Big Mart Sales

Casarano Federico, Cavallo Giuseppe, Granata Marilisa, Mangini Giulia, Mastrorilli Valeria, Rana Alessandro, Tarricone Adriana

AA 2022-23

Progetto finale laboratorio di Data Science

1. Scelta del dataset

Per il progetto del laboratorio di Data Science il nostro gruppo si è concentrato sull'analisi del dataset Big Mart Sales. Questo dataset contiene informazioni raccolte da BigMart, una catena di supermercati negli Stati Uniti, riguardo ai dati di vendita di 1559 prodotti distribuiti in 10 negozi situati in diverse città.

```
dataset <- read.csv('BigMartSales.csv')
```

2. Obiettivo

La scelta di questo dataset è stata guidata da vari motivi. In primo luogo, come studenti di statistica, siamo sempre alla ricerca di opportunità per affinare le nostre competenze analitiche e applicarle a scenari reali e Big Mart Sales ci offre una panoramica dettagliata delle dinamiche di vendita dei prodotti in diverse categorie, tipi di negozi e posizioni geografiche. In secondo luogo, BigMart ha raccolto questi dati al fine di comprendere quali prodotti si vendano maggiormente in quali tipi di negozi. Inoltre, lo scopo è indagare sull'impatto che l'esposizione del prodotto nei negozi ha sulle vendite dello stesso. Questa sfida ci affascina, poiché ci offre l'opportunità di esplorare come le caratteristiche specifiche dei prodotti e le diverse caratteristiche dei negozi possano influenzare le vendite.

Per raggiungere i nostri obiettivi di analisi, ci proponiamo di seguire due approcci principali. In primis, costruiremo un modello predittivo per stimare le vendite di ciascun prodotto in un determinato negozio o in negozi generici con caratteristiche diverse. Ciò ci consentirà di comprendere meglio quali fattori influenzano le vendite dei prodotti.

Poi, effettueremo un'analisi di clustering per raggruppare i prodotti in base alle variabili disponibili, tenendo conto anche delle diverse vendite nei vari negozi. Questo ci permetterà di individuare segmenti di prodotti con caratteristiche simili e valutare come tali segmenti si differenziano nelle vendite tra i vari negozi.

Siamo convinti che l'esplorazione di questo dataset e l'applicazione delle nostre competenze analitiche su Big Mart Sales ci offriranno un'opportunità unica per approfondire le strategie di mercato adottate dalle grandi catene di supermercati e comprendere meglio il loro impatto sulle vendite.

3. Descrizione del dataset

Il dataset descrive le caratteristiche di ciascuno dei prodotti e dei negozi. Per ogni prodotto viene descritto l'ID univoco di ogni prodotto, il peso del prodotto, il contenuto di grassi, la percentuale di esposizione del prodotto all'interno di un negozio, la categoria a cui appartiene e il prezzo di listino del prodotto. Per ogni negozio, invece, viene descritto l'ID univoco del negozio, l'anno in cui è stato aperto, la dimensione, il tipo di città in cui si trova e se è un negozio di alimentari o un supermercato. Inoltre, l'ultima delle variabili registra le vendite del prodotto nel negozio specifico.

```
str(dataset) #analizzare la struttura del dataset
```

```
## 'data.frame':    8523 obs. of  12 variables:
##  $ Item_Identifier          : chr  "FDA15" "DRC01" "FDN15" "FDX07" ...
##  $ Item_Weight              : num  9.3 5.92 17.5 19.2 8.93 ...
```

```
##  $ Item_Fat_Content         : chr  "Low Fat" "Regular" "Low Fat" "Regular" ...
##  $ Item_Visibility          : num  0.016 0.0193 0.0168 0 0 ...
##  $ Item_Type                : chr  "Dairy" "Soft Drinks" "Meat" "Fruits and Vegetables" ...
##  $ Item_MRP                 : num  249.8 48.3 141.6 182.1 53.9 ...
##  $ Outlet_Identifier        : chr  "OUT049" "OUT018" "OUT049" "OUT010" ...
##  $ Outlet_Establishment_Year: int  1999 2009 1999 1998 1987 2009 1987 1985 2002 2007 ...
##  $ Outlet_Size              : chr  "Medium" "Medium" "Medium" "" ...
##  $ Outlet_Location_Type     : chr  "Tier 1" "Tier 3" "Tier 1" "Tier 3" ...
##  $ Outlet_Type              : chr  "Supermarket Type1" "Supermarket Type2" "Supermarket Type1" "Groc
##  $ Item_Outlet_Sales        : num  3735 443 2097 732 995 ...
```

