Tecniche di classificazione, clustering e regole associative con il tool Weka sul dataset relativo salary

By Elka Segura

Ghofran Bessioud

Anno Accademico 2023/2024

Sommario

INTRODUZIONE

Il salario è elemento importante per i lavoratori, consente loro di acquisire i beni e i servizi di cui hanno bisogno per il proprio benessere e quello delle proprie famiglie; per gli imprenditori rappresenta un costo di produzione. Al giorno d'oggi, a causa della diversificazione dei prodotti e dei servizi offerti dalle aziende, diventa più difficile identificare i segmenti di mercato dei clienti che consumano determinati e specifici prodotti e servizi a causa di questa diversificazione. Le aziende specializzate nel marketing, così come nel settore bancario, hanno bisogno di conoscere il potere d'acquisto dei clienti per adattare prodotti e servizi alle loro preferenze. Uno degli indicatori che influenza il potere d'acquisto è lo stipendio, ecco perché attraverso questo progetto cerchiamo di applicare algoritmi di data mining che ci permettano di classificare i clienti in base all'importo che guadagnano annualmente. La classifica è concentrata in due: meno di 50mila dollari e più di 50 dollari. Il set di dati selezionato si riferisce a uno studio tratto dal censimento dei redditi del 1994 Prevede se il reddito supera i $ 50.000 all'anno in base ai dati del censimento. Noto anche come set di dati per adulti.

L'applicazione di tecniche di data mining a questi tipi di ricerca consente di delineare modelli di comportamento nei dati. Il concetto di data mining, si riferisce alla combinazione di tecniche di elaborazione dei dati per identificare modelli, tendenze e possibili coincidenze per proporre strategie e prevedere alcuni comportamenti futuri.

Il data mining è essenziale per analizzare grandi set di dati e definire modelli comportamentali in determinate situazioni. La chiave per rispondere ai bisogni è l'anticipazione, poiché l'analisi consente di pianificare le misure più appropriate per migliorare varie questioni.

L'estrazione della conoscenza è principalmente legata al processo di scoperta noto come Knowledge Discovery in Databases (KDD), che si riferisce al processo non banale di scoperta e identificazione di conoscenza, modelli validi e informazioni potenzialmente utili all'interno dei dati contenuti in alcuni repository di database. Si tratta di convertire i dati in informazioni. Tra le varie tecniche di classificazione ci sono Classificazione, Clustering, Alberi decisionale, Associazione e altre. La scoperta dei *pattern* sui grandi volumi dei dati si può ottenere facendo uso in maniera automatica o semiautomatica con la applicazione del data mining.

SELEZIONE E PRE-PROCESSING

Il pre-processing è fortemente legato da quelli che sono gli obiettivi che si vogliono raggiungere. Il pre-processing dei dati è l'insieme di tecniche e pratiche applicate ai dati grezzi prima di utilizzarli in qualsiasi progetto di Data Mining. Comprende una serie di attività volte a pulire, trasformare e organizzare i dati in modo che gli algoritmi di Data Mining possano estrarre informazioni accurate e significative.

L'importanza della preelaborazione dei dati nei progetti di data mining è immensa e si estende all'intero ciclo di vita di un progetto di analisi dei dati. Garantisce la qualità dei dati: Dati di bassa qualità, contenenti errori, valori anomali o valori mancanti, possono portare a conclusioni errate o a modelli inadeguati. La preelaborazione risolve questi problemi e garantisce che i dati utilizzati siano accurati e affidabili. Si facilita l'analisi esplorativa, prima di immergersi in analisi più complesse, è essenziale comprendere i dati nella loro forma più elementare e anche consente la visualizzazione e l'analisi esplorativa

dei dati garantendo che i dati siano coerenti e in un formato gestibile. Ciò rende più facile identificare tendenze, modelli e relazioni preliminari.

Gli algoritmi di Data Mining richiedono dati di input in un formato specifico. Ciò significa che nella maggior parte dei casi i dati grezzi non sono adatti per l'uso diretto. La preelaborazione dei dati prepara i dati per l'utilizzo negli algoritmi di data mining eseguendo attività come la codifica di variabili categoriali, la normalizzazione dei dati e la selezione di funzionalità pertinenti. Aiuta a evitare l'adattamento eccessivo, in cui un modello si adatta eccessivamente ai dati di addestramento e non si generalizza bene ai nuovi dati. La preelaborazione dei dati, in particolare la selezione delle funzionalità, aiuta a ridurre la dimensionalità dei dati e a semplificare i modelli.

Nei progetti di Data Mining non è sufficiente sviluppare modelli precisi. È inoltre essenziale comprendere i fattori che influenzano i risultati. La preelaborazione dei dati garantisce che i dati siano trasparenti e comprensibili. Ciò semplifica l'interpretazione dei risultati e consente ai professionisti della scienza dei dati di spiegare perché un modello prende decisioni specifiche. Questo processo di pre-processing dei dati può essere un processo più impegnativo, ma a lungo termine consente di risparmiare tempo e risorse. Investendo tempo nella corretta preparazione dei dati, si evitano problemi e si sarà in grado di ripetere determinati lavori più avanti nel progetto. Ciò è particolarmente importante nei progetti su larga scala, in cui gli errori nei dati possono essere costosi e difficili da correggere.

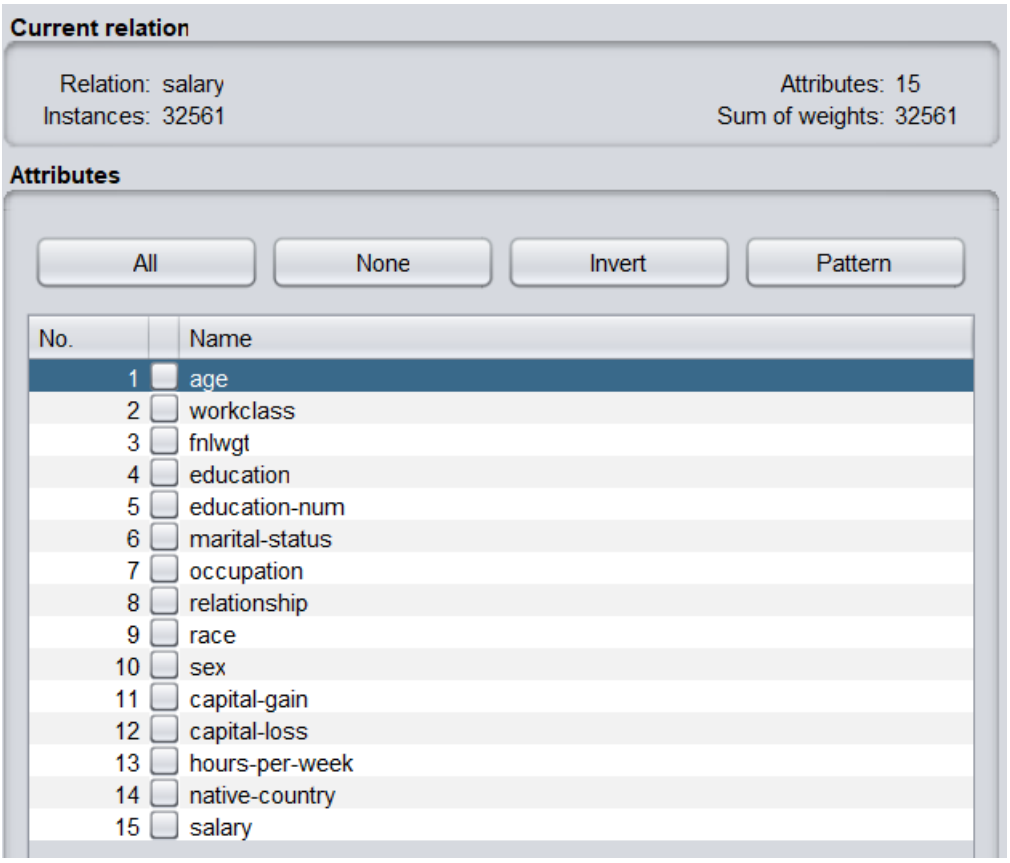
Uno degli obiettivi finali della scienza dei dati è la scoperta della conoscenza dai dati. Questa preelaborazione dei dati stabilisce una solida base per scoprire modelli, tendenze e relazioni nei dati. Senza un'adeguata preelaborazione è possibile perdere informazioni preziose.

La costruzione dei modelli ha due rami principali *supervisionato* e *non supervisionato*. Nell'apprendimento supervisionato, gli algoritmi lavorano con dati "etichettati", cercando di trovare una funzione che, date le variabili di input, assegni loro l'etichetta di output appropriata. L'algoritmo viene addestrato con dati "storici" e quindi "impara" ad assegnare l'etichetta di output appropriata a un nuovo valore, ovvero prevede il valore di output.

L'apprendimento non supervisionato si verifica quando i dati "etichettati" non sono disponibili per l'addestramento. Conosciamo solo i dati di input, ma non esistono dati di output che corrispondono a un determinato input. Pertanto, possiamo solo descrivere la struttura dei dati, per cercare di trovare un qualche tipo di organizzazione che semplifichi l'analisi. Hanno quindi carattere esplorativo.

In dataset selezionato per lo sviluppo di questo progetto si chiama *Cencus Income Dataset* dal sito Census Income - UCI Machine Learning Repository. L'obbietivo è applicare varie tecniche di data minig con l'idea di *scoprire quali attributi e modalità di attributi mi permette di prevedere se una persona guadagna più di 50.000 all'anno.*

Il dataset fornisce informazioni rilevanti riferite su persone, ad esempio, età, sesso, education, stato coniugale, tipo di lavoro, occupation, paese di nascita e altre. Contiene **32561** istanze con **15** attributi.



Informazioni sugli attributi

1) **age**: l'età di un individuo
2) **workclass**: la classe di lavoro a cui appartiene un individuo.
3) **fnlwgt**: il peso assegnato alla combinazione di caratteristiche (una stima di quante persone appartengono a questo insieme di combinazioni)
4) **education**: livello di istruzione più elevato
5) **education-num:** numero di anni per i quali è stata frequentata l'istruzione.
6) **marital-status**: matrimoniale di una persona.
7) **occupation:** professione della persona
8) **relationship:** relazione della persona nella sua familia
9) **race:** color o origine etnica.
10) **sex:** Female, Male.
11) **capital-gain:** capitale guadagnato da una persona
12) **capital-loss:** perdita di capitale per una persona
13) **hours-per-week**: numero di ore per le quali un individuo lavora a settimana
14) **native-country:** paese di nascita a cui appartiene una persona
15) **salary:** <=50K or >50K

Age

Selected attribute

Name: age Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-31]'	10572	10572.0
2	'(31-44]'	10904	10904.0
3	'(44-inf)'	11085	11085.0

Come vediamo l'attributo *age* è di tipo continuo con con un range compreso tra 17 e 90. In questo caso abbiamo proceduto a ridurre la numerosità, discretizzando la variabile in 3 range, per questo abbiamo applicato il filtro *Discretize.* (Nessun valore mancante):

Workclass

L'attributo *workclass* viene rappresentato nei dati di tipo nominal, con la definizione di 9 categorie: "Private", "Self-emp-not-inc", "Self-emp-inc", "Federal-gov", "Local-gov", "State-gov", "Without-pay", "Never-worked". Non presenta valori mancanti però esiste categoria segnata con il segno "?". Dato che l'attributo presenta 1836 istanze caratterizzate da questo segno, e non sarebbe opportuno eliminare tutte le istanze perché si perderebbe informazione, la modalità verrà ricodificata con "Other".

