18/02/2019

Progetto DT-ML

Analisi dei dati volta ad individuare il guadagno annuale di una persona

Paolo Marconi – matricola: 807172
Simone Monti – matricola: 807994
Gianluca Puleri – matricola: 807064

Sommario

Dominio e obiettivi dell'elaborato

Il dominio di riferimento è relativo al censimento di un gruppo di persone e di alcune delle caratteristiche personali e lavorative di questa popolazione.
In particolare, si è interessati a estrapolare dati come l'età, il sesso, il livello di istruzione, la professione, la situazione occupazionale e lo stato coniugale al fine di predire quali persone guadagnano più di 50k dollari l'anno.

In seguito, verranno descritte nel dettaglio tutte le ipotesi compiute e le scelte di design effettuate nella fase di creazione del dataset, focalizzando l'attenzione sui cambiamenti apportati ai dataset originali e successivamente al dataset finale utilizzato per la predizione.

Descrizione dei dataset

Per lo svolgimento del progetto sono stati utilizzati due open source dataset scaricati dalla piattaforma UCI – Machine Learning Repository. Il portale si presenta come una collezione di dataset largamente utilizzata da studenti, ricercatori e professori in tutto il mondo come prima fonte di dataset per lo sviluppo e l'analisi di algoritmi di Machine Learning. L'archivio è stato creato nel 1987 da David Aha in Irvine, California e oggigiorno è una delle piattaforme più utilizzate, tanto da entrare in molti articoli ufficiali della community di computer science. Dal 2007 il portale rientra nel progetto Rexa.info in collaborazione con l'università del Massachusetts Amherst.

I dataset utilizzati sono due:

- Adult Data Set
- Census-Income Data Set

Essi sono stati ottenuti grazie all'estrazione di dati dal portale U.S. Census Bureau[1].

Adult Data Set

Attributi del dataset

1. Age: *numerico intero*,
2. Fnlwgt: *numerico intero*,
3. Workclass: *nominale*,
4. Education: *nominale*,
5. education-num: *numerico intero*,
6. marital_status: *nominale*,
7. occupation: *nominale*,
8. relationship: *nominale*,
9. race: *nominale*,
10. gender: *nominale*,
11. capital_gains: *numerico intero*,
12. capital_losses: *numerico intero*,
13. hours_per_week: *numerico intero*,
14. country_self: *nominale*,

[1] https://www.census.gov/

15. **income_50k**: *nominale*.

In grassetto è stata evidenziata la variabile obiettivo da predire.

Misure di qualità utilizzate

Pertinence e readability:

È stata fatta un'attenta analisi del dataset ed è risultato molto intuitivo e di facile comprensione in quanto sia il nome degli attributi che i relativi valori sono facilmente comprensibili. L'unico attributo che è emerso essere scarsamente documentato è *Fnlwgt*. Esso è un attributo che assume valori formati da 6 cifre numeriche a cui purtroppo non è stato possibile attribuire alcun significato. Inoltre, l'attributo numerico *Education-num* è strettamente correlato (ridondante) all'attributo *Education* e per questo motivo è stato rimosso.

Attributi rimossi:

- *Fnlwgt*
- *Education-num*

Entrambe le caratteristiche sono state rimosse in quanto considerate non utili per lo svolgimento dello studio di Machine Learning.

Completezza

Numero di missing values:

1. Age: 0% (0),
2. Workclass: 0% (0),
3. Education: 0% (0),
4. marital_status: 0% (0),
5. occupation: 0% (0),
6. relationship: 0% (0),
7. race: 0% (0),
8. gender: 0% (0),
9. capital_gains: 0% (0),
10. capital_losses: 0% (0),
11. hours_per_week: 0% (0),
12. country_self: 0% (0),

13. **income_50k**: 0% (0).

Totale missing values: 0% (0).

Non è presente alcun missing value all'interno del dataset Adult.

Census-income Data Set

Attributi del dataset

1. Workclass: *nominale*,
2. Education: *nominale*,
3. Marital_status: *nominale*,
4. occupation: *nominale*,
5. race: *nominale*,
6. gender: *nominale*,
7. full_or_part_emp: *nominale*,
8. capital_gains: *numerico intero*,
9. capital_losses: *numerico intero*,
10. state_prev_res: *nominale*,
11. det_hh_summ: *nominale*,
12. det_hh_fam_stat: *nominale*,
13. fam_under_18: *nominale*,
14. country_self: *nominale*,
15. year: *numerico intero*,
16. **income_50k**: *nominale*,
17. YoB: numerico intero.

In grassetto è stata evidenziata la variabile obiettivo da predire.

Misure di qualità utilizzate

Pertinence e readability:

È stata fatta un'attenta analisi del dataset ed è risultato abbastanza intuitivo e di facile comprensione. Durante questa fase è stato individuato come superfluo l'attributo *det_hh_fam_stat,* in quanto, esso descrive in maniera troppo dettagliata lo stato familiare della persona. Risulta correlato a *det_hh_summ,* il quale, invece,

descrive il complesso familiare in maniera sommaria. Abbiamo optato per la rimozione dell'attributo *det_hh_fam_stat* poiché è risultato di difficile integrazione.

