GREEN INVESTMENT CLASSIFICATOR

Università di Cagliari

Giuseppe Grosso

Matteo Fercia

INDICE

Introduzione

Nel seguente progetto è stata fornita una valutazione del fenomeno della Renewable Energy con il principale scopo di realizzare un classificatore il quale, servendosi di informazioni di natura diversa prese da Internet, come tweet o altri indici estrapolati da Tabelle in rete, potesse essere in grado di valutare la convenienza di un investimento in quel determinato settore, per ciascuno dei 37 Paesi scelti per svolgere la seguente analisi.

L'intenzione era quella di valutare se dati non strutturati come i tweet o gli altri dati potessero fornire delle utili informazioni al fine di ottenere delle previsioni sulla response value scelta, ovvero il Risultato dell'Investimento, che è stata codificata come una variabile categorica binaria con possibili realizzazioni "Positive" e "Negative", e confrontare le predictions ottenute con real-life data per asserire se effettivamente, il metodo di apprendimento statistico adoperato ci consenta di ottenere dei buoni risultati.

Utilizzando anche la Cluster Analysis e la PCA si sono analizzate nel dettaglio le relazioni tra le variabili, cercando dei risultati che in qualche modo ci aiutassero a comprendere il fenomeno.

Primo capitolo: Creazione Dataset su Python

1.1 Estrazione dati social media

Per prima cosa, servendosi del tool **"SNScraper"** sono stati raccolti i tweet relativi ai diversi paesi inserendo all'interno del parametro **"query"** della funzione realizzata per estrapolare i tweet ricerca *"green energy"*, *"renewable energy"*, *"sustainable development"* e *"green economy"* insieme allo specifico geocode.

Per evitare di dover ripetere il procedimento 37 volte, ci siamo serviti della funzione **"string.format()"** per inserire come argomento la lista *"country_list"*, la quale è stata importata da un'altro file, in modo tale da iterare al suo interno e selezionare solo la stringa in posizione zero, ovvero il geocode relativo al paese e il raggio di ricerca dello scraper (impostato di 1000km per tutti); questo è stato fatto per i quattro script in cui venivano ricercate le rispettive query.

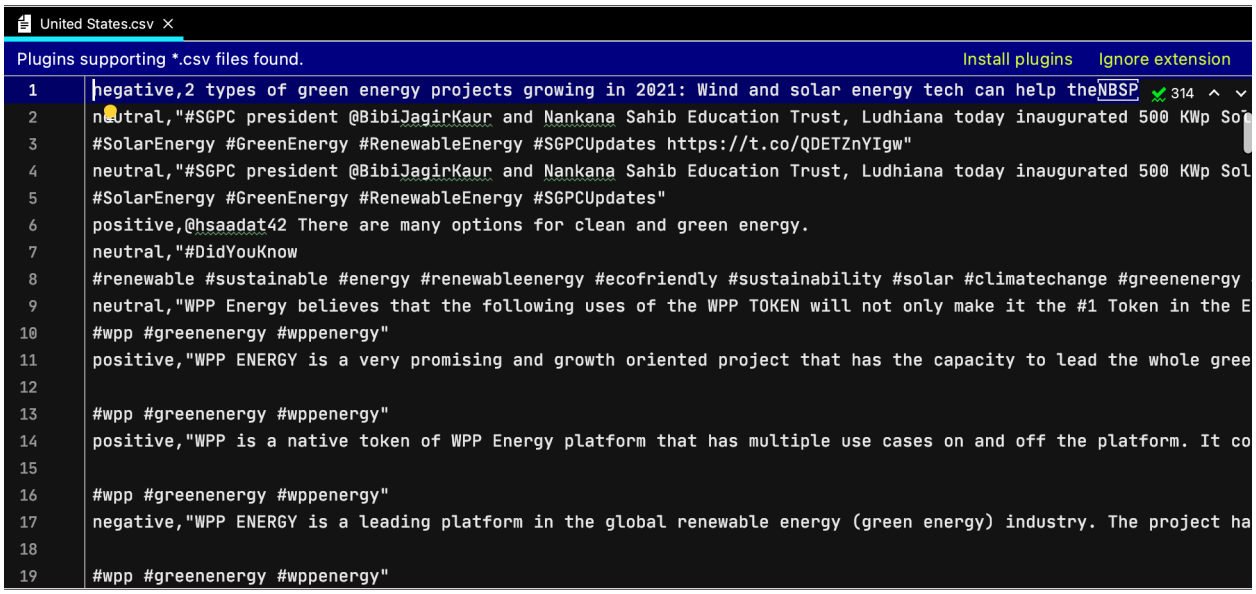
```
95  if __name__ == '__main__':
96      for item in country_list:
97          query = 'green economy:"{}"'.format(item[0])
98          count = 2500
99          with open('{}.csv'.format(item[1]), 'a') as csvfile:
100             for tweet in get_tweets_sentiment(query, count):
101                 writer = csv.writer(csvfile)
102                 writer.writerow((tweet['sentiment'], tweet['text']))
```

Come si vede dal breve estratto di codice, utilizzando come count 2500 ci siamo assicurati di "acchiappare" il numero massimo di tweet per paese, nell'arco di 2-3 settimane, che sono stati inseriti in un file CSV in modo da averne uno per ogni paese: anche in questo caso, ci siamo serviti della funzione **"string.format()"** inserendola stavolta come argomento del comando di apertura del CSV **"with open()"**, annidato al ciclo for che iterava all'interno di ogni elemento di *"country_list"*, per aprire in scrittura un apposito file CSV chiamato come l'elemento 1 della lista ovvero il nome del paese associato allo specifico **geocode**.

Ovviamente, prima di essere inseriti dentro al file csv, i tweet hanno subito diverse procedure di elaborazione testuale la cui funzione era quella di eliminare gli elementi superflui in modo tale da avere solo il testo: con la **RegEx** sono stati rimossi i link e i caratteri speciali dai tweet, i quali sono stati in seguito oggetto di **Parsing** per valutarne la struttura e successivamente inseriti come argomento di una funzione che si servisse della libreria **"TextBlob"** per eseguirne la sentiment analysis, al fine di ottenere una struttura dati composta dal sentiment, che poteva essere *"positive"*, *"negative"* o *"neutral"*, e il testo del tweet.

Una volta terminate tutte queste procedure, il risultato ottenuto veniva inserito nel file csv relativo al paese, con il comando **"csv.writer.writerow()"** come si può evincere dalle ultime righe del codice riportato sopra.

Il risultato ottenuto è il seguente:



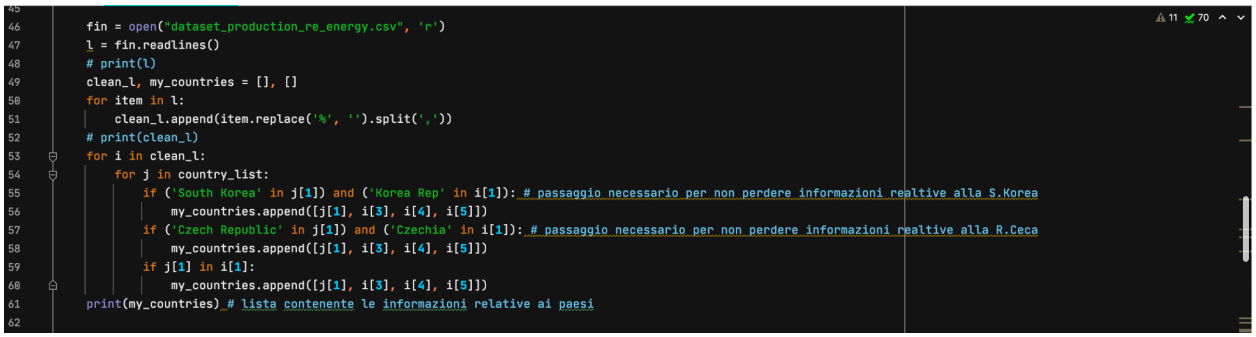
1.2 Estrazione dati da tabelle prese su Internet

Ottenuti i 37 file csv contenenti il testo del tweet e il sentiment associato, abbiamo proseguito estraendo informazioni di natura diversa da tabelle trovate su Internet.
Le tabelle che abbiamo estrapolato fanno riferimento a dati sui consumi e produzione di energia dei diversi paesi, nonché altri indici come popolazione, emissioni di anidride carbonica, aiuti pubblici ricevuti per il settore e altri.
Prima di passare all'estrapolazione delle suddette informazioni, è stata creata una struttura dati, più precisamente un dizionario, avente come chiave lo specifico paese e come valore una lista contenente a sua volta il numero di tweet positivi, negativi e neutri per lo stesso; per la creazione del dizionario, è stato necessario servirsi ancora una volta di **"with open()"**, aprendo stavolta i diversi csv creati in lettura e inserendo il contenuto in una lista utilizzando il metodo **"readlines()".** Subito dopo, con un ciclo **"for"** e tre istruzioni condizionali **"if"**, il programma riconosceva la stringa relativa al sentiment nella lista e aggiornava ogni volta le rispettive somme di *"positive"*, *"negative"* e *"neutral"* per ciascuno dei 37 paesi, andandole a salvare dentro al dizionario *"country_sentiment"*.
Una volta ottenuto il dizionario, è stata eseguita l'estrapolazione delle informazioni dalla prima tabella, relativa al consumo e alla produzione di Renewable Energy: utilizzando la libreria **"wikipedia",** inserendo l'url della pagina di Wikipedia abbiamo recuperato la tabella e servendosi della libreria **"pandas"** è stata convertita in un file csv nel seguente modo;

```
38      html = wp.page("List_of_countries_by_renewable_electricity_production").html().encode("UTF-8")
39      try:
40          df = pd.read_html(html)[1]
41      except IndexError:
42          df = pd.read_html(html)[0]
43      print(df.to_string())
44      df.to_csv("dataset_production_re_energy.csv", sep=',')
```

dopo aver ottenuto il file csv con le informazioni della tabella, aprendolo in lettura e riportando tutte le righe in una lista con **"readlines()"**, servendosi di un ciclo **"for"** abbiamo iterato dentro ogni elemento della lista con i paesi, poi con un secondo ciclo **"for"** annidato abbiamo iterato dentro la lista del csv, che è stata prima "pulita" dagli eventuali caratteri speciali con il metodo **"replace()"** e anche splittata in sottoliste con **"split()"**, e inserendo una condizione di uguaglianza tra gli elementi delle due liste che contenevano i nomi dei paesi, abbiamo estrapolato solo le relative informazioni inserendole in una nuova lista *"my_countries"* facendo

attenzione ai paesi il cui nome poteva risultare scritto in un modo differente, ripetendo la medesima procedura anche per le altre tabelle trovate in rete.



```
fin = open("dataset_production_re_energy.csv", 'r')
l = fin.readlines()
# print(l)
clean_l, my_countries = [], []
for item in l:
    clean_l.append(item.replace('%', '').split(','))
# print(clean_l)
for i in clean_l:
    for j in country_list:
        if ('South Korea' in j[1]) and ('Korea Rep' in i[1]): # passaggio necessario per non perdere informazioni realtive alla S.Korea
            my_countries.append([j[1], i[3], i[4], i[5]])
        if ('Czech Republic' in j[1]) and ('Czechia' in i[1]): # passaggio necessario per non perdere informazioni realtive alla R.Ceca
            my_countries.append([j[1], i[3], i[4], i[5]])
        if j[1] in i[1]:
            my_countries.append([j[1], i[3], i[4], i[5]])
print(my_countries) # lista contenente le informazioni relative ai paesi
```