Selected attribute

Name: workclass Type: Nominal
Missing: 0 (0%) Distinct: 9 Unique: 0 (0%)

No.	Label	Count	Weight
1	State-gov	1298	1298.0
2	Self-emp-not-inc	2541	2541.0
3	Private	22696	22696.0
4	Federal-gov	960	960.0
5	Local-gov	2093	2093.0
6	?	1836	1836.0
7	Self-emp-inc	1116	1116.0
8	Without-pay	14	14.0
9	Never-worked	7	7.0

In questo caso per ridurre la numerosità delle classi procederemo a unire le categorie della seguente forma: "Gov_job", "Other", "Private" o "Self-employed" con il filtro *MergeManyValues.*

Selected attribute

Name: workclass Type: Nominal
Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

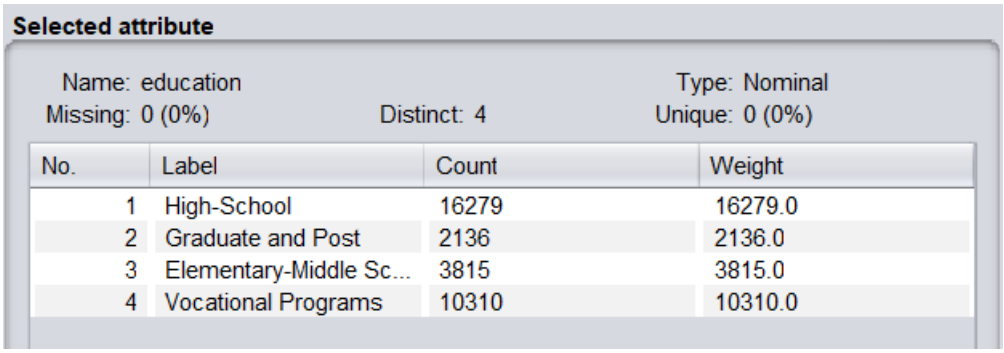
No.	Label	Count	Weight
1	Private	22696	22696.0
2	Other	1836	1836.0
3	Self-employed	3657	3657.0
4	Gov-job	4351	4351.0

Fnlwgt

L'attributo fnlwgt è il peso assegnato alla combinazione di caratteristiche (una stima di quante persone appartengono a questo insieme di combinazioni), presenta una tipologia numerica con un range compreso tra 12285 e 1484705. In questo caso, questo attributo verrà eliminato perché non è un buon predittore.

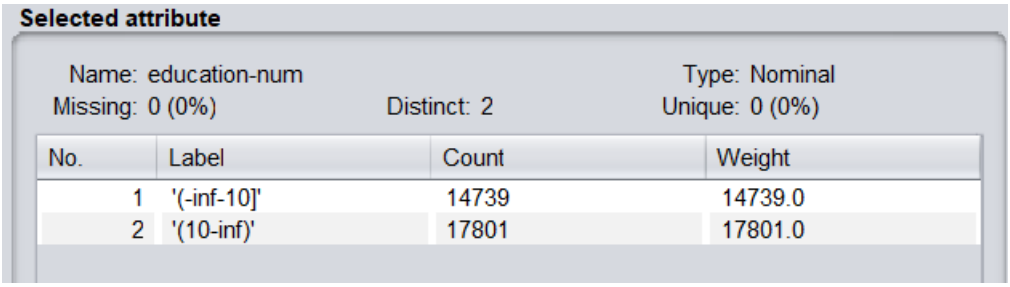
Education

L'attributo *education* viene rappresentato nei dati di tipo nominal, con la definizione di 16 categorie: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. Non presenta valori mancanti. Anche in questo caso per ridurre la numerosità delle classi procederemo a unire le categorie della seguente forma: "High-School", "Graduate and Post", "Elementary-Middle School" o "Vocational Programs" con il filtro *MergeManyValues.*



Education-num

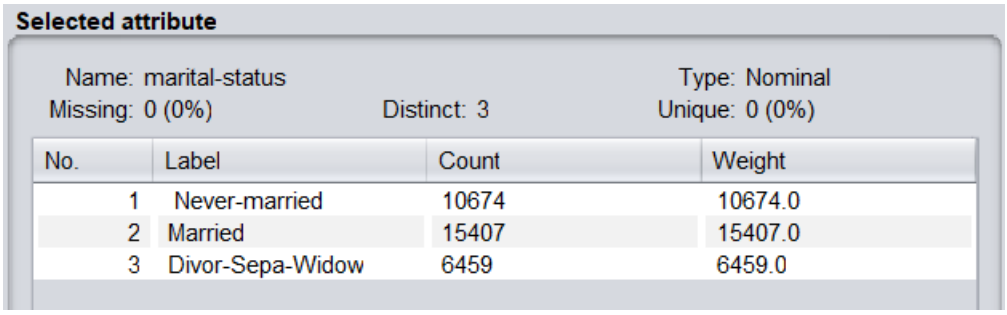
L'attributo *education* numero di anni per i quali è stata frequentata l'istruzione con minimo di 1 e un massimo 16. Per questo attributo viene applicato il flitro *Discretize* in 2 range. Non presenta valori mancanti.



Marital-status

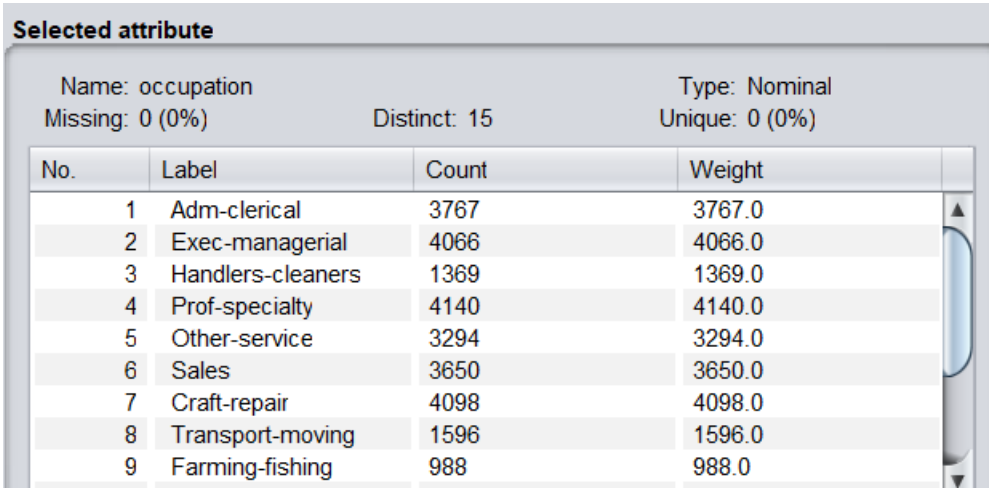
L'attributo *marital-status* viene rappresentato nei dati di tipo nominal, con la definizione di 7 categorie: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. Non presenta valori mancanti e anche in questo caso per ridurre la numerosità delle classi procederemo a unire le categorie della

seguente forma: "Married", "Never-Married" e "Divor-Sepa-Widow" Programs" con il filtro *MergeManyValues.*



Occupation

L'attributo *occupation* viene rappresentato nei dati di tipo nominal, con la definizione di 15 categorie: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. In particolare, a questa variabile non viene applicato alcun filtro e non presenta valori mancanti.



Relationship

L'attributo *relationship* viene rappresentato nei dati di tipo nominal, con la definizione di 6 categorie: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. In particolare, a questa variabile non viene applicato nessun filtro e non presenta valori mancanti.

Selected attribute

Name: relationship Type: Nominal
Missing: 0 (0%) Distinct: 6 Unique: 0 (0%)

No.	Label	Count	Weight
1	Not-in-family	8304	8304.0
2	Husband	13189	13189.0
3	Wife	1564	1564.0
4	Own-child	5058	5058.0
5	Unmarried	3444	3444.0
6	Other-relative	981	981.0

Race

L'attributo *race* viene rappresentato nei dati di tipo nominal, con la definizione di 5 categorie: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other e Black. Non presenta valori mancanti e viene applicato il filtro *MergeManyValues* per unire Asian-Pac-Islander, Amer-Indian-Eskimo, Other.

Selected attribute

Name: race Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count	Weight
1	White	27799	27799.0
2	Black	3121	3121.0
3	Other	1620	1620.0

Sex

L'attributo *sex* viene rappresentato nei dati di tipo nominal, con la definizione di 2 categorie: "Female" o "Male". In particolare, a questa variabile non viene applicato alcun filtro e non presenta valori mancanti.

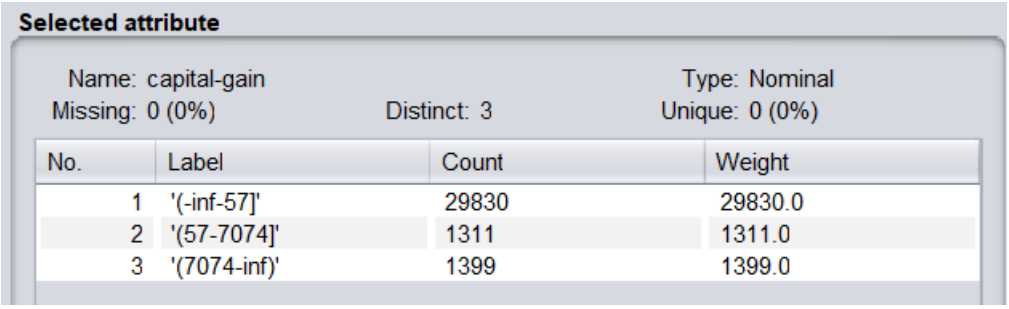
Selected attribute

Name: sex Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	Male	21776	21776.0
2	Female	10764	10764.0

Capital-gain

L'attributo *capital-gain* è capitale guadagnato da una persona. Per questo attributo viene applicato il flitro *Discretize* in 3 range. Non presenta valori mancanti.



