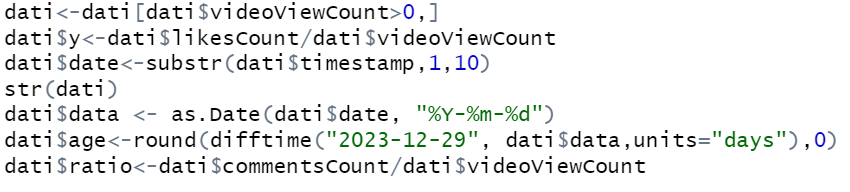
L’influencer che ho scelto è Kylie Jenner. Ho estratto i dati dei suoi reels riguardanti: il numero di commenti, dei like, delle visualizzazioni, la data dei video e la loro durata e li ho inseriti in un dataset.

Dopodiché ho impostato la directory di lavoro corrente così quando successivamente verranno caricati o salvati file, R li cercherà in questa directory e poi ho salvato il dataset nella variabile “dati”.

 In seguito ho filtrato il dataframe includendo solo le righe in cui il valore della colona "videoViewCount" è maggiore di zero perché quattro video presentano 0 visualizzazioni, perché non presenti.

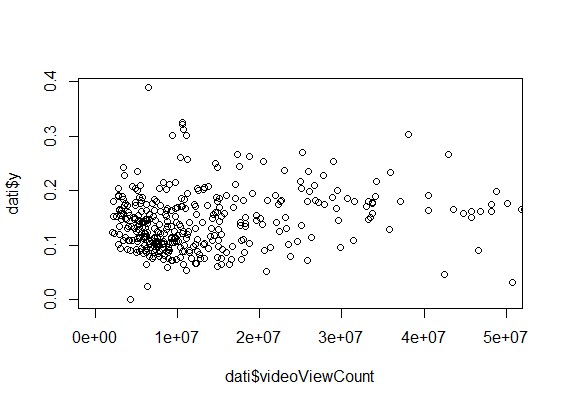
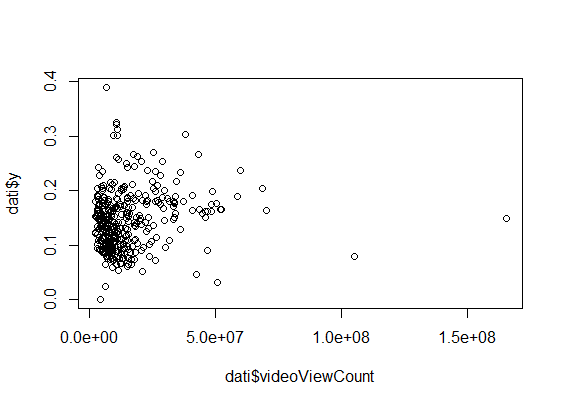
Ho calcolato, come richiesto, una nuova variabile chiamata “y” , che rappresenta il rapporto tra i likes e le visualizzazioni ed un’altra chiamata “ratio”, che rappresenta il rapporto tra i commenti e le visualizzazioni.

Inoltre ho creato una nuova colonna chiamata “date” estraendo i primi 10 caratteri dalla colonna "timestamp" e questo rappresenta la data. Ho poi convertito la colonna date in un oggetto di classe Date di R e l’ho salvata nella nuova colonna “data”

In seguitoho calcolato l'età delle osservazioni rispetto al 29 dicembre 2023, arrotondando al giorno più vicino ed ho calcolato la differenza di tempo in giorni tra la data specificata e la colonna data.

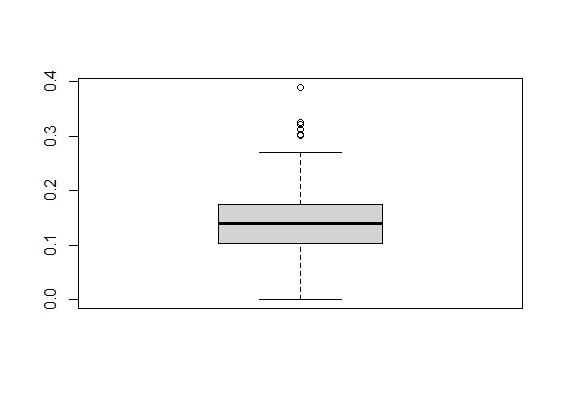
Ho fatto dei diagrammi a dispersione tra y e altre variabili.