Attributi rimossi:

- det_hh_fam_stat

Completezza

Numero di missing values:

1. Workclass: 0% (0),
2. Education: 0% (0),
3. Marital_status: 0% (0),
4. occupation: 0% (0),
5. race: 0% (0),
6. gender: 0% (0),
7. full_or_part_emp: 0% (0),
8. capital_gains: 0% (0),
9. capital_losses: 0% (0),
10. state_prev_res: ~91% (13697),
11. det_hh_summ: 0% (0),
12. fam_under_18: ~98% (14634),
13. country_self: ~2% (278),
14. year: 0% (0),
15. **income_50k:** 0% (0),
16. YoB: 0% (0).

Totale dei missing values: ~11% (28609).

- L'attributo *state_prev_res* rappresenta lo stato di residenza prima di un'emigrazione da parte dell'individuo, l'elevato numero di missing values è dovuto al fatto che la maggior parte dei soggetti registrati non ha cambiato stato di residenza.
- L'attributo *fam_under_18* indica la condizione familiare di un soggetto minorenne, l'elevato numero di missing values è dovuto al fatto che la maggior parte degli individui registrati non sono minorenni.

Sono state apportate al dataset le seguenti modifiche:

1. Sono stati sostituiti i valori mancanti contenuti nell'attributo *state_prev_res* con la stringa '*non mover*'
2. Se l'individuo è maggiorenne al posto del valore mancante è stata inserita la stringa '*adult'* nell'attributo *fam_under_18*.

Numero di missing values successivamente alle modifiche:

10. state_prev_res: 0% (0),
12. fam_under_18: 0.07% (11).

Numero totale di missing values: 0.1% (289).

Processo di integrazione

Per il processo di integrazione dei dati sono stati utilizzati i seguenti software:

- Talend Data Preparation,
- RStudio.

Talend Data Preparation

Attraverso il tool Talend Data Preparation abbiamo effettuato una prima preparazione dei dati in cui sono state risolte le eterogeneità sintattiche fra i dataset:

- sono stati rinominati con lo stesso nome gli attributi in comune tra i dataset,
- sono stati fatti corrispondere i valori degli attributi con stessa semantica ma differente sintassi (eliminazione maiuscole, simboli e utilizzo della funzione di replace).

RStudio

Per la vera e propria integrazione dei dati è stato utilizzato il software RStudio.

Eterogeneità

Adult dataset

- è stata aggiunto l'attributo *fam_under_18* con valore '*adult*' se l'attributo *age* è risultato maggiore o uguale a 18 o valore vuoto altrimenti,
- è stato sostituito l'attributo numerico *hours_per_week* con l'attributo nominale *full_or_part_emp*:
  - è stato attribuito il valore '*full time schedules*' se il valore contenuto in *hours_per_week* era maggiore o uguale a 35,
  - è stato attribuito il valore '*unemployed part time*' se il valore contenuto in *hours_per_week* era compreso tra 10 e 35,

- è stato attribuito il valore '*unemployed full time*' se il valore contenuto in *hours_per_week* era minore o uguale a 10,
- è stato attribuito il valore '*children or armed forces*' se il valore contenuto in *occupation* era uguale a '*armed forces*',
- è stato attribuito il valore '*children or armed forces*' se il valore contenuto in *age* era minore di 18.

Census-income dataset

- è stato calcolato il parametro *age* effettuando la differenza fra *year* (rappresentazione dell'anno di registrazione) e *Yob* (acronimo di "*year of birth*");
- è stato fatto corrispondere l'attributo *det_hh_sum* a *relationship* grazie alle seguenti trasformazioni:
  - è stato sostituito il valore '*spouse of householder*' con '*wife*',
  - è stato sostituito il valore '*non relative*' con '*not in family*',
  - sono stati sostituiti i valori '*child under 18 never married*', '*child 18 or older*', '*child under 18 ever married*' con '*own child*',
  - è stato sostituito il valore '*householder*' con '*husband*',
  - è stato sostituito il valore '*other relative of householder*' con '*other relative*',
  - sono stati sostituiti i restanti valori con '*unmarried*'.

Merge dei dataset

È stato inserito l'attributo *state_prev_res* in Adult in quanto unico attributo mancante rispetto agli attributi del dataset Census-income, come valore di default è stato utilizzato '*nonmover*', ipotizzando quindi che le persone non abbiano cambiato stato di residenza. Successivamente è stata fatta la union dei dataset, la quale, ha portato alla creazione di un unico dataset denominato MergedAdultCensus.

