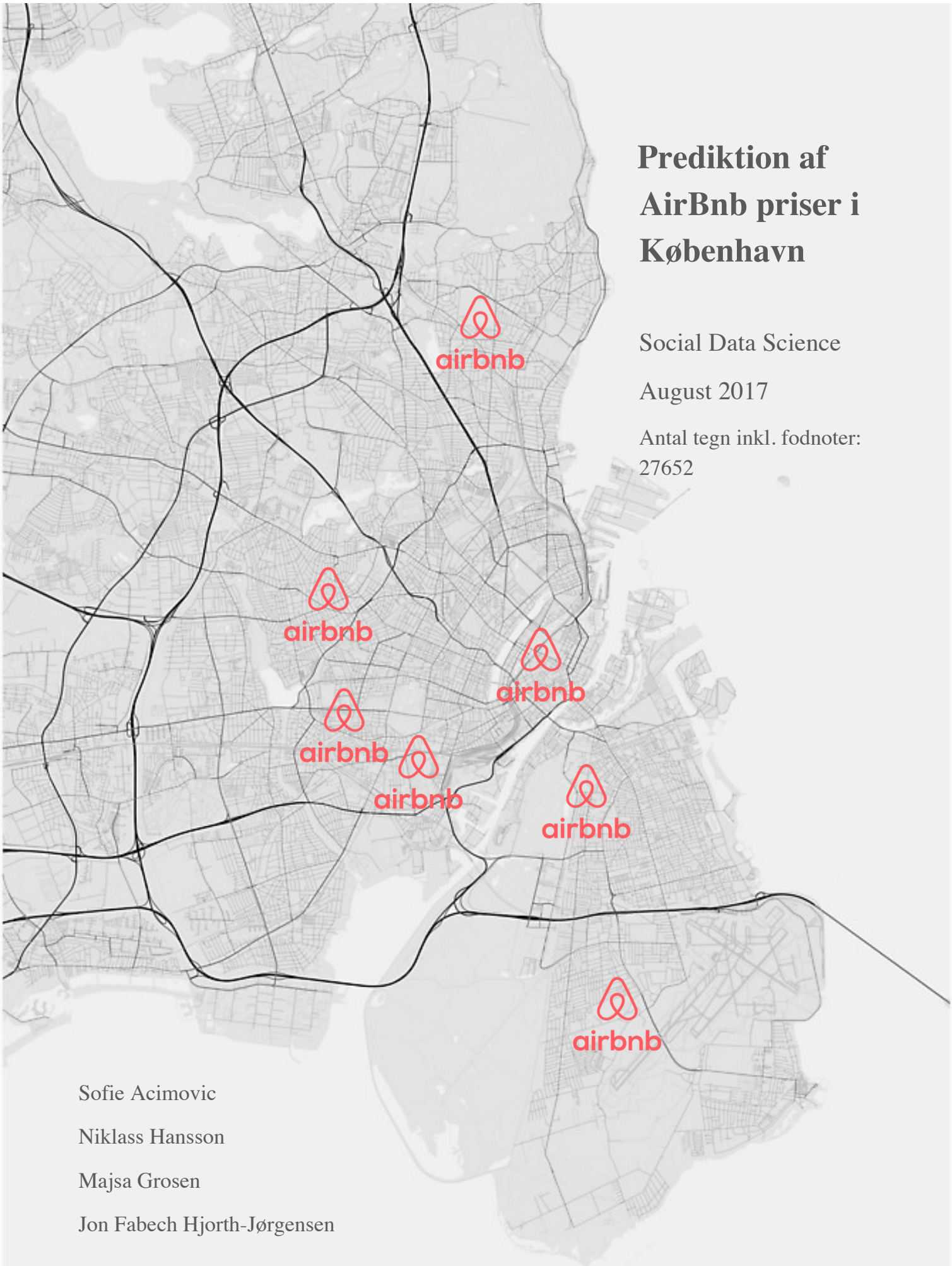


Prediktion af AirBnb priser i København

Social Data Science

August 2017

Antal tegn inkl. fodnoter: 27652

Sofie Acimovic

Niklass Hansson

Majsa Grosen

Jon Fabech Hjorth-Jørgensen

Indhold

1 Indledning

Debatten om udlejningsportalen Airbnb tog fart i Danmark i foråret 2017 efter at blandt andet Horesta på baggrund af tal fra AirBnb kunne konkludere, at der nu er flere AirBnb-senge i Danmark, end der er hotelværelser (Olsen Straka 27-05-2017). AirBnb brander sig som en platform, der skal fremme en deleøkonomisk udvikling. Én af diskussionerne vedrørende AirBnb har gået på hvilke mennesker, der har gavn af deleøkonomien. (Fransen 23-08-2017 ; Gkiousou 28-02-2017), da lejen man kan tage for sin lejligheden varierer.
Formålet med opgaven er, på baggrund af denne motivation, at træne machine learning modeller til at forudsige priser for leje af AirBnb-lejemål ud fra lejemålenes karakteristika og udlejers beskrivelse heraf.
I opgaven beskrives dataindsamling, valg af afhængig variable og feature modifikationer indledningsvis. Herefter præsenteres den deskriptive statistik for de mest centrale features og opgavens modeller præsenteres og prædikationsevnerne herfor diskuteres. Afslutningsvis konkluderes det at Random Forest regressionsmodellen er den bedste til at prædiktere priser for AirBnb lejemål med de givne features.

2 Data

Vi vil i følgende afsnit redegøre for, hvorledes vi har indsamlet den empiri, som ligger til grund for følgende opgave, samt hvordan vi har genereret yderlige variable som supplement til den indsamlede empiri.

2.1 Indsamling af data

Udgangspunktet for vores undersøgelse er data scrapet fra AirBnbs hjemmeside. Ved web scraping indhentes data fra hjemmesidens html og gemmes lokalt som tekstfil. Web scraping som metodisk værktøj til dataindsamling implicerer lav reliabilitet, da data ikke kan reproduceres; i vores tilfælde kommer der bl.a. konstant nye lejemål og brugere til og fra hjemmesiden. Vores kode er dog vedhæftet opgaven, og det er på den måde muligt at granske vores tilgang.
Et generelt validitetsproblem for onlinetjenester er, hvad der ofte betegnes som *the ideal user assumption*, hvor man ikke tager højde for, at brugere på websiden kan give falske oplysninger. (Lazer Radford, 2017:32). Vi vurderer dog umiddelbart at risikoen for dette er relativt lille, da udbyttet af dette er begrænset bl.a. grundet AirBnbs ratingsystem (hvor lejere kan rate og kommentere lejemål).

