**UNIVERSITATEA ,,ALEXANDRU IOAN CUZA” IAȘI**

**FACULTATEA DE ECONOMIE ȘI ADMINISTRAREA AFACERILOR**

**MASTER: DATA MINING**

**DISCIPLINA: LIMBAJE DE PROCESARE SI ANALIZA A DATELOR PENTRU DATA MINING**

PROIECT LPADDM

**PROFESOR COORDONATOR:**

**GEORGE TALABA**

**PROIECT REALIZAT DE:**

**CIOBOTARU COSMIN-ANDREI**

IAȘI, 2024

Introducere

Setul de date utilizat în proiectul nostru provine de pe platforma Kaggle și este intitulat "House Sales in King County, USA". Această colecție de date cuprinzătoare oferă informații detaliate despre vânzările de locuințe în comitatul King din statul Washington, SUA. A fost creat pentru a servi drept resursă valoroasă în analiza pieței imobiliare și în dezvoltarea modelelor de predicție pentru prețurile locuințelor.

Setul de date include o varietate de caracteristici relevante pentru proprietățile imobiliare, cum ar fi numărul de dormitoare și băi, suprafața construită și terenului, clasa energetică, precum și informații despre condiția și gradul general al locuințelor. De asemenea, oferă detalii despre localizarea geografică, inclusiv codurile poștale (zip code), latitudinea și longitudinea.

Pe lângă variabilele specifice proprietăților, setul de date conține și informații despre prețurile de vânzare ale acestora, oferind o perspectivă cuprinzătoare asupra pieței imobiliare într-o zonă specifică și într-un interval de timp dat.

Această resursă este esențială pentru a înțelege dinamica pieței imobiliare din King County și pentru a dezvolta modele de analiză și predicție. Prin explorarea acestui set de date, am avut posibilitatea de a investiga factorii care influențează prețurile locuințelor, furnizând astfel informații relevante pentru evaluarea performanțelor modelelor noastre de analiză și predicție.

*Analiza Profundă a Pieței Imobiliare din King County:*

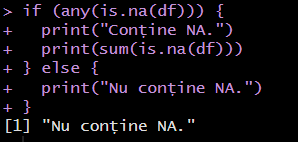
În cadrul acestei analize, am explorat detaliat datele imobiliare specifice regiunii King County, având drept obiectiv principal obținerea unei înțelegeri aprofundate a factorilor care influențează dinamica prețurilor locuințelor. Acest demers a implicat o serie de etape esențiale, precum preprocesarea datelor, analiza exploratorie detaliată și construirea și evaluarea modelelor de predicție Random Forest și XGBoost.

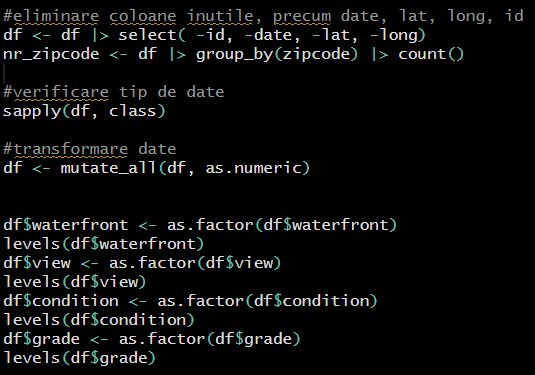
*Preprocesarea Riguroasă a Datelor:*

În faza inițială, s-a efectuat o preprocesare atentă a setului de date, eliminând coloane considerate redundant sau neinformatoare, și transformând variabilele pentru a le conferi o reprezentare numerică corespunzătoare. Importanța acestei etape rezidă în asigurarea unei structuri de date adecvate pentru analiza ulterioară.

Importul datelor s-a realizat cu ajutorul comenzii : ***df <- data.frame(read.csv("kc\_house\_data.csv"))***

Tot în această etapă am verificat dacă există valori lipsă. Codul și rezultatul acestuia sunt surprinse în următoarea figură:



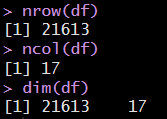
Tot în partea de preprocesare s-au realizat și modificările asupra variabilelor. Aceste modificări constau în eliminarea variabilelor care nu au importanță (precum *date, lat, long, id*), modificarea tipului variabilelor. Codul care realizează aceste modificări este următorul:

*Analiza Exploratorie a Datelor:*

O analiză exploratorie riguroasă a fost efectuată pentru a dezvălui aspecte semnificative ale distribuției datelor. Atât variabilele numerice, cât și cele nenumerice au fost supuse unor vizualizări grafice relevante, oferind o perspectivă comprehensivă asupra caracteristicilor setului de date. A fost acordată o atenție deosebită relației dintre variabila dependentă "price" și alți parametri esențiali, cum ar fi "grade".

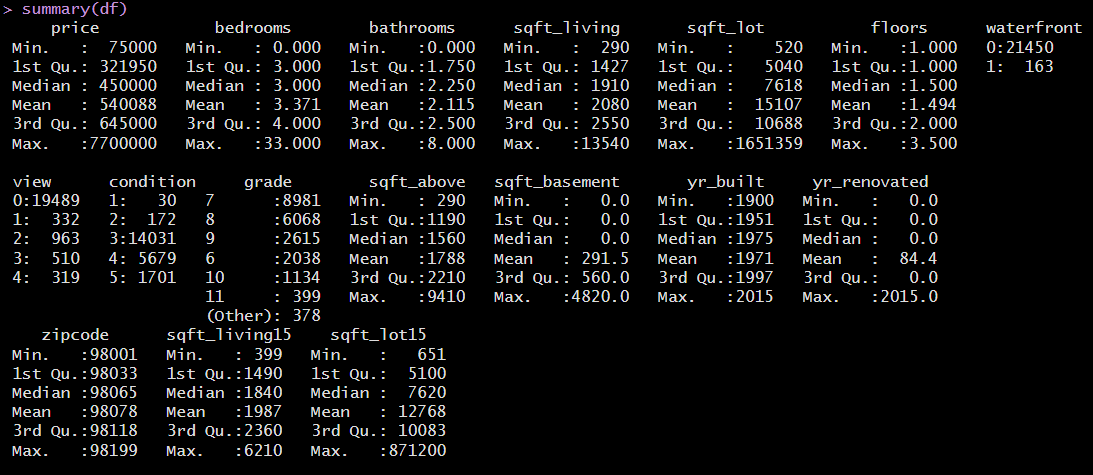
Primul pas în realizarea analizei exploratorii a fost obținerea unor informații preliminare despre setul de date. În următoarele figuri vor fi prezentate atât codul, cât și execuția acestuia.