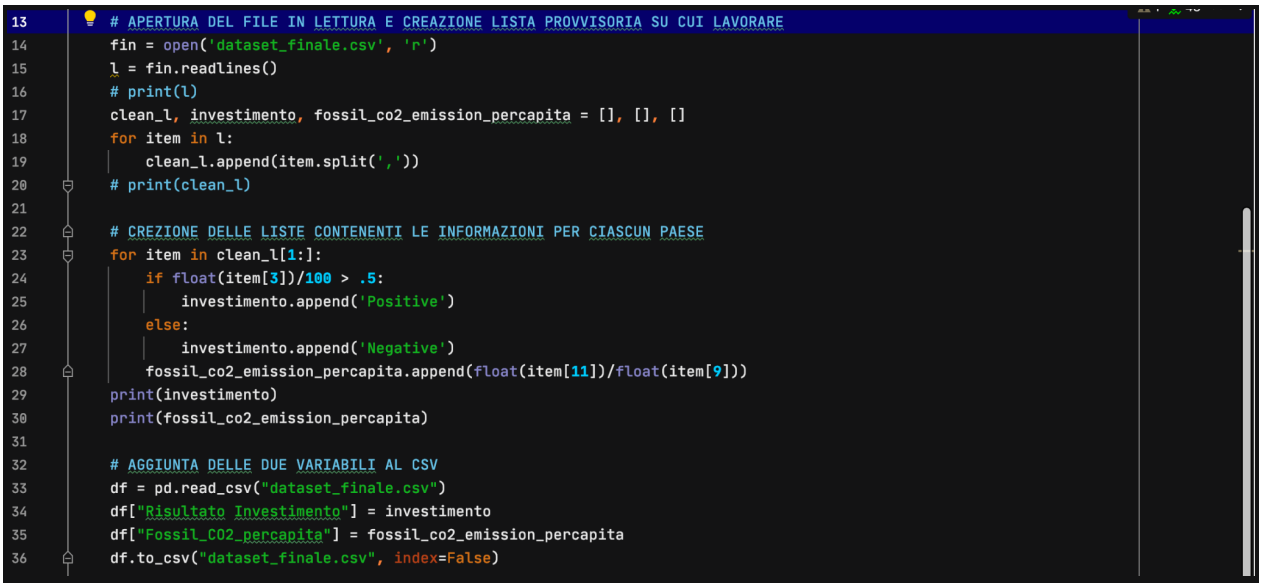
1.3 Creazione Dataset

In seguito abbiamo proseguito alla creazione di due dataset iniziali *"data_set.csv"* e *"data_set2.csv"* andando a scrivere nel primo le informazioni contenute nel dizionario *"country_sentiment"* e nella lista *"my_countries"* con **"csv.writer.writerow()",** e nel secondo le liste *"countries"* e *"countries3"* contenenti rispettivamente informazioni su consumo/produzione di energia elettrica e emissioni di CO2 utilizzando la stessa metodologia.

Una volta ottenute, abbiamo importato un terzo csv dal sito di "OECD", sul quale non sono state necessarie alcune modifiche importanti in quanto dal sito si poteva eseguire il custom della tabella e il download del file nel formato opportuno; dopodiché importando la libreria **"pandas"** è stato eseguito il merge dei diversi dataset ottenendo come risultato il file *"dataset_finale.csv"*, ovvero un prima versione del dataset da utilizzare per l'analisi statistica, il quale però necessitava di ulteriori modifiche.

La prima di queste modifiche consisteva nell'aggiungere un vettore contenente la quota di emissioni di anidride carbonica per capita, calcolata per ognuno dei 37 paesi, e l'aggiunta della response value *"Risultato Investimento"*, che è stata codificata come una variabile categorica con 2 possibili realizzazioni *"Positive"* e *"Negative"* assumendo che se la percentuale di Renewable Energy prodotta in quel paese fosse maggiore del 50% rispetto al totale dell'energia prodotta, allora l'investimento nel settore sarebbe stato conveniente.

Per far ciò abbiamo aperto in lettura il file *"dataset_finale.csv"*, inserendo con **readlines()** le righe del csv dentro una lista e successivamente, con un ciclo **"for"** che iterasse all'interno della stessa precedentemente splittata, abbiamo aggiunto a due liste appena inizializzate *"investimento"* e *"fossil_co2_emission_percapita"*, i valori "Positive" o "Negative" a seconda che la quota di produzione di RE fosse maggiore o minore di 0.5 nella prima, e il risultato del rapporto tra il totale delle emissioni per un paese e la relativa popolazione nella seconda. Infine, dopo averle ottenute, con l'utilizzo di **"pandas"** sono state aggiunte al dataset finale.



```
# APERTURA DEL FILE IN LETTURA E CREAZIONE LISTA PROVVISORIA SU CUI LAVORARE
fin = open('dataset_finale.csv', 'r')
l = fin.readlines()
# print(l)
clean_l, investimento, fossil_co2_emission_percapita = [], [], []
for item in l:
    clean_l.append(item.split(','))
# print(clean_l)

# CREZIONE DELLE LISTE CONTENENTI LE INFORMAZIONI PER CIASCUN PAESE
for item in clean_l[1:]:
    if float(item[3])/100 > .5:
        investimento.append('Positive')
    else:
        investimento.append('Negative')
    fossil_co2_emission_percapita.append(float(item[11])/float(item[9]))
print(investimento)
print(fossil_co2_emission_percapita)

# AGGIUNTA DELLE DUE VARIABILI AL CSV
df = pd.read_csv("dataset_finale.csv")
df["Risultato Investimento"] = investimento
df["Fossil_CO2_percapita"] = fossil_co2_emission_percapita
df.to_csv("dataset_finale.csv", index=False)
```

La seconda e anche ultima modifica riguardava l'aggiunta di una colonna che contenesse le quote di investimento in Renewable Energy di ciascun paese, da utilizzare per effettuare un confronto con le predizioni ottenute con il classificatore. Tuttavia, queste informazioni erano disponibili sul sito di "Irena" solo per le Regioni che contenevano i paesi utilizzati nell'analisi ("East Asia and Pacific", "OECD America", "Western Europe", "Latin America and The Carribeans", "Eastern Europe and Central Asia") e pertanto abbiamo calcolato quelle relative ai singoli paesi nel seguente modo: abbiamo prima ricavato il totale della produzione di energia per ciascuna regione sommando i valori ottenuti per i singoli paesi appartenenti alla regione con una semplice funzione **"ret_total_re_prod()"**, che accettava come unico argomento una lista e che iterasse all'interno della stessa andando a sommare i valori contenuti in seconda posizione di

ciascuna sottolista, che corrispondevano alla quota di produzione del singolo paese, andando a restituire il risultato finale ottenuto.

```
25  def ret_total_re_prod(list):
26      '''
27      Calcolo produzione totale di energia per regione
28      '''
29      total_re = 0
30      for item in list:
31          total_re += float(item[2])
32      return total_re
```

Successivamente servendosi della funzione **“ret_investment()”** che accettava in entrata una lista e due numeri (il Totale della produzione calcolata per una regione ottenuto con la funzione precedente e la quota di investimenti totali per le 5 macroaree ottenuta dal database di Irena), abbiamo ricavato la quota di investimenti per ciascun paese collocato in quelle 5 regioni di interesse facendo il rapporto tra la quota di energia rinnovabile prodotta da ciascun paese in una regione e il totale della produzione di energia nella stessa, e moltiplicando il risultato per la quota di investimenti totali per la suddetta regione, inserendo infine con il metodo **“append()”** una lista contenente il nome del paese e il prodotto ottenuto in un’altra lista, la quale conteneva tutti i valori calcolati per i paesi situati in quella regione e che costituiva inoltre il return della nostra funzione.

```
4   def ret_investment(lista, numero1, numero2):
5       '''
6       Calcoliamo la quota di investimento green per ciascun paese
7       '''
8       investimento_green = []
9       for item in lista:
10          investimento_green.append([item[0],(float(item[2]) / numero2) * numero1])
11      return investimento_green
```