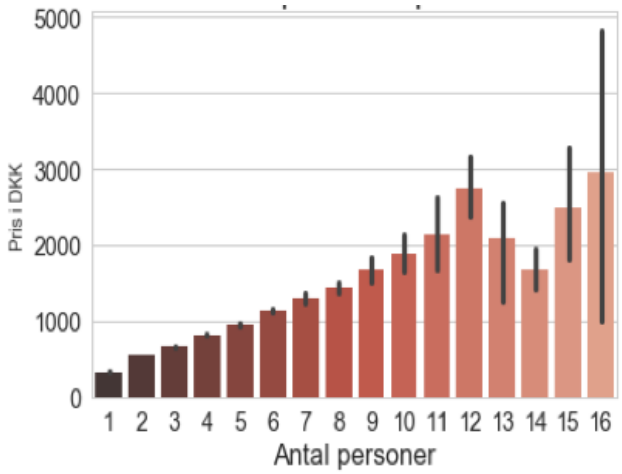
Data i denne opgave bygger på information om udbudte lejligheder til udlejning i Københavnsområdet på airbnb.com. Vi har selv bygget en web scraper der kan scrape AirBbnb. Vores analyse er dog baseret på data fra siden Inside AirBnb - et uafhængigt projekt, der via scraping indsamler data fra AirBnb, med det formål at stille data til rådighed for offentligheden (InsideAirBnb). Vi har valgt at anvende data fra Inside AirBnb, da de har indsamlet data for København af to omgange - både juni 2016 og 2017. Konkret anvendes "listings" datasættene. Dette giver os en større datamængde at arbejde med end når vi selv genererer data ud fra én dato, ét år. Desuden er det unødvendigt at replikere det arbejde der ligger i at bygge en web scraper, som kan hente de features som allerede eksisterer i datasættet fra Inside AirBnb. Reliabiliteten af InsideAirbnbs data understøttes af, at andre forskere tidligere har anvendt det i deres undersøgelser (Gant, 2016; Ma et al., 2017).
Ovennævnte data suppleres med data indsamlet via. egen web scraping af henholdsvis dsb.dk

og m.dk. Ved web scraping af dsb.dk og m.dk fås en komplet liste over adresser på DSB- samt metrostationer, hvilket omtales i afsnit 4.
Den primære datakilde indeholder udbudsaktiviteten for airbnb lejligheder i Københavnsområdet. Se bilag 4 for liste over variable i det oprindelige datasæt. Hver række indeholder al information om det pågældende lejemål, herunder pris (kr.), kapacitet, bydel, antal soveværelser, reviews samt længde- og breddegrader for de enkelte lejemål. Sidstnævnte giver os mulighed for at danne variable der måler afstanden fra lejemål og eventuelle relevante områder. Under databehandlingen fjernes irrelevante og overflødige variable og da data er indsamlet på tværs af to datoer, fjernes eventuelle dubletter.

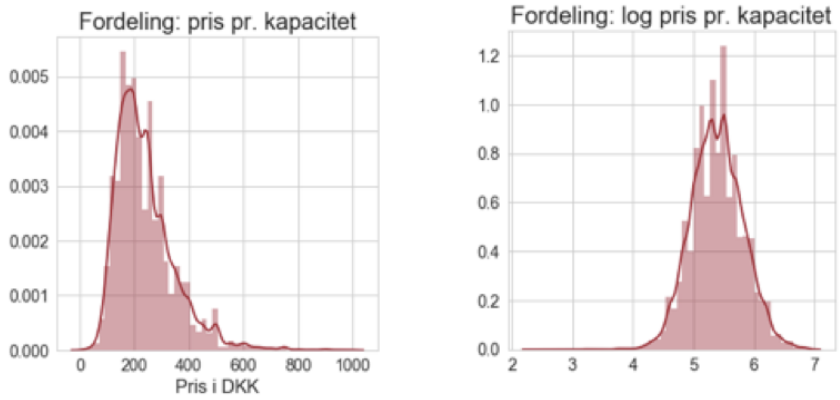
3 Afhængig variabel

Da opgaven har til formål at forudsige priser pr. nat for lejemål på AirBnb lejemål, benytter vi variablen *pris* som den afhængige variabel.
I figur 1 fremgår det som forventet at gennemsnitsprisen for et lejemål stiger med øget kapacitet. Den gennemsnitlige pris pr. nat for et lejemål er 723 DKK, mens den gennemsnitlige pris pr. nat pr. kapacitet er 240 DKK.
For at tage højde for denne effekt, anvendes pris pr kapacitet som analysens afhængige variabel.Plottes denne variabel ses der dog at forekomme ekstreme outliers, som kan skyldes målefejl. Disse fjernes ved at sætte en øvre og nedre prisgrænse der svarer til minimum pris pr. kapacitet pr. nat, observeret i datasættet fra egen web scraping samt en øvre grænse på 1500 kr. pr. nat. Dette resulterer i, at 33 observationer fjernes fra datasættet.

Figur 1: Kapacitet ifht. pris



Fordelingsplots for pris pr. kapacitet-variablen ses illustreret nedenfor. Her ses at en log-transformation af pris pr. kapacitet muligvis vil føre til et bedre fit i vores eventuelle modellering. Dens fordeling er noget højreskæv ifht. en typisk normalfordeling. Plottet til højre viser, at en log-transformation tackler skævheden(spredningen er mere normalfordelt), dog med en tung venstre hale. Der er dog rigtig få observationer med så lave værdier i pris pr. kapacitet så vi fortsætter med at bruge den transformerede variant i modelleringen.



4 Feature engineering

Vi har valgt at generere yderligere variable med henblik på, at kunne komme med mere præcise prædiktioner. I de følgende afsnit vil vi redegøre for hvordan vi har skabt henholdsvist kontinuere numeriske features samt tekst-features.

4.1 Numeriske features

Til den primære datakilde har vi beregnet og tilføjet en variabel for afstand til Rådhuspladsen. Denne variabel måler afstanden mellem Rådhuspladsen og det givne lejemål, og tilføjes idet prisen på et lejemål kan forventes at afhænge af afstanden til centrum. Denne variabel er lavet ved brug af funktionen *vincenty*, der giver afstanden mellem de givne længde- og breddegrader i km.
Ydermere suppleres datasættet med det web scrapede data for danske DSB- og metrostationer. Her har vi web scrapet stationernes adresser transformeret dem til de givne længde- og breddegrader ved hjælp af geocoding. Herefter beregnes afstanden til de enkelte stationer og for hvert lejemål gemmes data for den station med kortest afstand. Denne feature inkluderes i datasættet idet lejerens mulighed for at komme rundt i byen kan forventes at påvirke prisen i mindst lige så høj grad som bydel og afstand til centrum.
Airbnbs Fortrolighedspolitik omfatter, at udlejernes præcise adresse ikke deles med offentligheden (AirBnb). Da lejemålenes koordinater benyttes i modelleringen, bør det overvejes om dette er etisk forsvarligt, i forhold til at opretholde brugernes anonymitet. Airbnb angiver at udlejerens præcise adresse først videregives ved booking og ikke offentliggøres uden samtykke. Der anvendes dog ingen information i modelleringen, der ikke er offentligt tilgængeligt fra hjemmesiden. Ud fra figur 4 i afsnit 5, vises fordelingen af alle lejemål i København. Her kan vi dog se, at nogle lejemål ligger placeret i vandet, hvilket indikerer, at koordinaterne er tilføjet støj. Derfor benyttes data uden videre.

