The Predikator

En brutalt effektiv utredning av huspriser i 90-talets Kalifornien



Förnamn Efternamn

NBI/Handelsakademin

Kunskapskontroll, AI 2 – del 1

202406

Abstract

Summary:

Our project investigates the use of machine learning trained on the California Housing dataset from the Statlib repository to predict house prices. This dataset consists of observations from the 1990 California census, with each observation representing a ‘block group’ – a geographical zone and its accompanying data consisting of median house value (our target value), median income, housing median age, total rooms, total bedrooms, population, total households, latitude, and longitude. We carried out an initial data analysis of the dataset, looking for extreme observations (outliers) and missing data and removing or imputing respectively. We then carried out an exploratory data analysis to explore which features would be most promising when used in potential models and which new features could be created and used in said models. We then trained and evaluated the following three models on the data in Jupyter Notebooks using scikit-learn (supported by Pandas, Matplotlib and Seaborn libraries for computation, plotting and visualisation): linear regression, support vector regression and random forest regression.

Our results show that a random forest regression, with the hyperparameters max\_features: 2, n\_estimators: 30 (all other hyperparameters being the default values in scikit-learn 1.5.0) produces a result with the lowest RMSE of all three models(42590**)** and thus the lowest error rate (22.5%). When the error rate was calculated using the mean of the target value derived from the whole dataset, including outliers, we achieved an error rate of 20.6%.

Förkortningar och Begrepp

LR = Linear Regression (linjär regression)

ML = Machine Learning (maskininlärning)

RFR = Random Forest Regression

RMSE = Root Mean Square Error

SVM = Support Vector Machines

SVR = Support Vector Regression

Innehållsförteckning

[1 Inledning 1](#_Toc169185351)

[1.1 Syfte och frågeställningar 1](#_Toc169185352)

[2 Teori 2](#_Toc169185353)

[2.1 Regressionsmodeller 2](#_Toc169185354)

[2.1.1 Linjär regression 2](#_Toc169185355)

[2.1.2 Support Vector Regression 2](#_Toc169185356)

[2.1.3 Random Forest Regression 2](#_Toc169185357)

[3 Metod 3](#_Toc169185358)

[3.1 Strategi 3](#_Toc169185359)

[3.2 Tekniska verktyg 3](#_Toc169185360)

[3.2.1 Jupyter Notebook 3](#_Toc169185361)

[3.2.2 GitHub 3](#_Toc169185362)

[3.2.3 Visual Studio Code 3](#_Toc169185363)

[3.3 Analys 3](#_Toc169185364)

[3.3.1 Förbereda data 3](#_Toc169185365)

[3.3.2 Använda Random Forest Regressor för att hitta de viktigaste variablerna/features 3](#_Toc169185366)

[4 Resultat och Diskussion 5](#_Toc169185367)

[4.1 Valideringsresultat av modeller 5](#_Toc169185368)

[4.2 Resultat av bästa modell på testdata 6](#_Toc169185369)

[4.3 Jämförelse mellan resultat med olika features 6](#_Toc169185370)

[5 Slutsatser 7](#_Toc169185371)

[6 Teoretiska frågor 8](#_Toc169185372)

[7 Självutvärdering 11](#_Toc169185373)

[Appendix A 12](#_Toc169185374)

[Källförteckning 12](#_Toc169185375)

# Inledning

AI och maskininlärning är på frammarsch och det finns många områden denna nya, spännande teknologi kan appliceras på. Historiskt har vi varit beroende av mäklare för att värdera fastigheter och förutsäga vilka priser de kan tänkas landa på. Som vi alla vet är mäklare ett opålitligt skrå med hemlighetsfulla och potentiellt mörka metoder, som vi gärna hade klarat oss utan om det bara fanns ett alternativ. Mäklaren fungerar lite som en svart låda – man stoppar in data och ut kommer en värdering, ett möjligt pris och ett chockerande arvode, utan att vi har någon förståelse för eller insyn i processen. Tänk om vi hade kunnat använda AI istället för att anlita en mäklare, till betydligt lägre kostnad? Någon som detta skulle kunna vara intressant för är Arnold Schwarzenegger, som investerat mycket i real estate i Kalifornien. Eftersom hans karriär till skillnad från huspriserna peakade på 90-talet kan AI-modeller tränade på data från 90-talets Kalifornien också intressera Arnie.