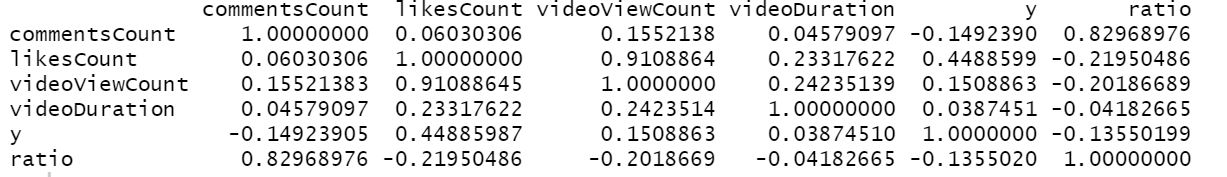
 

Per esempio nel diagramma con le visualizzazioni noto che ci sono un po’ di outliers e cerco quindi di zoommare ma il risultato migliora di poco. Ho più o meno gli stessi risultati sia guardando i diagrammi con age che con videoDuration.

Ho fatto poi il boxplot che mostra la distribuzione dei dati, evidenziando la mediana, i quartili e i valori anomali e ho notato che y si muove tra 0.10 e 0.30 con qualche outliers verso l’alto.



Inoltre ho fatto la matrice di correlazione tra i dati prendendo solo le colonne numeriche e guardando i risultati si capisce che :

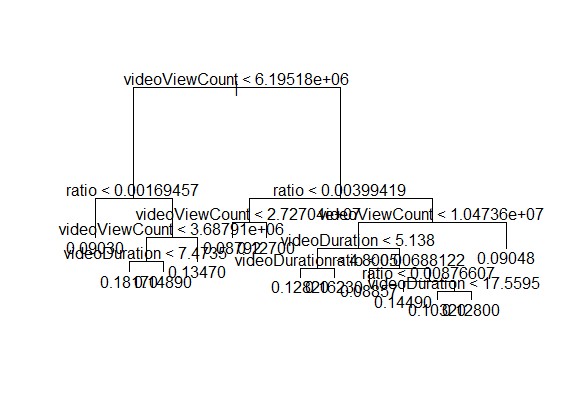
 commentsCount ha una correlazione positiva moderata con videoViewCount (0.155) e una correlazione negativa debole con y (-0.149); likesCount ha una forte correlazione positiva con videoViewCount (0.911) e una correlazione moderata positiva con y (0.449); videoViewCount ha una correlazione positiva elevata con likesCount (0.911) e una correlazione moderata positiva con y (0.151); videoDuration ha correlazioni deboli con tutte le altre variabili: moderata con likesCount (0.233), debole con commentsCount (0.046) e y (0.039); y ha correlazioni negative moderate con commentsCount (-0.149) e ratio (-0.136), e una correlazione positiva moderata con likesCount (0.449) e videoViewCount (0.151); ratio ha una correlazione negativa forte con likesCount (-0.220) e correlazioni moderate con commentsCount (0.830) e y (-0.136).

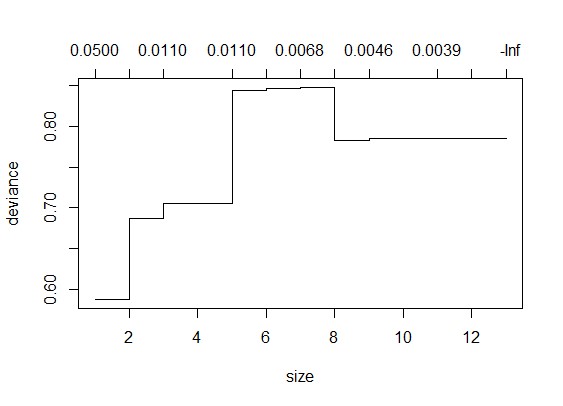
Ho costruito i due dataset di training,che rappresenterà il passato, e di test, che rappresenterà il futuro divisi rispettivamente in 75% e 25%. Dato che sono in ordine cronologico inverso quello di test sarà da 1 a n/4 e train sara da n/4 fino ad n. Passato e futuro nel senso che la predizione del “futuro” che faccio sul test dovrebbe corrispondere ai dati originali che ho della parte di test. Tuttavia questa predizione non è detto che nella realtà sarà corretta perche potrebbe essere che Kylie Jenner nel 2021 (anno in cui inizia il dataset di test) abbia fatto un boom, sia in senso negativo che positivo, per la quale i dati predetti non corrispondono a quelli reali/originali, ma vedremo poi che non è questo il caso.

Ora sono pronta a allenare i modelli.