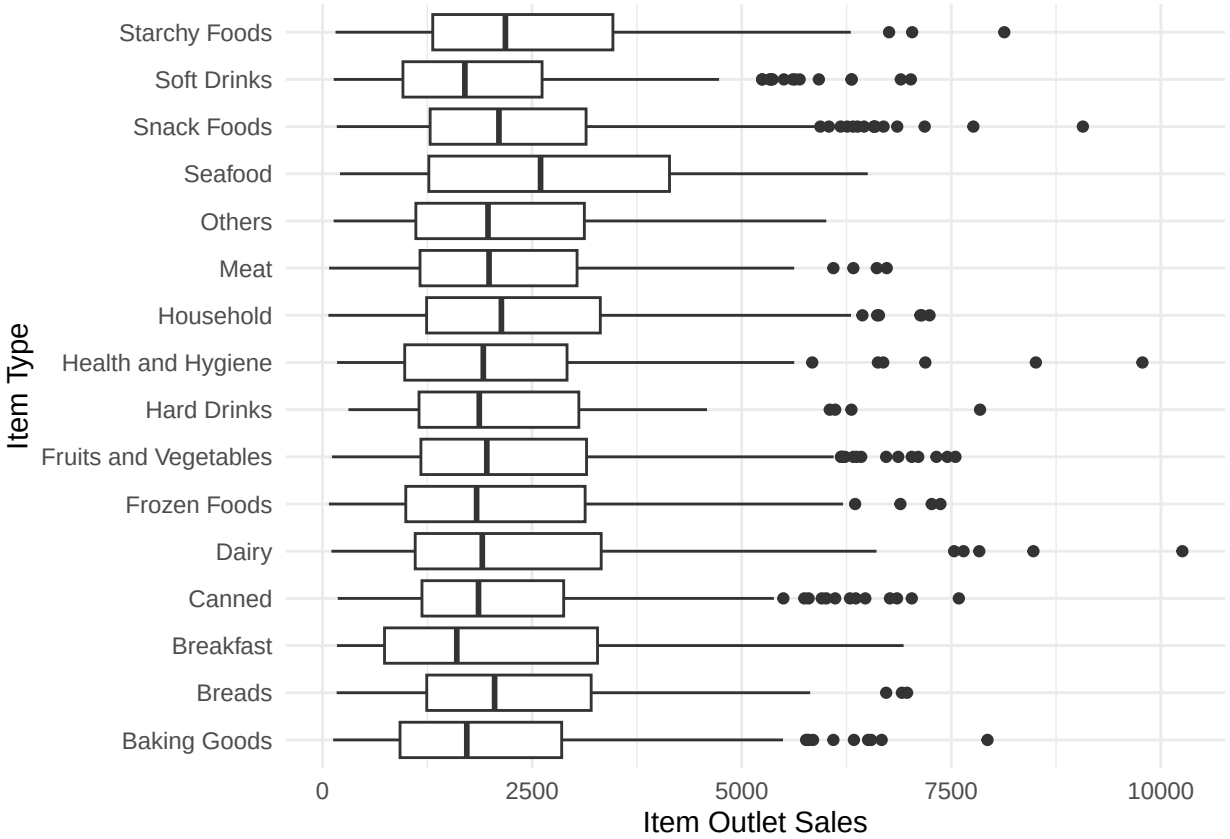
#Il dataset è composto da 8523 osservazioni e 12 variabili, di cui 5 quantitative e 7 qualitative.

Con il subset è stato filtrato il dataframe, escludendo le righe con Outlet_Size corrispondente a un valore nullo. Il nuovo dataframe ha, quindi, circa la metà delle osservazioni, cioè 4650. Analizzando la variabile Item_Fat_Content si è notato che alcuni valori non erano scritti in maniera uniforme, quindi si è proceduto a renderli tali.

##ANALISI ESPLORATIVA DEI GRAFICI

Trasformazione della variabili

```
ggplot(dataset_cleaned, aes(x = Item_Type, y = Item_Outlet_Sales)) +
  geom_boxplot() +
  labs(x = "Item Type", y = "Item Outlet Sales") +
  theme_minimal() +
  coord_flip()
```

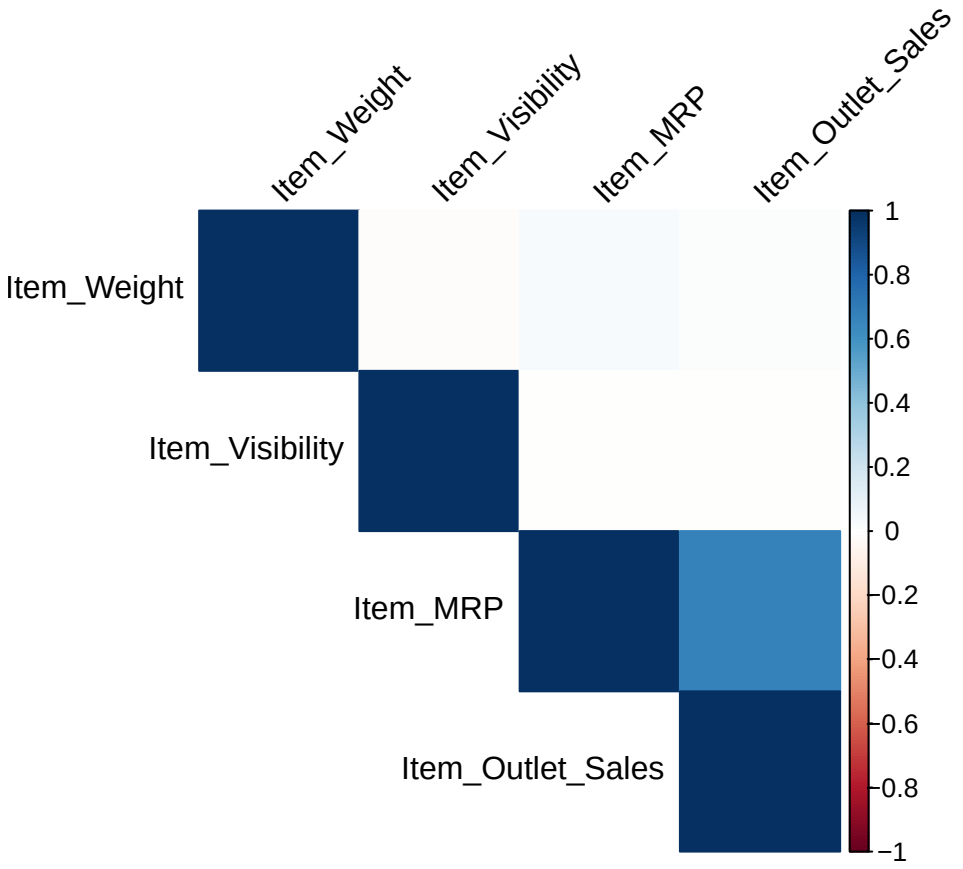


Creazione nuova variabile Prezzo_Unità

```
dataset_cleaned %<>%
  mutate(Prezzo_Unita = Item_MRP / Item_Weight)
```

