering på en av modellerna.

3.4.2 Grid Search

Efter att den bästa modellen hade valts utfördes Grid Search för att hitta de bästa parametrarna, vilket förväntas förbättra modellens prestanda. I detta projekt var den bästa modellen Random Forest och Grid Search genomfördes på denna enligt figur 12.

```
param_grid = {'max_depth': [5, 10, 15, 50],
              'n_estimators':[100, 200],}
grid_search = GridSearchCV(rf_model, param_grid, scoring='r2', cv=5)
grid_search.fit(X_train_final, y_train)
```

GridSearchCV

best_estimator_:
RandomForestRegressor

RandomForestRegressor

```
best_model = grid_search.best_estimator_
```

Figur 12: Grid Search på Random Forest.

3.4.3 Utvärdering av Best_Model

Den slutgiltiga bästa modellen utvärderades sedan på tränings och på testdatan. Detta genomfördes genom att få fram *RMSE* och R^2 enligt figur 13 och 14.

```
print("R² Train:", best_model.score(X_train_final, y_train))
print("R² Test:", best_model.score(X_test_final, y_test))
```

Figur 13: Utvärdering av best_model.

```
print("RMSE_Train: ",root_mean_squared_error(y_train, best_model.predict(X_train_final)))
print("RMSE_Test: ",root_mean_squared_error(y_test, best_model.predict(X_test_final)))
```

Figur 14: Utvärdering av best_model.

3.4.4 Slutgilitlig model

När all utvärdering var klar och modellen var färdigställd slogs tränings och testdatan ihop. Detta gjordes för att träna om modellen på all tillgänglig data och för att sedan produktionssätta modellen i nästa steg.

3.5 Produktionssättning

En Streamlit-applikation skapades för att produktionssätta den färdiga modellen. Applikationen fungerar enligt följande:

- Patienten fyller i ålder, BMI, kön, antal barn, region och rökstatus.
- Genom att trycka på knappen *Predict* visas den förväntade sjukvårdskostnaden.

4 Resultat och Diskussion

4.1 EDA-Analys

Från EDA-analysen framgick det att andelen yngre människor var större än andelen äldre. Dessutom var icke-rökare betydligt fler än antalet rökare. Sjukvårdskostnaden var ojämnt fördelat, med flest personer vid de låga kostnaderna och färre personer vid de högre kostnaderna. Dessa ojämlikheter kan påverka modellen prestanda och leda till avvikande resultat.

4.2 ML-modelleringen

4.2.1 RMSE & R2

Medelvärdet av R^2	
Enkel Linjär Regression	0.74
Random Forest	0.83
Decision Tree	0.71

Tabell 1: R2 för de tre valda modellerna.

Medelvärdet av RMSE	
Enkel Linjär Regression	-6043
Random Forest	-4844
Decision Tree	-6393

Tabell 2: RMSE för de tre valda modellerna.

R^2 på Random Forest (bästa model)	
R^2 Train	0.89
R^2 Test	0.85

Tabell 3: R2 värden på bästa modellen.

RMSE på Random Forest (bästa model)	
RMSE Train	3960
R^2 Test	4644

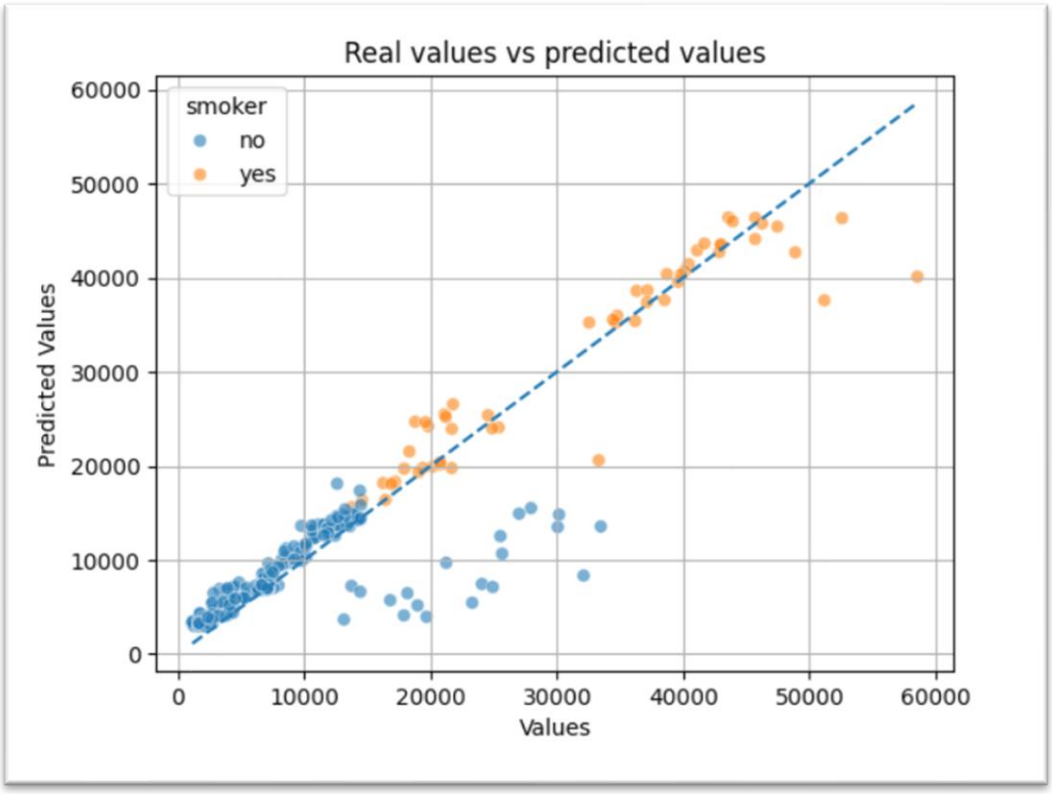
Tabell 4: RMSE värden på bästa modellen.