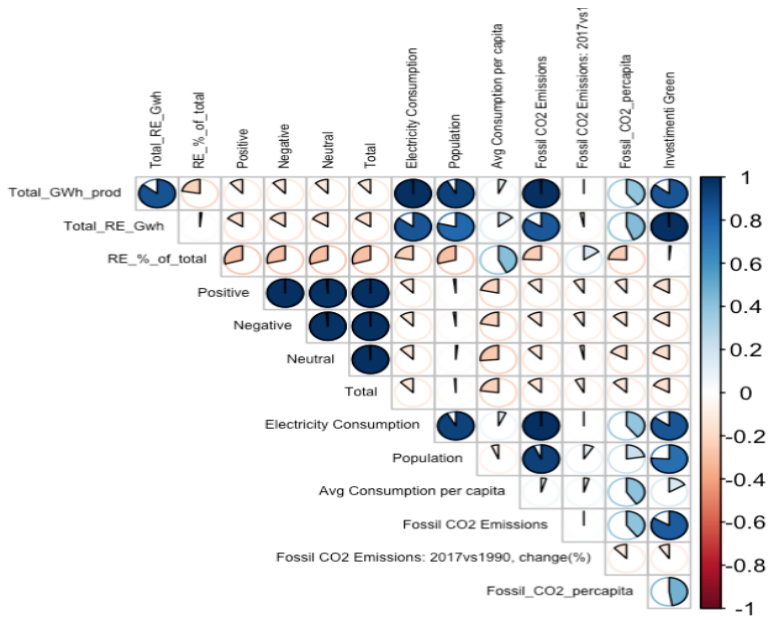
Una volta ottenute le 5 liste contenenti le informazioni specifiche a ciascun paese, importando la lista iniziale con i paesi scelti per l’analisi *"country_list"* e servendosi di un doppio ciclo for che iterasse in ogni elemento di *"country_list"* e in ogni elemento delle liste ottenute con la funzione precedente, abbiamo incluso esclusivamente le informazioni specifiche ai paesi di interesse inserendo un’istruzione condizionale che facesse un “check” sull’uguaglianza del nome dei paesi in entrambe le liste, facendo l’append in caso positivo su una nuova struttura dati inizializzata prima del processo **“temp_list”**, il quale è stato ripetuto per tutte le liste ottenute con la funzione **“ret_investment()”.**

Ottenuta infine quest'ultima, è stato eseguito una specie di sort manuale della stessa, in modo tale da avere i paesi al suo interno nello stesso ordine in cui erano già salvati all'interno del file csv e, servendosi nuovamente di **"pandas",** la nuova colonna è stata aggiunta al file *"dataset_finale.csv",* concludendo la prima parte del nostro lavoro.

Secondo capitolo: Analisi Statistiche su R

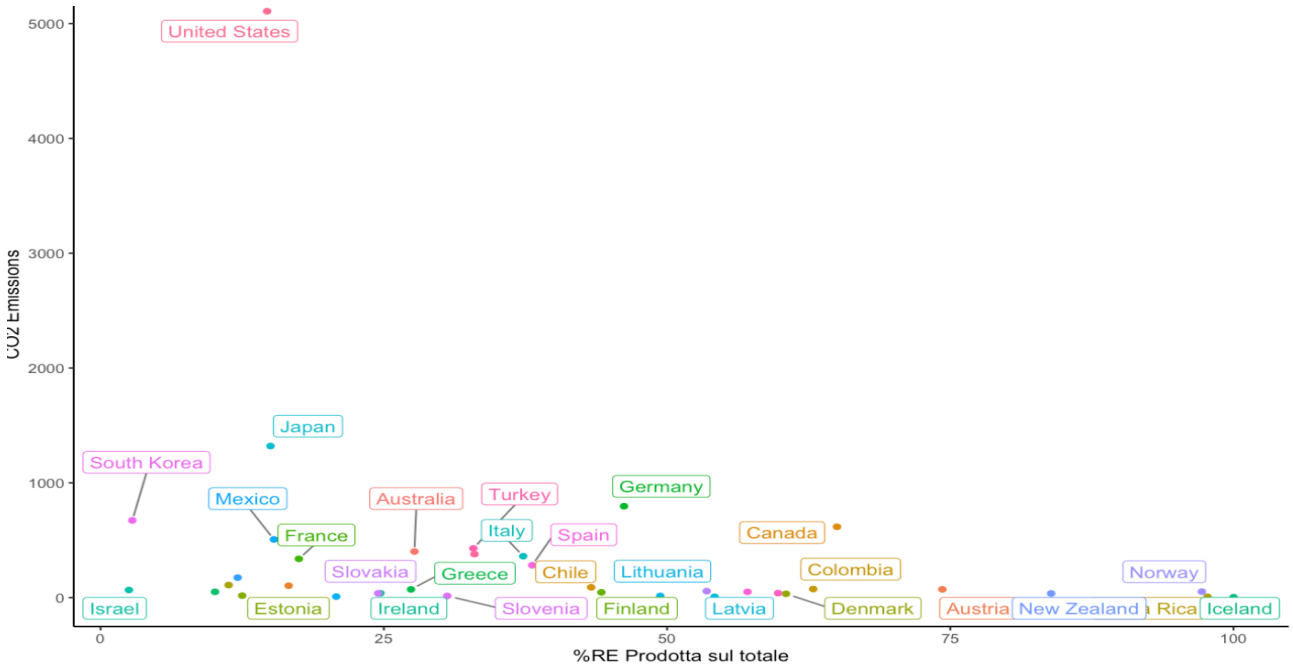
2.1 Analisi Esplorativa

Nella seconda parte, una volta costruito il dataset finale, abbiamo proceduto con l'analisi statistica dello stesso, iniziando con un'analisi esplorativa, per cercare di capirne le caratteristiche o eventuali relazioni tra le variabili scelte per spiegare il fenomeno. Di seguito alcuni dei grafici più significativi.

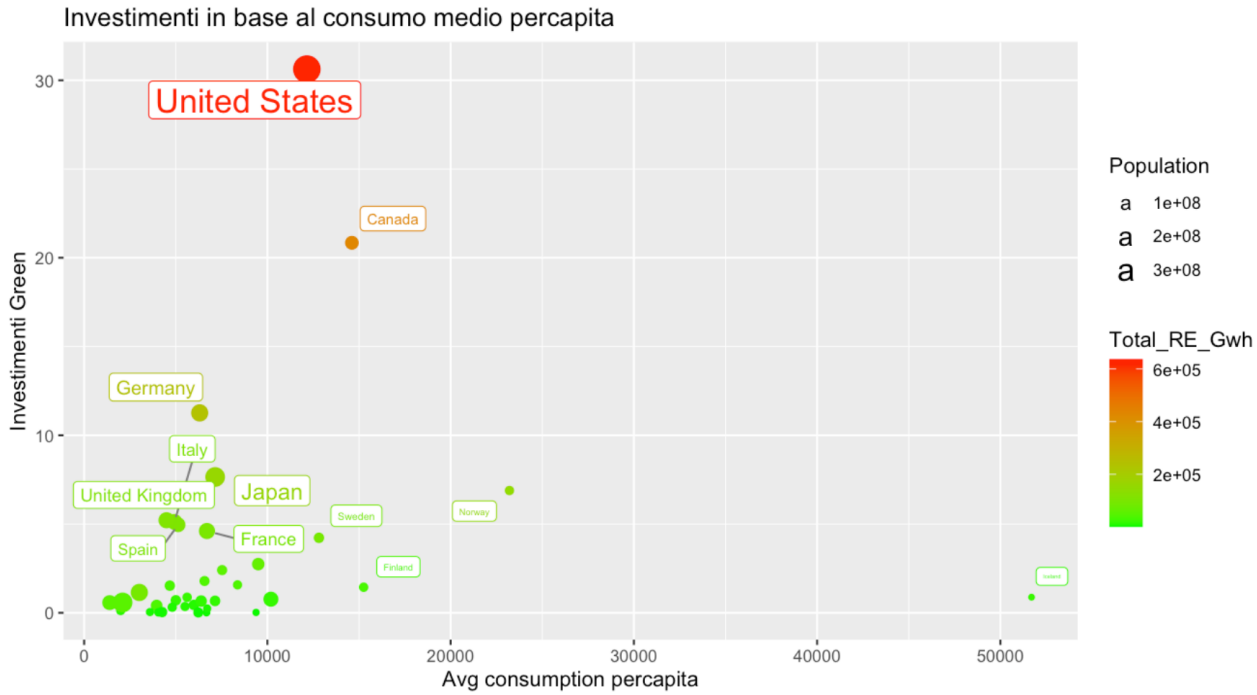


Dal **corrplot** è possibile osservare una forte correlazione positiva tra alcune variabili: tralasciando quella tra il sentiment dei tweet e il numero totale di tweet, in quanto quest'ultimo è una combinazione dei tre, è possibile osservare altre interessanti correlazioni: considerando il Totale della produzione di energia, notiamo una forte correlazione positiva con la Popolazione