Correlazione tra variabili

```
correlation <- cor(dataset_cleaned[, c("Item_Weight", "Item_Visibility", "Item_MRP",
                                       "Item_Outlet_Sales")])
corrplot(correlation, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45)
```



##REGRESSIONE LINEARE

```
data <- dataset_cleaned

# Creazione del modello di regressione lineare
model <- lm(Item_Outlet_Sales ~ Item_MRP, data = dataset_cleaned)

# Analisi del modello
summary(model)
```

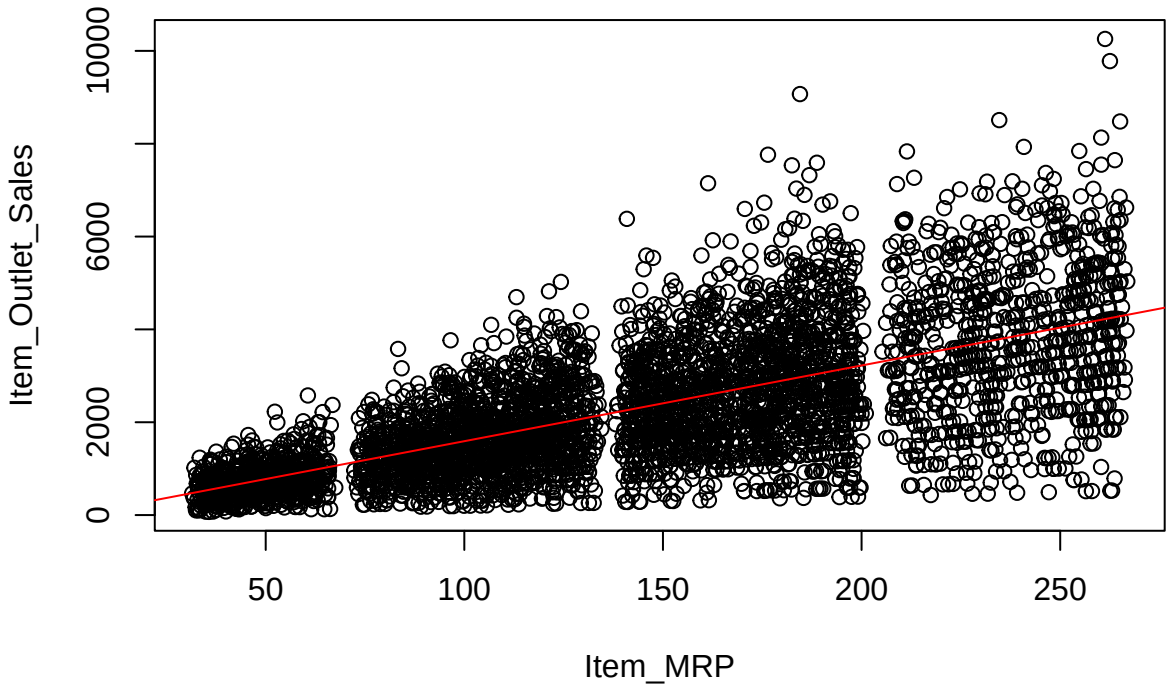
```
##
## Call:
## lm(formula = Item_Outlet_Sales ~ Item_MRP, data = dataset_cleaned)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3723.6  -612.0   -64.5   561.6  6099.9
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.1939    39.9861   -0.98    0.327
## Item_MRP     16.3089     0.2582   63.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1099 on 4648 degrees of freedom
## Multiple R-squared:  0.4618, Adjusted R-squared:  0.4617
## F-statistic:  3989 on 1 and 4648 DF,  p-value: < 2.2e-16
```