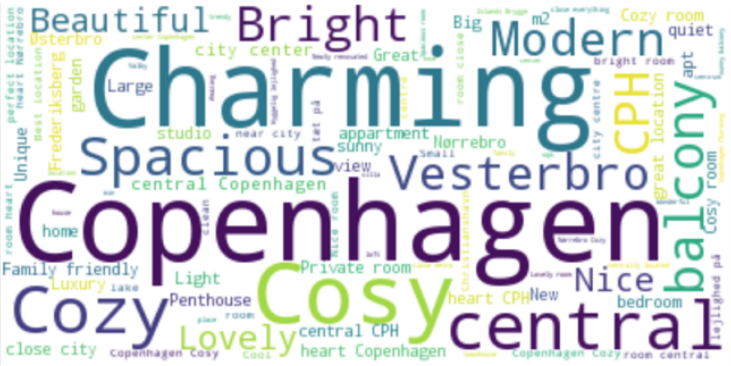
Data indeholder yderligere variablen amenities der er en concatenated string af lejemålets forskellige faciliteter. Ud fra denne variabel har vi genereret ni binære features - køkken, vaskemaskine, wifi, basics, tv, free parking, smoking allowed, morgenmad samt en variabel der tæller faciliteter udover disse. Vi inkluderer disse features, da vi forventer, at de påvirker den pris udlejer kan tage for lejemålet og dermed vores evne til at forudsige prisen.

4.2 Tekst-features

Til at understøtte vores prædiktioner, benyttes der også en tekstanalyse, som en del af vores features. Dette sker med en forventning om at lejemålsbeskrivelser der fremhæver bestemte karakteristika kan påvirke interessen for det givne lejemål.
For at indfange effekten af at betone bestemte elementer i lejemålets titel på AirBnBs hjemmeside har vi udformet seks binære features af ord. Disse fremhæver hver især forskellige træk ved det givne lejemål (se bilag 5).

Vi har etableret de seks kategorier på baggrund af en fire-delt analyse. Først etableres et indledende overblik over hyppigt anvendte ord, vha. en wordcloud, illustreret i figur 3.

Figur 3: wordcloud for udlejningstitel



Dernæst dannes for det andet en liste over ord på over tre bogstaver, der fremgår af mere end 500 titler. Af denne tokenization-proces, hvor titlerne splittes op i enkeltord, fremgår mange af de samme ord. Ud fra denne liste af ord samt wordclouden udvælges en række ord, der er sigende for lejemålet. Vi frasorterer ord som *"apartment", "copenhagen" og "house"*, som allerede indfanges af andre variable, eller er ens for alle lejemål. Da titlernes længde er begrænset, er mængden af stopwords, ligeledes, begrænset.

Som tredje led udføres en mere dybdegående analyse vha. Natural Language Toolkit (NLTK)-funktionen *similar*, der fremhæver ord i det givne datasæt, der bruges på samme måde som de fremhævede ord fra wordclouden og ordlisten.
I trin fire konstrueres ordkategorierne. Der lægges her vægt på at danne kategorier af ord der beskriver de samme træk ved lejemålene og at finde ord der henfører samme mening (eller stavefejl) som ordene i trin 2 og på den måde er indikatorer for de samme egenskaber.

Ved at belyse brugen af ord fra flere forskellige vinkler opnår vi en større målingsvaliditet for kategorierne, da vi sikrer os, at ordene i hver kategori måler den samme egenskab. Vi styrker desuden reliabiliteten ved at skabe indikatorer der reelt set har samme betydning, da tilfældige målefejl ift. brug af eksempelvis *"nice"* fremfor *"lovely"* undgås. Når vi til sidst danner de endelige binære features stemmer vi relevante ord med NLTKs snowball stemmer, hvormed vi fanger alle udgaver af ordene (Foster et al, 2017: 191).

For også at indfange sammensætninger af ord, har vi udført bi- og tri-gram-analyser, af ord der ofte optræder sammen (Foster et al, 2017: 191). Da resultaterne af bi- og tri-gram-analyserne fremhævede sammensætninger af de samme ord, som vi fandt frem til af tokenizations processens første tre steps, har vi valgt ikke at arbejde videre med resultaterne fra bi- og tri-gramsanalysen.

Vi har desuden valgt ikke at udføre en TF-IDF-test, som vægter ords betydning ved at se på hvor ofte de anvendes i et givent dokument ift. en bredere palet af dokumenter (Foster et al, 2017: 192). Dette skyldes at langt hovedparten af ordene kun fremgår en gang af hver titel. Vi har valgt tilgangen beskrevet ovenfor i stedet for at fremhæve ord der går igen på tværs af beskrivelser. Endvidere gør titlernes længde og relativt simple opbygning, at vi vurderer, at mere avancerede supervised- og unsupervised learning metoder (Grimmer Stewart, 2013: 275) er overflødige.

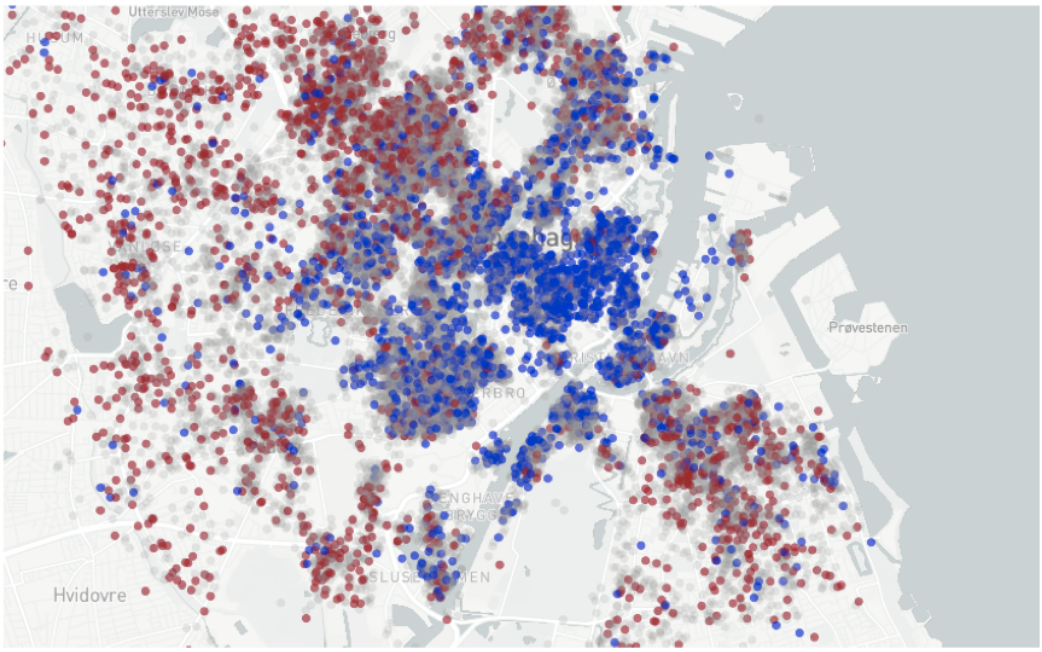
5 Deskriptiv analyse

I følgende afsnit vil vi indlendende undersøgelse vores endelige datasæt ved at fremlægge centrale variable og vise bivariate sammenhænge.