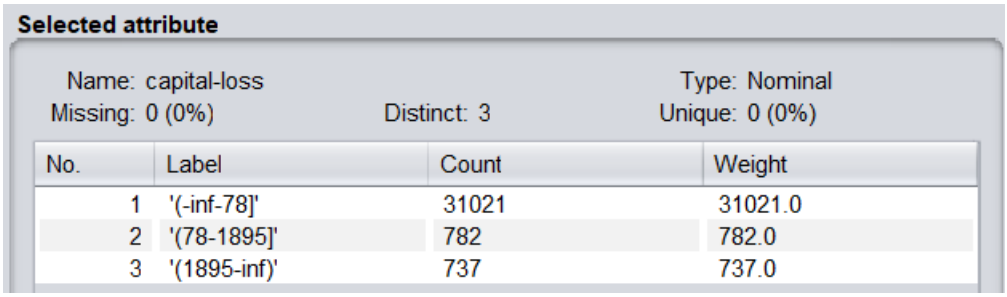
Selected attribute

Name: capital-gain — Type: Nominal
Missing: 0 (0%) — Distinct: 3 — Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-57]'	29830	29830.0
2	'(57-7074]'	1311	1311.0
3	'(7074-inf)'	1399	1399.0

Capital-loss

L'attributo *capital-loss* è il capitale perso da una persona con minimo di 0 e un massimo 4356. Per questo attributo viene applicato il flitro *Discretize* in 3 range. Non presenta valori mancanti.



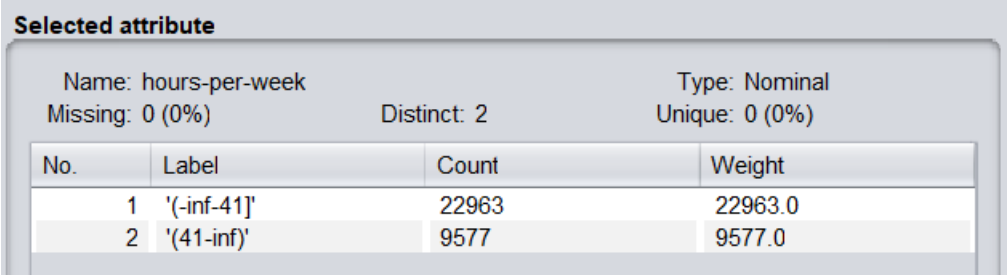
Selected attribute

Name: capital-loss — Type: Nominal
Missing: 0 (0%) — Distinct: 3 — Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-78]'	31021	31021.0
2	'(78-1895]'	782	782.0
3	'(1895-inf)'	737	737.0

Hours-per-week

L'attributo *hours-per-week* è numero di ore per le quali un individuo lavora a settimana con minimo di 1 e un massimo 99. Per questo attributo viene applicato il flitro *Discretize* in 2 range. Non presenta valori mancanti.



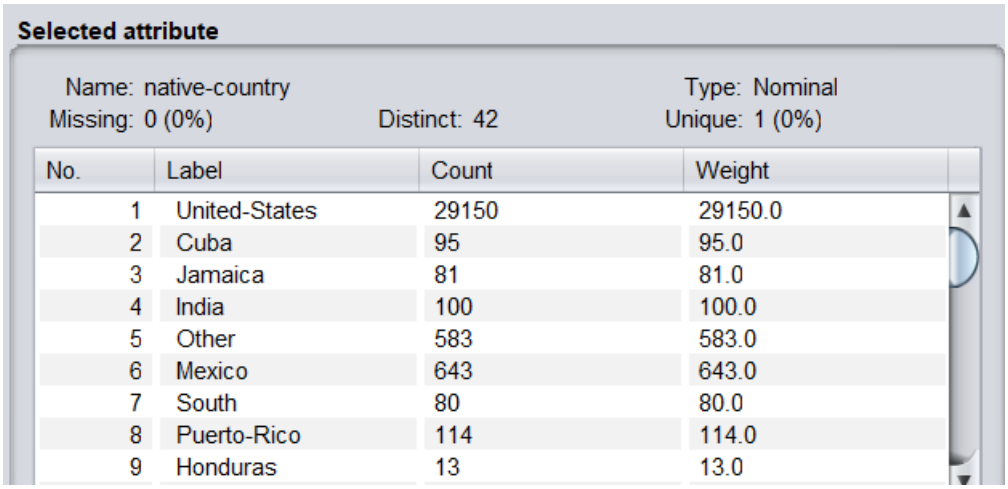
Selected attribute

Name: hours-per-week — Type: Nominal
Missing: 0 (0%) — Distinct: 2 — Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-41]'	22963	22963.0
2	'(41-inf)'	9577	9577.0

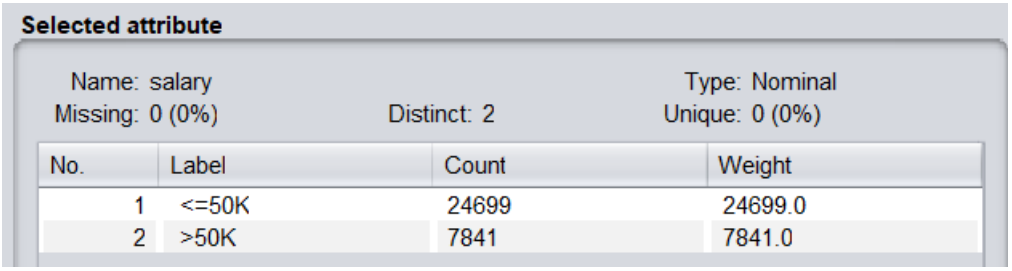
Native-country

L'attributo *native-country* viene rappresentato nei dati di tipo nominal, con la definizione di 42 paese: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands. Non presenta valori mancanti però esiste una categoria segnata con il segno "?". Dato che l'attributo presenta 583 istanze caratterizzate da questo segno, e non sarebbe opportuno eliminare tutte le istanze perché si perderebbe informazione, la modalità verrà ricodificata con "Other".

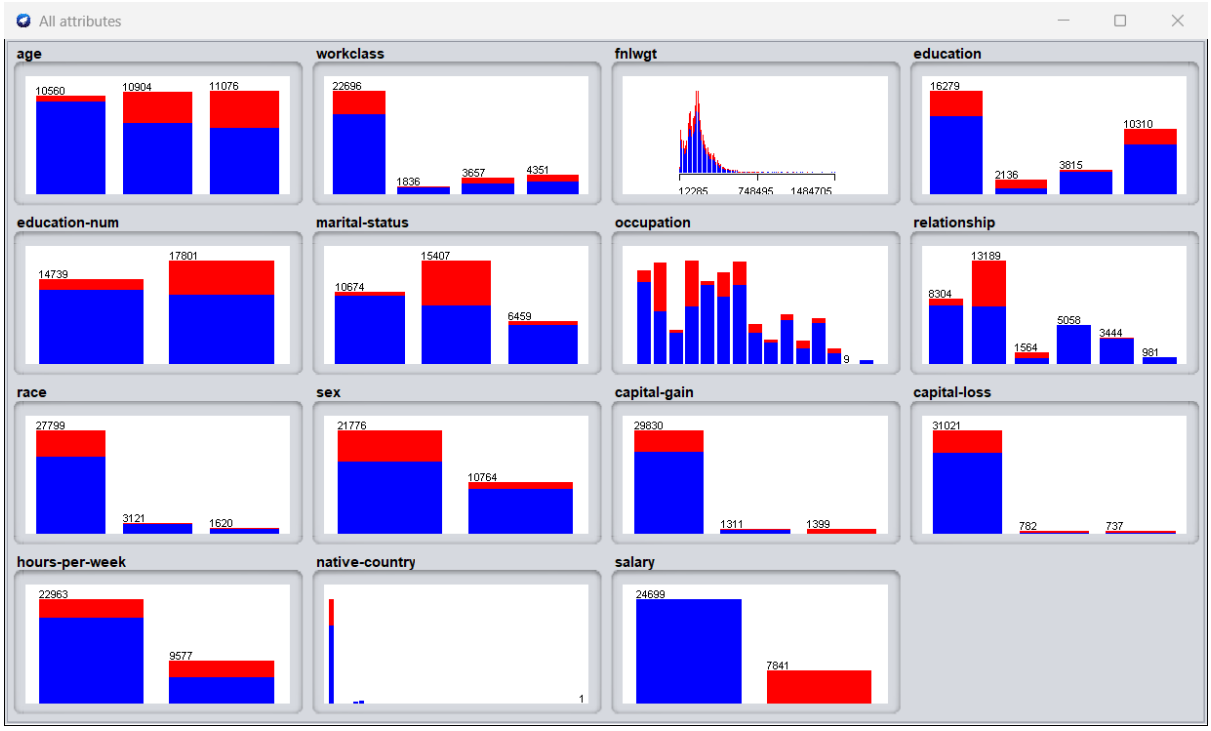


Salary

L'attributo *salary* è la nostra variabile target viene rappresentato nei dati di tipo nominal e presenta due modalità <=50K si guadagna meno di 50.000 dollars o >50 si guadagna più di 50.000 dollars. Non presenta valori mancanti.



La distribuzione di tutte le variabili è presentata nel modo seguente:



Il colore rosso identifica se la persona guadagna più di 50k e il colore blu se la persona guadagna meno di 50k.

Su questa informazione possiamo commentare che il 66.92 % sono di sesso “Male” e il 33.08% sono “Female”. Il sesso con più frequenze di persone che guadagnano più di 50k è “Male”. Dopo di aver discretizzato la variabile *age* identifichiamo che le persone maggiore di 44 anni hanno la maggiore frequenza di persone che guadagna più di 50k. Rispetto alla variabile *education i*l’50% sono “High-School”, il’6% "Graduate and Post", l’11,72% "Elementary-Middle School e il’31,68 “Vocational Programs”. Corrispondente all'attributo *eduaction-num*, il 45,3% hanno studiato meno di 10 anni mentre il’54.7 hanno studiato più di 10 anni. Da questo ultimo si può evidenziare che la maggior frequenza di persona che guadagna più di 50K hanno studiato più di 10 anni. Rispetto all’attributo *marital-status i*l’32.80% sono “Never-Married”, l’47.35% "Married" e il’19.85% “Divor-Seps-Widow”. La modalità dell’attributo *realtionship* con maggiore frequenza di persone che guadagnano più di 50K sono “husband”. Con rispetto all’attributo *occupation,* le occupazioni “Exce-mangerial” e “Prof-speciality” hanno la maggiore frequenza di persone che guadagnano più di 50K. Sull’attributo *workclass* le persone che sono etichettati con private presentano la maggior frequenza di coloro che guadagnano più di 50K . Tra l'intera popolazione considerata, si osserva che il 75,09% delle persone guadagna meno di 50.000 dollari all’anno, mentre il 24,10% registra un reddito superiore a questa soglia.

