**Prediktiv analys av datasetet ”housing”**

Kunskapskontroll för AI 2 -del 2

Sandra Persson

OPA22, höst 23



**Innehåll**

Del 1: Teoretiska frågor 2

Del 2: Rapport 4

Introduktion 4

Bakgrund 4

Syfte och frågeställning 4

Beskrivning av data 4

Metod och modeller 5

Preprocessing 5

Modeller 5

Resultat och analys 6

Slutsats 6

Del 3: Redogörelse 7

**Del 1: Teoretiska frågor**

1. *Hur kan vi definiera ”Maskininlärning”?*

Som namnet antyder innebär det att ge maskiner förmågan att lära sig. Alltså att bygga system som blir bättre med erfarenhet, genom att de själva upptäcker mönster i data utan att vi explicit programmerar dem, och att sedan kunna använda den informationen för att göra prediktioner utifrån ny data.

1. *Vad är supervised learning?*

Då vi under inlärningen använder data som innehåller facit, alltså data där vi vet vad variablerna har för värden eller etiketter. Ett exempel är de dataset vi använder till inlämningsuppgiften, där vi vet vilka siffror varje handskrivet nummer är och vilket huspris varje observation har. Uppgiften blir då att predicera vilken etikett eller värde ny data har.

1. *Vad är skillnaden mellan Regressionsproblem och Klassificieringsproblem?*

Definitionen enligt boken är att regressionsproblem är ett problem där vi förutspår data som är kontinuerlig, och ett klassificeringsproblem är ett problem där vi förutspår data som är diskret. Jag tycker personligen att den definitionen är förvirrande och föredrar att tänka på regressionsproblem som ett problem där den beroende variabeln är numerisk, kan anta ett värde på en glidande skala och där intervallerna mellan varje möjligt värde är likadan. Det skulle alltså kunna var både kontinuerlig och diskret data, t.ex. antal kunder ett företag har.

Ett klassificeringsproblem är helt enkelt ett problem där den beroende variabeln är en kategori, alltså där det finns ett begränsat antal värden den kan ha.

1. *Kan du ge ett exempel på vad regressionsmodeller respektive klassificeringsmodeller kan användas till?*

En regressionsmodell skulle kunna användas till att predicera hur mycket en kund kommer att spendera baserat på annan data vi har om kunden.

Det hade varit roligt att använda en klassificeringsmodell för att styra en liten Arduino-robotbil med hjälp av tanken. Man hade kunnat fästa elektroder på huvudet för att mäta hjärnans elektriska aktivitet och sedan använda modellen för att skilja på när en person tänker på höger/vänster/framåt/bakåt. Ett hemmabyggt Brain-Computer Interface helt enkelt.

1. *Vad är Root Mean Squared Error (RMSE)?*

RMSE används som ett mått för att kunna utvärdera hur bra en regressionsmodell är, alltså hur bra den har lyckats predicera data. För att få en intuitiv förståelse för vad RMSE är skulle man kunna tänka på det som ett mått som visar medelvärdet för hur mycket fel modellen gör, där vi med fel menar skillnaden på det faktiska värdet av den beroende variabeln jämfört med det predicerade värdet. Mer specifikt visar den standardavvikelsen för felen.

RMSE räknas ut genom att först räkna ut medelvärdet av de kvadrerade felen (Mean Squared Error), och sedan ta roten ur det värdet vilket gör det lättare att tolka för oss människor. Ett mindre RMSE är bättre än ett större, eftersom det betyder att modellen har gjort mindre fel.

1. *Vad är en ”confusion matrix”?*

En confusion matrix är en matris som kan användas för att utvärdera resultatet av en klassificeringsmodell. Den visar frekvenser av predicerade klasser jämfört med de faktiska klasserna, och kan alltså bland annat användas för att räkna ut hur många av prediktionerna som var korrekta. Man kan få ut flera intressanta mått från matrisen, såsom precision (hur stor andel av de positiva prediktionerna var korrekta) och recall (hur stor andel av en klass lyckades vi predicera korrekt).