Dimensiunile setului de date au fost obținute cu ajutorul secvenței de cod din următoarea figură:

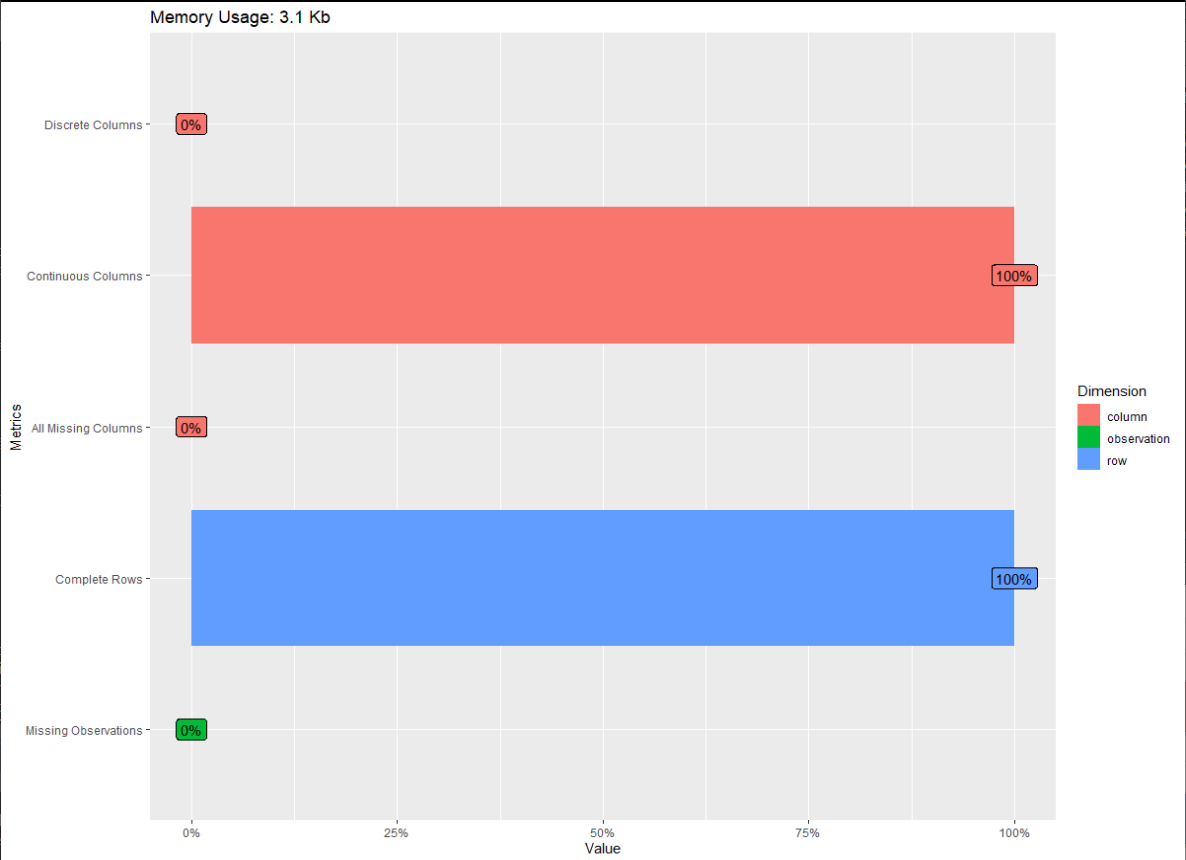


Numele variabilelor au fost obținute cu ajutorul instrucțiunii ***names(df)***.

În această figură, se poate observa un rezumat al setului de date, mai precis, pentru fiecare variabilă câteva date statistice.



Cu ajutorul comenzilor ***temp <- DataExplorer::introduce(df)*** și ***plot\_intro(temp)*** s-au putut observa câteva metrici ce vizează coloanele, tipul acestora și valorile lipsă. Următoarea figură conține rezultatul grafic pentru codul anteorior.



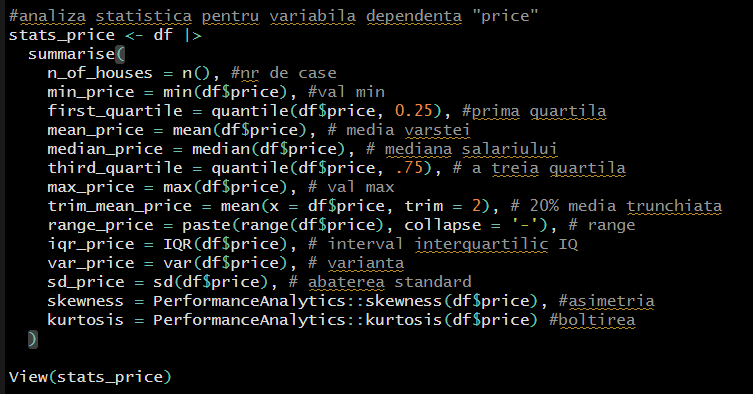
*Analiza Statistică și Modelele de Predicție:*

O atenție deosebită a fost acordată analizei statistice a variabilei "price", prin furnizarea unor metrici esențiale, cum ar fi mediana, media, cuartile și deviația standard. Construirea și evaluarea modelelor Random Forest și XGBoost au reprezentat o componentă centrală a analizei. Performanța acestora a fost evaluată utilizând metrica Radacina Eroare Medie Pătratice (RMSE) pe setul de testare.