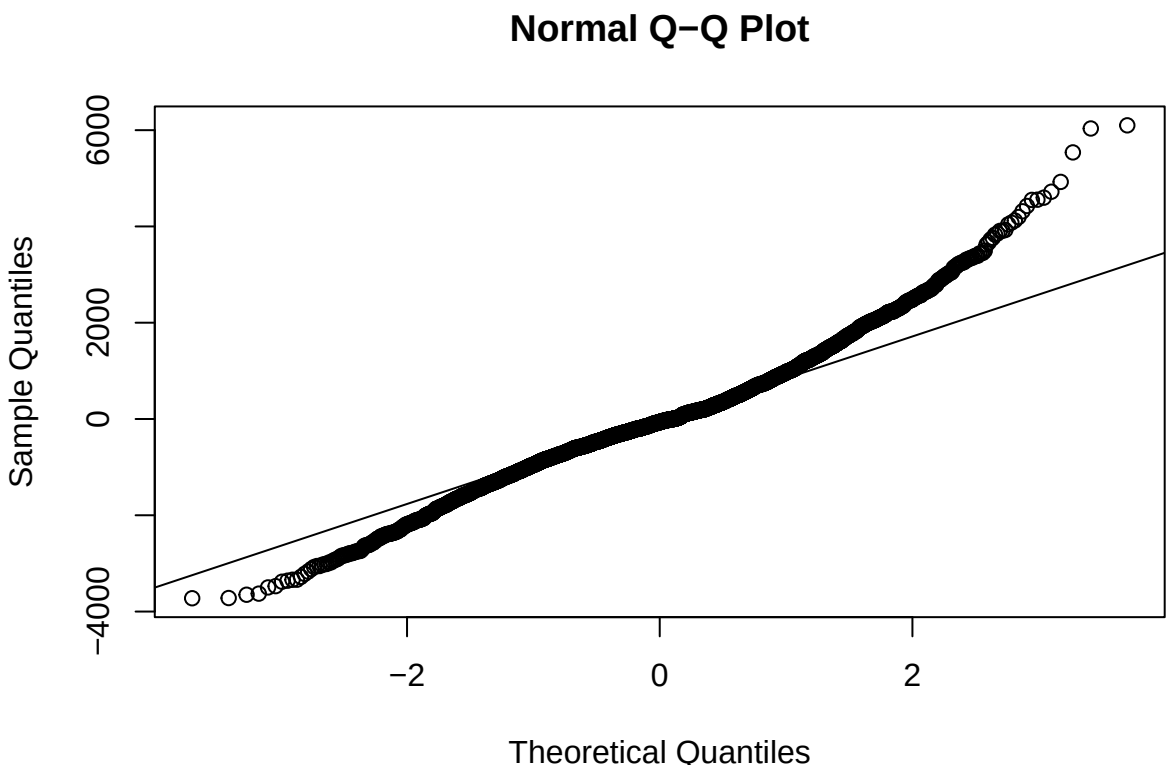
```
# Valutazione delle prestazioni del modello
r_squared <- summary(model)$r.squared
cat("Il coefficiente di determinazione (R-squared) del modello è", r_squared, "\n")
```

```
## Il coefficiente di determinazione (R-squared) del modello è 0.4618478
```

```
# Retta di regressione
plot(Item_Outlet_Sales ~ Item_MRP, data = data)
abline(model$coefficients, col = "red")
```




```
# Distribuzione in quantili
qqnorm(model$residuals)
qqline(model$residuals)
```



Ci siamo occupati di costruire un modello di regressione lineare chiamato model: eravamo interessati alle vendite del prodotto nel negozio specifico, quindi la nostra variabile di outcome da prevedere è Item_Outlet_Sales. Abbiamo fatto più prove, utilizzando sempre diverse covariate, a volte anche aggiungendone più di una. Facendo poi il riepilogo dei risultati e calcolandoci R^2, misura di fit del modello, ci siamo resi conto che la variabile più significativa che permette di avere un R^2 maggiore è Item_MRP, cioè la variabile che rappresenta il prezzo massimo al dettaglio (prezzo di listino) del prodotto. È anche un modello molto semplice in quanto utilizza solo una covariata, quindi si può adattare facilmente ad altri valori, essendo molto poco overfittato.

Abbiamo provato ad utilizzare una metodologia differente, per intravedere possibili modelli più efficienti in termini di risultati.

##REGRESSIONE LASSO

Dopo aver eseguito una conversione del dataset in un oggetto di tipo table, effettuato alcune modifiche di preparazione alle variabili di interesse del modello di regressione Lasso, si è diviso il dataset in train set e test set.

Creazione del training set e del test set

```
set.seed(1234)
train_indices <- sample(nrow(dataset_cleaned), 0.7 * nrow(dataset_cleaned))
train_data <- dataset_cleaned[train_indices, ]
test_data <- dataset_cleaned[-train_indices, ]
```

Effettuando la regressione sulla componente dipendente non trasformata, i risultati ottenuti non sono stati esaustivi. I residui presentavano una forma di eteroschedasticità a cono, aperto verso destra. Indica che la varianza dei residui aumenta man mano che i valori predetti aumentano. Ciò significa che l'errore del modello tende a essere sottostimato quando i valori predetti sono bassi e sovrastimato quando i valori predetti sono alti.

Per ovviare al problema della eteroschedasticità asimmetrica nei residui del modello, si applica una trasformazione logaritmica della variabili d'interesse l'analisi, con relativa modifica delle matrici "train" e "test", in

modo tale da rilevare un modello migliore che provi ad eliminare la presenza di eteroschedasticità.

Trasformazione logaritmica della variabile dipendente

```
# Creazione del training set e del test set
set.seed(1234)
train_indices <- sample(nrow(dataset_cleaned), 0.7 * nrow(dataset_cleaned))
train_data <- dataset_cleaned[train_indices, ]
test_data <- dataset_cleaned[-train_indices, ]
```

```
# Trasformazione logaritmica della variabile dipendente
train_data$Item_Outlet_Sales_log <- log(train_data$Item_Outlet_Sales)
test_data$Item_Outlet_Sales_log <- log(test_data$Item_Outlet_Sales)

# Applicazione delle trasformazioni alle variabili predittive
x_train_transformed <- as.matrix(train_data[, -c("Item_Outlet_Sales", "Item_Identifier")])
y_train_transformed <- train_data$Item_Outlet_Sales_log
x_test_transformed <- as.matrix(test_data[, -c("Item_Outlet_Sales", "Item_Identifier")])

# Addestramento del nuovo modello di regressione Lasso sulla variabile trasformata
lasso_model_transformed <- cv.glmnet(x_train_transformed, y_train_transformed, alpha = 1, nfolds = 5)

# Predizione sui dati di test utilizzando il nuovo modello Lasso
lasso_predictions_transformed <- exp(predict(lasso_model_transformed,
                                             newx = x_test_transformed, s = "lambda.min"))

# Calcolo del coefficiente di determinazione (R-squared) sulla variabile originale
r_squared_transformed <- cor(test_data$Item_Outlet_Sales, lasso_predictions_transformed)^2
print(paste("R-squared (trasformata):", round(r_squared_transformed, 4)))
```