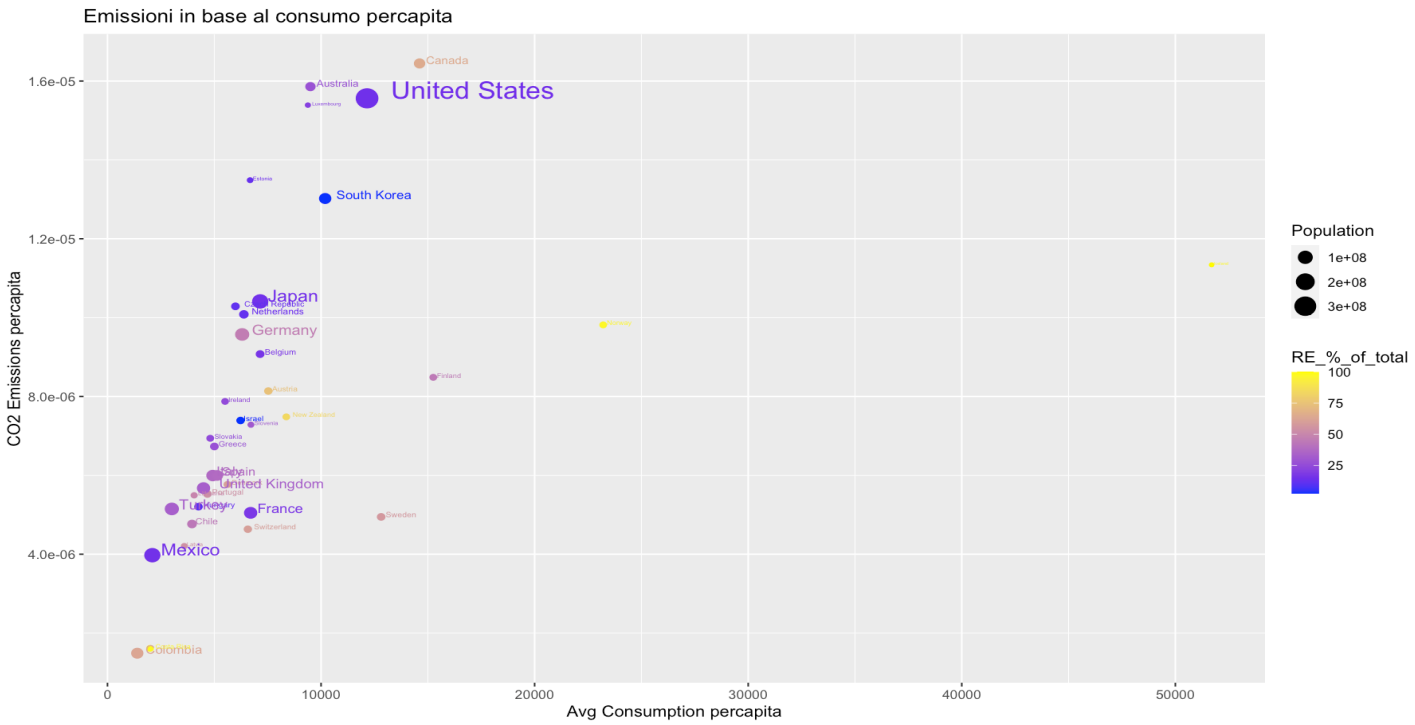
del paese (0.92). Con il consumo di elettricità (1) e con la quota di investimenti in green (0.86). Sulla base di queste relazioni è possibile concludere che i paesi ad investire maggiormente in green energy siano quelli con una popolazione abbastanza ampia e conseguentemente abbiano un ampio fabbisogno energetico. Considerando invece la quota di "Investimenti in Green" possiamo notare una serie di relazioni positive tra cui quella con il Tot dell'energia prodotta, spiegata poco fa, e anche quella scontata con il Tot di Renewable Energy prodotta dal paese (0.99), ma anche "Electricity Consumption" (0.85), "Population" (0.75), e "Fossil CO2 Emissions "(0.84). Queste relazioni ci portano a concludere che un paese investirà maggiormente in green quando il suo fabbisogno di energia dipendente a sua volta dalla popolazione sarà elevato, e le relative emissioni di CO2 saranno elevate.



Dal grafico è possibile osservare come ad eccezione degli Stati Uniti che presenta una quota di emissioni elevatissima rispetto alla produzione di Renewable energy, gli altri paesi abbiano una quantità di emissioni molto più contenuta, diversificandosi rispetto alla % di RE prodotta. Tra questi, possiamo notare come Islanda, Costa Rica e Norvegia siano i paesi con la percentuale di Renewable Energy prodotta sul totale più alta



Dal grafico è possibile notare la quota di investimenti in base al consumo per capita e dimensione del paese per popolazione;tralasciando i commenti su Canada e Stati Uniti, che sono gli stessi per i grafici precedenti, dal plot attuale possiamo osservare alcune situazioni interessanti: innanzitutto possiamo notare come, ad esclusione di Canada e Stati Uniti, gli altri paesi caratterizzati da una discreta quota di investimenti in RE siano perlopiù paesi situati nel continente Europeo, ad eccezione del Giappone, e tra questi compaiono anche paesi come Svezia e Norvegia che non sono caratterizzati da un'elevata popolazione. L'altra informazione interessante riguarda l'elevato consumo di energia per capita dell'Islanda, il quale risulta più alto addirittura di quello degli Stati Uniti.



Rispetto al grafico precedente, mettendo in relazione le CO2 Emissions per capita con il consumo medio possiamo notare come altri paesi rispetto a Canada e Stati Uniti, specialmente di dimensioni ridotte, come Australia, South Korea, Lussemburgo ed Estonia, siano caratterizzati da un dato di emissioni abbastanza elevato, soprattutto rispetto a paesi come il Giappone e la Germania, fortemente industrializzati e da cui ci aspetta una maggiore produzione di emissioni di anidride carbonica.

2.2 Regressione Lineare

Prima di passare alla classificazione, abbiamo preferito eseguire una Regressione sulla quota di investimenti di ciascun paese, sebbene questo predittore sia stato ricavato manualmente servendosi di informazioni relative al produzione e investimenti totali dei paesi e pertanto è possibile aspettarsi una certa relazione con queste variabili, per vedere se effettivamente quanto osservato nell'analisi esplorativa sia confermabile o meno: la prima regressione coinvolge la maggior parte dei predittori, e come ci si può aspettare, il modello risultante è un

modello “overfittato” con un **R2** pari a 1 e un **MSE** pari a 0.002 ma con conseguente scarsa performance predittiva per via dell’overfitting; dal summary inoltre notiamo come proprio la variabile *“Total_RE_Gwh”* sia quella più significativa, anzi l’unica variabile significativa.

Essendo questo risultato strano ad una prima vista, decidiamo di eseguire ulteriori modelli di regressione andando a considerare predittori diversi. Nel secondo modello, abbiamo eseguito una regressione multipla considerando sempre la variabile *“Total_RE_Gwh”* insieme alle variabili *“Electricity Consumption”*, *“Population”* e *“Fossil CO2 Emission”,* ottenendo dei risultati simili al modello precedente, e pertanto non utilizzabili.

In seguito sono state performate tre Regressioni Lineari semplici, utilizzando rispettivamente le variabili *“Population”, “Fossil CO2 Emission”* e *“Electricity Consumption”*, ottenendo dei risultati sorprendenti in quanto queste tre variabili, pur non essendo significative nei modelli precedenti, influenzati dal Totale della RE prodotta, contribuiscono a spiegare il 70% circa del fenomeno, come asseriscono gli **R2** dei rispettivi modelli pari a 0.60, 0.70 e 0.72 e livelli di **MSE training.**

A questo punto, abbiamo scelto come ultimo modello una regressione multipla in cui consideriamo i tre predittori ottenendo dei risultati simili ai precedenti: con un R2 pari a 0.7469 le tre variabili spiegano il 75% circa della variabilità associata al fenomeno, e la considerazione più importante riguarda il fatto che escludendo da questa regressione, così come dalle regressioni precedenti la variabile *“Tot_RE_Gwh”* i risultati siano più significativi da un punto di vista statistico, con un **MSE training** pari a 9.240.

```
lm5=lm(ds1$`Investimenti Green`~ ds1$Population + ds1$`Electricity Consumption` + ds1$`Fossil CO2 Emissions`, data = ds1, na.action = na.omit)
summary(lm5)
str(lm5)
preds5 = predict(lm5, ds1[-c(1,8)])
mean( (preds5 - ds1$`Investimenti Green`)^2 ) ## MSE =  9.240915
```

```
> summary(lm5)

Call:
lm(formula = ds1$`Investimenti Green` ~ ds1$Population + ds1$`Electricity Consumption` +
    ds1$`Fossil CO2 Emissions`, data = ds1, na.action = na.omit)

Residuals:
    Min      1Q  Median      3Q     Max
-4.6160 -1.1593 -0.8848  0.5651 14.2901

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  1.147e+00  6.874e-01   1.668   0.1047
ds1$Population               5.047e-10  2.686e-08   0.019   0.9851
ds1$`Electricity Consumption`  2.314e-05  1.021e-05   2.265   0.0302 *
ds1$`Fossil CO2 Emissions`    -1.186e-02  8.658e-03  -1.370   0.1800
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.219 on 33 degrees of freedom
Multiple R-squared:  0.7469,    Adjusted R-squared:  0.7239
F-statistic: 32.47 on 3 and 33 DF,  p-value: 5.825e-10
```

I risultati ottenuti che apparentemente sembrano buoni sono tuttavia calcolati sul training set, e per verificarne effettivamente la validità, calcoliamo il **Cross-Validation Error**, servendosi della **K-fold CV**:

```r
set.seed(123)
k = 10
folds = sample(1:k, nrow(ds1[-c(1, 8)]), replace=TRUE)
tmp = cbind(ds1, folds)
cv.errors = vector()
for(j in 1:k) {
  cv.fit = lm(`Investimenti Green`~ Population + `Electricity Consumption` + `Fossil CO2 Emissions`,
                data = ds1[folds != j, ]) # tutti i dati tranne il chunk j
  pred = predict(cv.fit, ds1[folds==j,]) # predizione proprio su j che era stato lasciato fuori
  cv.errors[j] = mean( (ds1$`Investimenti Green`[folds==j] - pred)^2 ) # calcolo e registro l'errore
}

mean.cv.errors = mean(cv.errors)
mean.cv.errors ## MSE = 50.84339
```

dallo screenshot possiamo notare la procedura utilizzata, che si serve di un ciclo **for** per iterare nel range che va da 1 a 10, il nostro parametro **"K"**, ed eseguire la regressione lineare non più su tutte le osservazioni del training set, come fatto prima, ma solo sui K-1 blocchi di osservazioni, escludendone uno che svolgerà il ruolo di test set o *"validation set"*. Una volta effettuato il modello lineare, il ciclo procede a calcolare le predizioni sul blocco scelto come validation set e infine calcola l'errore e lo inserisce in un vettore inizializzato in precedenza.

Il ciclo va quindi a ripetere la procedura andando a scegliere ogni volta uno diverso dei 10 blocchi come validation set aggiornando ogni volta il vettore *"cv.errors"* che conterrà 10 validation errors differenti, sui quali si andrà a calcolare la media, ricavando il **Cross-Validation Error** che nel nostro caso è risultato pari a 50.843.

Come si può notare la stima ottenuta per il test error, pari a 50.843, è maggiore rispetto a quella calcolata sul training set, pari a 9.240, la quale pertanto rappresenterà una sottostima del **test MSE** dovuta principalmente al fatto che nel dataset non disponiamo di un numero elevato di osservazioni rispetto ai predittori considerati, pertanto i risultati ottenuti sul training set, oltre a sottostimare in maniera significativa il vero test set, saranno anche caratterizzati da un certo grado di variabilità; in alternativa alla **Cross Validation**, vista la situazione, si sarebbero potuti applicare dei metodi di **Subset Selection** o dei metodi di **Shrinkage** per osservare quali predittori servissero maggiormente al fine di ottenere un risultato significativo.