4.2.2 Diskussion

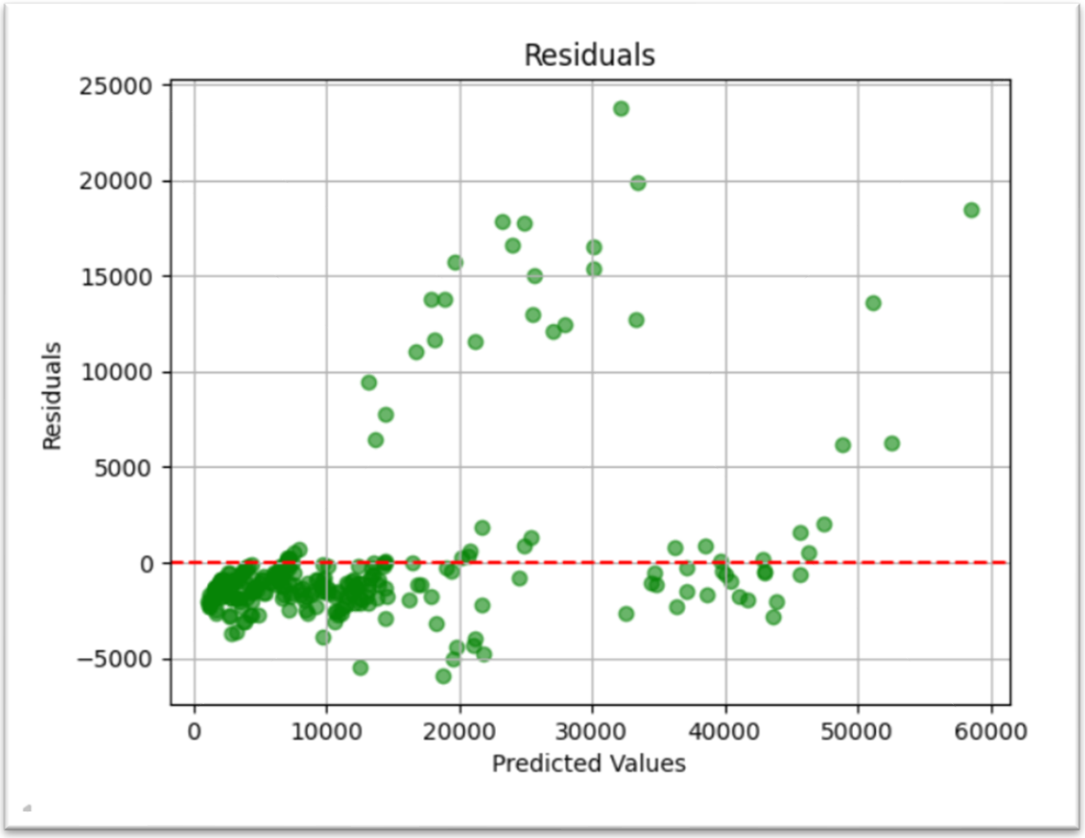
Enligt tabell 1 och 2 visade sig Random Forest ha de bästa R^2 och *RMSE* värdena, vilket antyder att den presterar bäst när den ska prediktera nya sjukvårdskostnader. Anledningen till att Random Forest är den bästa modellen är att den kombinerar resultaten från olika beslutsträd, vilket gör att den kan fånga både icke-linjära och komplexa samband. Sjukvårdskostnaden påverkas av flera faktorer som samverkar på olika sätt, vilket gör att linjär regression har svårare att fånga sådana samband. Efter att Random Forest valts förbättrades resultatet ännu mer med hjälp av Grid Search, vilket kan ses i tabell 3 och 4.