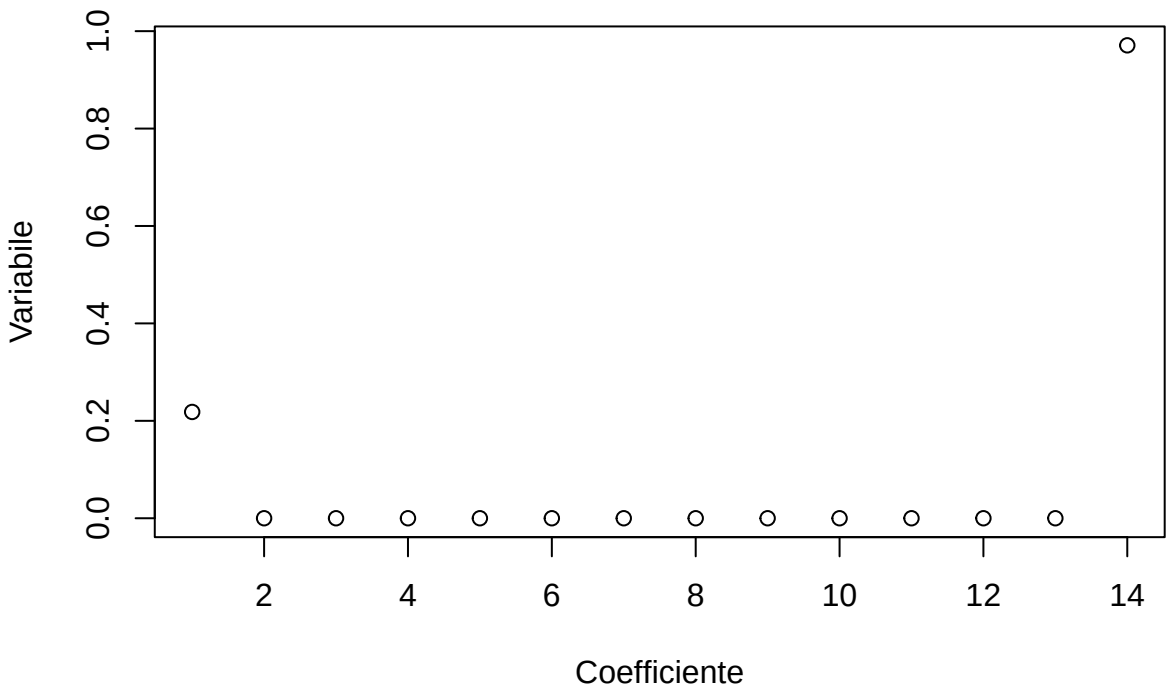
```
## [1] "R-squared (trasformata): 0.9999"
```

```
rmse <- sqrt(mean((test_data$Item_Outlet_Sales - lasso_predictions_transformed)^2))
print(paste("RMSE (trasformata):", round(rmse, 4)))
```

```
## [1] "RMSE (trasformata): 69.5555"
```

```
# Estrazione i coefficienti del modello Lasso
lasso_coef <- coef(lasso_model_transformed, s = "lambda.min")

# Visualizzazione dell'importanza delle variabili
plot(lasso_coef, xlab = "Coefficiente", ylab = "Variabile", main = "Importanza delle variabili")
```




```
# Validazione incrociata per stimare l'errore di generalizzazione
cv_error <- cv.glmnet(x_train_transformed, y_train_transformed, alpha = 1)
print(paste("Errore di generalizzazione minimo:",
            cv_error$cvm[cv_error$lambda == cv_error$lambda.min]))
```

```
## [1] "Errore di generalizzazione minimo: 0.000488282205921544"
```

```
# Calcolo dell'errore percentuale medio (MAPE)
mape <- mean(abs((test_data$Item_Outlet_Sales - lasso_predictions_transformed)
                 / test_data$Item_Outlet_Sales)) * 100
print(paste("MAPE (trasformata):", round(mape, 2)))
```

```
## [1] "MAPE (trasformata): 1.78"
```

L'errore percentuale medio (MAPE) sulle variabili trasformate è del 1.78%. Questo indica che le previsioni del modello Lasso trasformato hanno un errore medio del 1.78% rispetto ai valori effettivi della variabile dipendente trasformata.*

Complessivamente, i risultati suggeriscono che il modello Lasso trasformato con la variabile dipendente logaritmicamente trasformata ha una buona adattabilità ai dati, una buona capacità di generalizzazione e un errore di previsione ridotto.