SELEZIONE DEGLI ATTRIBUTI

Nella maggior parte delle applicazioni pratiche dell'apprendimento supervisionato, non tutti gli attributi sono ugualmente utili per prevedere la destinazione. A seconda dell'attività di apprendimento impiegata, attributi ridondanti e/o irrilevanti possono comportare la generazione di modelli meno accurati. Il compito di identificare manualmente gli attributi utili in un set di dati può essere noioso, ma è possibile applicare metodi di selezione automatica degli attributi. In questo caso utilizziamo GainRatioAttributeEval. valuta gli attributi misurando il loro rapporto di guadagno dell'informazione rispetto alla classe. Come mostrato nell'output, i primi 5 attributi che forniscono informazioni in termini di classe di previsione sono capital-gain, marital-status, realtionship, capital-loss e hours per week. Dopo aver utilizzato l'intero set di dati con diversi algoritmi, abbiamo deciso di non prendere in considerazione le variabili workclass, race, native-country e fnlwgt poiché il loro contributo alla previsione della classe era minimo e gli stessi risultati sono stati ottenuti tenendole in considerazione.

```
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

average merit      average rank  attribute
 0.175 +- 0.001     1    +- 0      11 capital-gain
 0.1   +- 0.001     2    +- 0       6 marital-status
 0.077 +- 0         3    +- 0       8 relationship
 0.049 +- 0.001     4.2  +- 0.4    12 capital-loss
 0.046 +- 0.001     4.9  +- 0.54   13 hours-per-week
 0.045 +- 0         5.9  +- 0.3     1 age
 0.042 +- 0.001     7    +- 0       5 education-num
 0.041 +- 0         8    +- 0      10 sex
 0.029 +- 0         9    +- 0       4 education
 0.026 +- 0        10    +- 0       7 occupation
 0.011 +- 0        11    +- 0       2 workclass
 0.009 +- 0        12.3  +- 0.46    9 race
 0.009 +- 0        12.7  +- 0.46   14 native-country
 0     +- 0        14    +- 0       3 fnlwgt
```

Modelli di apprendimento supervisionato

CLASSIFICATORE ZeroR

Il classificatore Zero Rules (ZR) è un metodo che assegna un caso a una classe in modo deterministico. Cioè, dà sempre lo stesso risultato per un dato insieme di input. In ZR, viene presa una decisione sulla classe per ciascuna classe nel set di dati di addestramento. Successivamente, viene effettuato il conteggio di quanti casi in ciascuna classe sono compatibili con questa decisione di classe. Infine, viene selezionata la classe con il maggior numero di casi compatibili. Questo metodo è utile nei casi in cui altri metodi di machine learning più sofisticati non possono essere utilizzati a causa di vincoli di risorse, tempi di elaborazione, complessità del problema, ecc. Inoltre, il ZR è resistente al rumore nel set di dati di training.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       24699               75.9035 %
Incorrectly Classified Instances      7841               24.0965 %
Kappa statistic                          0
Mean absolute error                      0.3658
Root mean squared error                  0.4277
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.759      1.000    0.863      ?        0.500     0.759      <=50K
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.241      >50K
Weighted Avg.    0.759    0.759    ?          0.759    ?          ?        0.500     0.634

=== Confusion Matrix ===

     a     b   <-- classified as
 24699     0 |     a =  <=50K
  7841     0 |     b =  >50K
```

L'applicazione del classificatore ZeroR restituisce i risultati, come mostrato in precedenza. Se guardiamo la percentuale delle corrette classificazioni suddivise per classi (TP Rate o True Positive Rate), si vede che la prima classe è corretta al 100% (<=50k, TP Rate=1), corrispondente al valore della classe A (guadagna <=50k). La seconda classe fallisce completamente (>50k, TP Rate=0). Questo è logico, visto il modo in cui funziona ZeroR, in cui solo la classe maggioritaria riesce a farlo bene. Con ciò è noto che per l'insieme dei dati, il 75,90% corrisponderà a persone che guadagna meno di 50k e il 24,10% corrisponderà a persone che guadagna più di 50k. Di seguito potete vedere la matrice di confusione dove si osserva che tutti i dati sono classificati come persone che guadagnano meno di 50k.

CLASSIFICATORE IBK

L'algoritmo KNN (IBK) presuppone che cose simili esistano molto vicine. In altre parole, cose simili sono vicine tra loro. È un classificatore di apprendimento supervisionato, che utilizza la prossimità per effettuare classificazioni o previsioni sul raggruppamento di un singolo punto dato. Il valore k nell'algoritmo k-NN definisce quanti vicini verranno controllati per determinare la classificazione di uno specifico punto. Per i problemi di classificazione, viene assegnata un'etichetta di classe in base a un voto di maggioranza, ovvero viene utilizzata l'etichetta rappresentata più frequentemente attorno a un determinato punto dati. Se k=1, l'istanza verrà assegnata alla stessa classe del vicino più vicino. Definire k può essere un atto di bilanciamento poiché valori diversi possono portare a un adattamento eccessivo o insufficiente. Per selezionare il K appropriato per i dati, eseguiamo più volte l'algoritmo KNN con diversi valori di K e scegliamo il K che riduce il numero di errori riscontrati mantenendo la capacità dell'algoritmo di fare previsioni accurate quando vengono forniti nuovi dati che non abbiamo mai visto prima. Man mano che riduciamo il valore di K a 1, le nostre previsioni diventano meno stabili. Al contrario, aumentando il valore di K, le nostre previsioni diventano più stabili a causa del

voto a maggioranza/media e quindi è più probabile che facciamo previsioni più accurate fino a un certo punto. Per applicare l'algoritmi utilizziamo il modo classico e la cross-validation con k=10.

IBK con k=1 con il metodo classico

```
=== Evaluation on training set ===

Time taken to test model on training data: 52.41 seconds

=== Summary ===

Correctly Classified Instances       28378               87.2096 %
Incorrectly Classified Instances      4162               12.7904 %
Kappa statistic                          0.6288
Mean absolute error                      0.1707
Root mean squared error                  0.2921
Relative absolute error                 46.6518 %
Root relative squared error             68.302  %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.943    0.350    0.894      0.943    0.918      0.633    0.937     0.980      <=50K
                 0.650    0.057    0.783      0.650    0.710      0.633    0.937     0.837      >50K
Weighted Avg.    0.872    0.280    0.868      0.872    0.868      0.633    0.937     0.945

=== Confusion Matrix ===

     a     b   <-- classified as
 23284  1415 |     a =  <=50K
  2747  5094 |     b =  >50K
```

IBK con k=3 con il metodo classico

```
=== Evaluation on training set ===

Time taken to test model on training data: 59.48 seconds

=== Summary ===

Correctly Classified Instances       27857               85.6085 %
Incorrectly Classified Instances      4683               14.3915 %
Kappa statistic                          0.5777
Mean absolute error                      0.1929
Root mean squared error                  0.311
Relative absolute error                 52.7292 %
Root relative squared error             72.7271 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.937    0.399    0.881      0.937    0.908      0.584    0.919     0.974     <=50K
                 0.601    0.063    0.752      0.601    0.668      0.584    0.919     0.791     >50K
Weighted Avg.    0.856    0.318    0.850      0.856    0.850      0.584    0.919     0.930

=== Confusion Matrix ===

     a     b   <-- classified as
 23143  1556 |     a =  <=50K
  3127  4714 |     b =  >50K
```

IBK con k=5 con il metodo classico

```
=== Evaluation on training set ===

Time taken to test model on training data: 104.66 seconds

=== Summary ===

Correctly Classified Instances       27734               85.2305 %
Incorrectly Classified Instances      4806               14.7695 %
Kappa statistic                          0.5656
Mean absolute error                      0.1989
Root mean squared error                  0.3159
Relative absolute error                 54.3767 %
Root relative squared error             73.8555 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.936    0.410    0.878      0.936    0.906      0.572    0.913     0.972     <=50K
                 0.590    0.064    0.744      0.590    0.658      0.572    0.913     0.778     >50K
Weighted Avg.    0.852    0.327    0.846      0.852    0.846      0.572    0.913     0.925

=== Confusion Matrix ===

     a     b   <-- classified as
 23107  1592 |     a =  <=50K
  3214  4627 |     b =  >50K
```

IBK con k=1 con cross-validation

```
IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       27391               84.1764 %
Incorrectly Classified Instances      5149               15.8236 %
Kappa statistic                          0.5342
Mean absolute error                      0.2025
Root mean squared error                  0.3365
Relative absolute error                 55.3692 %
Root relative squared error             78.6784 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.929    0.433    0.871      0.929    0.899      0.540    0.883     0.956     <=50K
                 0.567    0.071    0.717      0.567    0.633      0.540    0.883     0.725     >50K
Weighted Avg.    0.842    0.346    0.834      0.842    0.835      0.540    0.883     0.901

=== Confusion Matrix ===

     a     b   <-- classified as
 22944  1755 |     a =  <=50K
  3394  4447 |     b =  >50K
```

IBK con k=3 con cross-validation

```
IB1 instance-based classifier
using 3 nearest neighbour(s) for classification


Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       27416               84.2532 %
Incorrectly Classified Instances      5124               15.7468 %
Kappa statistic                          0.534
Mean absolute error                      0.2062
Root mean squared error                  0.3304
Relative absolute error                 56.3599 %
Root relative squared error             77.2538 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.932    0.438    0.870      0.932    0.900      0.541    0.890     0.960     <=50K
                 0.562    0.068    0.723      0.562    0.632      0.541    0.890     0.739     >50K
Weighted Avg.    0.843    0.349    0.835      0.843    0.835      0.541    0.890     0.907

=== Confusion Matrix ===

     a     b   <-- classified as
 23013  1686 |     a =  <=50K
  3438  4403 |     b =  >50K
```

IBK con k=5 con cross-validation

```
IB1 instance-based classifier
using 5 nearest neighbour(s) for classification


Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       27434               84.3085 %
Incorrectly Classified Instances      5106               15.6915 %
Kappa statistic                          0.5346
Mean absolute error                      0.2082
Root mean squared error                  0.3292
Relative absolute error                 56.9174 %
Root relative squared error             76.9827 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.933     0.440     0.870      0.933     0.900      0.542     0.892     0.961     <=50K
               0.560     0.067     0.726      0.560     0.632      0.542     0.892     0.742     >50K
Weighted Avg.  0.843     0.350     0.835      0.843     0.836      0.542     0.892     0.908