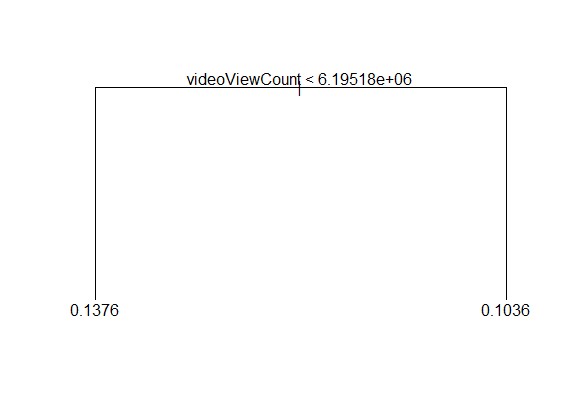
Il primo modello è la regressione lineare: ho utilizzato le variabili videoViewCount, videoDuration, age e ratio come predittori per la variabile di risposta y nei dati di addestramento (train) . Ho poi utilizzato il modello addestrato per fare predizioni sulla variabile di risposta y nei dati di test (test) ed ho salvato il risultato nella variabile pred\_lm. Infine ho calcolato il mean squared error (MSE), che rappresenta la media dei quadrati degli errori tra le predizioni del modello (pred\_lm) e i valori osservati nei dati di test (test$y) e l’errore è 0.01087297.

Il secondo modello è l’albero di regressione: ho diviso il dataset di train in due parti come se avesse anche lui a sua volta una parte di train e una di test facendo quindi la calibrazione e l’ho diviso in train1 che va dall'osservazione 96 alla 284, e train2 che va dalla prima all'osservazione 95.

Ho creato un albero di decisione chiamato tree1, la variabile di risposta y è predetta in base alle variabili esplicative videoViewCount, videoDuration, age, e ratio nel dataset train1. Visualizzo il grafico e aggiungo i nomi delle variabili ai nodi dell’albero. 

Ho poi potato l'albero tree1 utilizzando il dataset train2 e noto dal plot che il minimo di varianza è quello con 2 rami. 

Perciò poto l’albero a due rami e vedo che la variabile più significativa è viewcount.



Infine ho fatto previsioni sull'insieme di test utilizzando l'albero potato mt1 e calcolando il Mean Squared Error (MSE) per valutare le performance del modello rispetto alle vere osservazioni e l’errore è 0.006288603, più piccolo rispetto al modello precedente.

Il terzo modello è la random forest: ho creato il vettore err con 3 elementi, tutti inizializzati a zero. Questo vettore verrà utilizzato per memorizzare i valori del MSE ottenuti durante l'iterazione del ciclo for.Ho creato un ciclo for per vedere facendo la random forest ogni volta con variabili diverse qual è l’albero con l’errore più basso e quindi così ho capito quante variabili devo usare. 

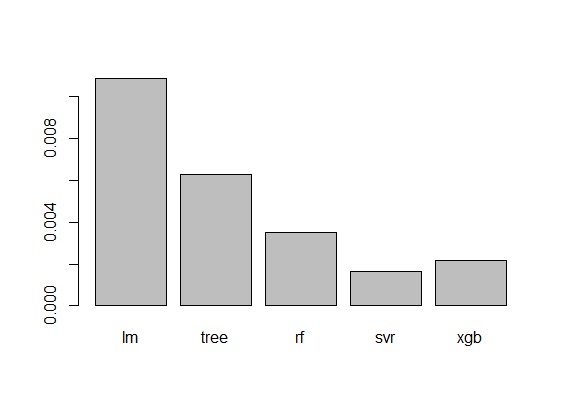
Il modello migliore lo ho quando utilizzo 4 variabili quindi faccio la random forest usando 4 variabili alla volta. Addestra il modello di foresta casuale con le variabili predittive videoViewCount, videoDuration, e age, utilizzando il set di addestramento train1. Si imposta mtry = 4 per specificare che ogni divisione del nodo dovrebbe considerare casualmente tutte e quattro le variabili. Calcola il Mean Squared Error (MSE) per le predizioni della foresta casuale sui dati di test e vedo che l’errore è 0.003268656 ancora più basso rispetto agli altri due.

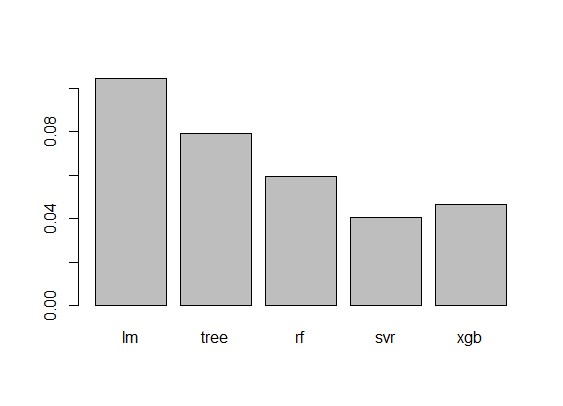
Il quarto modello è il Support Vector Regression: ho addestrato il modello utilizzando le variabili predittive videoViewCount, videoDuration, age e ratio , con il set di addestramento train. L’ho poi utilizzato per fare predizioni sui dati di test. Infine ho calcolato il Mean Squared Error (MSE) per le predizioni del modello SVR sui dati di test e l’errore è 0.001639424 ancora più basso degli altri.