##RANDOM FOREST

La griglia dei parametri rappresenta tutte le possibili combinazioni dei valori dei parametri che sono stati valutati durante l'ottimizzazione del modello Random Forest. Questo ci permette di esaminare diverse configurazioni dei parametri per identificare quella che produce le migliori prestazioni del modello.

```
# Addestramento del modello Random Forest
rf_mod <- train(
  x = train_data[, -c("Item_Identifier", "Item_Outlet_Sales")],
  y = train_data$Item_Outlet_Sales,
  method = 'ranger',
```

```
  trControl = trainControl(method = "cv", number = 5),
  tuneGrid = tgrid,
  num.trees = 400,
  importance = "permutation")

# Punteggio di validazione medio (RMSE)
mean_rmse <- mean(rf_mod$resample$RMSE)
print(paste("Punteggio di validazione medio (RMSE):", round(mean_rmse, 4)))
```

```
## [1] "Punteggio di validazione medio (RMSE): 43.6765"
```

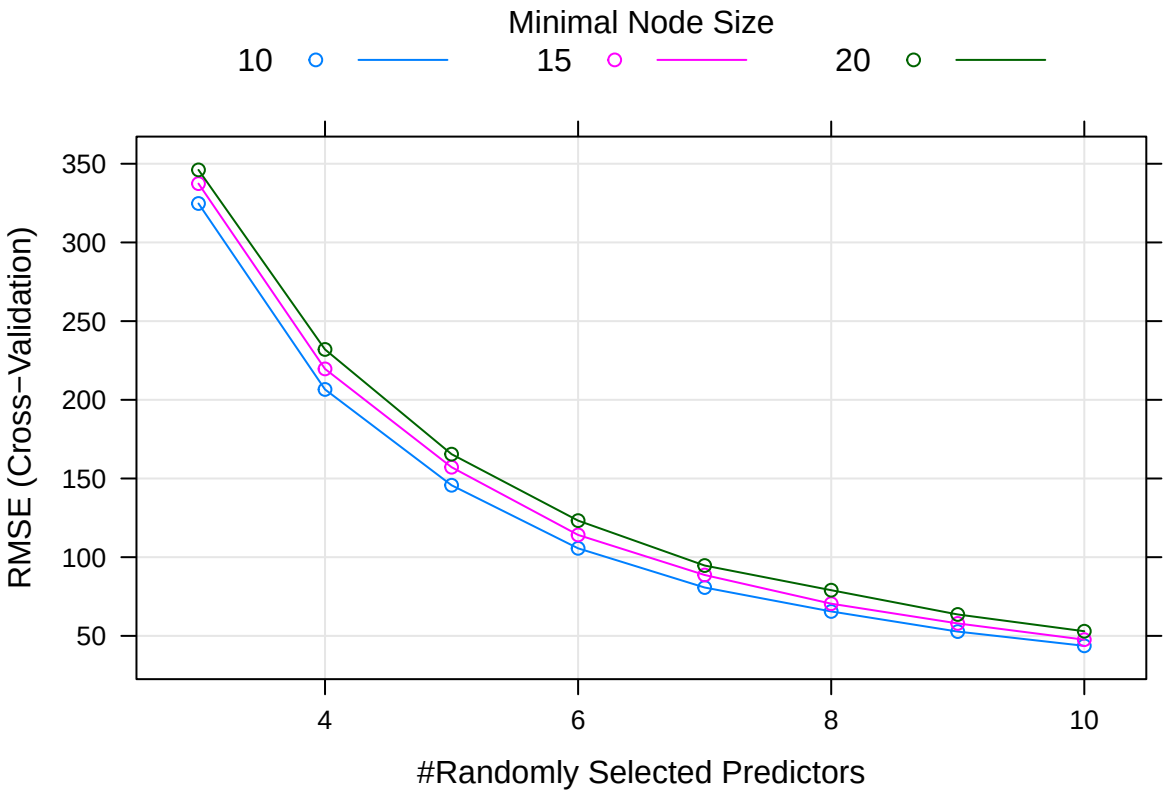
```
# Migliori parametri del modello
best_params <- rf_mod$bestTune
print("Migliori parametri del modello:")
```

```
## [1] "Migliori parametri del modello:"
```

```
print(best_params)
```

```
##    mtry splitrule min.node.size
## 22   10  variance            10
```

```
# Grafico dei risultati
plot(rf_mod)
```




```
variable_importance <- varImp(rf_mod)
```

L'output restituisce l'importanza delle variabili utilizzate dal modello Random Forest, misurata in termini di quanto contribuiscono alla riduzione dell'errore di predizione. Ogni variabile ha un punteggio di importanza relativa, indicato come "Overall" nella scala da 0 a 100.

Nell'output fornito, le variabili più importanti per la predizione dell'Item_Outlet_Sales_log (la variabile di

output trasformata nel modello) sono:

Item_MRP: con un'importanza del 44.8% Prezzo_Unità: con un'importanza del 6.19% Item_Weight: con un'importanza del 0.59%. Altre variabili come Outlet_Identifier, Item_Visibility, Outlet_Type, Outlet_Location_Type e Item_Fat_Content hanno un'importanza inferiore, ma contribuiscono comunque al modello. Da questi risultati, si può dedurre che le variabili legate al prezzo dei prodotti (Item_MRP e Prezzo_Unità) sono le più influenti per la predizione delle vendite dei prodotti nei negozi.

In sintesi, il modello Random Forest ha identificato le variabili più importanti per la predizione delle vendite degli articoli nei negozi, focalizzandosi principalmente sul prezzo dei prodotti. Per analizzare ulteriormente i dati, sarebbe opportuno considerare il clustering degli articoli in base alle covariate disponibili, tenendo conto anche delle diverse vendite nei diversi negozi attraverso l'espansione del dataset con la funzione "spread()".