Det endelige datasæt indeholder 25.615 unikke lejemål i Københavnsområdet og den gennemsnitlige pris pr. kapacitet pr. nat er 240 DKK. Gennemsnitsprisen pr. kapacitet ses at variere henover både lejemålstype og bydele. Den hyppigste lejemålstype er *apartments* som udgør 92 procent af datasættet, mens den hyppigste rumtype er *Entire home/apt.* som udgør 74 procent.

For at få et overblik over den geografiske fordeling af lejemålene samt priserne herfor, dannes der et kort ved brug af *plotly* og de givne koordinater. I figur 4 ses udbuddet af Airbnb lejemålene at sprede sig langt ud mod yderkanten af Københavnsområdet. Mens de blå plots angiver de 10% dyreste lejemål angiver de røde prikker de 10% billigste. De blå områder klynger sig hovedsageligt sammen omkring centrum af København. Dette stemmer ligeledes overens med figur 6 der viser en tendens til, at prisen pr. kapacitet er lavere for udlejningsmål med større afstand til Rådhuspladsen.

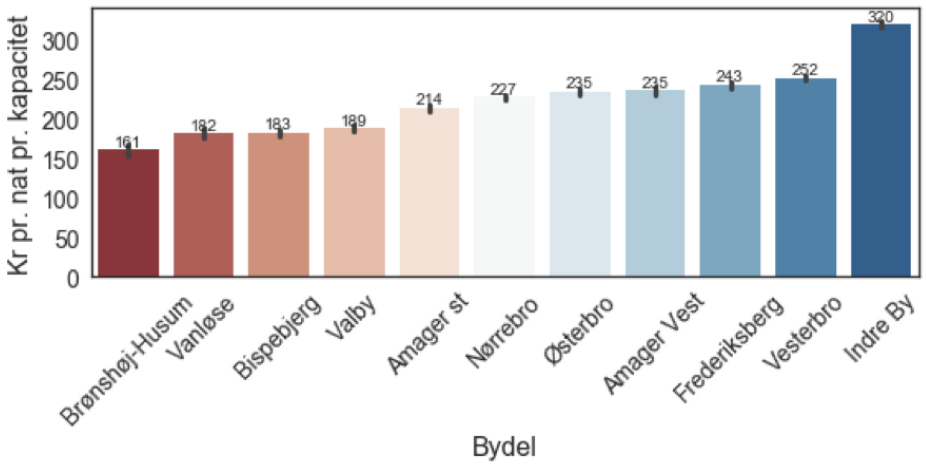
Figur 4: Pris pr. kapacitet - billigste og dyreste decil



Da datasættet ligeledes indeholder en klassificering af bydelene i København, kan fordelingen af gennemsnitsprisen ses på dette niveau i figur 5. Her ses det tydeligt at bydelen hvori lejemålet

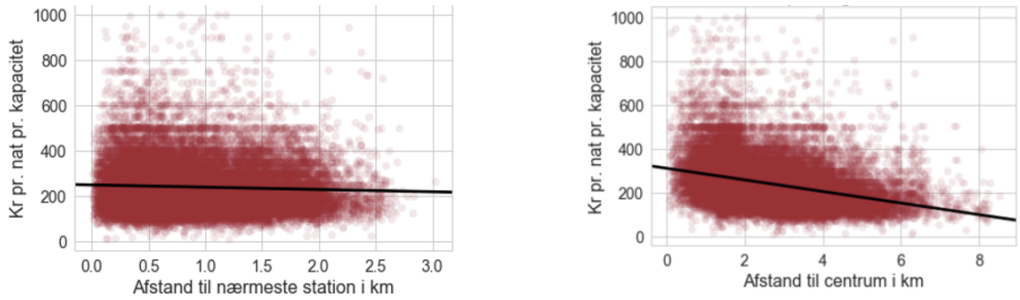
ligger, har en stor betydning for prisen. Indre By har en klart højere gennemsnitspris end de resterende bydele og at priserne omkring indre by er relativt ens. De gennemsnitlig billigste lejemål ses at befinde sig i yderkanten af København.

Figur 5: Pris pr. kapacitet fordelt på bydel



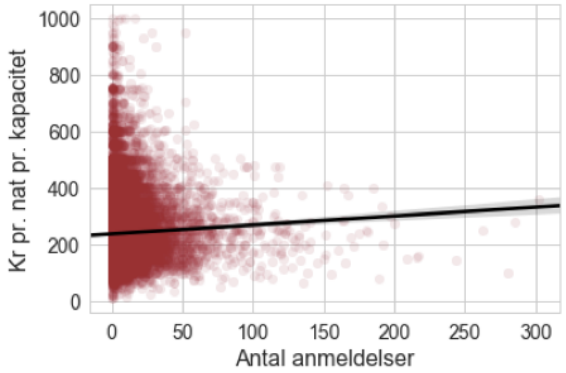
Ligeledes ses prissætningen for lejemålet at være påvirket af afstanden til s-togs og metrostationer. Her ses prisen pr. kapacitet generelt at være højere, jo tættere lejemålet er på offentlige transportmuligheder.

Figur 6: Pris pr. kapacitet ifht. afstand til station og centrum



Udover lejemålets geografiske placering, ses sammenhængen mellem pris pr. kapacitet og antal reviews at være interessant. I figur 7 ses korrelationen mellem reviews og pris pr. kapacitet at være positiv. Lejligheder med få reviews ses derfor I gennemsnit at have en lavere pris pr. kapacitet end de lejligheder med flere reviews. En af grundene til dette kan være større gennemsigtighed jo flere reviews et givent lejemål får. Af figuren ses det også at spredningen er stor for de med få reviews, relativt til de med mange. Dette kan hænge sammen med at de dyreste lejligheder ikke bliver besøgt lige så ofte og derfor ikke får et efterfølgende review.