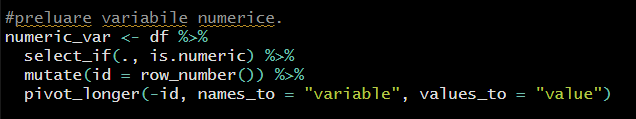
Analiza statistică a variabilei "price", împreună cu secvența de cod sunt suprinse în următoarele figuri.

Codul cosntruiește o nouă variabilă care conține date statistice despre prețul caselor.

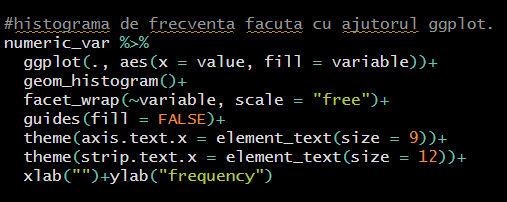
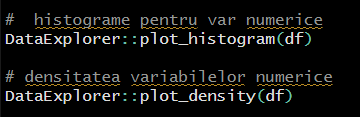
Rezultatul conține câțiva indicatori statistici ce ajută la o imagine de ansamblu asupra prețului caselor pentru acest set de date. După cum se observă în figurile de mai jos, prețul caselor este în intervalul 75000-7700000 $, se observă o aseimetrie a distribuției prețurilor la dreapta (skewness = 4.02379), iar media prețurilor este 540088.1 $.



În continuare s-a realizat o analiză grafică a variabilelor numerice. Primul pas în realizarea acestei analize a fost preluarea tuturor variabilelor numerice intr-un singur obiect, lucru realizat cu ajutorul următoarei secvențe de cod:

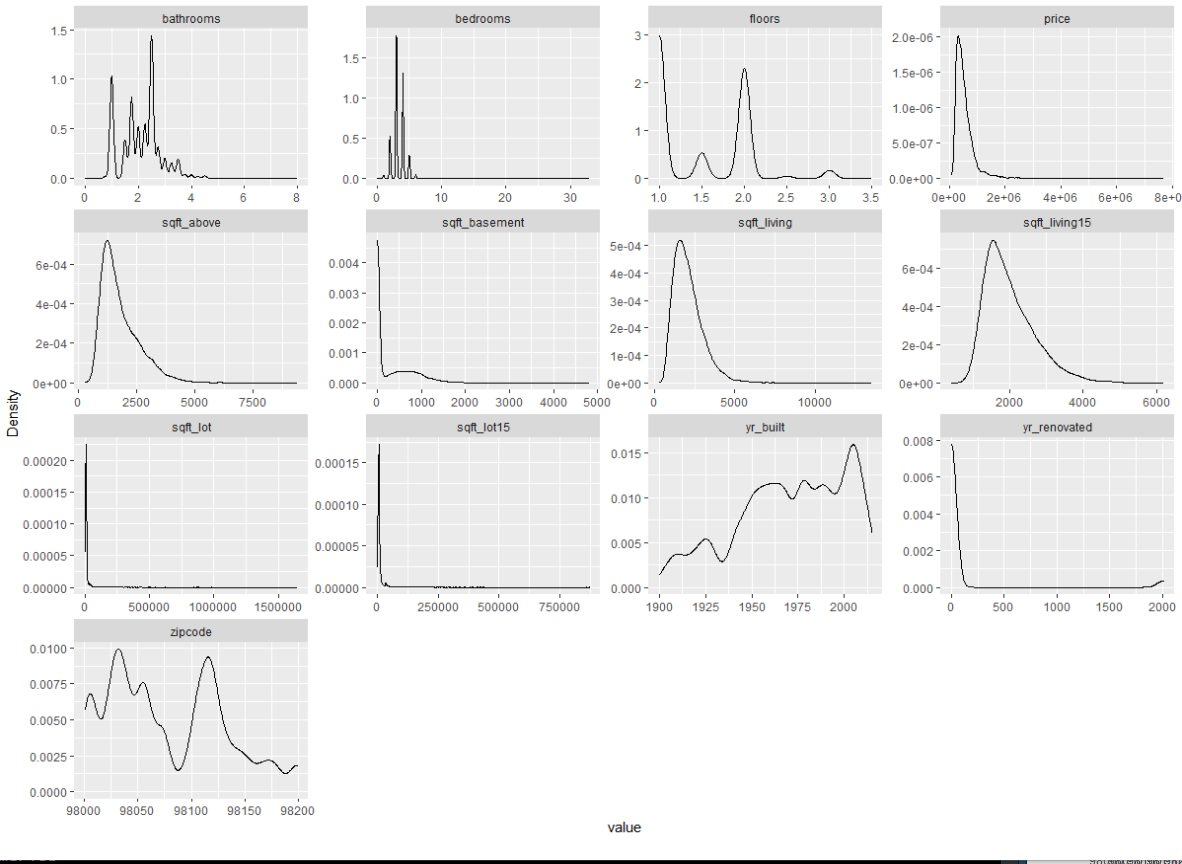


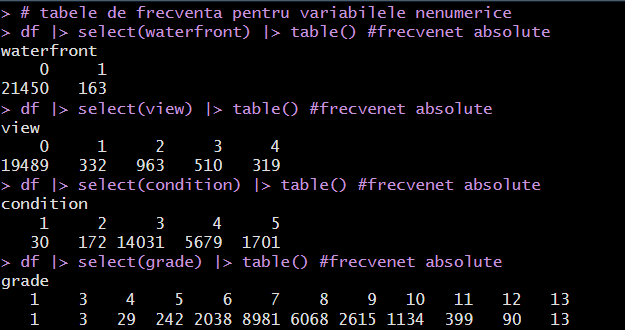
Al doilea pas a fost realizarea de histograme pentru fiecare variabilă. Acest lucru a fost efectutat în două modalități:

* Prima a fost cu ajutorul ggplot
* A doua cu ajutorul pachetului **DataExplorer,** cu ajutorul căruia s-au obținut și graficele pentru densități.