##CLUSTER ANALYSIS

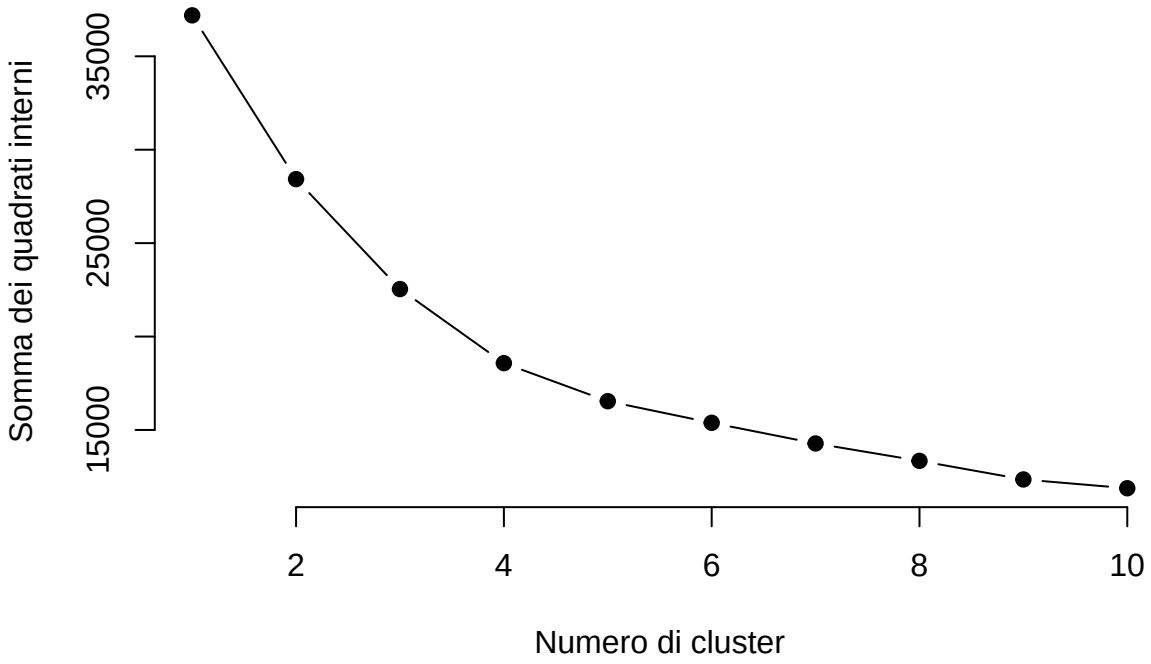
```
# Selezione delle covariate rilevanti per il clustering
selected_vars <- c("Item_Weight", "Item_Visibility", "Item_MRP",
                   "Outlet_Establishment_Year", "Outlet_Location_Type",
                   "Outlet_Type", "Item_Outlet_Sales")
clustering_data <- dataset_cleaned1[selected_vars]
# Trasformazione dei dati
# Trasformazione logaritmica delle vendite
```

```
# Normalizzazione dei dati
clustering_data <- scale(clustering_data)

# Selezione del numero di cluster
# Utilizzo del metodo del gomito
wss <- numeric(10)
for (i in 1:10) {
  kmeans_model <- kmeans(clustering_data, centers = i, nstart = 10)
  wss[i] <- kmeans_model$tot.withinss
}
```

Questi valori rappresentano la somma dei quadrati delle distanze dei punti all'interno di ciascun cluster per diversi numeri di cluster nell'analisi K-means. I valori di wss vengono utilizzati per determinare il numero ottimale di cluster nell'analisi K-means. Un valore di wss più piccolo indica una migliore suddivisione dei dati, poiché indica che i punti all'interno di ciascun cluster sono più simili tra loro.

```
# Plot
plot(1:10, wss, type = "b", pch = 19, frame = FALSE, xlab = "Numero di cluster", ylab = "Somma dei quadi
```