1. *Om man delar upp datan i träning, validering och test – hur används respektive del?*

Träningsdelen används för att träna modeller, det är alltså den del man använder till metoden fit() i scikit learn. Valideringsdatan används för att utvärdera och jämföra modeller, inklusive för att välja hyperparametrar. När vi väl har tränat och valt en modell används slutligen testdatan för att se hur väl vår modell generaliserar, alltså hur bra den kan predicera ny data.

1. *Vad är en parameter för något? Ge ett exempel.*

En parameter är något som lärts från datan, alltså något vi får fram när vi tränar modellen. Ett exempel är *θ1* i linjär regression.

1. *Vad är en hyperparameter för något? Ge ett exempel.*

En hyperparameter är något som används för att styra inlärningen, alltså något som sätts innan träningen. Ett exempel är antal träd (n\_estimators) i scikit learns Random forest regressor.

1. *Vad är GridSearchCV i Scikit-learn? Ge ett exempel på vad det kan användas till.*

GridSearchCV är en metod som används för att välja hyperparametrar. Den använder sig av cross validation, alltså ett sätt utvärdera modellen genom att dela upp datan i flera mindre delar och sedan utvärdera på varje del separat. Varje möjlig kombination av olika hyperparametrar, som man själv specificerar, utvärderas. Om man använder standardinställningen kommer GridSearchCV att träna om modellen på all träningsdata när den har hittat den bästa möjliga kombinationen av hyperparametrar.

**Del 2: Rapport**

**Introduktion**

**Bakgrund**

Året är 1990. Du har precis sett ett avsnitt av The Fresh Prince of Bel Air, tagit på dig dina ljusa jeans och din rutiga flanellskjorta och bestämt dig för att Kalifornien verkar vara ett riktigt coolt ställe att bo på. Det finns ett problem bara. Du har ingen aning om vad ett hus i Kalifornien kostar.

Hade det här varit idag hade du helt enkelt öppnat din dator, googlat fram data om hus i Kalifornien och använt dig av maskininlärning för att snabbt kunna förutspå vad ett hus som verkar intressant kommer att kosta.

Datasetet jag kommer att använda i den här analysen kommer från boken *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems* (Géron, 2019), vilket är ett anpassat set som från början kommer från 1990. Det är ett klassiskt dataset för att lära sig Machine Learning och visar bland annat median-huspriser angivet i US dollar för varje distrikt, eller ”block” i Kalifornien.

**Syfte och frågeställning**

Syftet med analysen är att bygga en modell som kan predicera median-huspriser utifrån de andra variablerna i datasetet. För att hitta en modell som kan predicera huspriser så korrekt som möjligt kommer jag att börja med att jämföra flera olika typer av modeller i Scikit learn och sedan arbeta vidare med den som visar bäst resultat utifrån RMSE scores.

**Beskrivning av data**

Jag börjar med en kort EDA för att undersöka vilka slags variabler datan har. Samt för att kontrollera om det finns saknad data, att värdena ser rimliga ut och hur distributionen ser ut för varje variabel.

Datasetet har totalt 10 variabler, varav en nominal (ocean proximity) och resterande numeriska (longitude, latitude, housing median age, total rooms, total bedrooms, population, households, median income och median house value). Variabelnamnen är beskrivande men värt att minnas är att varje observation gäller för ett distrikt i Kalifornien. Antal observationer, alltså antal distrikt, är 20 640.

Variabeln ’total bedrooms’ har saknade värden, vilket jag kommer att behöva behandla under preprocessing. Histogram och boxplots visar att flera variabler har en ”skewed” distribution samt flera outliers. Den beroende variabeln, ’median house value’, har en intressant distribution där max värdet är samma som mode, det vanligaste värdet är alltså det högsta värdet. Jag ser i boken att det beror på att datan kar kapats vid 500001 dollar.

Innan jag tittar på samband mellan variabler, delar jag upp datan i ett tränings-set och ett test-set för att undvika att ”tjuvkika” på mönster i testdatan. En plot över korrelationen mellan de numeriska variablerna i träningsdatan visar att den beroende variabeln bara visar ett linjärt samband med medianinkomst. Den visar även att flera av de oberoende variablerna visar en stark korrelation med varandra vilket innebär att en variabel i varje par antagligen är överflödig eftersom den inte tillför någon information. Till exempel visar antal rum och antal sovrum en stark positiv korrelation. Latitud och longitud visar en stark negativ korrelation, vilket helt enkelt beror på den geografiska formen av Kalifornien.