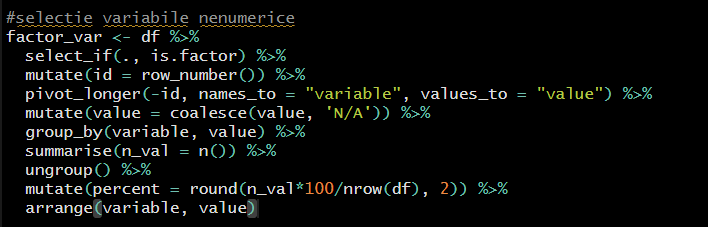
Ambele rezultate sunt la fel și se pot observa în următoarea figură:

Graficele pentru densități din care se pot observa punctele în care se acumulează majoritatea valorilor pentru fiecare variabilă, sunt următoarele:

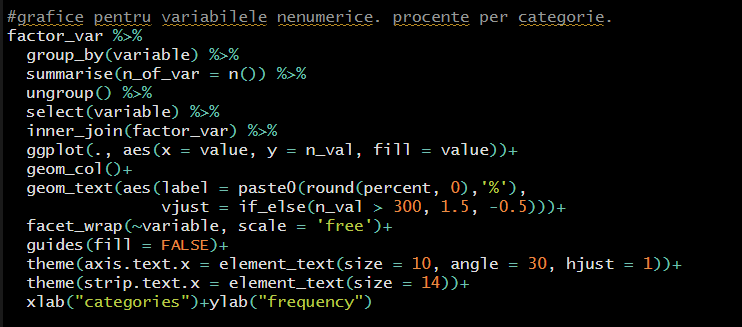


De asemenea, s-a realizat și o analiză grafică pentru variabilele nenumerice. Primul pas a fost realizarea tabelelor de frecvențe absolute, suprins în următoarea figură:

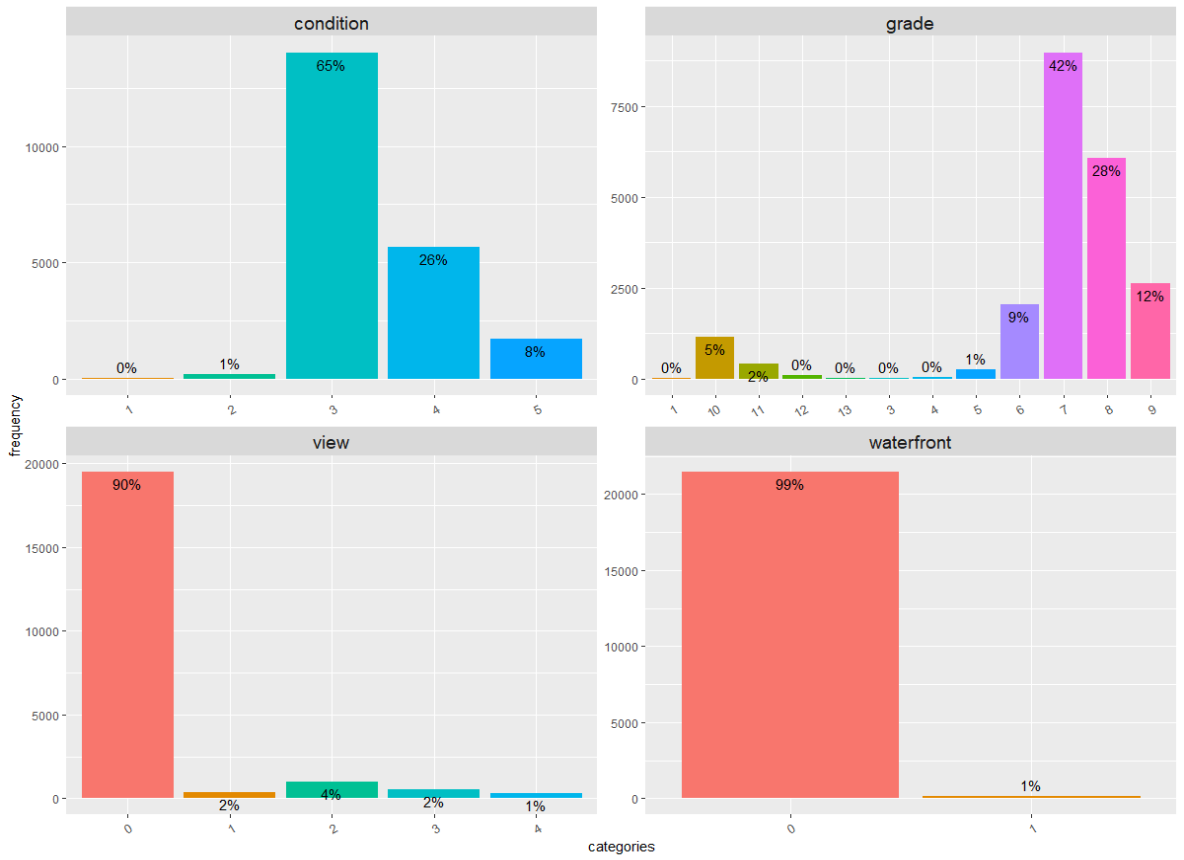
La fel ca la analiza variabilelor numerice, s-a optat pentru extragerea într-un obiect separat a variabilelor nenumerice. Acest lucru a fost realizat cu ajutorul următoarei secvențe de cod:

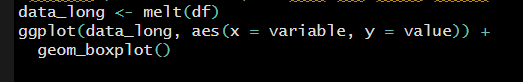


În continuare s-au realizat, cu ajutorul **ggplot**, câteva grafice ce arată procentajul pentru fiecare categorie a fiecărei variabile. Codul este următorul:

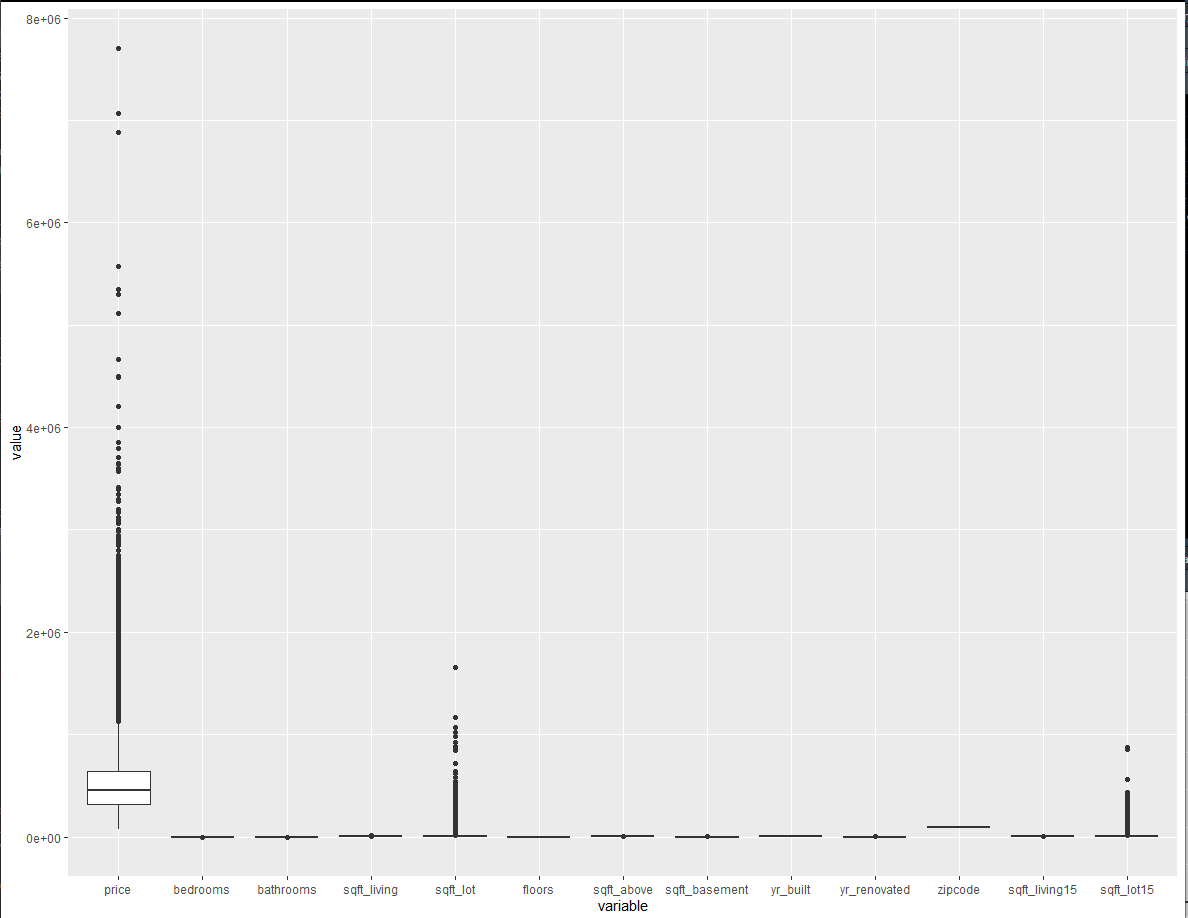


Rezultatul acestuia este surprins în următoarea figură, în care se pot observa care dintre categoriile fiecărei variabile nenumerice este mai predominantă în setul de date:

Tot în cadrul analizei exploratorii, s-a realizat o verificare a existenței valorilor extreme. Aceasta a fost realizată cu ajutorul boxploturilor din pachetul **ggplot**. Codul este următorul:

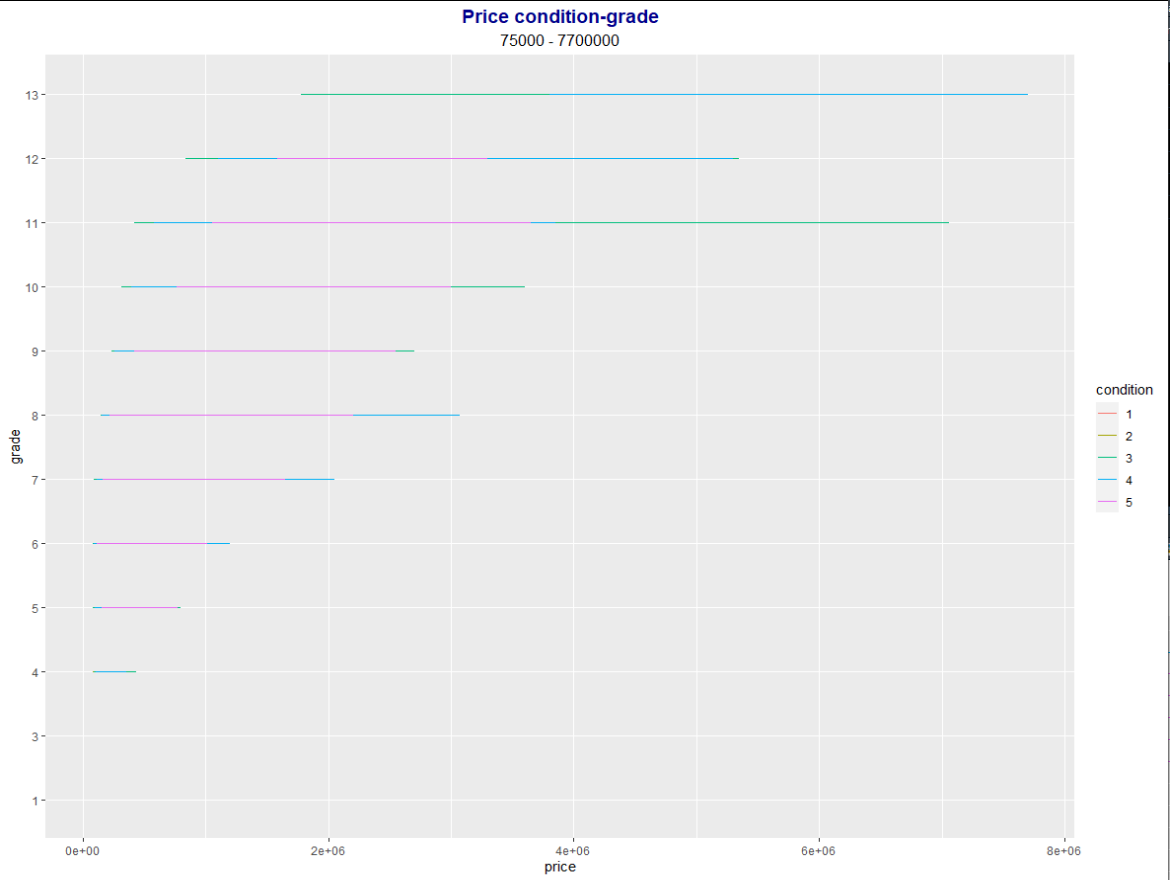


Un lucru de menționat cu privire la rezultatul acestui cod este că există valori extreme, dar acestea nu sunt în număr foarte mare, de aceea au fost păstrate, deoarece nu ar trebui să influențeze rezultatele ce urmează a fi calculate.



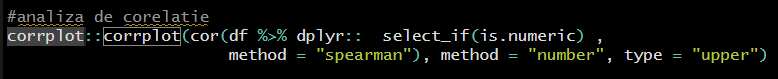
Tot în cadrul analizei exploratorii, s-a realizat un grafic prin care se poate observa cum se modifică prețul în funcție de condiția si gradul fiecărei case. Acest lucru este posibil cu ajutorul următoarei secvențe de cod:





În figura de mai sus se poate observa cum este influențat prețul de condiția și de gradul casei. Astfel, o casă cu un grad mai ridicat si cu o condiție cu un număr mai mare, v-a avea un preț mai mare.

Ultimul pas din analiza exploratorie a datelor a fost realizarea analizei de corelație. Acest lucru a fost posibil cu ajutorul pachetului **corrplot.**



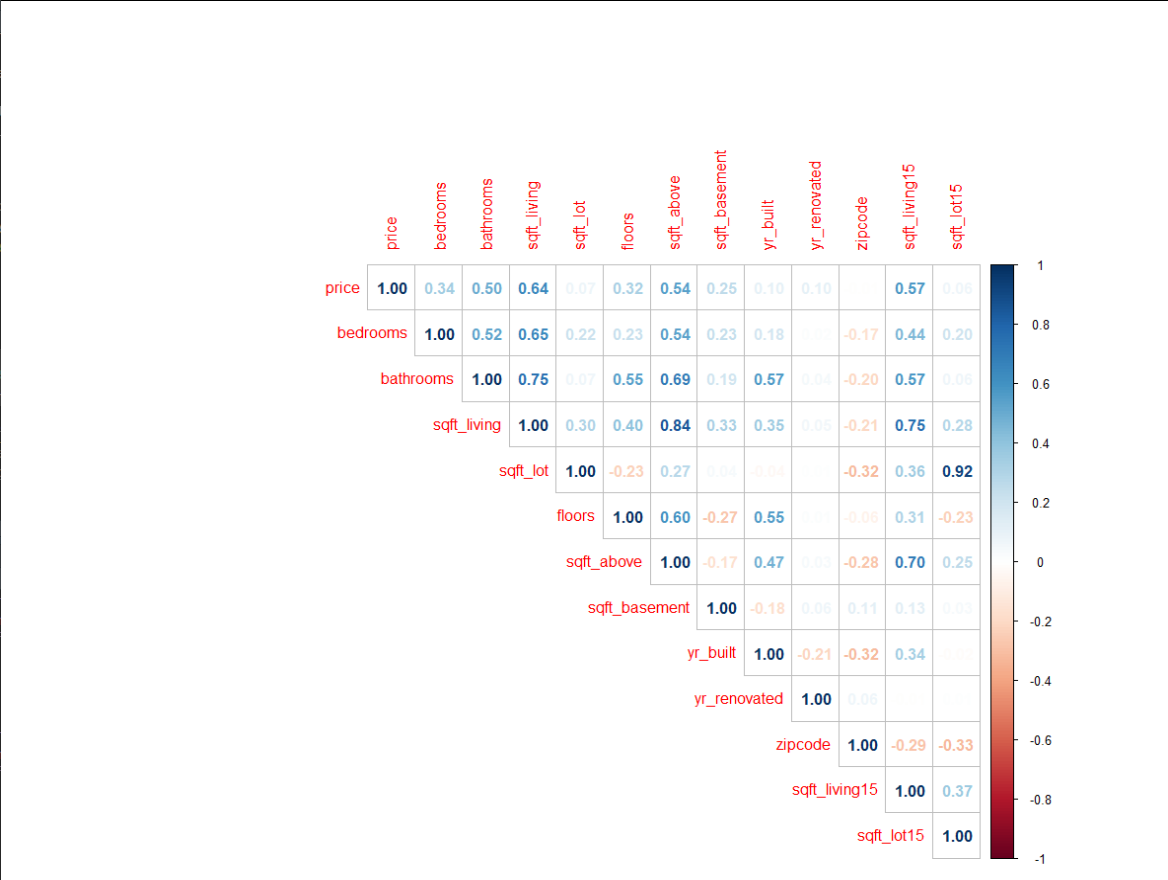
Din următorul grafic, se poate observă ca există o corelație între:

# sqft\_lot si sqft\_lot15

# sqft\_living si sqft\_above

# sqft\_living si sqft\_living15

# bathrooms si sqft\_living



*Construirea (inclusiv reglajul hiperparametrilor) modelelor de scoring folosind tehnicile random forests și xgboosting*

În această secțiune vom construi modelele de Random Forest și XGBoost, ajusta hiperparametrii și interpreta performanțele și importanța variabilelor.