2.3 Classificazione

Come abbiamo sottolineato l'obiettivo della nostra analisi è quello di cercare un classificatore che sia in grado di capire sulla base dei dati raccolti se in un paese è conveniente investire nel settore green/renewable oppure no.

2.3.1 Regressione Logistica

Il primo metodo che utilizziamo è la **Regressione Logistica**, cambiando la codifica della variabile di risposta *"Risultato Investimento"* in "0" e "1". Procediamo considerando principalmente 2 modelli, in quanto il terzo era per osservare un eventuale legame tra le variabili utilizzate ma con scarsi risultati.

Come per la Regressione Lineare, anche qui iniziamo con il modello con quasi tutti i predittori, per evidenziare l'eventuale presenza di variabili con maggiore significatività; tuttavia i risultati ottenuti sono anomali, in quanto dal summary dei modelli possiamo osservare come tutte le variabili considerate nei diversi modelli presentino un **"p-value"** pari a 1 o comunque prossimo a 1, indicandone la non significatività, e valori degli **Standard Error** abbastanza elevati, asserendo che le stime ottenute siano abbastanza distorte. Nonostante ciò, dalla confusion matrix e dal calcolo dell'error rate, possiamo vedere come i modelli classifichino le osservazioni perfettamente, non commettendo errori: questo perchè, anche in questo caso, tutti i modelli, a prescindere dal numero di predittori considerati, andranno in overfitting e i conseguenti risultati, seppur all'apparenza buoni, saranno in realtà caratterizzati da una performance predittiva molto scarsa.

```
> table(glm.pred, `Risultato Investimento`)
          Risultato Investimento
glm.pred   Negative Positive
  Negative       25        0
  Positive        0       12
```

Consideriamo pertanto un approccio alternativo, ovvero utilizzando la **K-Fold CV**:

```
library(caret)
## Define training control

train_control <- trainControl(method = "cv", number = 10, savePredictions = TRUE)

## Train the model on training set
model <- train(`Risultato Investimento` ~ .,
               data = ds1[,-c(1,8)],
               trControl = train_control,
               method = "glm",
               family=binomial())

model ## Accuracy=0.7833333
1-0.7833333 ## Cross-Validation Error = 0.2166667
```

come possiamo osservare dal codice, con la libreria **caret** possiamo applicare direttamente la Regressione Logistica alla lista *"train control"* invece di rifare il procedimento utilizzato per la Regressione Lineare che si serviva di un ciclo for che ripetesse la procedura e salvasse i risultati in un vettore continuamente aggiornato; **caret** permette di eseguirlo un'unica volta, specificando *"glm"* come metodo da utilizzare.

Dallo screenshot vediamo come i risultati associati alla procedura siano migliori rispetto quelli ottenuti in precedenza con **Accuracy** e **Error Rate** pari, rispettivamente, a 0.783 e 0.217.

L'unico problema associato a questa procedura è che non siamo in grado di estrarre un vettore con le predizioni o una confusion matrix da utilizzare per calcolare **Precision**, **Recall** e **F1 score**, i quali pertanto sono stati ricavati usando le predizioni ottenute sui dati training, influenzando purtroppo i tre indici, che sono risultati pari a 1.

2.3.2 K-Nearest Neighbour

Il secondo metodo che utilizziamo è la **KNN**. Costruiamo 3 modelli cambiando il valore di **K** rispettivamente in 1,5,15, ovviamente all'aumentare del parametro **K** il nostro modello ridurrà la **Variance** e viceversa aumenterà il **Bias**. Infatti nel primo modello il tasso di errata classificazione è pari a 0, con K=1, poiché i gruppi di classificazione sono composti ciascuno da 1 osservazione, pertanto al variare del training set il modello subirà sicuramente delle variazioni importanti. Nei due modelli successivi il tasso di errata classificazione passa da 0.3076923 per K=5 sul test set fino ad arrivare a 0.3513514 con K=15 sempre sul test set. Pertanto

aumentando **K** otteniamo risultati più robusti da un punto di vista statistico in termini di prediction accuracy/riduzione della Variance, ad un costo trascurabile in termini di leggero aumento del Bias.

Ora utilizziamo sempre la **KNN** però sfruttiamo la cross validation per capire qual è il valore è di K al quale è associato il modello con accuracy più alta.

```
library(caret)
trControl <- trainControl(method  = "cv", number  = 5, savePredictions = TRUE)
fit <- train(`Risultato Investimento` ~ .,
             method     = "knn",
             tuneGrid   = expand.grid(k = 1:15),
             trControl  = trControl,
             metric     = "Accuracy",
             data       = ds1[,-c(1,8,19)])

fit ## 0.6750000
## Dal fit possiamo vedere come eseguendo la knn con K=12 il nostro classificatore
## presenta la prediction accuracy più elevata.
```

Il modello con K=12 risulta essere il più efficiente.

A questo punto calcoliamo il modello sul test set con K=12 ed otteniamo un error rate del 0.153 ed un accuracy del 0.846 che risulta essere un ottimo risultato in quanto su 10 osservazioni il nostro modello ne classifica correttamente 8.

```
p.YTrain2 = knn(XTrain, XTrain, YTrain, k=15)
table(p.YTrain2, ds1$`Risultato Investimento`)
mean(p.YTrain2!=ds1$`Risultato Investimento`) ## ErrorRate = 0.3513514
mean(p.YTrain2==ds1$`Risultato Investimento`) ## Accuracy 0.6486486
p.YTest2 = knn(XTrain, XTest, YTrain, k=15)
mean(p.YTest2!=ds1[test,]$`Risultato Investimento`) #[1] 0.4615385
mean(p.YTest2==ds1[test,]$`Risultato Investimento`) #[1] 0.5384615

p.YTrain3 = knn(XTrain, XTrain, YTrain, k=12)
table(p.YTrain3, ds1$`Risultato Investimento`)
mean(p.YTrain3!=ds1$`Risultato Investimento`) ## ErrorRate = 0.3513514
mean(p.YTrain3==ds1$`Risultato Investimento`) ## Accuracy 0.6486486
p.YTest3 = knn(XTrain, XTest, YTrain, k=12)
mean(p.YTest3!=ds1[test,]$`Risultato Investimento`) #[1] 0.1538462
mean(p.YTest3==ds1[test,]$`Risultato Investimento`) #[1] 0.8461538
```

Chiudiamo la nostra analisi KNN calcolando **Precision**, **Recall** e **F1 score**:
Precision = 0.4545455, Recall = 0.4166667, F1 score = 0.4347826

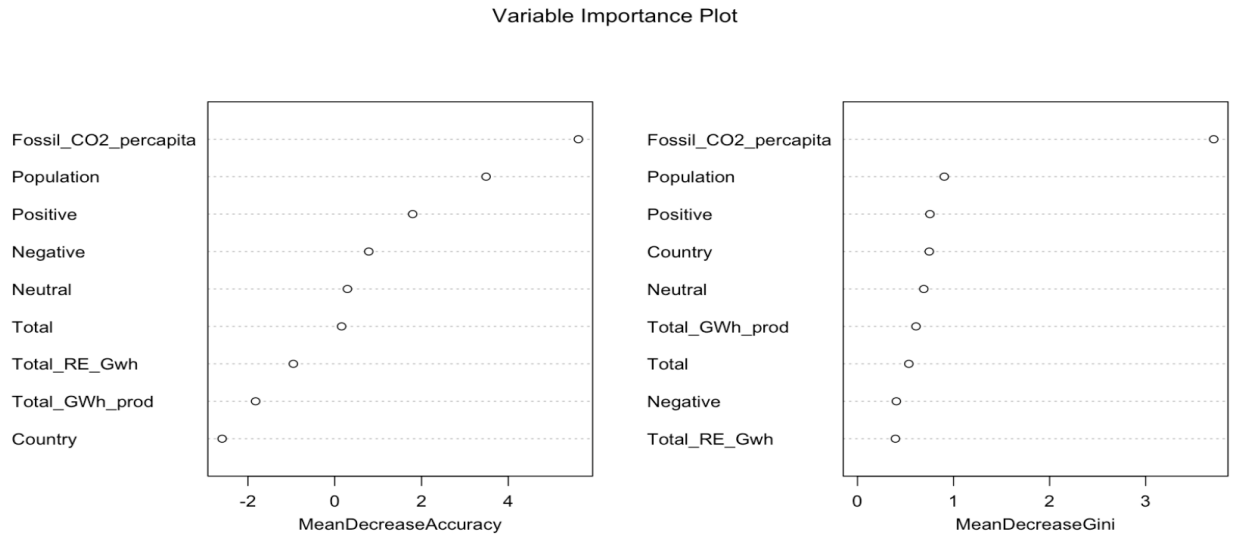
```
precision <- posPredValue(predictions, y, positive="Positive") # 0.4545455
precision
recall <- sensitivity(predictions, y, positive="Positive") # 0.4166667
recall

F1 <- (2 * precision * recall) / (precision + recall) # 0.4347826
```

I risultati non sono estremamente positivi in quanto abbiamo dei tassi intorno allo 0.5 di **Precision** e soprattutto del **Recall** score. La **Precision** di 0.45 ci dice che su 100 casi in cui il nostro modello classifica un paese come 'Positive' per investire, in realtà solo in 45 circa di questi 100 casi risulta essere vero. La **Recall** invece ci dice che su 100 casi che veramente sono Positive il nostro modello riesce a classificarne solamente 41 circa. **L'F1 score** è un indice che riassume i due parametri precedenti.

2.3.3 Random Forest

Il terzo metodo che abbiamo usato per la classificazione è **Random Forest**, per prima cosa trasformiamo la variabile "*Risultato Investimento*" in una variabile booleana 0,1 rispettivamente "Negative" e "Positive", dividiamo infine il nostro dataset in Training e Test set.

A questo punto applichiamo **RF** diverse volte cambiando il parametro *"mtry"* (il quale serve a regolare il numero di predittori che casualmente saranno considerati nel processo) per osservare un'eventuale variazione dei risultati, le due variabili più importanti sono *"Fossil Co2 per capita"* e *"Population"* che hanno valori più alti di Variable importance in tutti i modelli RF creati.