## Syfte och frågeställningar

Syftet med denna rapport är att använda ML för att skapa modeller som kan prediktera medelhuspriser så bra som möjligt. För att uppfylla syftet så kommer följande frågeställningar att besvaras:

1. Vilka features är viktigast och vilka kan slås ihop för att skapa nya, mer betydelsefulla features?
2. Vilken modell ger bäst prediktioner och med vilka hyperparametrar?
3. Hur kan vi utvärdera felmarginalen?

# Teori

## Regressionsmodeller

### Linjär regression

Linjär regression är en enkel och grundläggande modell inom maskininlärning och statistik. Modellen antar att det finns ett linjärt samband mellan den oberoende variabeln (i vårt fall oberoende variabler) och den beroende variabeln.

### Support Vector Regression

Support Vector Regression (SVR) är en variant av Support Vector Machines (SVM) som används för regressionsproblem. SVR syftar till att hitta en funktion som har högst av alla fel inom ett visst tröskelvärde (ε) samtidigt som den är så platt som möjligt.

### Random Forest Regression

Random Forest Regression är en modell som används för att förutsäga värden på en kontinuerlig variabel, såsom huspriser. I stället för att använda ett enda beslutsträd, kombinerar Random Forest många träd för att förbättra noggrannheten i sina prognoser.

# Metod

3.1 Strategi  
  
Vi har importerat data från en CSV-fil och rensat data genom att ta bort och fylla i rader som inte uppfyller vissa kriterier. Sedan gjorde vi korrelationsanalys och hittade de viktigaste features genom att använda Random Forest Regressor. När vi tränade och utvärderade modeller, använde vi Linear Regression, Support Vector Regressor och Random Forest Regressor. Slutligen hittade vi de bästa hyperparametrarna, tränade om modellerna och körde den bästa modellen på testdatan.

## 3.2 Tekniska verktyg

### 3.2.1 Jupyter Notebook

Det finns flera alternativ när det gäller verktyg för att skriva Python-kod. Jupyter Notebook är ett av de mest populära och kraftfulla verktygen för att skriva Python. Fördelen med Jupyter Notebook är att man kan använda det i webbläsaren, tillsammans med exempelvis Anaconda.

### 3.2.2 GitHub

GitHub är en webbplats som används för att hantera Git repositories och dela källkod, vilket gör att flera personer kan jobba och skriva kod tillsammans för att bygga en webbsida eller en applikation. Vi använder GitHub för att rätt hantera och arbeta med kod tillsammans.

### 3.2.3 Visual Studio Code

Visual Studio Code är en programutvecklingsmiljö med öppen källkod utvecklad av Microsoft för Windows, Linux och Mac OS. Vi valde att använda det eftersom vi kunde använda LiveShare och koda tillsammans, men det fungerade inte så bra som vi hade tänkt. Vi använde istället GitHub, skapade en branch och arbetade tillsammans i samma branch.

## 3.3 Analys

3.3.1 Förbereda data  
Vi har tagit bort > 500,000 från kolumnen median\_house\_value och > 51 från kolumnen median\_house\_age. Konverterat ocean\_proximity med ordinal encoding (0-3), tagit bort ISLAND och fyllt tomma värden med medelvärdet i kolumnen total\_bedrooms.

3.3.2 Använda Random Forest Regressor för att hitta de viktigaste variablerna/features  
Vi har använt Random Forest Regressor för att hitta de viktigaste features att sedan träna med olika modeller. Sedan skrev vi ut och visualiserade diagram med Matplotlib.  
  
3.3.3 Träna modeller  
Vi har skapat tränings-, validerings- och testdata och tränat Linear Regression, Support Vector Regressor och Random Forest Regressor på dem.  
  
  
  
3.3.4 Köra bästa modellen på testdatan  
Efter vi har tränat och validerat testdata med tre olika modellerna, körde vi den bästa testdata igen med Random Forest Regressor eftersom den visade sig ha bäst prestanda av de tre modellerna.

# Resultat och Diskussion

## 4.1 Valideringsresultat av modeller

|  |  |
| --- | --- |
| **RMSE för olika modeller** | |
| Linear Regression | 61273 |
| Support Vector Regressor | 81087 |
| Random Forest Regressor | 42590 |

Tabell 4.1: Root Mean Squared Error (RMSE) för de tre valda modellerna.