I vår kursbok skapar de nya variabler genom att kombinera variabler, vilket känns som en bra idé eftersom det skapar mer informativa variabler. Jag väljer att skapa variablerna ’bedrooms\_per\_rooms’ som visar antal sovrum per rum i distriktet, och ’rooms\_per\_household’ som visar antal rum per hushåll. Båda variablerna visar en starkare korrelation med vår beroende variabler jämfört med originalvariablerna. Framförallt visar ’bedrooms\_per\_rooms’ en korrelation på r = -0.26.

**Metod och modeller**

**Preprocessing**

Innan jag går vidare med modellval delar jag upp datan i x och y data samt skapar en pipeline för preprocessing. Saknade värden i numeriska variabler ersätter jag med medianen. Anledningen till att jag väljer median som strategi är att den variabel som har saknade värden, total bedrooms, visar en skewed distribution. De numeriska variablerna skalar jag med hjälp av robust scaler, som skalar datan genom att subtrahera medianen och sedan dividera med IQR, vilket gör den mer robust mot outliers jämfört med standard scaler. Den kategoriska variabeln gör jag om till en numerisk med hjälp av One-hot-encoding. Två varianter av datan skapas, en som använder dummy variable encoding (genom drop=’first’) för att användas i linjär regression, och en vanlig one-hot-encoding för övriga modeller.

**Modeller**

Som ett första steg väljer jag att jämföra flera olika modeller genom att bara använda deras default parametrar. Vissa av modellerna skulle kunna ses som onödiga (tre olika varianter av regulariserad linjär regression) men jag väljer att använda alla helt enkelt för att lära mig mer om varje modell och för att kunna jämföra score för alla. De typer av modeller jag testat är följande:

* Linjär regression: Predicerar den beroende variabeln genom att beräkna en viktad summa av de oberoende variablerna, inklusive ett intercept. Målet är att hitta de vikter och det intercept som minimerar de kvadrerade felen mellan vad modellen predicerar och de faktiska värdena, det vi kallar en ”cost function”.
* Ridge regression: en regulariserad variant av linjär regression som läger till ett extra straff till cost function. Det gör att modellen inte överanpassar sig till träningsdatan (”overfitting”), vilket skulle kunna leda till dålig generalisering.
* Lasso regression: Även det en regulariserad linjär regression som lägger till ett straff till cost function. Lasso tenderar även att sätta vikterna för de minst viktiga variablerna till nära 0, vilket gör att den automatiskt väljer de viktigaste variablerna.
* Elastic net regression: En mix av Ridge och Lasso där man själv kan välja ration mellan de båda.
* Linear SVR: En typ av Support Vector Machine som kan användas för linjär regression.
* SVR poly: En icke-linjär Support Vector Regression (SVR) som använder sig av en polynomial kernel
* SVR rbf: Även det en ickelinjär SVR. Använder sig av en ”radial basis function” kernel.
* Decison tree regressor: Predicerar den beroende variabeln genom att skapa en träd-likande struktur som består av ”decision nodes” och ”leaf nodes”. Vid varje decision node tas ett beslut om vilken nod man ska röra sig vidare till baserat på ett tröskelvärde hos de oberoende variablerna. En leaf node visar sedand et predicerade värdet hos y.
* Random forest regressor: En metod som använder sig av flera decision trees, alltså en form av ”ensemble learning” där flera olika modeller används.
* K Nearest Neighbour regressor: Predicerar y baserat på dess närmsta grannar, alltså de observationer som mest liknar den nya observationen.

**Resultat och analys**

RMSE för varje modell samt standardavvikelse för RMSE kan ses i tabellen nedan.

|  |  |  |
| --- | --- | --- |
| Modell | Mean RMSE | Standard deviation |
| Linear regression | 67803.18 | 1591.58 |
| Lasso regression | 67803.18 | 1591.17 |
| Ridge regression | 67802.25 | 1591.27 |
| Elastic net regression | 80806.85 | 1119.34 |
| Linear SVR | 215504.84 | 2045.03 |
| SVR poly | 118707.75 | 1221.52 |
| SVR rbf | 118465.32 | 1616.93 |
| Decison tree regressor | 70806.41 | 1641.67 |
| Random forest regressor | 49880.43 | 380.02 |
| K Nearest Neighbour regressor | 63220.48 | 1413.94 |

Baserat på RMSE väljer jag att gå vidare med Random Forest Regressor som visar det lägsta värdet på 49 880.43. Den har dessutom en lägre standardavvikelse, 380.02, jämfört med de andra modellerna.