Calcoliamo poi l'accuracy del modello che risulta essere per tutti e 3 i modelli 0.6153846, un risultato non altissimo che indica una previsione corretta del nostro modello nel 65% dei casi circa. Calcoliamo anche **Precision**, **Recall** ed **F1 score** che sono rispettivamente 1, 0.1666667 e 0.2857143.

2.3.4 Support Vector Machine

Il terzo metodo di classificazione che utilizziamo è il **Support Vector Machine**, dividiamo anche qui il dataset in Training e Test, creiamo due modelli cambiando il **Budget** ovvero **il parametro C** che ci permette di regolare il soft margin, al primo modello assegnamo C = 10 al secondo C = 1.

```
mean(pred_test1==ds_svm_test$`Risultato Investimento`) #accuracy 0.8461538
precision_svm <- posPredValue(predictions_svm, y_svm, positive="1") # 0.6666667
recall_svm <- sensitivity(predictions_svm, y_svm, positive="1") # 0.6666667
F1_svm <- (2 * precision_svm * recall_svm) / (precision_svm + recall_svm) #0.6666667
```

```
svmfit = svm(`Risultato Investimento`~., data=ds_svm_train, kernel="linear", cost=10,
             scale=TRUE) #svm, kernel linear usa un svm lineare, cost punti da usare per i margini, scale
names(svmfit)
svmfit$index
summary(svmfit)
```

Il primo modello presenta un'accuracy leggermente migliore del secondo modello 0.8461538 contro 0.8181818 mentre **Precision**, **Recall** ed **F1 score** sono uguali per entrambi e sono discretamente essendo sopra i 0.5.

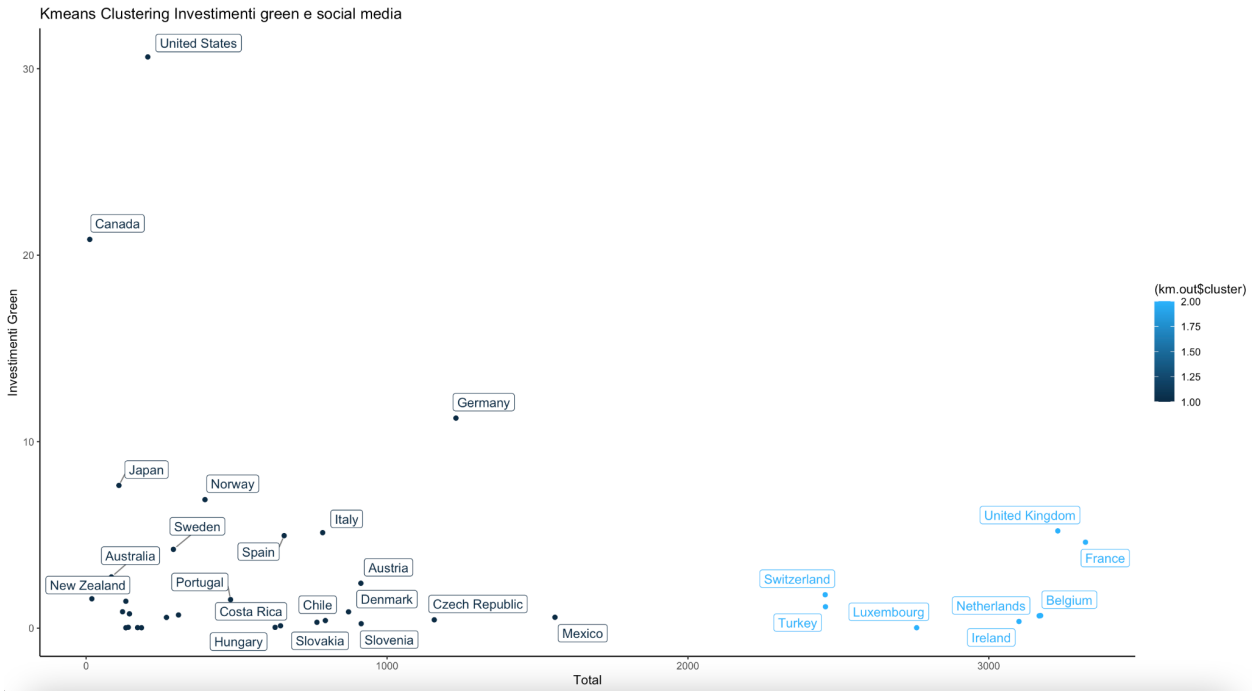
```
y_svm <- ds_svm_test$`Risultato Investimento`
predictions_svm <- pred_test2

precision_svm <- posPredValue(predictions_svm, y_svm, positive="1") # 0.6666667
recall_svm <- sensitivity(predictions_svm, y_svm, positive="1") # 0.6666667

F1_svm <- (2 * precision_svm * recall_svm) / (precision_svm + recall_svm) # 0.6666667
```

Infine per migliorare i risultati applichiamo la **Leave-One-Out Cross Validation** con cui otteniamo un modello che sembra perforare molto bene con un tasso di errata classificazione dello 0.1052632.

2.4 Cluster analysis

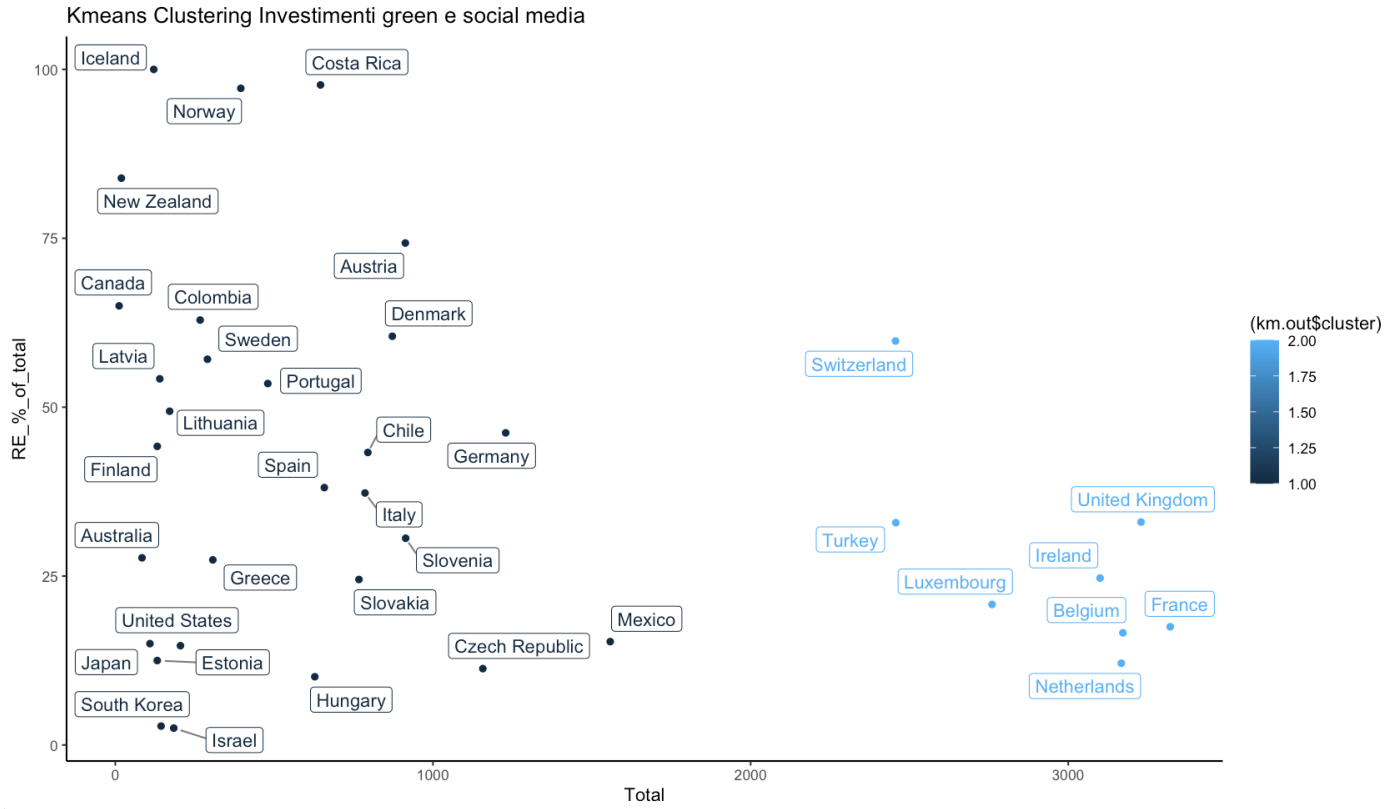
2.4.1 K-Means Clustering

Con il **K-Means Clustering** cerchiamo di trovare dei gruppi di osservazioni che per qualche ragione sono accomunate da caratteristiche simili. Il primo tentativo che facciamo è con le variabili *"Total"* (n. di tweets trovati per le query utilizzate) e *"Investimenti Green"*, scegliendo come parametro K=2, il quale deve necessariamente essere stabilito prima della procedura e che indica il numero di Cluster che vogliamo ottenere: vengono evidenziati due gruppi, uno evidenziato di celeste che concentra paesi molto vicini nel nord Europa (a parte la Turchia) e l'altro che invece racchiude tutti gli altri paesi; la principale discriminante tra i due gruppi è la variabile Total, vediamo come i social media abbiano un impatto superiore sul settore green principalmente nel contesto europeo.