=== Confusion Matrix ===

     a     b   <-- classified as
 23045  1654 |     a =  <=50K
  3452  4389 |     b =  >50K
```

Per valutare i modelli, gli algoritmi utilizzano la matrice di confusione, che permette di identificare quanto il modello era sbagliato e quanto prevedeva bene la classe. Abbiamo i valori tipici TP, TN, FP e FN con cui vengono calcolati Accuracy, RECALL e Precisione del modello. L'accuratezza del modello è la proporzione di volte in cui l'algoritmo ha previsto correttamente, rispetto al totale dei dati valutati. La sensibilità o precisione misura la percentuale dei veri positivi identificati correttamente come tali. La specificità o recall misura la proporzione di veri negativi correttamente identificati come tali. Dai risultati precedenti possiamo evidenziare che il classificatore migliore stimato corrisponde a **IBK1** Classico con una accuratezza dell'87,21%. Ovviamente questo risultato comporta un overfitting dei risultati poiché l'intero set di dati è stato utilizzato per stimare e anche con K=1 porta al classificatore a un adattamento eccessivo. Tra tutti i modelli stimati abbiamo selezionato IBK3 Classico che rappresenta una accuratezza non lontana dalla migliore ottenuta, con valori accettabili di RECALL e Precisione. Un altro parametro che viene esposto è l'area sotto i valori della curva ROC. Questo valore è compreso tra 0 e 1. Quando il classificatore riesce a separare perfettamente le classi, l'area sotto la curva è 1. Quando il classificatore non riesce a separare le classi meglio dell'assegnazione casuale, l'area sotto la curva è 0,5. Per il modello selezionato l'area ROC è 0,919 Per quanto riguarda la stima del modello con validazione incrociata, i classificatori stimati presentano risultati simili, non superando il classificatore stimato in termini di Acuratezza, precisione e RECALL.

Metodo	**Accuracy**	**Precision**	**Recall**	**Kappa**	**ROC**
IBK1 Classico	87.21%	.868	.872	.6288	.937
IBK3 Classico	85.61%	.850	.856	.5777	.919
IBK5 Classico	85.23%	.846	.852	.5656	.913
IBK1 CV	84.17%	.834	.842	.5342	.883

IBK3 CV	84.25%	.835	.843	.534	.890
IBK5 CV	84.30%	.835	.843	.5346	.892

CLASSIFICATORE J48

L'algoritmo J48 base del C4.5, integrato in Weka, è uno degli algoritmi di data mining più diffusi negli studi che includono algoritmi di classificazione. Questa tecnica sceglie in ogni nodo un attributo che di fatto suddivide l'insieme campione in sottoinsiemi, in modo da ottenere maggiori informazioni. Pertanto, l'attributo con il maggior guadagno informativo è quello scelto come parametro decisionale. Ogni partizione o separazione di un set di dati (S) viene effettuata testando tutti i possibili valori delle istanze in ciascuna dimensione o attributo e quindi viene selezionata la partizione migliore in base ad alcuni criteri. Questo processo viene eseguito in modo ricorsivo. Alcuni dei criteri per scegliere la migliore partizione dei set di dati si basano su Entropy(E), Information Gain (G), Split Info(SI) e GainRatio(GR). Questo metodo può essere applicato sia per split multiplo che binario, che per il nostro caso mettiamo in confronto le due modalità. Con queste tipo d'algoritmo si può ottenere diversi resultati, dipende dai parametri specifici che influenzeranno la complessità dell'albero e anche da quanto è grande l'errore che faccio in ogni divisione di ciascun nodo. Se ogni foglia terminale fosse caratterizzata da una classe di appartenenza sarebbe la situazione ideale, ma non sempre avviene. In considerazione di quest'ultimo vengono fissati criteri di arresto in cui viene determinato l'arresto dell'algoritmo. In questo caso abbiamo impostato la numerosità dell'unità nei nodi terminali a 200, testando anche a 100 per confrontare i risultati.

Dagli alberi costruiti abbiamo ottenuto risultati simili, anche in questo abbiamo deciso di non includere l'attributo occupazione per alcuni alberi e confrontare così i risultati. La motivazione è perché questo attributo è molto eterogeneo con 15 modalità di classi, è di tipo sconnesso quindi non è possibile determinare un ordinamento tra i valori ed è difficile raggrupparlo in meno modalità. Nel caso dell'albero, lo scopo è quello di ottenere un classificatore che non sia né così generale né così specifico perché commetterebbe un overfitting del modello.

Metodo	**Accuracy**	**Precision**	**Recall**	**Kappa**	**ROC**
J48 Classico 200	84.87%	.841	.849	.5466	.863
J48 CV 200	84.72%	.840	.847	.5441	.859
J48 Classico 100	**84.94%**	**.842**	**.849**	**.5478**	**.863**
J48 CV 100	84.79%	.840	.848	.5463	.862
J48 Classico 100 (-occupazione)	**84.27%**	**.843**	**.843**	**.5277**	**.868**
J48 Classico 200 (-occupazione)	84.17%	.833	.842	.5172	.861
J48 CV 200 (-occupazione)	84.05%	.832	.841	.5169	.858
J48 CV 100 (-occupazione)	84.13%	.833	.841	.5143	.864
J48 Binario Classico 100 (-occupazione)	84.11%	.834	.841	.5389	.862
J48 Binario CV 100 (-occupazione)	83.97%	.832	.840	.5303	.859

J48 Classico 100

```
=== Evaluation on training set ===

Time taken to test model on training data: 0.03 seconds

=== Summary ===

Correctly Classified Instances       27640               84.9416 %
Incorrectly Classified Instances      4900               15.0584 %
Kappa statistic                          0.5478
Mean absolute error                      0.2229
Root mean squared error                  0.3339
Relative absolute error                 60.9402 %
Root relative squared error             78.0651 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.943    0.444    0.870      0.943    0.905      0.558    0.863     0.935      <=50K
                 0.556    0.057    0.754      0.556    0.640      0.558    0.863     0.707      >50K
Weighted Avg.    0.849    0.351    0.842      0.849    0.841      0.558    0.863     0.880

=== Confusion Matrix ===

     a     b   <-- classified as
 23279  1420 |     a =  <=50K
  3480  4361 |     b =  >50K
```

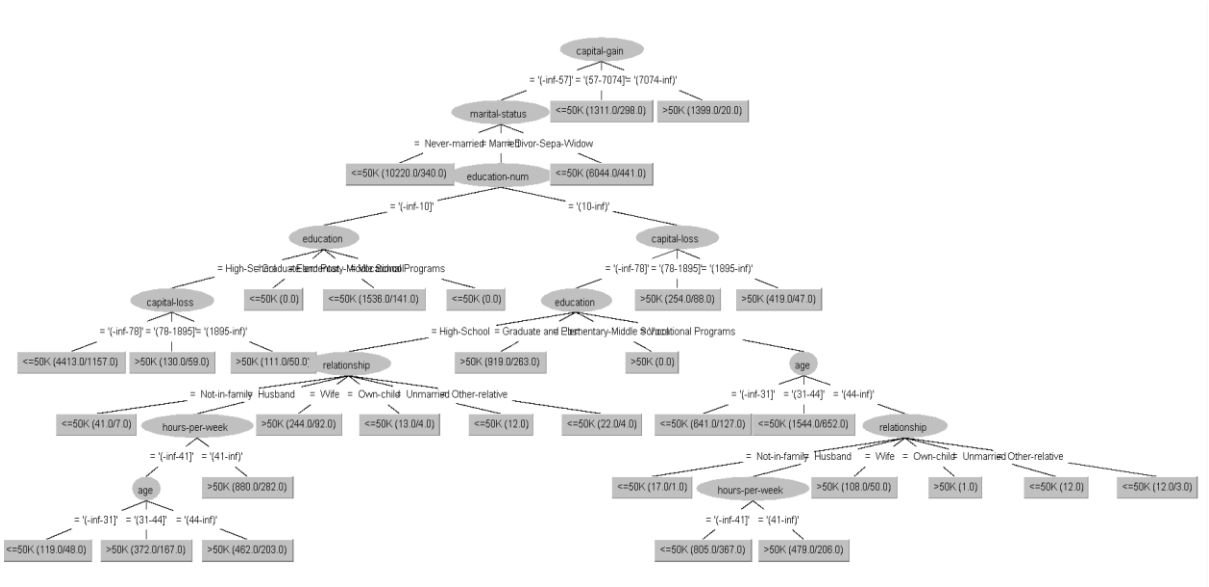
J48 Classico 100 (-occupazione)

```
=== Evaluation on training set ===

Time taken to test model on training data: 0.02 seconds

=== Summary ===

Correctly Classified Instances       27423               84.2747 %
Incorrectly Classified Instances      5117               15.7253 %
Kappa statistic                          0.5277
Mean absolute error                      0.2235
Root mean squared error                  0.3343
Relative absolute error                 61.1047 %
Root relative squared error             78.1704 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.938    0.458    0.866      0.938    0.901      0.538    0.868     0.937      <=50K
                 0.542    0.062    0.736      0.542    0.624      0.538    0.868     0.710      >50K
Weighted Avg.    0.843    0.362    0.834      0.843    0.834      0.538    0.868     0.883

=== Confusion Matrix ===

     a     b   <-- classified as
 23172  1527 |     a =  <=50K
  3590  4251 |     b =  >50K
```

Dai risultati precedenti possiamo evidenziare che il classificatore migliore stimato corrisponde a J48 Classico 100 con una accuratezza dell'84,94%, con Area ROC pari 0.863. In generale, i resultati dei modelli stimati non sono distante tra di loro anche senza tener

conto dell'attributo occupazione, per questo scegliamo il classificatore J48 Classico 100 (-occupazione) con una accuratezza dell'84,27% ma con un Area ROC leggermente superiore pari a 0.868. Il modello generalmente tende a classificare persone che guadagnano più di 50k come persone che guadagnano <=50k.

Dalla matrice di confusione si hanno che 23172 unità appartenenti alla classe "<=50K " e che sono stati classificati bene mentre il 3590 indica coloro che sono stati classificati come "<=50K" ma in realtà sono ">50K". C'è anche che 1527 sono stati classificati come ">50K" ma in realtà sono "<=50K. Sulla diagonale principale si ha il totale delle istanze classificate correttamente che corrisponde a 27423 unità. Di seguito viene riportato l'albero di classificazione associato al modello appena descritto.



Per quanto riguarda l'applicazione della suddivisione binaria non si ottengono risultati migliori.

Andiamo a vedere le prime regole di classificazione che rende il metodo. La prima variabile che suddivide il nodo radice è la variabile "capita-gain", il che significa che è l'attributo che contribuisce meglio a predire la modalità della mia variabile dipendente. Come regola di decisione a priori possiamo dire che le persone che hanno tra [57-7074] di capital-gain l'algoritmo li classifica che guadagnano <=50k e le persone [>7074] di capital-gain li classifica che guadagnano >50k. Allora una persona che di capital-gain si trova [-inf-57] e non è mai stata sposata né è vedova/separata guadagna <=50k.

Random Forest

I modelli Random Forest sono costituiti da una serie di alberi decisionali individuali, ciascuno addestrato con un campione leggermente diverso dei dati di addestramento generati dal bootstrap. La previsione di una nuova osservazione si ottiene sommando le previsioni di tutti i singoli alberi che compongono il modello.

L'algoritmo Random Forest è una modifica del bagging che migliora i risultati decorrelando ulteriormente gli alberi generati nel processo. I metodi random forest e bagging seguono lo stesso algoritmo con l'unica differenza che, nella random forest, prima di ogni suddivisione, m predittori vengono selezionati casualmente. La differenza

nel risultato dipenderà dal valore m scelto. Se m=p i risultati della foresta casuale e del bagging sono equivalenti. Per problemi di classificazioni, la selezione degli m predittori è la radice quadrata del numero totale dei predittori. m≈$\sqrt{p}$.