Figur 7: Pris pr. kapacitet ifht. antal anmeldelser



6 Modellering

Vi vil i dette afsnit vise, hvordan vi træner nogle udvalgte modeller til at prædiktere *log(pris_kapacitet)* for AirBnB-lejemål i Københavnsområdet, hvilke baselinemodeller vi sammenligner de trænede modeller med samt hvilket mål vi bruger som redskab i vores sammenligning. I analysen benyttes *Simple Average og Ordinary Least Squares med udvalgte features* som baseline-modeller. De resterende 4 modeller; *OLS regression, Ridge regression, Lasso regression og Random forest regression* baseres på det fulde feature set.

6.1 Tilgang til modelleringen

Da vores afhængige variabel er kontinuert anvender vi regression, og vi kan dermed ikke bruge de intuitive mål der typisk bruges i klassifikations-tilgangen til supervised learning problemer – så som en *confusion matrix¸ precision eller recall*. I stedet bruger vi den hyppigt anvendte *Root Mean Square Error (RMSE)*:

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

I vores tilfælde anvender vi RMSE som mål for forskellen mellem den afhængige variabels prædikterede værdi for hver enkelt observation i testdatasættet ud fra de valgte modeller og dens observerede værdi. Des lavere RMSE des bedre vil vi dermed være i stand til at forudsige prisen for udlejningsmål på Airbnb. Ved at sammenligne modellernes RMSE, findes den model der er bedst til at prædiktere uset data.
Før modellerne opbygges splitter vi vores data i to dele – en træningsdel der udgør 70% af observationerne, og en testdel der udgør de resterende 30%.

I træningsdatasættet træner vi regressionsmodellernes hyperparametre ved hjælp af krydsvalidering. Hyperparametre er en klasse af parametre der skal fastsættes inden træning af data, modsat koefficienter, hvis værdi udledes undervejs.
I krydsvalideringen har vi splittet vores træningsdatasæt i 10 folds. For hvert fold angiver vi en specifik værdi for hyperparameteren. Modellen trænes herefter på 9 ud af de 10 fold og testes herefter på det sidste fold, hvorefter RMSE udregnes. Denne proces gentages indtil alle folds har fungeret som testfolds, med varierende værdier for hyperparameteren for hver iteration af

krydsvalideringen. Den optimale værdi er den med den laveste tilsvarende RMSE. Derefter opbygges modellen med hele træningsdata, de fundne optimale parametre indsættes, og derefter udregnes RMSE på uset testdata.
De modeller der ikke indeholder hyperparametre trænes på træningsdata, og deres RMSE udregnes på test data.
For at kunne bruge vores data sammen med Pythons *sckit* pakke, omkodes niveauerne i de kategoriske variable vha. *pd.get_dummy* metoden. Dette medfører, at vi for de 3 første katagoriske variable, *host_is_superhost, host_has_profile_pic og host_identity_verfied* får 3 nye binære variable, mens vi for hver de af 4 sidste katagoriske variable, *neighbourhood_cleansed, property_type, room_type og bed_type* får flere binære variable, da disse har flere niveauer end 2. Dette medfører, at vores samlede antal features stiger fra 43 til 61.

6.2 Simple Average

Simple average er, som navnet antyder, den simpleste model. For at finde RMSE, trækkes værdien for hver enkelt observation i *log(price_kapacitet)* i test data fra middelværdien af tilsvarende output i træning data. Heraf udregnes RMSE for forskellen mellem de to. Vi må antage, at denne model underfitter pga. dens simple form og dermed ikke fanger strukturen i data godt nok (Foster, 2017:154). RMSE for den simple gennemsnitsmodel estimeres til 0.429894.

6.3 OLS regression

Med OLS-regression er formålet at minimere residualledet, som er forskellen mellem den observerede og den prædikterede værdi på den afhængige variabel. Under en række stærke antagelser vil OLS være den bedste lineære unbiased estimator. I vores tilfælde kan vi dog ikke antage, at estimatoren er unbiased. Blandt andet antager vi, at flere af vores features er korreleret med modellens fejlled *(log(price_kap)*, og dermed residualerne, er ikke fuldstændig normalfordelte) samt at flere af de inkluderede features korrelerer med hinanden, hvilket vil skabe multikollinearitet. Korrelationen med fejlledet kan blandt andet skyldes, at der er flere karakteristika for udlejer, eks. alder og uddannelse, som vi ikke tager højde for.
Endvidere vil vi automatisk indføre bias for vores evaluering af OLS modellerne, da deres RMSE udregnes for test data.

6.3.1 OLS med udvalgte features

Vi har for denne model udvalgt en række features, som vi forventer har en særlig signifikant effekt på prisen. Disse er: *neighbourhood_cleansed_dummies, property_type_dummies, room_type_dummies, bathrooms, bedrooms, beds, bed_type_dummies og nærmeste_Stog*. For OLS-modellen med udvalgte features opnår vi en RMSE på 0.0.378592. Dette er en lavere værdi end for *Simple Average modellen* og derved en mere nøjagtig prædikation.

6.3.2 OLS med alle features

Denne model opbygges og testes med samme fremgangsmåde som for den begrænsede OLS-model, dog med det fulde sæt af features. Værdierne for de estimerede koefficienter kan ses i bilag 1. RMSE findes at være 0.368910 og er dermed igen en forbedring af vores evne til at prædiktere prisen for udlejning gennem Airbnb.

6.4 Ridge regression

Med Ridge regressionen minimerer vi følgende udtryk:

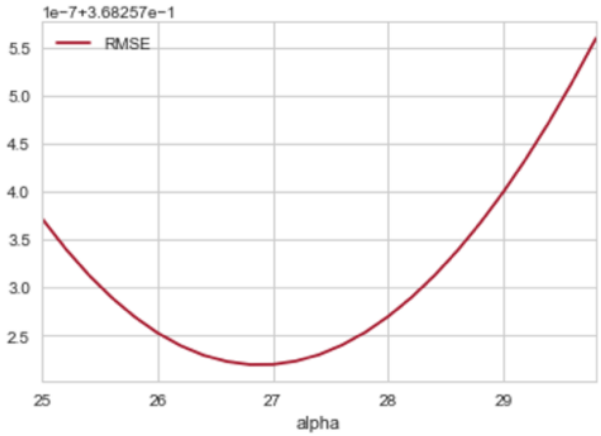
$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2$$

hvor *Residual Sum of Squares (RSS)* er OLS-delen, mens alpha er en hyperparameter der skal estimeres. Med Ridge regression forsøger man for det første, at minimere RSS og dermed at estimere de koefficienter der passer bedst muligt til data. For det andet introducerer man et "straf-led" som bliver mindre, når koefficienterne kommer tættere på 0:

$$(\alpha \sum_{j=1}^{p}\beta_j^2)$$

Estimaterne mindskes dermed for koefficienter der går mod nul. Med hyperparameteren alpha forsøger man at styre den relative effekt af de to led på estimatet for regressionskoefficienten. Når alpha = 0 vil Ridge regressionen give de samme estimater som OLS (IBID). Omvendt vil Ridge estimaterne nærme sig nul, når alpha går imod uendeligt. (Friedman et al, 2001: 215). Da Ridge regression indeholder en hyperparameter bruger vi krydsvalidering på træningsdata for at finde den optimale værdi for denne.