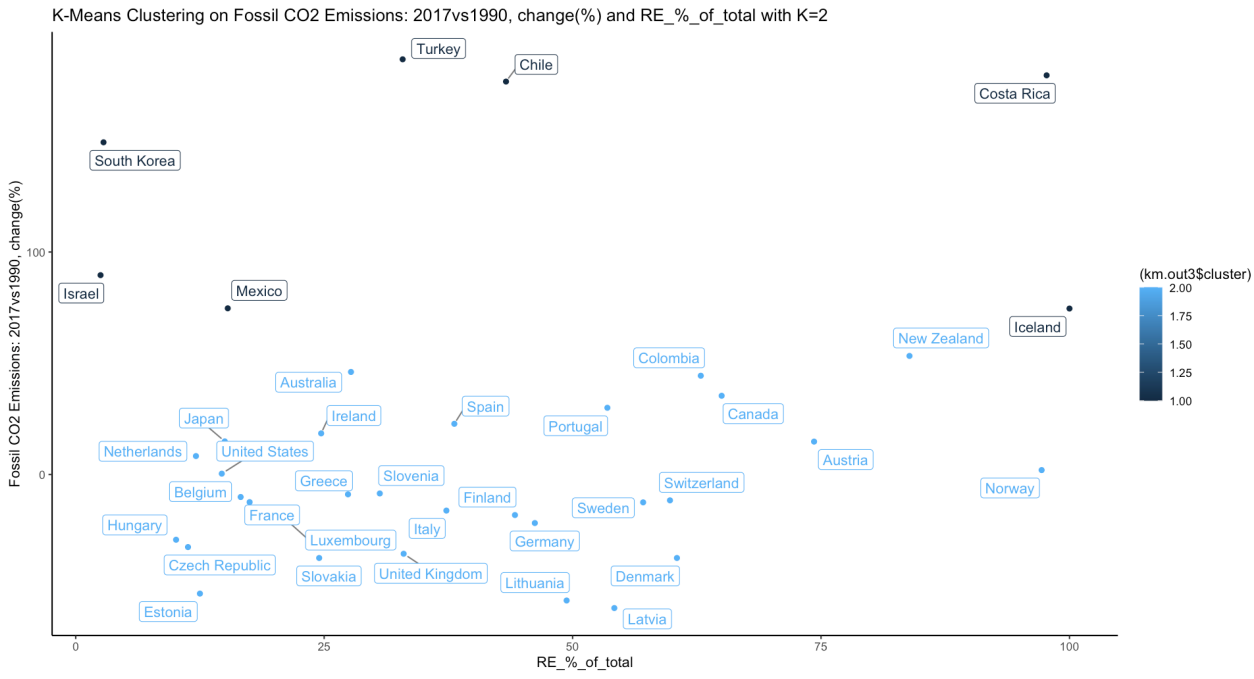
Nella prima clusterizzazione, la variabile sull'asse delle verticale era "*Investimenti Green*" per valore nominale, ripetiamo l'analisi cambiando la variabile in *"RE_%_of_total"* in questo modo abbiamo un'idea più chiara del peso degli investimenti in green sul totale dei vari paesi.
I cluster che emergono sono comunque gli stessi della prima analisi, l'unica differenza è che il primo cluster (colore scuro) presenta una nube di punti molto meno concentrata di prima.

Possiamo notare come tra i paesi che non presentano un elevato numero di tweet, Islanda, Costa Rica, Norvegia e Nuova Zelanda siano caratterizzati da una percentuale di produzione di Renewable Energy molto maggiore rispetto a paesi più grandi e maggiormente industrializzati, con un maggior grado di sviluppo tecnologico tipo Stati Uniti, Germania, Giappone ecc. Una possibile motivazione di questo trend si potrebbe ricercare nella locazione geografica di quei paesi, situati in zone del pianeta favorevoli all'utilizzo di fonti di produzione di energia green come impianti fotovoltaici, impianti eolici e così via.



In questo terzo K-means Clustering analizziamo le variabili *"RE_%_of_total"* e *"Fossil_Co2_emission"* (variazione), i cluster che emergono sono abbastanza sbilanciati, nel primo (colore chiaro) rientrano quasi tutti i paesi, mentre nel secondo (colore scuro) molti meno; i due cluster sono separati principalmente a causa della variabile relativa alle emissioni di Co2.

Tra le osservazioni che rientrano nel secondo cluster, possiamo notare come Costa Rica e Islanda siano i paesi che oltre a presentare la percentuale di RE prodotta maggiore siano anche quelli con il risultato migliore in termini di variazione percentuale delle CO2 Emissions, confermando quanto asserito su questi due paesi nel cluster precedente, mentre Norvegia e Nuova Zelanda pur presentando ottimi valori in termini di produzione di energia rinnovabile, pecchino per quanto riguarda la variazione della produzione di emissioni nel tempo, implicando che per loro lo "switch" al settore della Renewable Energy sia avvenuto magari in tempi o modalità differenti.



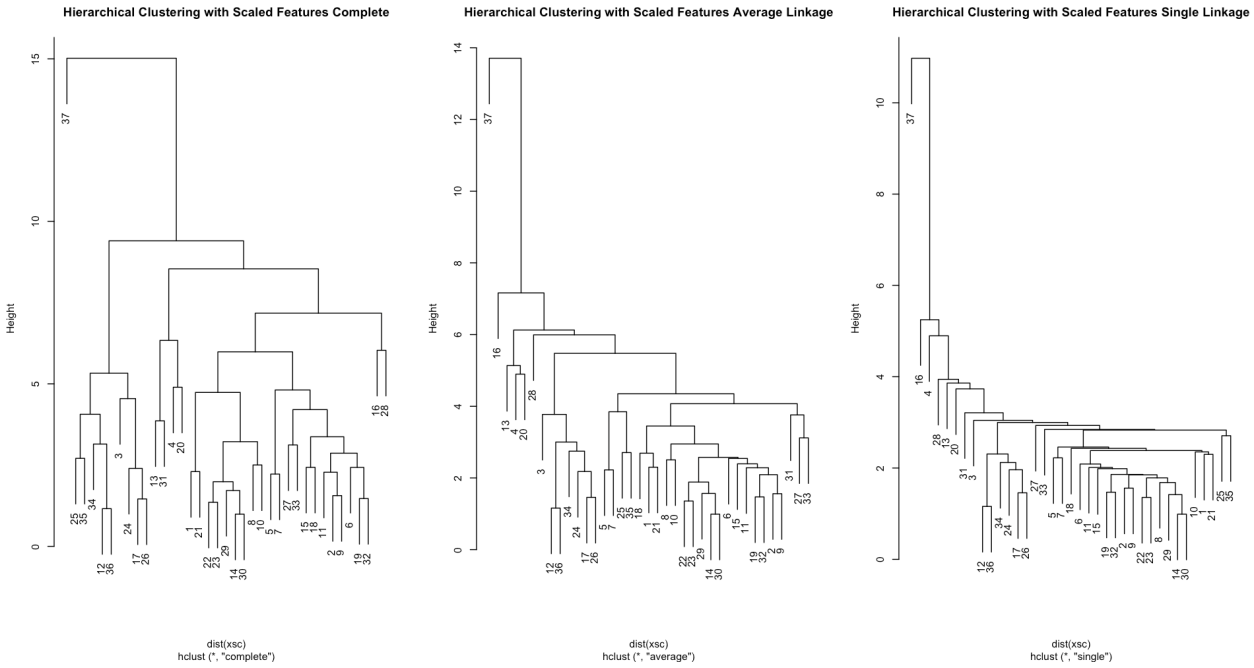
2.4.2 Hierarchical Clustering

Continuiamo la nostra analisi con il **Clustering Gerarchico** per osservare se la clusterizzazione ottenuta sia migliore rispetto quella effettuata con il metodo precedente oppure se i risultati ottenuti siano simili; rappresentati nei tre quadranti della figura vediamo le tre clusterizzazioni effettuate con i tre tipi di **"linkage"** considerati per la fusione dei gruppi di osservazioni rispettivamente **"complete"**, **"average"** e **"single"**. Prima di passare all'analisi dei risultati, è bene precisare che in questo caso le osservazioni sono state oggetto di standardizzazione, effettuata con il comando **"scale()"** di **R.**

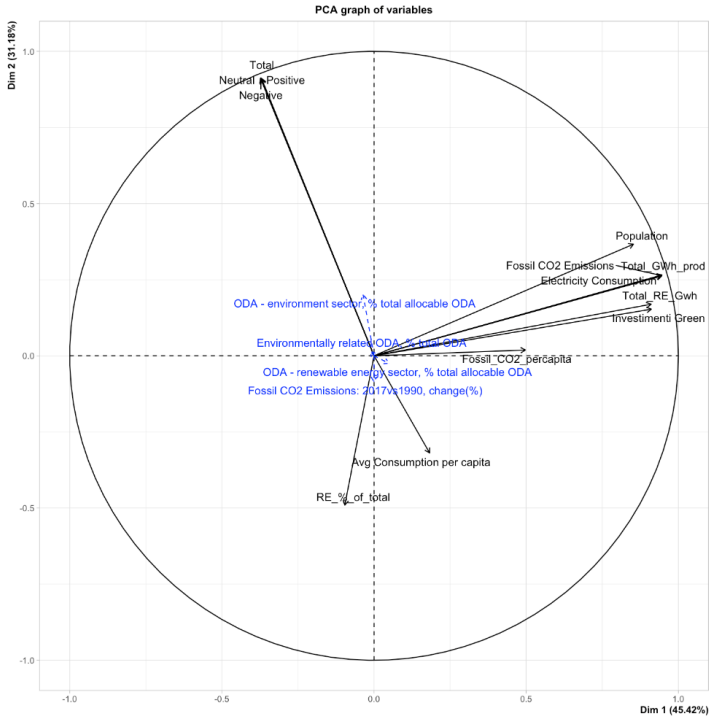
I risultati migliori da un punto di vista interpretativo li possiamo osservare nel clustering che si serve del **"complete linkage":** possiamo notare come molte osservazioni e i successivi gruppi di osservazioni siano abbastanza diversi tra loro, in quanto la fusione si verifica nel mezzo del dendrogramma. Ricordando che nell'asse verticale è rappresentata la distanza delle osservazioni, questo significa che in realtà, solo poche osservazioni sono effettivamente simili tra loro e sono quelle la cui fusione avviene nel bottom della struttura, tutte le altre invece, come abbiamo scritto poco fa, saranno raggruppate ad un'altezza superiore, fino ad arrivare al top del dendrogramma, in cui si verifica la fusione con l'osservazione 37 che è la più diversa di tutte.

Pertanto anche se dal grafico possiamo individuare tre o quattro cluster principali, considerando gli USA il numero di cluster che si ottengono dalla divisione del dendrogramma che possiamo avere è 2: il primo cluster contiene solamente l'osservazione 37, che come abbiamo detto ha caratteristiche molto diverse da tutti gli altri paesi, mentre nel secondo troviamo tutte le altre osservazioni.