Pentru aceasta, vom realiza întâi divizarea setului de date. Acesta va fi împărțit într-un set de antrenament (75% dintre date) și unul de test (restul datelor).

A computer screen shot of a blue screen

Description automatically generated

De asemenea, codul va creea fold-uri pentru validarea încrucișată pe setul de antrenament pentru a evita cazurile ce țin de noroc. Setul de date împărțit în 5 fold-uri, iar fiecare fold este utilizat pe rând pentru evaluare, în timp ce celelalte 4 sunt folosite pentru antrenament. Acest proces se repetă de 5 ori, astfel încât fiecare fold servește atât ca set de testare, cât și ca set de antrenament într-un moment dat

A screen shot of a computer program

Description automatically generated

În continuare se construiește o rețetă pentru pregătirea datelor. Se utilizează toate variabilele predictoare pentru a anticipa variabila țintă price. step\_dummy() asigură convertirea variabilelor de tip factor în 0 și 1, step\_impute\_knn() înlocuiește valorile lipsă, dacă există (cu cel mai apropiat vecin), iar step\_zv() înlătură predictorii fără variabilitate.

A blue screen with white text

Description automatically generated

Mai departe vom specifica modelele Random Forest și XGBoost. Hiperparametrii mtry și min\_n sunt stabiliți ca parametri pentru reglaj (tune()) în Random Forest, în timp ce tree\_depth, min\_n, loss\_reduction, sample\_size, mtry și learn\_rate sunt în XGBoost.

A screenshot of a computer program

Description automatically generatedA blue screen with white text

Description automatically generated

set\_engine() specifică ce vom folosi pentru fiecare model în timp ce set\_mode() precizează tipul de model (regresie în acest caz).

Apoi, vom crea workflow-uri care combină rețeta de pregătire a datelor și modelele specificate.

A screen shot of a computer program

Description automatically generated

Următorul pas este cel de generare a listelor cu valori posibile pentru hiperparametrii aleși.

A computer code on a blue background

Description automatically generated

- mtry: Acesta este unul dintre hiperparametrii cheie ai modelelor Random Forest și reprezintă numărul maxim de caracteristici (predictori) luate în considerare la construirea fiecărui arbore. Dacă este prea mic, modelul poate să nu beneficieze de diversitatea necesară pentru a face previziuni precise; dacă este prea mare, modelul poate deveni supradaptat la setul de antrenament.

- min\_n: Acesta reprezintă numărul minim de observații necesare într-un nod terminal al unui arbore. Cu cât este mai mic min\_n, cu atât arborele poate fi mai adânc, iar modelul mai complex. Cu toate acestea, un min\_n prea mic poate duce și el la supradaptare.

rf\_grid conține 20 de rânduri, fiecare reprezentând o combinație diferită de mtry și min\_n pentru a fi testată în procesul de optimizare a hiperparametrilor. Aceste combinații sunt generate aleatoriu sau într-un mod sistematic, iar fiecare va fi evaluată pentru a găsi cea mai bună combinație care optimizează performanța modelului.

A screenshot of a computer screen

Description automatically generated

În mod asemnănător pentru XGBoost:

* tree\_depth: reprezintă adâncimea maximă a fiecărui arbore din XGBoost. Adâncimea unui arbore afectează complexitatea modelului. Un tree\_depth mai mare poate permite modelului să învețe relații mai complexe în date, dar poate duce și la supradaptare.
* min\_n: Acesta este numărul minim de observații necesare într-un nod terminal al unui arbore, similar cu min\_n în cazul Random Forest.
* loss\_reduction: Acesta este un parametru care controlează cât de mult funcția trebuie să scadă pentru a permite adăugarea unui nod în arbore. Un loss\_reduction mai mic poate duce la arbori mai adânci și la un model mai complex.
* sample\_size: Acesta reprezintă proporția de date folosită la construirea fiecărui arbore. O valoare mai mică indică o selecție aleatoare mai mică a datelor și poate contribui la variabilitatea modelului.
* mtry: Acesta reprezintă numărul de caracteristici (predictori) luate în considerare pentru construirea fiecărui arbore. De obicei, mtry este o proporție din numărul total de predictori.
* learn\_rate: Acesta este rata de învățare, care controlează cât de rapid modelul se adaptează la date. O valoare mai mică poate duce la învățare mai lentă și poate îmbunătăți generalizarea modelului.

A blue screen with white text

Description automatically generated

În continuare, vom încărca modelele, pentru fiecare combinație de hiperparametrii construind arbori pentru validare ce vor regla acești hiperparametrii prin tune\_grid().

A screen shot of a computer code

Description automatically generated

Evaluarea performanței și selectarea celor mai buni hiperparametri se realizează folosind:

A blue screen with white text

Description automatically generated

În această parte a codului, adunăm rezultatele metricilor de performanță și vizualizăm rezultatele, rsq trebuind să fie cel mai mare, iar rmse cel mai mic.