Figur 8: alpha ifht. RMSE



Figur 8 viser den alpha-værdi for hvilken RMSE for vores Ridge regression er lavest. alpha findes til at være 26.8 og den tilsvarende RMSE på test data er 0.368938. Ridge regression er dermed marginalt dårligere end OLS regression. De straffede (da alpha > 0) koefficienter for modellen kan ses i bilag 2.

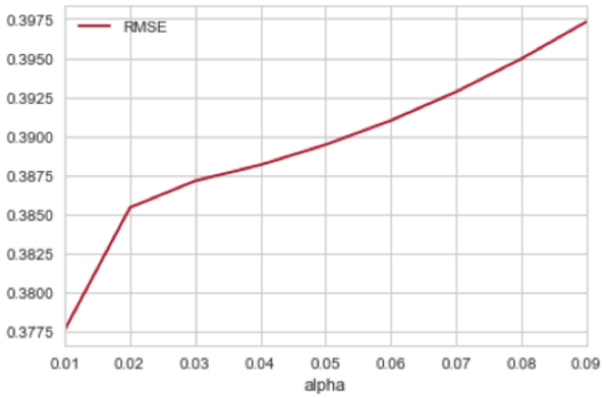
6.5 Lasso regression

Hvor Ridge regression inkluderer alle features, er et af de afgørende karakteristika ved Lasso regression, at den implementerer *Loss1*-regularization der kan minimere koefficienten til at være præcist lig nul, fremfor bare at være tæt på nul (Friedman et al., 2001: 219). Lasso regression straffer dermed potentielt overfitting i højere grad end Ridge regression, og er dermed bedre til at sikre imod for komplekse modeller (Foster, 2017: 154). Lasso-koefficienterne er dem der minimerer følgende udtryk:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \alpha\sum_{j=1}^{p}|\beta_j| = RSS + \alpha\sum_{j=1}^{p}|\beta_j|$$

(Friedman et al., 2001: 219)
Straffen for overfitting kan være relevant I vores model, da vi estimerer koefficienter for 61 features. Da Lasso, ligesom Ridge, også indeholder en hyperparameter, bruges samme algoritme.

Figur 9: alpha ifht. RMSE



Ud fra figur 9 er det tydeligt, at den laveste RMSE-værdi opnås ved en alpha-værdi tæt på 0. Dette vil sige, at vi i praksis snakker om en OLS regression.
Ved krydsvalideringen finder vi alpha = 0.01 og den tilsvarende RMSE på test data er 0.368910 – på 6 decimaler præcis det samme som for vores OLS med det fulde antal features. Koefficienterne for Lasso modellen kan dermed mere eller mindre antages at være de samme som for OLS med det tilsvarende feature set, da alpha er omtrentligt 0.

6.6 Random Forest regression

Random Forest regression er en samling (forest) af Decision Trees. En simpel opsummering af algoritmen for et Decision Tree er:

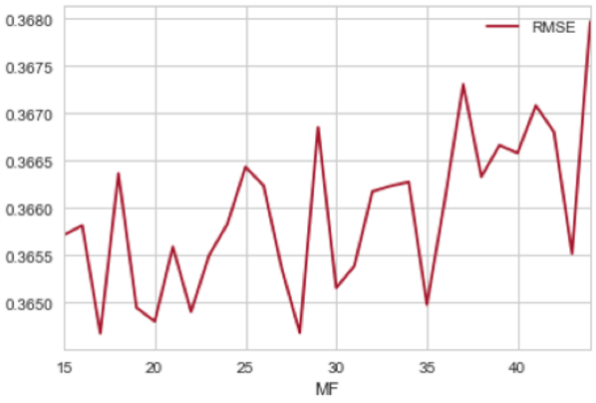
1) Læg den feature med den største effekt på den afhængige variabel i træets rod (øverste node).
2) Del træningsdata i subsets. Disse sets skal alle indeholde samme værdi for et givent feature.
3) Gentag 1 og 2 indtil alle grene indeholder et leaf (slut node).

Den relative performance mellem Random Forest regression og mere traditionelle lineære modeller, som eks. OLS, Lasso og Ridge afhænger af sammenhængen mellem den afhængige variabel og de specificerede features. Hvis sammenhængen er tilnærmelsesvis lineær i sin form vil de traditionelle lineære modeller fungere bedre (Friedman, 2001: 314). Hvis sammenhængen er mere kompleks og stærkt ikke-lineær vil Random Forest regressionen i gennemsnit klarer sig bedre (Friedman, 2001: 314). Random Forest regression straffer dog ikke overfitting på samme måde som Lasso og Ridge gør det. Den bruger i stedet bagging, da hvert enkelt træ i skoven er en *weak learner* (en dårlig predictor), men samlet set (bagging) kan træerne udgøre en *strong learner*.

Vi specificerer hvert træ uden begrænsning på *max_depth*, hvilket betyder at hvert leaf (endnode) enten er ren, eller at de kun indeholder én observation. Dette er problematisk at gøre, hvis

man kun arbejder med et enkelt Decision Tree, da dette kan lede til alvorlig overfitting. Dette tager Random Forest regression dog højde for vha. bagging - parameteren for denne sætter vi til 10.
Vi vælger dog at optimere hyperparameteret for antal variable hvert træ skal vælge tilfældigt. Den optimale værdi findes fortsat ved krydsvalidering.

Figur 10: Antal af features ifht. RMSE



Figur 10 viser, at det vil være optimalt at bruge lidt under 1/3 del af vores features svarende til 17 features. Ved at fitte med denne optimale værdi på træningsdata findes den tilsvarende RMSE på testdata til at være 0.362843. Random Forest regressionen er dermed den bedste model til at prædiktere prisen målt ud fra RMSE.
Feature importance for de forskellige features ses i bilag 3.

6.7 Opsummerende diskussion af modellering

Som det fremgår af tabel 1, der opsummerer de respektive modellers RMSE, er alle de inkluderede modeller bedre end den simple gennemsnitsmodel. Ved at inkludere features for Airbnb-lejemålenes karakteristika og udlejers beskrivelse heraf forbedrer vi altså vores evne til at prædiktere prisen pr. kapacitet pr. nat.

Tabel 1

Model	RMSE
Simple average model	0,429894
Simpel OLS	0,378597
Full Feature OLS	0,368910
Ridge	0,368938
Lasso	0,368910
Random Forrest	0,362843

Det fremgår desuden af tabellen, at Random Forest regressionsmodellen er den bedste af de inkluderede modeller til at prædiktere prisen pr. kapacitet pr. nat for airbnb lejemål. Det tyder dermed på, at sammenhængen mellem vores features og *log(price_kapacitet)* er mere kompleks end det antages i de lineære modeller (*OLS, Ridge og Lasso*).
Vi kunne have opnået endnu lavere RMSE for vores modeller ved at tilpasse optimeringen yderligere.