4.3 Avvikelser & Residualer

4.3.1 Visualiseringar



Figur 15: Verkliga vs predikterade värden



Figur 16: Residualer

4.3.2 Diskussion

Modellen presterar mestadels bra enligt ovanstående visualiseringar, särskilt vid de låga kostnader där de predikterade värdena ligger nära de verkliga. I mittenområdet mellan 15 000–30 000, börjar avvikelserna bli större, vilket gör att modellen underskattar värden. Residual visualiseringen visar samma mönster med fler residualer i detta intervall. Anledningen till detta är fortfarande lite oklart eftersom inget tydligt mönster har identifierats som orsakar det. I figur 15 har man ändrat variablen för att undersöka om de övriga variablerna har någon påverkan, men det visades sig inte. En möjlig anledning till detta kan vara ojämnlikheterna i fördelning av vissa variabler som identifierades i EDA-analysen.

4.4 Streamlit Applikation

Figur 17 visar hur Streamlit-applikationen ser ut. Hur den används har beskrivits tidigare i rapporten.

Figur 17: Applikationen

5 Slutsatser

Avslutningsvis är den bästa modellen för datasetet **Random Forest**. Den presterar bäst eftersom den kan fånga icke-linjära och komplexa samband. Sjukvårdskostnaden påverkas av 6 andra variabler som kombineras på olika sätt och det är just i sådana fall Random Forest presterar bra.

Efter användning av modellen och utförandet av prediktioner i applikationen, har man kommit fram till att rökstatus och patientens ålder har starkast korrelationen till sjukvårdskostnaden. Vid ändring av dessa variabler ser man att sjukvårdskostnaden ändras drastiskt, vilket är logiskt ur ett realistiskt perspektiv då rökning och ålder påverkar mycket hälsan.

Utvärderingsresultaten på den färdiga modellen var hyfsat bra, men det finns naturligtvis alltid utrymme för förbättringar. Genom att lägga till ny data i datasetet kan den ojämna fördelningen av vissa variabler balanseras, vilket gör att de grupper som tidigare underpresterade får större påverkan på modellens resultat. Detta kan leda till modellen presterar ännu bättre och att residualerna möjligtvis minskar.

En annan metod som vanligtvis förbättrar resultaten är att rensa datan och ta bort eventuella extremvärden. Denna metod utfördes i detta projekt men gav sämre resultat, därför behölls alla värden. Datasetet som nämnts tidigare i denna rapport är litet. Detta innebär att små förändringar vid rensning kan påverka resultaten och leda till mindre korrekta prediktioner.

6 Självutvärdering

1. Vad har varit roligast i kunskapskontrollen?

Roligast med kunskapskontrollen var att genomföra hela projektet enligt alla machine learning steg och sedan sätta modellen i produktion. Det gav en inblick i hur ett AI-projekt funkar, vilket var mycket intressant.

2. Vilket betyg anser du att du ska ha och varför?

Jag anser att jag bör få minst ett godkänt eller ett högre betyg, eftersom jag har genomfört hela uppgiften. Jag har gjort alla delar i ett machine learning projekt och skapat en streamlit applikation. Jag har även skrivit en rapport som dokumenterar arbetet tydligt samt analyserat och identifierat förbättringar.

3. Vad har varit mest utmanande i arbetet och hur har du hanterat det?

Det var svårt i början att följa alla steg i machine learning processen och förstå i vilken ordning de skulle genomföras. Ju mer man arbetade med kunskapskontrollen, desto tydligare blev flödet. Boken hade ett exempel på hur ett sådant projekt genomförs, vilket var till stor hjälp.

4. Hur har grupparbetet gått?

Grupparbetet gick bra. Vi var snabbt överens om målen och hade kontinuerligt kontakt under arbetes gång för att jämföra och hjälpa varandra.

Källförteckning

Grid Searching From Scratch using Python. (2024, Maj 21). Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/machine-learning/grid-searching-from-scratch-using-python/

Medical Cost Personal Datasets. (2018, Februari 21). Retrieved from Kaggle: https://www.kaggle.com/datasets/mirichoi0218/insurance

Prgomet, A., Johnson, T., Solberg, A., & Rundberg Streuli, L. (2025-07-29). *Lär dig AI från grunden - Tillämpad maskininlärning med Python.* Pedagogicus Publishing.

Random Forest Algorithm in Machine Learning. (2025, September 01). Retrieved from Geeksforgeeks: https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning/

Verma, D. (2023, Maj 28). *R2 Score: Linear Regression*. Retrieved from Medium: https://medium.com/@deependra.verma00/r2-score-linear-regression-e095a1188e87