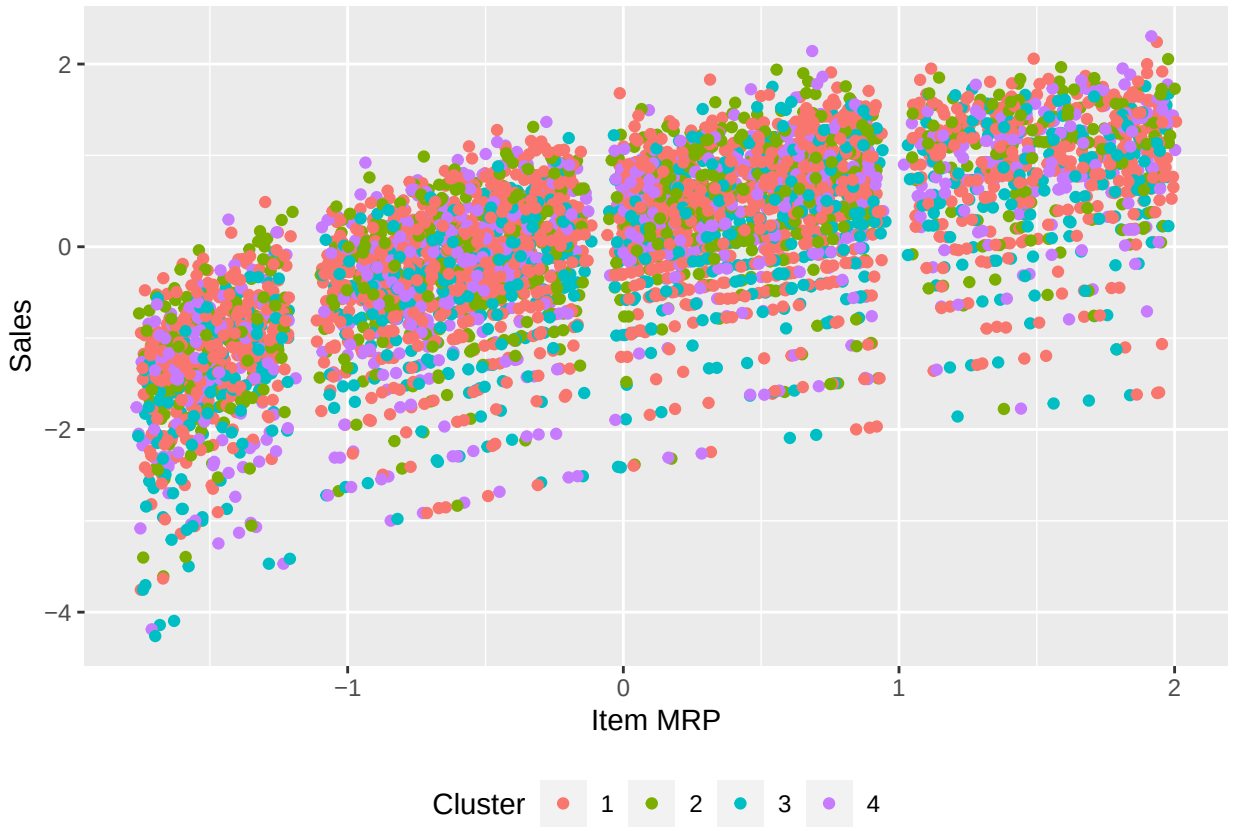



```
# Applicazione dell'algoritmo di clustering (con K-means)
k <- 4  # Numero di cluster scelto dal metodo del gomito
kmeans_model <- kmeans(clustering_data, centers = k, nstart = 10)
```

```
# Valutazione dei cluster
silhouette <- silhouette(kmeans_model$cluster, dist(clustering_data))
mean_silhouette <- mean(silhouette[, 3])
print(paste("Indice di Silhouette medio:", round(mean_silhouette, 2)))
```

```
## [1] "Indice di Silhouette medio: 0.32"
```

```
# Visualizzazione dei cluster
# Scatter plot
cluster_data <- data.frame(clustering_data, Cluster = as.factor(kmeans_model$cluster))
ggplot(cluster_data, aes(x = Item_MRP, y = Item_Outlet_Sales, color = Cluster)) +
  geom_point() +
  labs(x = "Item MRP", y = "Sales", color = "Cluster") +
  theme(legend.position = "bottom")
```




```
# Interpretazione dei cluster
cluster_summary <- aggregate(clustering_data, by = list(Cluster = kmeans_model$cluster), FUN = mean)
print(cluster_summary)
```

```
##   Cluster  Item_Weight Item_Visibility      Item_MRP Outlet_Establishment_Year
## 1       1 -0.001402623    -0.001468713 -0.0086315359                -0.1611273
## 2       2 -0.014841833     0.012617609  0.0225270867                 0.6509125
## 3       3 -0.005422716     0.007039806 -0.0006038689                 1.3276124
## 4       4  0.023008652    -0.016669003 -0.0046514416                -1.6498671
##   Outlet_Location_TypeTier 2 Outlet_Location_TypeTier 3
## 1                 -0.4999462                 -0.8164088
## 2                  1.9997849                 -0.8164088
## 3                 -0.4999462                  1.2246132
## 4                 -0.4999462                  1.2246132
##   Outlet_TypeSupermarket Type2 Item_Outlet_Sales
## 1                   -0.4992742       0.043450887
## 2                   -0.4992742       0.118770614
## 3                    2.0024768      -0.197613188
## 4                   -0.4992742      -0.008465968
```

In conclusione, in entrambi i modelli, i risultati mostrano buone capacità di previsione. Il modello di regressione Lasso trasformato con la variabile dipendente logaritmicamente trasformata ha ottenuto un alto coefficiente di determinazione (R-squared) e un basso errore radice quadrata medio (RMSE), indicando una buona adattabilità ai dati e un'accuratezza nella previsione delle vendite. Il modello Random Forest ha mostrato prestazioni simili, con valori di RMSE e R-squared che indicano una buona capacità di previsione. Sono stati applicati approcci diversi, ma entrambi hanno ottenuto risultati soddisfacenti.

Infine, abbiamo esaminato le caratteristiche distintive dei quattro cluster identificati. Ogni cluster ha presentato delle differenze nelle variabili selezionate. Ad esempio, il Cluster 1 ha mostrato una media

leggermente superiore per il peso dei prodotti e negozi aperti in anni più recenti, principalmente situati in città di Tier 3 e di tipo 2. Il Cluster 2, invece, ha presentato medie leggermente inferiori per le variabili peso dei prodotti, visibilità e prezzo, con negozi aperti in anni più recenti e principalmente situati in città di Tier 2 e di tipo 2. Il Cluster 3 ha evidenziato una media leggermente superiore per il peso dei prodotti, negozi aperti in anni più lontani, principalmente situati in città di Tier 1 e di tipo supermarket. Infine, il Cluster 4 ha mostrato una media leggermente inferiore per il peso dei prodotti, negozi aperti in anni più lontani, principalmente situati in città di Tier 3 e di tipo 2.