Escludendo ipoteticamente dall'analisi l'osservazione incriminata, il numero ottimale di cluster sarebbe 3, ricavati tagliando il dendrogramma ad un'altezza pari a 8 circa: in un primo cluster troveremo le osservazioni 25, 35, 34, 3, 12, 36, 24, 17 e 26; nel secondo le osservazioni 13, 31, 4 e 20 e infine nel terzo tutte le altre.



2.4.3 Principal Component Analysis



Proseguiamo la nostra analisi non supervisionata con la **Principal Component Analysis**, dove andiamo a cercare i maggiori responsabili della variabilità dei dati.

Alla sinistra vediamo il grafico delle variabili dove quelle evidenziate in nero sono variabili attive (hanno concorso a formare le PC) mentre quelle in blu sono varaibili supplementari (non hanno concorso a formare le PC).

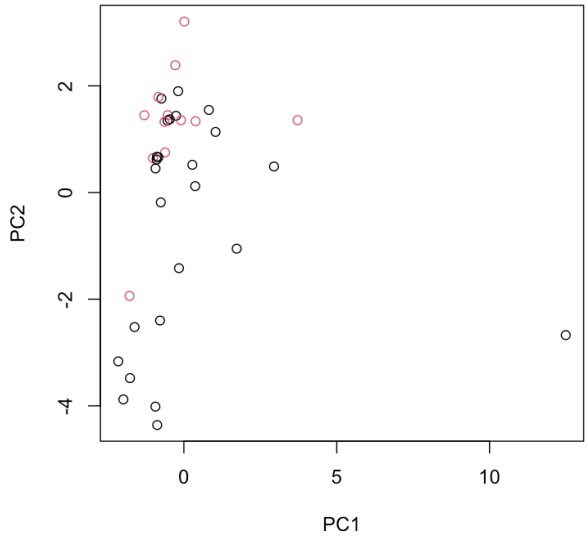
Notiamo una fortissima relazione tra le variabili *"Population"*, "*Fossil CO2*

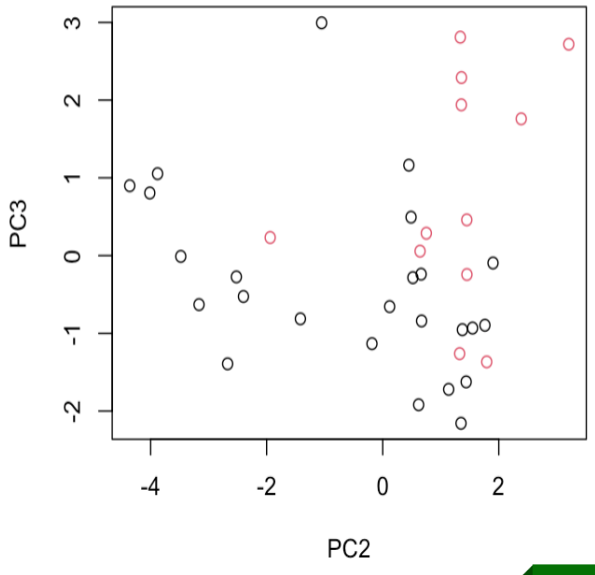
Emissions”, *“Electricity consumption”*, *“Total_RE_Gwh”*, *“Investimenti Green”* che risultano direttamente collegate tra loro ma anche fortemente relazionate con la prima componente principale. Invece le variabili relative ai dati social media sono tutte molto correlate tra loro ed insieme correlate con la PC n2, invece le variabili supplementari evidenziate in blu non sembrano avere correlazione con le PC. Le prime due componenti principali spiegano insieme una % di inerzia molto elevata pari al 76.6%.

```
#Eigenvalues
#                      Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7   Dim.8   Dim.9
#Variance               5.905   4.053   1.350   1.015   0.433   0.149   0.077   0.009   0.003
#% of var.            45.424  31.180  10.386   7.810   3.331   1.145   0.593   0.070   0.026
#Cumulative % of var.  45.424  76.603  86.990  94.799  98.130  99.275  99.868  99.938  99.963
```

Utilizziamo poi la funzione **“prcomp”** per svolgere una seconda PCA, i risultati sono molto simili alla prima confermando le relazioni trovate in precedenza.
Rappresentiamo poi lo spazio delle unità per le prime 4 componenti principali e coloriamo le osservazioni a seconda della loro convenienza o meno riguardo all’investimento in green.

Non osserviamo un pattern particolare se non che la maggior parte delle osservazioni sono correlate con la PC2, dunque la PC1 è quasi totalmente definita da un unico punto all’estremo che è rappresentato dagli USA, nelle PC 3 e 4 invece tutti i punti sono distribuiti in maniera più omogenea sicuramente a causa del fatto che la variabilità di USA è stata utilizzata nel primo piano fattoriale e quindi nei restanti rimane quella di tutti gli altri paesi.

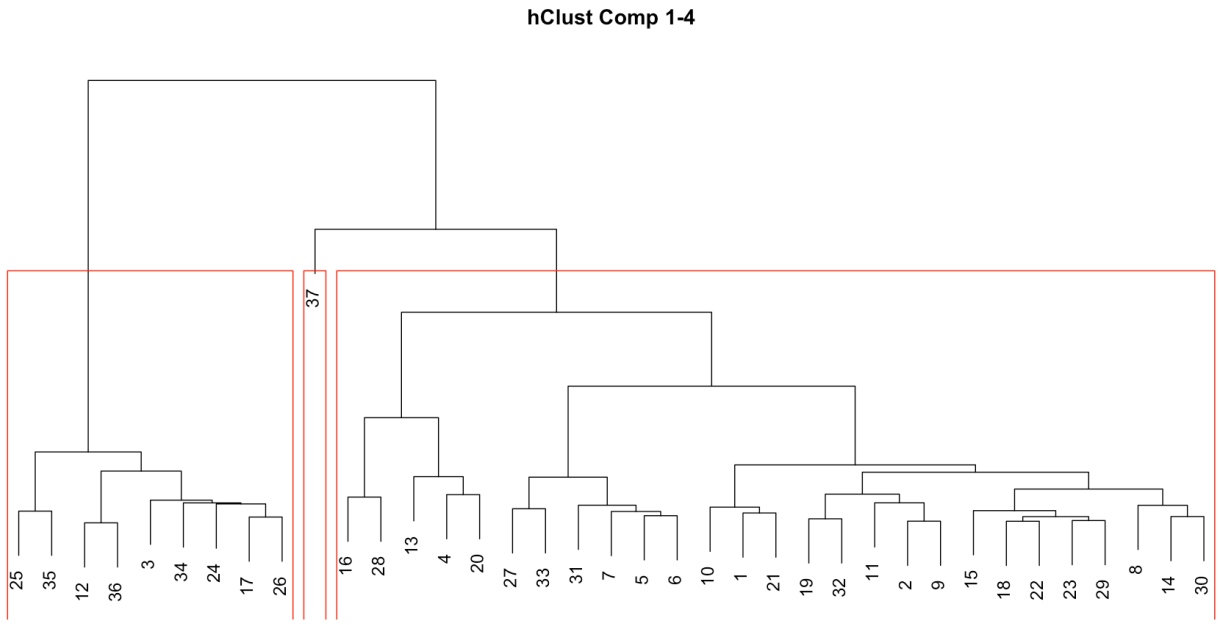




2.4.4 Hierarchical Clustering with Principal Component

Concludiamo la nostra analisi provando a ricercare dei cluster all'interno dei punti unità dei 4 assi fattoriali e suddividendo le osservazioni in 3 cluster, a differenza delle clusterizzazioni precedenti, in cui si prediligeva la divisione delle osservazioni in 2 cluster.

È necessario specificare che anche in questo caso, l'osservazione 37 (Stati Uniti) rappresenta un cluster a sé, e come possiamo osservare dalla figura, rappresenta anche un'osservazione molto diversa rispetto alle altre, in quanto la sua fusione con gli altri gruppi si verifica praticamente presso il top del dendrogramma; negli altri due cluster invece possiamo vedere come la maggior parte delle fusioni si verifichino verso il bottom del dendrogramma, sottolineando come le osservazioni ivi contenute siano abbastanza vicine tra loro, in termini di similarità.



Conclusioni

In conclusione, i risultati ottenuti sono abbastanza soddisfacenti anche se con qualche precisazione che deve essere fatta: per quanto riguarda la creazione del dataset, ci siamo resi conto innanzitutto che alcune variabili introdotte non sono risultate influenti per quanto concerne il nostro obiettivo, ad esempio le variabili legate agli ODA (Official Development Assistance), altre invece sono state basate su importanti assunzioni, come la variabile di risposta *"Risultato Investimento"* con la quale si valuta un paese conveniente se la quantità di RE prodotta è superiore ad una certa soglia, o la variabile *"Investimenti Green",* che rappresenta la quota effettiva di investimenti nel settore green di un paese, ricavata anch'essa manualmente in quanto non reperibile da altre fonti per i singoli paesi. Ovviamente non si tratta di assunzioni surreali però è bene specificarle in quanto parte dei risultati sicuramente dipenderà da esse.

Per quanto riguarda l'analisi statistica dei paesi e in particolare la classificazione, abbiamo notato come la mancanza di osservazioni abbia influenzato in buona misura i risultati ottenuti, portando spesso ad un overfit del classificatore nel training set, situazione che abbiamo cercato di risolvere utilizzando i metodi di resampling in ogni procedura adottata; nonostante ciò, i classificatori utilizzati hanno portato dei discreti risultati, dimostrando che le nostre assunzioni non fossero del tutto sbagliate, come si accennava poco fa.

Questo ci porta a pensare che i risultati ottenuti, seppur discreti, siano ovviamente migliorabili a seguito di considerazioni più approfondite e dettagliate che cerchino di concentrarsi di più sugli aspetti che effettivamente siano in grado di migliorare la precisione dei classificatori e della loro performance predittiva.

Quindi in conclusione la convenienza o meno di investire in un paese dipende principalmente da variabili come le emissioni di Co2, la popolazione ed il fabbisogno energetico di un paese, come riportato nei modelli trovati.