```
=== Evaluation on training set ===

Time taken to test model on training data: 2.52 seconds

=== Summary ===

Correctly Classified Instances       28378               87.2096 %
Incorrectly Classified Instances      4162               12.7904 %
Kappa statistic                          0.6319
Mean absolute error                      0.1771
Root mean squared error                  0.2942
Relative absolute error                 48.4141 %
Root relative squared error             68.7803 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.939    0.339    0.897      0.939    0.918      0.635    0.936     0.979     <=50K
                 0.661    0.061    0.775      0.661    0.714      0.635    0.936     0.832     >50K
Weighted Avg.    0.872    0.272    0.868      0.872    0.868      0.635    0.936     0.944

=== Confusion Matrix ===

     a     b   <-- classified as
 23193  1506 |     a =  <=50K
  2656  5185 |     b =  >50K
```

Questo risultato è uguale a quello ottenuto con il classificatore IBK1 raggiungendo la stessa accuratezza dell'87.02%, con parametri Kappa statistic e Area ROC quasi uguali.

AdaBoost

AdaBoost (Adaptive Boosting) è un algoritmo di apprendimento automatico supervisionato utilizzato per migliorare l'accuratezza dei modelli di classificazione deboli. L'algoritmo AdaBoost addestra in modo iterativo una sequenza di classificatori deboli su diversi sottoinsiemi di dati, assegnando pesi più elevati ai dati che sono stati classificati in modo errato nelle iterazioni precedenti. Quindi combina i risultati di questi classificatori deboli in un classificatore forte ponderato, in cui i classificatori deboli con prestazioni migliori hanno un peso maggiore nella classificazione finale. L'algoritmo AdaBoost è noto per la sua capacità di migliorare significativamente la precisione dei modelli di machine learning, soprattutto in compiti di classificazione complessi con set di dati grandi e rumorosi. Inoltre, è facile da implementare e può essere adattato a diversi tipi di algoritmi deboli di machine learning, rendendolo popolare nella pratica del machine learning. (J48:100)

```
Time taken to test model on training data: 0.36 seconds

=== Summary ===

Correctly Classified Instances       28274               86.89   %
Incorrectly Classified Instances      4266               13.11   %
Kappa statistic                          0.6235
Mean absolute error                      0.1767
Root mean squared error                  0.2973
Relative absolute error                 48.2943 %
Root relative squared error             69.5111 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.936    0.342    0.896      0.936    0.916      0.626    0.933     0.978     <=50K
                 0.658    0.064    0.765      0.658    0.707      0.626    0.933     0.825     >50K
Weighted Avg.    0.869    0.275    0.864      0.869    0.865      0.626    0.933     0.941

=== Confusion Matrix ===

     a     b   <-- classified as
 23118  1581 |     a =  <=50K
  2685  5156 |     b =  >50K
```

La Radom Forest come la AdaBoost hanno risultati soddisfacenti, rispetto a quelli ottenuti con gli altri metodi. Fino adesso, gli algoritmi tendono a classificare, persone che guadagnano >50k come persone che guadagnano <=50. I falsi positivo vengono ridotti con AdaBoost 2685 e con la Random Forest 2656.

Questo caso non è un caso delicato come i problemi di predizione del cancro, malattie autodegenerative o problemi in cui, per natura, la sensibilità dei classificatori ha un peso importante. Nel caso del problema di studio che stiamo affrontando in questo progetto, è di natura socio-economica e istituzioni come banche o agenzie di marketing sviluppano strategie per identificare segmenti di mercato che, a causa del loro tenore di vita, generano bisogni di prodotti consumi, investimenti, marchi, di standard elevati. Identificare questo segmento di persone contribuisce al posizionamento di servizi e prodotti appositamente predisposti per fidelizzare i propri consumi.

Naive Bayes

Naive Bayes è un modello di previsione basato sulla probabilità bayesiana. Il modello è molto semplice, ma potente, in quanto è il risultato diretto dei dati e del loro trattamento con semplici statistiche bayesiane di probabilità condizionata. Bisogna tenere conto che si assume, per semplicità, che le variabili siano tutte eventi indipendenti.

Il modello di probabilità condizionale bayesiano è rappresentato come: $P(A|B)=P(A \cap B)/P(B)$

Cioè, la probabilità che sia così A dato B è uguale alla probabilità dell'intersezione di A con B ($A \cap B$) corrisponde alla probabilità di B.

Estendendo questa formulazione arriveremmo al teorema di Bayes la cui espressione più tipica è la seguente: $P(A \mid B) = P(B \mid A) * P(A) / P(B)$

NaiveBayes Classico

```
=== Evaluation on training set ===

Time taken to test model on training data: 0.13 seconds

=== Summary ===

Correctly Classified Instances       26616               81.7947 %
Incorrectly Classified Instances      5924               18.2053 %
Kappa statistic                          0.5492
Mean absolute error                      0.1986
Root mean squared error                  0.359
Relative absolute error                 54.2785 %
Root relative squared error             83.9485 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.832    0.225    0.921      0.832    0.874      0.558    0.898     0.965      <=50K
                 0.775    0.168    0.594      0.775    0.672      0.558    0.898     0.754      >50K
Weighted Avg.    0.818    0.212    0.842      0.818    0.825      0.558    0.898     0.914

=== Confusion Matrix ===

     a     b   <-- classified as
 20543  4156 |     a =  <=50K
  1768  6073 |     b =  >50K
```

NaiveBayes CV

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       26626               81.8254 %
Incorrectly Classified Instances      5914               18.1746 %
Kappa statistic                          0.549
Mean absolute error                      0.1988
Root mean squared error                  0.3593
Relative absolute error                 54.3322 %
Root relative squared error             84.0058 %
Total Number of Instances            32540

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.833    0.228    0.920      0.833    0.874      0.558    0.898     0.965      <=50K
                 0.772    0.167    0.595      0.772    0.672      0.558    0.898     0.754      >50K
Weighted Avg.    0.818    0.213    0.842      0.818    0.826      0.558    0.898     0.914

=== Confusion Matrix ===

     a     b   <-- classified as
 20574  4125 |     a =  <=50K
  1789  6052 |     b =  >50K
```

Col metodo probabilistico "NaiveBayes" l'output restituisce un valore per l'accuratezza pari al 81.79% e 81.82% respettivamente. Dalla Confusion Matrix si legge che incrementa il numero di mal classificate con rispetto agli altri metodi applicati. Il modello

generalmente tende a classificare persone che guadagnano meno di 50k come persone che guadagnano >50k.

Le tabelle seguenti mostrano la distribuzione di probabilità delle variabili "Capital-gain", "Marital-status"e "Relationship" che sono le variabili che forniscono il maggior guadagno d'informazione sulla predizione nella classifica di salary. Si può vedere che le persone che hanno un capital-gain [-inf-57] hanno una probabilità del 95,8% di avere uno stipendio <=50k, mentre se questa persona è nell'intervallo [7074-inf] la probabilità è che abbia uno stipendio <=50k è quasi 0, questo ultimo anche è prevedibile. Per l'attributo marital-status si osserva che una persona sposata ha una probabilità dell'85,9% di avere uno stipendio >50K. Per l'attributo relationship si osserva che una persona con il ruolo di husband ha una probabilità dell'85,9% di avere uno stipendio >50K.

Probability Distribution Table For capital-gain

salary	'\'(-inf-57]\''	'\'(57-7074]\''	'\'(7074-inf)\''
' <=50K'	0.958	0.041	0.001
' >50K'	0.786	0.038	0.176

Probability Distribution Table For marital-status

salary	' Never-married'	Married	Divor-Sepa-Widow
' <=50K'	0.412	0.351	0.237
' >50K'	0.063	0.859	0.078

Probability Distribution Table For relationship

salary	' Not-in-fa...	' Husband'	' Wife'	' Own-child'	' Unmarried'	' Other-rel...
' <=50K'	0.302	0.294	0.033	0.202	0.131	0.038
' >50K'	0.109	0.755	0.095	0.009	0.028	0.005

Tra tutte i modelli stimati il miglior classificatore è la Radom Forest.

Classificatore	**Accuracy**	**Precision**	**Recall**	**Kappa**	**ROC**
Random Forest	**87.21%**	**.868**	**.872**	**.6319**	**.936**
AdaBoost	86.89%	.864	.869	.6235	.933
IBK3 Classico	85.61%	.850	.856	.5777	.919
IBK5 Classico	85.23%	.846	.852	.5656	.913
J48 Classico 100	84.94%	.842	.849	.5478	.863

IBK5 CV	84.30%	.835	.843	.5346	.892
IBK3 CV	84.25%	.835	.843	.534	.890
NaivesBayes CV	81.82%	.842	.818	.549	.898
NaviesBayes Classico	81.79%	.842	.818	.5492	.898
ZeroR	75.90%	0	.759	0	.50

Modelli di apprendimento non supervisionato

I metodi non supervisionati sono algoritmi che basano il loro processo di addestramento su un insieme di dati senza etichette o classi precedentemente definite. Nessun valore oggettivo o di classe è cioè noto a priori, né categoriale né numerico. L'apprendimento non supervisionato è dedicato alle attività di raggruppamento, chiamate anche clustering o segmentazione, in cui il suo obiettivo è trovare gruppi simili nel set di dati. Esistono due gruppi principali di metodi o algoritmi di clustering. I metodi gerarchici, che producono un'organizzazione gerarchica delle istanze che compongono il set di dati, consentendo così diversi livelli di raggruppamento e i metodi partizionali o non gerarchici, che generano gruppi di istanze che non rispondono ad alcun tipo di organizzazione. Esempi di questi tipi di metodi sarebbero le k-means.