En bild som visar skärmbild, text, linje, Graf

Automatiskt genererad beskrivningEn bild som visar text, skärmbild, linje, Graf

Automatiskt genererad beskrivning

En bild som visar text, skärmbild, diagram, Rektangel

Automatiskt genererad beskrivningEn bild som visar text, skärmbild, linje, Graf

Automatiskt genererad beskrivning

Figur 4‑2 Predikterande värden i SVR

Figur 4‑1 Predikterande värden i Linjär regression

Figur 4‑4 Jämförelse av rmse och error rate i olika modeller

Figur 4‑3 Predikterande värden i RFR

## 4.2 Resultat av bästa modell på testdata

Vår bästa modell blev som också framgår av visualiseringarna Random Forest Regressor. Efter att ha kört modellen med hyperparametrarna Max\_features: 2 och N\_estimators: 30 fick vi en **RMSE på 42590**. Vi fick också **en felmarginal på 22.5%** beräknad på data-setet utan outliers, och det är också den felmarginalen som visas i figurerna.

**Felmarginalen beräknad på hela data-setet blev 20.6%**

## 4.3 Jämförelse mellan resultat med olika features

Resultatjämförelse mellan två olika data-set där skillnaden är olika features som framgår av *figur 4-7*

*En bild som visar text, skärmbild, Rektangel, diagram

Automatiskt genererad beskrivningEn bild som visar text, skärmbild, diagram, Rektangel

Automatiskt genererad beskrivning*

Figur 4-5 Features exklusive longitud och latitud

Figur 4‑5 Features exklusive longitud och latitudFigur 4‑5 Exklusive longitud o latitud

Figur 4‑5 Features exklusive longitud och latitud

Figur 4‑6 Features inklusive longitud och latitud

En bild som visar text, skärmbild, Teckensnitt, nummer

Automatiskt genererad beskrivning

Figur 4‑7 Skillnaden på de olika data-seten

# Slutsatser

Vilka features är viktigast och vilka kan slås ihop för att skapa nya, mer betydelsefulla features?

De features som har störst påverkan på y-värdet median house value är median income, ocean proximity, population per household, samt longitude och latitude. Population och households får större inverkan som ett viktat värde än var för sig. Dessa samband framträder tydligare genom att utvärdera feature importance med hjälp av RFR än genom att analysera korrelationsmatriser, som bara plockar upp linjära samband.

Vilken modell ger bäst prediktioner och med vilka hyperparametrar?

Random Forest Regressor ger bäst prediktioner, betydligt bättre än LR och framförallt SVR, med hyperparametrarna max\_features 2 och n\_estimators 30. Egentligen ger allt högre n\_estimators något bättre resultat, men till priset av allt mindre tidseffektivitet. ”Diminishing returns”, alltså.

Hur kan vi utvärdera felmarginalen?

Felmarginalen kan utvärderas med RMSE, som ger en ungefärlig differens, och dividerat med medelvärdet av median\_house\_value multiplicerat med 100 får vi en felmarginal i procent.

Slutligen kan vi konstatera att valet av features och modeller kan ha stor påverkan på utfallet, vilket vi verkligen har märkt under arbetets gång. Tydliga korrelationer kan visa sig ha mindre betydelse medan korrelationer som är svårare att upptäcka kan visa sig vara viktiga, som vi märkte i skillnaderna mellan vad korrelationsmatriser och feature importance visade.

# Teoretiska frågor

**1. Vad är en CSV fil?**

En särskild filtyp som är i form av en tabell med värden separerade med kommatecken (CSV är en akronym som står för det engelska namnet ”comma-separated values”: kommaseparerade värden). Filen kan skapas eller redigeras i Excel.

**2. Kalle delar upp sin data i ”Träning”, ”Validering” och ”Test”, vad används respektive del för?**

Träning används för att träna modeller. Validering används för att utvärdera modellerna och välja den bästa. Test används för att testa den valda modellen och få ett estimat av "generalization error".

**3. Julia delar upp sin data i träning och test. På träningsdatan så tränar hon tre modeller;**

**”Linjär Regression”, ”Lasso regression” och en ”Random Forest modell”. Hur skall hon**

**välja vilken av de tre modellerna hon skall fortsätta använda när hon inte skapat ett**

**explicit ”validerings-dataset”?**

Hon kan använda K-Fold cross validation.