Eftersom vår EDA visade att många variabler inte hade ett samband med vår oberoende variabel väljer jag att börja med att göra en feature selection med hjälp av ”feature importance” i Random forest. Jag börjar med att göra ett ganska hårt urval där jag bara sparar de 8 variablerna med bäst score. Detta för att ta bort både de sämsta variablerna samt total\_bedrooms och total\_rooms som vi ju använt till nya variabler. Det visar sig vara en dålig strategi eftersom det ger ett högre RMSE på 50 221.24. Jag testar även att behålla de 11 bästa variablerna men även det ger sämre RMSE.

Jag behåller därmed alla features i datan och går vidare med optimering av min valda modell med hjälp av Grid search CV. De bästa hyperparametrarna visade sig vara n\_estimators=200 , alltså 200 träd, max\_depth=None , min\_samples\_leaf=1 och max\_features=4. Jag tränar om den optimerade modellen på test datan för att se hur väl den generaliserar. RMSE på testdatan är 50 281.11

**Slutsats och förslag på potentiell vidareutveckling**

Den slutliga finetuneade modellen har ett RMSE på 50 280.11, det vill säga sämre än RMSE på valideringsdatan, vilket är väntat. För att kunna tolka RMSE jämför jag det med y-test, alltså testdelen för vår beroende variabel, som har ett medelvärde på 205 500. RMSE är 36.94% i relation till medelvärdet. Jag tror att det skulle vara möjligt att bygga en bättre modell, alltså en som predicerar värdet bättre, med hjälp av nedanstående förslag till vidareutveckling.

Från kartan i boken ser det ut som att plats för huset verkar ha betydelse för priset, men frågan är om latitud och longitud är bra prediktorer. Det hade varit intressant att slå ihop dessa två till en ny variabel med hjälp av klustring. Det hade också varit intressant att testa ett neuralt nätverk samt fler ensemble learning modeller genom att välja ut och optimera några av de modeller jag testat och sedan använda dessa. Jag hade även velat arbeta vidare med optimering av modeller för att lära mig mer om vilka hyperparametrar jag borde välja och hur det påverkar modellen.

**Referenser**

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems* (2nd ed.). Sebastopol, CA: O'Reilly Media.

**Del 3: Redogörelse**

*Utmaningar under arbetet*

Den största utmaningen var nog att jag kände mig betydligt mindre motiverad än jag brukar eftersom vi redan svarat på de teoretiska frågorna under en lektion och redan använt oss av datasetet. Men sen kom jag fram till att det fortfarande kommer att vara väldigt lärorikt att laborera med datan och testa olika metoder och modeller.

Under analysen tyckte jag att det svåraste var att välja i vilken ordning alla steg skulle göras. Skulle jag börja med feature selection och sedan testa några olika typer av modeller eller skulle jag först utvärdera modellerna och sedan göra en feature selection om det behövdes. Jag bestämde mig till slut för en väldigt enkel metod där jag bara loopade genom olika modeller och valde den bästa, gick vidare till feature selection och sedan modelloptimering.

*Vilket betyg jag anser att jag skall ha och varför*

Jag anser att jag bör ha VG på uppgiften eftersom jag utöver att ha valt datasetet housing även resonerat kring vilka metoder jag använt mig av och varför, lagt tid på att förstå varje steg i analysen (även om jag förstås bara skrapat på ytan av allt som finns att lära sig), samt reflekterat över och analyserat både datan i sig och de resultat jag fått.

*Tips jag hade ”gett till mig själv” i början av kursen nu när jag slutfört den*

Jag skulle ha lagt mer tid på att skriva kod redan från början och på att testa alla typer av modeller. Däremot är jag nöjd med att jag började skapa flashcards redan i början av kursen, vilket gjorde alla begrepp mindre förvirrande och hjälpte för att komma ihåg det vi lärt oss förra terminen.