Clustering

Il clustering è un'attività il cui scopo principale è raggruppare insiemi di oggetti senza etichetta, al fine di costruire sottoinsiemi di dati noti come Cluster. Ogni cluster all'interno di un grafico è formato da un insieme di oggetti o dati che dal punto di vista dell'analisi sono simili tra loro, ma che presentano elementi differenziali rispetto ad altri oggetti appartenenti al data set e che possono formare un cluster indipendente. Questo tipo di processo viene applicato nei modelli di machine learning non supervisionati. Grazie alla sua implementazione, il sistema può analizzare i dati, eseguire l'attività e trovare possibili errori nel suo funzionamento. Il clustering, in questo caso, serve a segmentare i dati in gruppi di dimensioni simili in base alle caratteristiche per facilitare questo processo.

K-Means

L'algoritmo k-medie è un metodo di clustering che divide un set di dati in k gruppi o cluster. I dati sono raggruppati in modo tale che i punti nello stesso cluster siano più simili tra loro rispetto ai punti in altri cluster. Nell'universo degli algoritmi di apprendimento non supervisionato, K-means è probabilmente il più riconosciuto. Il motivo per cui esiste questo metodo è perché oggi la quantità totale di dati creati, acquisiti, copiati e consumati a livello globale è di circa 100 Zettabyte e continuerà a crescere. Con l'algoritmo k-means è possibile raccogliere grandi quantità di informazioni simili in un unico posto, il che aiuta a trovare modelli e fare previsioni in grandi insiemi di dati.

Per utilizzare l'algoritmo K-medie, viene prima specificato il numero di cluster desiderati (k). Ad esempio, impostando "k" uguale a 2 raggrupperai il tuo set di dati in 2 gruppi,

mentre impostando "k" uguale a 4 raggrupperai i tuoi dati in 4 gruppi. Ogni gruppo è rappresentato dal proprio centro o baricentro, che corrisponde alla media aritmetica dei punti dati assegnati al gruppo. In questo modo, l'algoritmo funziona attraverso un processo iterativo finché ciascun punto dati non è più vicino al centroide del proprio cluster rispetto ai centroidi di altri cluster, riducendo al minimo la distanza all'interno del cluster ad ogni passaggio.

Procedura

Specificare il numero di cluster desiderati (k): Il primo passo è specificare in quanti cluster vogliamo dividere il set di dati. Questo numero si chiama k. Selezionare k punti casuali dal set di dati come centroidi iniziali di ciascun cluster: Successivamente, k punti casuali vengono scelti dal set di dati per fungere da centroidi iniziali di ciascun cluster. Questi centroidi sono il punto centrale o la media di ciascun cluster. Assegna ciascun punto nel set di dati al cluster il cui baricentro è più vicino: l'algoritmo assegna quindi ciascun punto nel set di dati al cluster il cui baricentro è più vicino. Per fare ciò, viene calcolata la distanza tra ciascun punto e ciascun baricentro e il punto viene assegnato al cluster il cui baricentro ha la distanza più piccola. Ricalcola i centroidi di ciascun cluster come media di tutti i punti nel cluster: una volta che tutti i punti sono stati assegnati a un cluster, i centroidi di ciascun cluster vengono ricalcolati come media di tutti i punti nel cluster. Ciò significa che la posizione del baricentro viene aggiornata per riflettere il nuovo clustering. La procedura precedente viene ripetuta iterativamente finché i centroidi dei cluster non cambiano più o finché non viene raggiunto il numero massimo di iterazioni.

L'algoritmo può riprodurre diversi risultati, da quelli ottenuti variando i parametri si è deciso di prendere il risultato che ha l'errore quadratico medio del gruppo più piccolo. Se fissa il valore k=12

```
Initial starting points (random):

Cluster 0: '\'(31-44]\'','Graduate and Post','\'(10-inf)\'',Married,' Exec-managerial',' Wife',' Female','\'(-inf-57]\'','\'(-inf
Cluster 1: '\'(-inf-31]\'',High-School,'\'(-inf-10]\'',Divor-Sepa-Widow,' Sales',' Unmarried',' Female','\'(-inf-57]\'','\'(-inf-
Cluster 2: '\'(-inf-31]\'',High-School,'\'(-inf-10]\'',' Never-married',' Handlers-cleaners',' Not-in-family',' Male','\'(-inf-57
Cluster 3: '\'(31-44]\'','Vocational Programs','\'(10-inf)\'',Married,' Craft-repair',' Husband',' Male','\'(-inf-57]\'','\'(-inf
Cluster 4: '\'(44-inf)\'',High-School,'\'(10-inf)\'',Married,' Prof-specialty',' Husband',' Male','\'(-inf-57]\'','\'(-inf-78]\''
Cluster 5: '\'(44-inf)\'',High-School,'\'(10-inf)\'',Married,' Machine-op-inspct',' Wife',' Female','\'(-inf-57]\'','\'(-inf-78]\
Cluster 6: '\'(31-44]\'',High-School,'\'(-inf-10]\'',Married,' Transport-moving',' Husband',' Male','\'(-inf-57]\'','\'(-inf-78]\
Cluster 7: '\'(-inf-31]\'','Elementary-Middle School','\'(-inf-10]\'',' Never-married',' Transport-moving',' Own-child',' Male','
Cluster 8: '\'(-inf-31]\'',High-School,'\'(10-inf)\'',Married,' Prof-specialty',' Husband',' Male','\'(-inf-57]\'','\'(78-1895]\'
Cluster 9: '\'(44-inf)\'',High-School,'\'(-inf-10]\'',Married,' Machine-op-inspct',' Husband',' Male','\'(-inf-57]\'','\'(-inf-78
Cluster 10: '\'(44-inf)\'','Graduate and Post','\'(10-inf)\'',Married,' Exec-managerial',' Husband',' Male','\'(7074-inf)\'','\'(
Cluster 11: '\'(31-44]\'',High-School,'\'(-inf-10]\'',Divor-Sepa-Widow,' Handlers-cleaners',' Not-in-family',' Male','\'(-inf-57]'
```

```
Final cluster centroids:
                                                Cluster#
Attribute                       Full Data             0             1             2             3
                                (32540.0)     (1848.0)      (4163.0)      (4891.0)      (4088.0)
=============================================================================================================
age                            '(44-inf)'    '(31-44]'     '(-inf-31]'   '(-inf-31]'   '(31-44]'
education                      High-School Vocational Programs  High-School  High-School Vocational Programs
education-num                  '(10-inf)'    '(10-inf)'    '(-inf-10]'   '(-inf-10]'   '(10-inf)'
marital-status                   Married       Married     Divor-Sepa-Widow Never-married   Married
occupation                   Prof-specialty  Exec-managerial    Sales      Other-service  Craft-repair
relationship                     Husband       Wife        Unmarried     Not-in-family    Husband
sex                                Male        Female        Female          Male          Male
capital-gain                   '(-inf-57]'   '(-inf-57]'   '(-inf-57]'   '(-inf-57]'   '(-inf-57]'
capital-loss                   '(-inf-78]'   '(-inf-78]'   '(-inf-78]'   '(-inf-78]'   '(-inf-78]'
hours-per-week                 '(-inf-41]'   '(41-inf)'    '(-inf-41]'   '(-inf-41]'   '(-inf-41]'
salary                             <=50K         >50K          <=50K         <=50K         <=50K
```

	4	5	6	7	8	9	10	11
	(2775.0)	(1727.0)	(2133.0)	(3591.0)	(1670.0)	(3435.0)	(884.0)	(1335.0)
	'(44-inf)'	'(44-inf)'	'(31-44]'	'(-inf-31]'	'(31-44]'	'(44-inf)'	'(44-inf)'	'(31-44]'
	High-School	High-School	High-School	Vocational Programs	High-School	High-School	Graduate and Post	High-School
	'(10-inf)'	'(10-inf)'	'(-inf-10]'	'(10-inf)'	'(10-inf)'	'(-inf-10]'	'(10-inf)'	'(-inf-10]'
	Married	Married	Married	Never-married	Married	Married	Married	Divor-Sepa-Widow
	Prof-specialty	Adm-clerical	Transport-moving	Other-service	Prof-specialty	Craft-repair	Exec-managerial	Other-service
	Husband	Wife	Husband	Own-child	Husband	Husband	Husband	Not-in-family
	Male	Female	Male	Male	Male	Male	Male	Male
	'(-inf-57]'	'(-inf-57]'	'(-inf-57]'	'(-inf-57]'	'(-inf-57]'	'(-inf-57]'	'(7074-inf)'	'(-inf-57]'
	'(-inf-78]'	'(-inf-78]'	'(-inf-78]'	'(-inf-78]'	'(-inf-78]'	'(-inf-78]'	'(-inf-78]'	'(-inf-78]'
	'(41-inf)'	'(-inf-41]'	'(41-inf)'	'(-inf-41]'	'(-inf-41]'	'(-inf-41]'	'(-inf-41]'	'(-inf-41]'
	<=50K	<=50K	>50K	<=50K	>50K	<=50K	>50K	<=50K

```
Time taken to build model (full training data) : 0.19 seconds

=== Model and evaluation on training set ===

Clustered Instances

 0       1848 (  6%)
 1       4163 ( 13%)
 2       4891 ( 15%)
 3       4088 ( 13%)
 4       2775 (  9%)
 5       1727 (  5%)
 6       2133 (  7%)
 7       3591 ( 11%)
 8       1670 (  5%)
 9       3435 ( 11%)
10        884 (  3%)
11       1335 (  4%)
```

Come interpretazione del risultato dell'applicazione dell'algoritmo K-medie possiamo osservare secondo l'obiettivo iniziale fissato all'inizio di questo progetto. L'idea è quella di caratterizzare in base agli attributi selezionati come predittori se una persona guadagna più di 50k. Come abbiamo visto finora, i primi 4 attributi che mi aiutano a prevedere la modalità della variabile target sono capital-gain, stato civile, relazione e capital-loss. Uno dei risultati come regola decisionale nell'applicazione degli algoritmi di classificazione è stato che la persona [>7074] con capital-gain la classifica come avente guadagna >50k. Allora una persona che di capital-gain si trova [-inf-57] ed è non è mai stata sposata né è vedova/separata guadagna <=50k. Uno dei problemi che abbiamo riscontrato a priori è che se la persona è [-inf-57] come capital-gain, come rischio identificare che guadagna >50k.