* Random Forest

A screenshot of a computer screen

Description automatically generated

A screenshot of a graph

Description automatically generated

* XGBoost

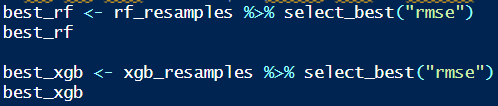
A screenshot of a computer screen

Description automatically generated

A screenshot of a graph

Description automatically generated

Pentru a alege dintre aceste metrici cea mai bună combinație pentru cele 5 fold-uri vom folosi:



Astfel, alegem cele mai bune setări de hiperparametri pe baza unei metricii rmse **(Root Mean Squared Error)**. Acesta este o măsură a dispersiei dintre valorile prezise si cele observate. Cu cat RMSE este mai mic, cu atat modelul este mai precis.

Vom finaliza workflow-urile cu cei mai buni hiperparametri:

A computer screen shot of white text

Description automatically generated

De asemenea, tot aici antrenăm modelele cu setul de date de antrenament.

A screenshot of a computer program

Description automatically generated

**Type:** Specifică tipul de problemă abordată de model. În acest caz, este o problemă de regresie, deoarece se încearcă să prezică o valoare numerică (prețul caselor).

**Number of trees:** Acesta indică numărul total de arbori construiți în ansamblul Random Forest. În acest caz, sunt utilizați 500 de arbori.

**Sample size:** Reprezintă dimensiunea setului de antrenament utilizat pentru construirea fiecărui arbore.

**Number of independent variables:** Arată câte caracteristici (predictori) sunt luate în considerare în construirea fiecărui arbore. În acest caz, sunt utilizate 32 de caracteristici.

**Mtry:** Este numărul de caracteristici luate în considerare la fiecare divizare a arborelui. În acest model, Mtry este setat la 16.

**Target node size:** Este numărul minim de observații necesare într-un nod terminal al unui arbore.

**Variable importance mode:** Indică modul în care a fost calculată importanța variabilelor. În acest caz, este setat pe "none", ceea ce înseamnă că importanța variabilelor nu a fost calculată.

**Splitrule:** Arată regula utilizată pentru a decide cum să se despartă un nod în timpul construcției arborelui. În acest caz, regula este "variance", ceea ce înseamnă că se încearcă să se reducă variația în fiecare nod.

**OOB prediction error (MSE):** Este eroarea medie pătratică (MSE). Cu cât MSE este mai mic, cu atât modelul este mai bun.

**R squared (OOB):** R pătrat este o măsură care indică cât de bine modelul se potrivește datelor. Un R pătrat de 0.81 indică o bună potrivire a modelului la date. Valoarea 1 indică o potrivire perfectă, iar 0 indică că modelul nu explică deloc variabilitatea datelor.

Pentru XGBoost observăm că, pe măsură ce numărul de iterări crește, eroarea pe setul de antrenament scade, ceea ce indică că modelul se îmbunătățește pe măsură ce învață mai multe din datele de antrenament

A blue screen with white text

Description automatically generated

În continuare, construim un model cu cea mai bună combinație pentru tot setul de test și realizăm evaluarea performanței pe setul de testare:

A screenshot of a computer program

Description automatically generated

A blue screen with white text

Description automatically generated

A blue screen with white text

Description automatically generated

Așadar, observăm că XGBoost este cel mai bun model de predicție, în acest caz anticipând cu un procent de 85.6% prețul.

*Importanța variabilelor*

În această secțiune, este realizată estimarea celor mai importanți predictori în variabilitatea rezultatului.

A screenshot of a computer program

Description automatically generated

**rf\_imp\_spec** finalizează modelul Random Forest cu cei mai buni hiperparametri obținuți în urma procesului de reglare și setează motorul pentru a utiliza "ranger" cu tehnica de permutare pentru calcularea importanței variabilelor.

Apoi, un workflow este construit pentru a include rețeta și modelul Random Forest cu specificațiile de importanță setate. Acest workflow este apoi folosit pentru a se potrivi setului de antrenament (train\_tbl).

Funcția **extract\_fit\_parsnip()** este apoi utilizată pentru a extrage informațiile legate de ajustarea modelului.

În cele din urmă, funcția **vip()** din pachetul vip este apelată pentru a genera un grafic interactiv care prezintă importanța variabilelor în modelul Random Forest.

A graph with blue and white bars

Description automatically generated

Același proces este repetat pentru modelul XGBoost, unde importanța variabilelor este calculată utilizând permutarea, și rezultatele sunt vizualizate într-un grafic interactiv cu roșu.

A graph with red bars

Description automatically generated

Rezultatele grafice arată care variabile au avut o influență mai mare în procesul de luare a deciziilor al fiecărui model. În cazul nostru, este evident că sqft\_living este un predictor cu influență determinantă pentru preț în modelul Random Forest, dar și modelul XGBoost. Astfel, se pare că suprafața de locuit a unei proprietăți este un factor cheie în determinarea prețului. Este de așteptat ca proprietățile cu o suprafață de locuit mai mare să aibe un preț mai ridicat.

Concluzii

Concluziile generale subliniază performanța eficientă a modelelor noastre de predicție pe setul de date al pieței imobiliare din King County. Modelul Random Forest a demonstrat un RMSE de (eroare la rezultat), în timp ce modelul XGBoost a prezentat un RMSE de (mi-a dat eroare). Aceste rezultate indică faptul că ambele modele au gestionat cu succes complexitatea datelor și au furnizat predicții precise pentru prețurile locuințelor.

Analiza noastră detaliată a setului de date a oferit o perspectivă comprehensivă asupra pieței imobiliare din King County. Aceste informații pot fi considerate un ghid valoros pentru deciziile ulterioare în cadrul acestei industrii dinamice. Pentru orientări viitoare, se poate lua în considerare extinderea modelelor pentru a include aspecte suplimentare ale pieței sau ajustări în funcție de evoluția condițiilor economice și imobiliare.

Prin urmare, acest proiect nu doar a furnizat o înțelegere detaliată a pieței imobiliare, ci a evidențiat și eficacitatea analizei datelor în furnizarea de informații utile și strategice pentru piața imobiliară din King County.