I Lasso og Ridge regressionerne kunne vi have forsøgt at optimere hyperparameteren endnu mere ved at lave ændringerne for alpha for hver iteration af krydsvalideringen mindre og mindre. Dette havde dog kun medført relativt begrænsede forbedringer i RMSE, da vores Lasso regression med alpha = 0,01 i forvejen havde samme RMSE på seks decimaler. Tilsvarende kunne vi for Random Forest have øget antallet af træer - opnået en stærkere learner - i hver skov i vores optimering for antal features, men dette ville hurtigt være en meget tidskrævende process når skoven skulle trænes.
Desuden har vi arbejdet med et forholdsvist stort antal af features der, som tidligere kommenteret, formentlig medfører en vis grad af multikollinearitet. For at lave modellerne mindre komplekse samt for at tackle evt. multikollinearitet, kunne vi gennem unsupervised learning have udført en principal komponent analyse på vores features før modelleringen. Ved at bruge de principale komponenter som vores features i stedet for de oprindelige features, kunne vi meget vel have opnået endnu bedre resultater for vores modeller, ved blandt andet at imødekomme den potentielle udfordring vedrørende multikollinearitet.

7 Konklusion

Formålet med denne opgave har været at træne machine learning modeller til at forudsige priser for leje af AirBnB-lejemål ud fra karakteristika for disse lejemål samt udlejers beskrivelse heraf. Vi har gjort dette med udgangspunkt i data fra den uafhængige side InsideAirbnb. Vi valgte at anvende data fra InsideAirbnb frem for det data vi selv havde scrapet fra Airbnbs hjemmeside, for at få adgang til flere observationer.
Vi valgte pris pr. kapacitet som afhængig variabel, da prisen varierede kraftigt på tværs af forskelle i kapacitet. Vi valgte endvidere at lave en logaritmisk transformation af denne variabel for at opnå en mere normalfordelt spredning.
Som yderligere features ift. det oprindelige datasæt introducerede vi afstand til Rådhuspladsen, afstand til nærmeste station, en række specifikke variable for faciliteter i lejemålet samt en række variable for udlejers beskrivelse af lejemålet.
I afsnittet for deskriptiv statistik så vi på en række centrale variable i forbindelse med at prædiktere prisen på lejemål. Blandt andet så vi, at lejemål med høj pris pr. kapacitet er placeret i centrum af København, samt at pris er negativt korreleret med hhv. afstand til Rådhuspladsen og nærmeste station (S-tog/metro).
I vores modellering af data begyndte vi med at opsplitte data i 2 dele - træning og test. Modellerne skulle opbygges vha. træningsdata, mens de skulle evalueres på testdata. Det evalueringsmål der blev brugt var RMSE. Vi kom med et udkast af 2 baseline modeller, Simple Average og OLS regression med udvalgte features. Derefter præsenterede vi 4 andre modeller: OLS regression med fulde features, Ridge regression, Lasso regression og Random Forest regression. Forventningen var, at RMSE for de sidste 4 modeller ville være lavere, og dermed bedre, end for de 2 baseline modeller. Dette viste sig at være tilfældet.
Ud af de 3 lineære modeller bygget på det fulde feature set var OLS ligeså god som Lasso; Lasso modellens mekanisme med at slå ubetydelige koefficienter til 0 var uden reel effekt, da krydsvalideringen gav et nærmest ubetydeligt alpha. For Ridge modellen fandt vi et betydeligt større alpha. Her blev koefficienter straffet, men RMSE var højere end for de to andre modeller.
Den bedste model vurderet ud fra RMSE var Random Forest, hvilket tyder på, at sammenhængen mellem *log(price_kapacitet)* og features skal modelleres på en ikke-lineær facon. Vi forventer dog, at en række mulige modifikationer i træningsprocesserne for hhv. *Lasso, Ridge og Random Forest* regressionerne samt en PCA kunne have mindsket RMSE yderligere. Skulle vi prædiktere priser af AirBnB annoncer, givet de har de samme features vi har arbejdet med,

så ville valget derfor falde på vores Random Forest regression.

8 Litteraturliste

Hovedlitteratur

- AirBnb: *"AirBnb Pricacy Policy"* https://www.airbnb.dk/terms/privacy_policy (besøgt d. 20-08-2017)
- Fransen, Niels (01-05-2017). Politikken *"Airbnb er de i forvejen privilegeredes kyniske pengemaskine"* http://politiken.dk/debat/art5930507/Airbnb-er-de-i-forvejen-privilegeredes-kyniske-pengemaskine (besøgt d. 23-08-2017)
- Foster, Ian Ghani, Rayid Jarmin, Ron Kreuter, Frauke Lane, Julia (2017). *Big Data and Social Science: A Practical Guide to Methods and Tools,* Boca Raton: CRC Press
- Friedman, Jerome Trevor Hastie Robert Tibshirani (2001). *An Introduction to Statistical Learning.* Vol. 1. Springer, Berlin: Springer series in statistics.
- Gkiousou, Sofia (28-02-2017). *"AirBnb: Danmark skal være et deleøkonomisk foregangsland",* http://www.altinget.dk/by/artikel/airbnb-danmark-skal-vaere-et-deleoekonomisk-foregangsland (besøgt d. 23-08-2017)
- Inside AirBnb: *"About Inside AirBnb"* http://insideairbnb.com/about.html (besøgt d. 23-08-2017)
- Olsen, Michael Straka, Rasmus (27-05-2017). Politikken *"Horesta: Nu er der flere Airbnb-senge end hotelværelser i Danmark",* http://politiken.dk/indland/art5889552/Nu-er-der-flere-Airbnb-senge-end-hotelv%C3%A6relser-i-Danmark (besøgt d. 23-08-2017)
- Lazer, David Radford, Jason (2017) *The Annual Review of Sociology,* Data ex Machina: Introduction to Big Data.