Confrontiamo questi risultati con i risultati ottenuti da k-means. Chi trova [>7074] capital-gain guadagna più di 50K, anche il k-means lo classifica così come se envidenzia

nel cluster 10. Osserviamo che il cluster 1 appartengono persone che hanno [-inf-57] capital-gain, sono sposate, di età compresa tra [31-44] hanno il ruolo di moglie, lavorano più di 40 ore settimanali, il numero di anni per i quali è stata frequentata l'istruzione ha oltre 10 anni in programmi professionali e ricopre posizioni dirigenziali dirigenziali, guadagnando più di 50k. Osserviamo inoltre che il cluster 7 appartengono persone che hanno [-inf-57] capital-gain, sono sposate, di età compresa tra [31-44], hanno il ruolo di marito, lavorano più di 40 ore settimanali, il numero di anni per i quali è stata frequentata l'istruzione è meno di 10 anni dalla Higth-School e lavora nel settore dei trasporti guadagna più di 50k.

Osserviamo anche che al cluster 9 appartengono persone che hanno [-inf-57] capital-gain, sono sposate, di età compresa tra [31-44], hanno il ruolo di marito, lavorano meno di 40 ore settimanali, il numero di anni per i quali è stata frequentata l'istruzione è minore di 10 anni di High-School e sono professionisti specializzati guadagnano più di 50k.

Va notato come comportamento che le donne sono svantaggiate secondo i resultati mostrati e questo è dimostrato anche dalla realtà secondo altre ricerche che si occupano di disuguaglianza di genere sul posto di lavoro. Il comportamento che emerge indica che, affinché una donna possa conseguire un reddito elevato, superiore ai 50.000 dollari, è necessario che dimostri competenze più avanzate acquisite attraverso un periodo prolungato di studio e preparazione professionale, spesso superiore alla media degli uomini. Questo potrebbe richiedere un impegno maggiore in termini di anni di istruzione e formazione per le donne rispetto alla controparte maschile inoltre vediamo anche le donne devono lavorare più di 40 ore settimanali in posizioni Dirigenti-manageriali per conseguire un reddito alto.

Regole Associative: Algoritmo Apriori

Gli algoritmi delle regole di associazione mirano a trovare relazioni all'interno di un insieme di transazioni, in particolare elementi o attributi che tendono a verificarsi insieme. Ciascuno degli eventi o degli elementi che fanno parte di una transazione è noto come item e un insieme di essi è noto come itemset. Una transazione può essere composta da uno o più items; nel caso di più elementi, ogni possibile sottoinsieme di essi costituisce un itemset diverso.

Apriori è stato uno dei primi algoritmi sviluppati per la ricerca delle regole associative e continua ad essere uno dei più utilizzati, prevede due fasi: Identificare tutti gli insiemi di elementi che si verificano con una frequenza superiore a un determinato limite (itemset frequenti). Convertire questi itemset frequenti in regole di associazione. Queste regole associate prendono in considerazione i seguenti due termini:

Supporto: il supporto dell'item o dell'itemset X è il numero di transazioni che contengono X diviso per il numero totale di transazioni.

Confidenza: la confidenza di una regola "Se X allora Y" è definita in base all'equazione

confidenza(X=>Y)=supporto(unione(X,Y))/supporto(X),

dove unione(XY) è il itemset che contiene tutti gli items di X e Y. La confidenza viene interpretata come la probabilità P(Y|X), ovvero la probabilità che una transazione che contiene l'item X, contenga anche l'item di Y. Se la confidenza si fissa al 0.9, sarebbe che la regola è soddisfatta il 90% delle volte.

Trovare itemsets frequente (itemsets con una frequenza maggiore o uguale a un certo supporto minimo) non è un processo banale a causa dell'esplosione combinatoria di possibilità, tuttavia, una volta identificate, è relativamente semplice generare regole di associazione che presentino una confidenza minima. L'algoritmo Apriori effettua una ricerca esaustiva per livelli di complessità (dalla dimensione più piccola a quella più grande di itemsets). Per ridurre lo spazio di ricerca, applicare la regola "se a *non è frequente, nessuno dei suoi superset (set di elementi più grandi che contengono il primo) può essere frequente". Considerato in un altro modo, se un insieme è poco frequente, anche tutti gli insiemi in cui si trova quest'ultimo sono poco frequenti.

Considerando rispettivamente un valore minimo supportato di 0,1 e una confidenza minima di 0,9, generando 15 regole.

```
Generated sets of large itemsets:

Size of set of large itemsets L(1): 5

Size of set of large itemsets L(2): 8

Size of set of large itemsets L(3): 5

Best rules found:

 1. hours-per-week='(-inf-41]' salary= <=50K 18978 ==> capital-loss='(-inf-78]' 18434    <conf:(0.97)> lift:(1.02) lev:(0.01) [341] conv:(1.63)
 2. salary= <=50K 24699 ==> capital-loss='(-inf-78]' 23953    <conf:(0.97)> lift:(1.02) lev:(0.01) [406] conv:(1.54)
 3. capital-gain='(-inf-57]' salary= <=50K 23666 ==> capital-loss='(-inf-78]' 22920    <conf:(0.97)> lift:(1.02) lev:(0.01) [358] conv:(1.48)
 4. hours-per-week='(-inf-41]' 22963 ==> capital-loss='(-inf-78]' 22062    <conf:(0.96)> lift:(1.01) lev:(0.01) [170] conv:(1.19)
 5. hours-per-week='(-inf-41]' salary= <=50K 18978 ==> capital-gain='(-inf-57]' 18228    <conf:(0.96)> lift:(1.05) lev:(0.03) [830] conv:(2.1)
 6. salary= <=50K 24699 ==> capital-gain='(-inf-57]' 23666    <conf:(0.96)> lift:(1.05) lev:(0.03) [1023] conv:(1.99)
 7. capital-gain='(-inf-57]' hours-per-week='(-inf-41]' 21408 ==> capital-loss='(-inf-78]' 20507    <conf:(0.96)> lift:(1) lev:(0) [98] conv:(1.11)
 8. capital-loss='(-inf-78]' salary= <=50K 23953 ==> capital-gain='(-inf-57]' 22920    <conf:(0.96)> lift:(1.04) lev:(0.03) [961] conv:(1.93)
 9. capital-gain='(-inf-57]' 29830 ==> capital-loss='(-inf-78]' 28311    <conf:(0.95)> lift:(1) lev:(-0) [-126] conv:(0.92)
10. sex= Male 21776 ==> capital-loss='(-inf-78]' 20626    <conf:(0.95)> lift:(0.99) lev:(-0) [-133] conv:(0.88)
11. sex= Male capital-gain='(-inf-57]' 19689 ==> capital-loss='(-inf-78]' 18539    <conf:(0.94)> lift:(0.99) lev:(-0.01) [-230] conv:(0.8)
12. hours-per-week='(-inf-41]' 22963 ==> capital-gain='(-inf-57]' 21408    <conf:(0.93)> lift:(1.02) lev:(0.01) [357] conv:(1.23)
13. capital-loss='(-inf-78]' hours-per-week='(-inf-41]' 22062 ==> capital-gain='(-inf-57]' 20507    <conf:(0.93)> lift:(1.01) lev:(0.01) [282] conv:(1.18)
14. salary= <=50K 24699 ==> capital-gain='(-inf-57]' capital-loss='(-inf-78]' 22920    <conf:(0.93)> lift:(1.07) lev:(0.04) [1430] conv:(1.8)
15. capital-loss='(-inf-78]' 31021 ==> capital-gain='(-inf-57]' 28311    <conf:(0.91)> lift:(1) lev:(-0) [-126] conv:(0.95)
```

Dai risultati si può vedere che le persone che lavorano meno di 41 ore settimanali e che guadagnano meno di 50.000 sono caratterizzate come capital-gain [-inf-57] con una confidenza del 97% e come capital-loss [-inf-78] con una confidenza del 96%. Rispetto alle statistiche, lo stipendio medio annuo negli USA si aggira intorno ai 50mila-65mila, negli ultimi anni è cresciuto. Il precedente comportamento derivante dalle regole associative riflette il lavoratore medio che guadagna abbastanza per pagare le spese e le tasse generate dall'affitto, dalle automobili, dal cibo e dai crediti acquisiti con l'acquisizione di beni. È un argomento molto interessante il modo in cui le economie dei paesi sviluppati trovano il modo di sopraffare ai segmenti di persone, attraverso delle istituzioni finanziarie, le banche e le aziende che si dedicano al comportamento umano, ai prodotti, ai marchi e ai servizi che incoraggiano l'acquisto compulsivo per ritrovarsi con i soldi appena sufficienti alla fine del mese.

Uno dei principi dei soldi più sfruttati dai finanzieri è la circolazione, il termine soldi genera soldi. Il lavoratore medio ha una giornata lavorativa di 8 ore al giorno impegnata in lavoro manuale o intellettuale per le aziende, lavoro pagato con denaro. Data la mancanza di cultura finanziaria personale, attributi come capital-gain/capital-loss sono

bassi perché la maggior parte dipende dallo stipendio e non dagli investimenti. Come evidenziato dagli algoritmi di classificazione, le persone che hanno guadagnato o perso molti soldi sono state classificate con uno stipendio più di 50.000 all'anno.

CONCLUSIONE

Come abbiamo visto in precedenza, gli algoritmi ZeroR, K-NN(IBK), Arbol J48, Random Forest e Adaboost sono stati applicati come algoritmi supervisionati. Come algoritmi non supervisionati, k-means e Aprori Algorithm. Tra tutti i modelli applicati nel caso della classificazione, Random Forest è considerato il migliore con una accuratezza dell'87,20%. Questo metodo è un algoritmo molto robusto in cui non corriamo il rischio di overfitting e lentezza nella generazione del modello e nella previsione. Con l'applicazione delle k-mean abbiamo ottenuto comportamenti che gli algoritmi di classificazione concordano con le regole decisionali ed è emerso anche un altro comportamento che a prima vista non era evidente con gli algoritmi di classificazione, ad esempio la disuguaglianza di genere sul posto di lavoro: una donna che ha uno stipendio elevato, che in questo caso sarà superiore a 50k, deve dimostrare capacità espresse in più anni di studio e di preparazione professionale rispetto alla media degli uomini, lavorando più di 40 ore settimanali in posizioni Dirigenti-manageriali. Rispetto all'applicazione dell'Algoritmo Apriori emerge il comportamento di un lavoratore medio, che non è difficile distinguere dagli algoritmi precedenti. Tutto, dipende dall'obiettivo dello studio per definire che un algoritmo è migliore di un altro anche se abbiamo indici come accuratezza, Recall, precisione, kappa, curva Roc e altri.