L’ultimo modello è XGBoost(eXtreme Gradient): ho selezionato le colonne 3, 5, 9 e 10 dal dataset di addestramento che corrispondono alle variabili predittive videoViewCount, videoDuration, age e ratio e le ho assegnate a una nuova variabile chiamata dataset. Ho poi convertito la colonna age all'interno del dataset di addestramento in formato numerico. Infine ho convertito il dataset “dataset” in una matrice chiamandola “mat”. Ho fatto la stessa cosa ma dal dataset di test e ho chiamato il nuovo dataset “dataset\_test”, anche qui poi ho convertito la colonna age in formato numerico ed ho convertito “dataset\_test” in una matrice chiamandola “mat\_test”. Ho poi addestrato il modello XGBoost utilizzando il dataset di addestramento mat specificando per esempio Nround in modo che performasse 1000 volte e ogni volta migliorasse e con early stopping rounds gli dico che se dopo 3 giri il miglioramento è lo stesso allora si deve fermare.

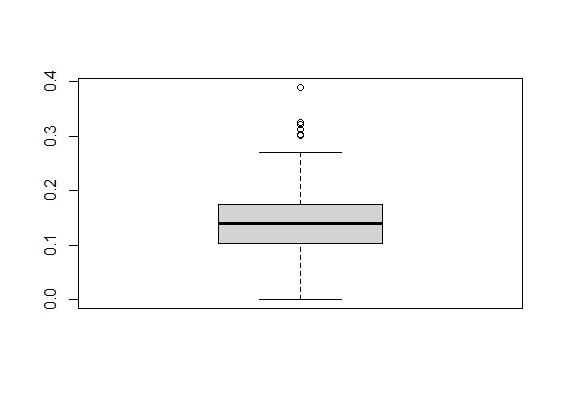


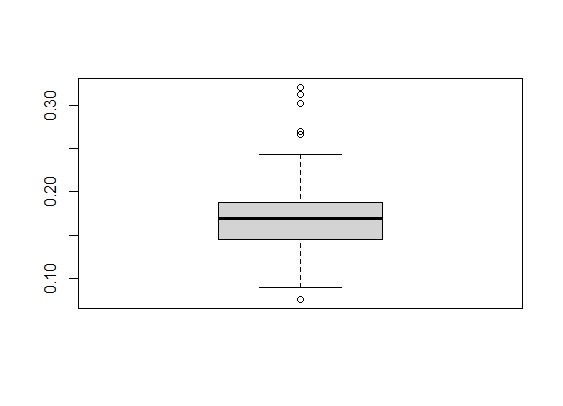
Ho utilizzato poi il modello addestrato per fare previsioni sui dati di test (mat\_test) ed ho calcolato l'errore quadratico medio (MSE) confrontando le previsioni del modello (pred\_xgb) con le vere risposte nei dati di test (test$y) e l’errore è 0.002150182, più alto del precedente.

Ho poi messo a confronto l’errore quadratico medio (MSE) di ogni modello facendo un barplot e qui si vede chiaramente come il miglior modello è SVR e il peggiore è la regressione lineare. 

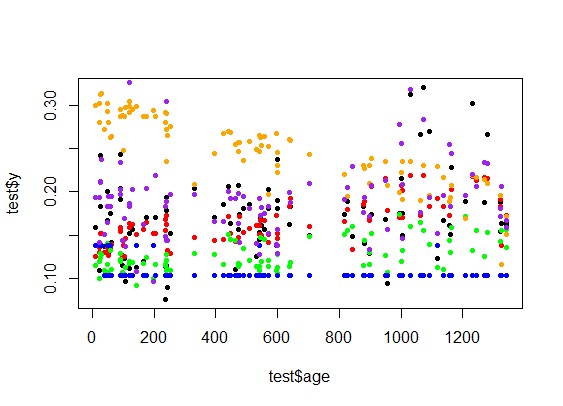
Ho fatto poi la radice dell'errore quadratico medio (RMSE) e anche qui il peggior modello è la regressione lineare mentre il migliore è SVR. 