Data

- Inside AirBnb: *"Get the data"* http://insideairbnb.com/get-the-data.html (besøgt d. 17-08-2017) Anvendt data:listings.csv.gz for København fra hhv. 15. juni 2016 og 2017
- Metro: *"Adresser på alle metrostationer"* http://www.m.dk/!/om+metroen/rejseinformation/zone (scrapet d. 21-08-2017)
- S-togs-stationer: *"Adresser på alle DSB-stationer"* https://www.dsb.dk/kundeservice/stationer/ (scrapet d. 21-08-2018)
- Airbnb https://www.airbnb.dk/ (scrapet d. 18.-21.-08-2017)

9 Bilag

Bilag 1
Koefficienter for OLS med det fulde feature set

```
[  8.73405515e-02  -2.34399467e-02  -9.51192250e-02   7.36949105e-06
   1.55919739e-04  -2.10536461e-02  -2.95993402e-05  -3.55756333e-03
   3.25137139e-04   3.73674961e-03  -3.49344272e-03   1.09014025e-02
  -6.46006742e-04  -7.33208965e-03   1.40796869e-02  -1.09076117e-02
   1.42248048e-03   5.63757827e-03   7.41118915e-03  -2.88436760e-02
   1.83602871e-02  -1.87526765e-02  -3.41962564e-02   4.92767801e-02
   1.92306785e-02  -5.87660744e-02   2.63152100e-02   1.36141533e-03
  -1.35492490e-02  -8.09555972e-02  -3.96737612e-02  -2.08814331e-02
  -9.36676606e-03  -2.72946161e-02   3.85308099e-02   1.09446350e-02
   8.07110311e-02  -1.51892551e-03  -7.33251320e-03   7.13388685e-03
  -9.01122112e-03  -1.01011766e-01  -5.99355003e-02   2.42095995e-02
   1.96314941e-01  -2.74783233e-02  -7.97384091e-02  -6.43539648e-02
   3.36713146e-02   8.01994426e-02  -5.11240891e-02   9.80590138e-02
  -4.69349247e-02   1.92302677e-01   3.88048965e-02  -2.31107574e-01
  -8.48589628e-02   6.68557878e-02  -1.10411003e-02  -1.24663217e-02
   4.15105970e-02]
```

Bilag 2
Koefficienter for Ridge Regression

```
array([  8.48443855e-02,  -2.37810871e-02,  -9.37360665e-02,
         8.23077471e-06,   1.57852641e-04,  -2.13871597e-02,
        -4.20075074e-05,  -3.18548973e-03,   3.87539461e-04,
         3.96940238e-03,  -3.42217323e-03,   1.26695921e-02,
         1.56350287e-03,  -8.87681512e-03,   1.40465821e-02,
        -1.27794235e-02,   1.22480518e-03,   4.60924668e-03,
         7.40225693e-03,  -2.60530210e-02,   1.65804198e-02,
        -2.04397359e-02,  -2.86959800e-02,   4.71414556e-02,
         1.66977624e-02,  -5.70828912e-02,   2.38052306e-02,
         1.39877739e-03,  -1.27084351e-02,  -8.18977288e-02,
        -4.24096616e-02,  -2.15003972e-02,  -6.35714370e-03,
        -2.64430964e-02,   4.12272300e-02,   7.48760151e-03,
         7.61908367e-02,   5.11171480e-03,  -9.84290408e-03,
         5.73471962e-03,  -9.58400592e-03,  -1.02924430e-01,
        -5.39486688e-02,   2.40175170e-02,   1.92544004e-01,
        -3.00991504e-02,  -7.82052156e-02,  -5.79115179e-02,
         3.03417999e-02,   8.00349484e-02,  -4.79140641e-02,
         9.31394208e-02,  -4.52253567e-02,   1.75801285e-01,
         2.65070559e-02,  -2.02308341e-01,  -2.23133183e-02,
         2.48052907e-02,  -8.61628577e-03,  -2.23200509e-02,
         2.84443644e-02])
```

Bilag 3
Vigtigheden af features for Random Forest

```
[ 0.01124961  0.02260877  0.0841504   0.02598456  0.04401633  0.01329718
  0.04368305  0.03318679  0.03202057  0.02385324  0.00996507  0.01457375
  0.00857447  0.00918843  0.01333055  0.01130563  0.04304892  0.00652108
  0.00973185  0.00308925  0.00941006  0.00402472  0.00754486  0.01092494
  0.00637371  0.00335363  0.00394744  0.04651589  0.075131    0.17251737
  0.00663647  0.01075534  0.00845712  0.00740065  0.00692869  0.00335589
  0.00294498  0.00064507  0.00946878  0.00509075  0.00313602  0.00901226
  0.00359312  0.00556041  0.03651953  0.00566561  0.00300929  0.00290501
  0.00454642  0.00774996  0.00344158  0.00307518  0.00193231  0.02547881
  0.00974805  0.00351591  0.00139903  0.0004532   0.00107154  0.00136152
  0.00201839]
```

Bilag 4: Liste over variable i det originale datasæt

Anvendte variable er markeret med fed. Features som vi selv har beregnet er ikke inkluderet i listen, men er beskrevet i afsnittet for feature engineering.

id
listing_url
scrape_id
last_scraped
name
summary
space
description
experiences_offered
neighborhood_overview
notes
transit
access
interaction
house_rules
thumbnail_url', 'medium_url', 'picture_url',
'xl_picture_url',
host_id
host_url
host_name
host_since
host_location
host_about
host_response_time
host_response_rate
host_acceptance_rate
host_is_superhost
host_thumbnail_url
host_picture_url
host_neighbourhood
host_listings_count
host_total_listings_count
host_verifications
host_has_profile_pic
host_identity_verified
street
neighbourhood
neighbourhood_cleansed
neighbourhood_group_cleansed
city
state
zipcode
market
smart_location
country_code
country
latitude
longitude
is_location_exact
property_type
room_type
accommodates
bathrooms
bedrooms
beds
bed_type
amenities
square_feet
price
weekly_price

monthly_price
security_deposit
cleaning_fee
guests_included
extra_people
minimum_nights
maximum_nights
calendar_updated
has_availability
availability_30
availability_60
availability_90
availability_365
calendar_last_scraped
number_of_reviews
first_review
last_review
review_scores_rating
review_scores_accuracy
review_scores_cleanliness
review_scores_checkin
review_scores_communication
review_scores_location
review_scores_value
requires_license
license
jurisdiction_names
instant_bookable
cancellation_policy
require_guest_profile_picture
require_guest_phone_verification
calculated_host_listings_count
reviews_per_month

Bilag 5: Tekst features

Navn	Beskrivelse	Ord
Stedet	*Om man nævner stedet*	*Nørrebro, Vesterbro, Østerbro, Frederiksberg, Christianshavn*
Beskrivelse	*Generel beskrivelse af lejligheden*	*Brigth, brigth, Charming, charming, Cozy, Cosy, cozy, cosy, lovely, Lovely, spacious, Spacious, beautiful, Beautiful, large, Large, nice, Nice, light, Light, modern, Modern, big, Big*
Placering	*Om man beskriver lejlighedens placering som værende central*	*Central, central, Center, center, Heart, heart, Centre, centre, Centrum, centrum*
Området	*Om man beskriver området*	*Area, area, location, Location, close, Close, near, Near, neighborhood, Neighborhood*
Ekstra egenskaber	*Om man beskriver ekstra egenskaber ved lejemålet*	*Balcony, balcony, Garden, garden, View, view*
Transportmuligheder	*Om man beskriver transportmuligheder*	*metro, Metro, Transport, transport, airport, Airport*