**4. Vad är ”regressionsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden?**

Ett regressionsproblem innebär att förutsäga en kontinuerlig variabel. Linjär regression, support vector machine regression och random forest regression är exempel på modeller som kan användas till regressionsproblem. Några potentiella tillämpningsområden kan vara att förutsäga en persons lön baserat på andra egenskaper såsom ålder, utbildning mm, eller huspris baserat på husåldern, geografiskt läge, medianlön i området där huset finns mm.

**5. Hur kan du tolka RMSE och vad används det till:**

A square and square formula

Description automatically generated with medium confidence

RMSE, root mean squared error, är roten ur medelvärdet för felmarginalen i kvadrat hos prediktionerna för en modell. Ju lägre RMSE är desto bättre är modellen på att förutsäga target value.

**6. Vad är ”klassificieringsproblem? Kan du ge några exempel på modeller som används och potentiella tillämpningsområden? Vad är en ”Confusion Matrix”?**

Ett klassificeringsproblem handlar om att förutsäga vilken kategori något tillhör.

Modeller: Logistisk Regression, Decision Tree, Random Forest, SVM.

Tillämpningar: Spamfilter, Bildigenkänning, Diagnoser

Confusion Matrix: En tabell som visar hur bra en modell är på att gissa rätt.

**7. Vad är Streamlit för något och vad kan det användas till?**

Streamlit är ett verktyg för att enkelt skapa och dela webbaserade appar för data och maskininlärning.

**8. Vad kännetecknar en nominell variabel?**

En nominell variabel kategoriserar data utan rangordning dvs att varje kategori är unik och det går inte att jämföra eller sortera baserat på kategorierna. Till exempel är färger (röd, grön, blå) nominella variabler.

**9. Vad kännetecknar en ordinal variabel?**

En ordinal variabel kategoriserar också data men till skillnad från en nominell variabel har en ordinal variabel rangordning. Till exempel är åldersintervall (’under-18’, ’18-24’, ’25-34’, ’35-44’, ’över 64’ år) ordinala variabler.

**10. Förklara (gärna med ett exempel): Ordinal encoding, one-hot encoding, dummy variable encoding.**

**Ordinal encoding**: Kategorier omvandlas till heltal med ordning.

|  |  |
| --- | --- |
| Kategori | Tal |
| Låg | 1 |
| Medium | 2 |
| Hög | 3 |

**One-hot encoding**: Varje kategori omvandlas till en binär kolumn. Exempel: "Röd", "Blå", "Grön" blir:

|  |  |  |
| --- | --- | --- |
| Röd | Blå | Grön |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

**Dummy variable encoding**: Liknar one-hot encoding men en kategori (referenskategori) utelämnas för att undvika multikolinjäritet. Exempel: "Röd", "Blå", "Grön" blir:

|  |  |
| --- | --- |
| Blå | Grön |
| 0 | 0 |
| 1 | 0 |
| 0 | 1 |

**11. Om vi använder vanlig linjär regression, skall vi använda one-hot-encoding eller dummyvariable encoding**

Dummy variable encoding (för att undvika multikolinjäritet).

# Självutvärdering

1. **Utmaningar ni haft under arbetet samt hur ni hanterat dem.**Vi stötte på många problem med GitHub, särskilt integrationen med Jupyter Notebooks. Vi hittade inte något mönster hos problemen eller ens någon bra lösning, så det var bara att härda ut.   
     
   Vi testade många verktyg för att kunna samarbeta i Jupyter Notebooks samtidigt, bland annat Jetbrains Datalore, Code With Me i PyCharm, och LiveShare i Visual Studio Code; vi tyckte att den sistnämnda var minst krånglig.
2. **Vilket betyg anser ni att ni skall ha och varför.**Det blir en spännande överraskning **☺**
3. **Något ni vill lyfta fram till Antonio?**Vi skulle vilja tacka för en jättebra kursstart, och hoppas på en lika braförsättning till hösten**.**Ser fram emot nästa kurs. Kör hårt, ha kul!
4. **Om ni hade gjort om kursen, hade ni gjort något annorlunda?**

Det hade aldrig varit fel med en vecka extra för att fördjupa sina kunskaper.

Appendix A  
  
  
  


# Källförteckning

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*.

*scikit-learn: machine learning in Python — scikit-learn 1.5.0 documentation*. (n.d.). <https://scikit-learn.org/stable/>