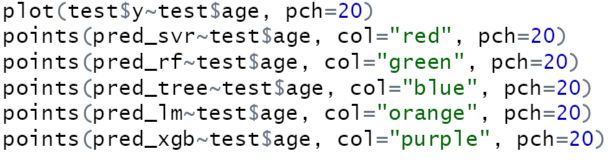
Ho visto poi i due boxplot che mostrano la distribuzione dei dati veri e quelli di test così ho visto la deviazione standard che sta anche nella variabile risposta. Se y va da 0.10 a 0.30 e la devianza sta intorno a 0.4 vuol dire che è buono ma non buonissimo perché c’è un po di variabilita nell’errore ma è abbastanza buono. Nei dati originali y va da 0.10 a 0.40 (compresi outliers) mentre nei dati predetti va da 0.10 a 0.30 quindi diciamo che i dati non sono proprio stimati perfettamente ma sono molto simili.

dati originali

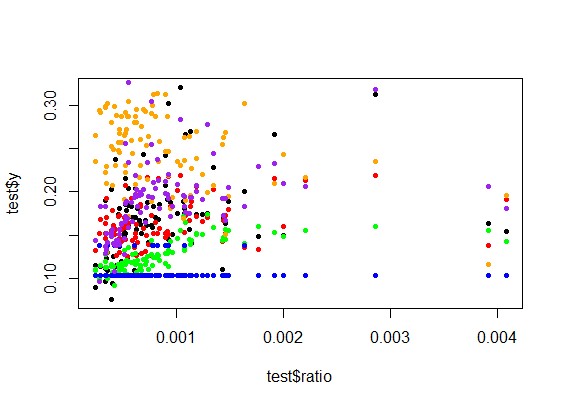
dati predetti

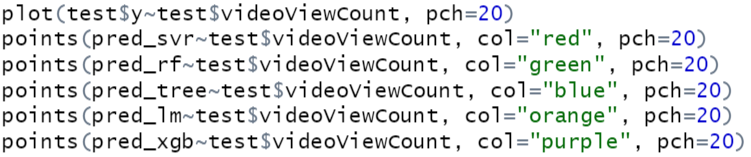
Infine ho valutato le predizioni rispetto ai dati delle variabili age, videoViewCount e ratio con i vari modelli per vedere proprio più nel dettaglio quanto i dati stimati si allontanano dagli originali.



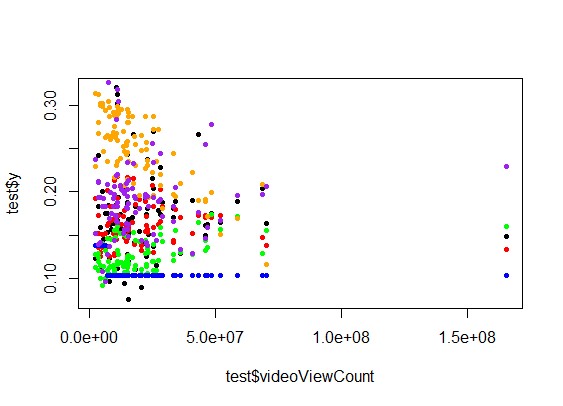


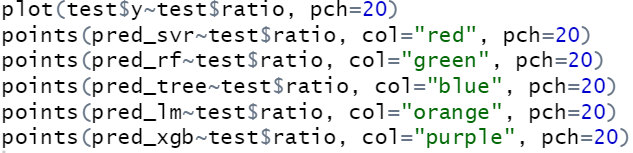
Qui, per esempio, si vede che la random forest(verde)sottostima e che la regressione lineare(arancione) sovrastima o ancora che l’albero di regressione(blu) prevede la maggior parte dei dati in maniera sbagliata ma , come avevamo visto già dal barplot, i modelli i cui dati stimati sono più simili ai reali sono SVR(rosso) e XGBoost (viola) e questo per la variabile age. Ma possiamo notare come anche nelle altre variabili è così.





Anche con la variabile videoViewCount, ovvero le visualizzazioni, notiamo come i migliori dati predetti sono quelli di SVR(rosso) e XGBoost (viola) e che anche qua l’albero di regressione(blu) prevede i dati in maniera sbagliata e che la random forest(verde)sottostima e che la regressione lineare(arancione) sovrastima.





Infine anche qui, con la variabile ratio, si vedono le stesse cose. Perciò così ho visto anche a livello grafico, variabile per variabile, qual è il modello i cui dati predetti sono più simili ai dati originali e come già visto dal barplot il modello migliore è SVR.