Descrizione dataset finale

MergedAdultCensus

Attributi

1. workclass: *nominale*,
2. education: *nominale*,
3. marital_status: *nominale*,
4. occupation: *nominale*,
5. race: *nominale*,
6. gender: *nominale*,
7. full_or_part_emp: *nominale*,
8. capital_gains: *numerico intero*,
9. capital_losses: *numerico intero*,
10. state_prev_res: *nominale*,
11. fam_under_18: *nominale*,
12. country_self: *nominale*,
13. age: *numerico intero*,
14. relationship: *nominale*,
15. **income_50k**: *nominale*.

In grassetto è stata evidenziata la variabile obiettivo da predire.

Misure di qualità utilizzate

Pertinence e readability:

Il dataset finale risulta facilmente comprensibile e molto intuitivo, tutti gli attributi sono utili ai fini del processo di Machine Learning.

Completezza

- workclass: 0% (0),

- education: 0% (0),
- marital_status: 0% (0),
- occupation: 0% (0),
- race: 0% (0),
- gender: 0% (0),
- full_or_part_emp: 0% (0),
- capital_gains: 0% (0),
- capital_losses: 0% (0),
- state_prev_res: 0% (0),
- fam_under_18: <1% (304),
- country_self: <1% (278),
- age: 0% (0),
- relationship: <1% (8),
- **income_50k**: 0% (0).

Totale dei missing values: <1% (590).

Il numero di missing values nel dataset finale è inferiore all' 1%.

Descrizione degli attributi

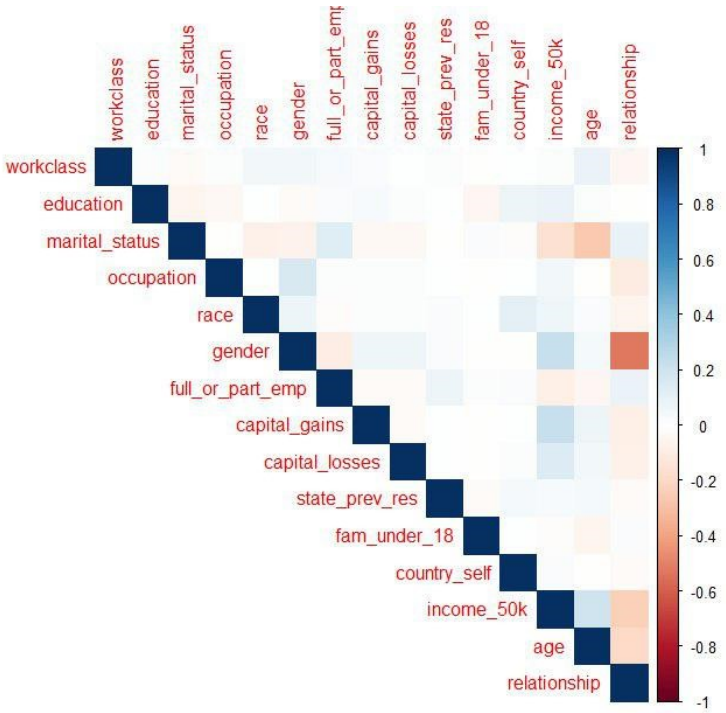
- workclass: tipologia di lavoratore,
- education: livello di educazione raggiunto,
- marital_status: stato matrimoniale,
- occupation: ambito di lavoro,
- race: razza,
- gender: genere,
- full_or_part_emp: tipologia orario lavorativo,
- capital_gains: guadagni per investimento capitali,
- capital_losses: perdite per investimento di capitale,
- state_prev_res: paese di residenza prima di un'emigrazione,
- fam_under_18: situazione familiare di un ragazzo minorenne,
- country_self: stato di residenza,
- age: età,
- relationship: stato coniugale,
- **income_50k**: guadagno superiore o inferiore a 50k annui.

Creazione dei training set

Preso il dataset MergedAdultCensus appena descritto, è stato scelto di suddividerlo in 2 parti per entrambi i modelli che verranno presi in considerazione: training set, composto dal 70% del dataset e test set, composto dal restante 30%.

Analisi esplorativa del training set

Viene presentato il grafico della correlazione tra gli attributi del dataset, dove il valore 1 indica una correlazione massima e il valore 0 un'assenza di correlazione:



Da come si evince dal grafico gli unici attributi che risultano essere lievemente correlati tra loro sono gender e relationship, in cui è presente una leggera correlazione negativa. È auspicabile ipotizzare che lo stato coniugale dipenda dal sesso di una persona. La variabile da predire risulta essere leggermente correlata con l'attributo gender e capital gains.

Descrizione e motivazione dei modelli di machine learning utilizzati

SVM

SVM è un algoritmo di apprendimento automatico supervisionato che può essere utilizzato per problemi di classificazione o regressione. Usa un metodo chiamato kernel per trasformare i dati e quindi, sulla base di queste trasformazioni, trovare un iperpiano separatore ottimale tra i possibili output. In poche parole, esegue alcune trasformazioni di dati estremamente complesse, quindi individua come separare i dati in base alle etichette o agli output definiti.
I punti di forza di quest'algoritmo sono l'elevata efficienza di apprendimento e la capacità di apprendere funzioni di separazione non lineari complesse.
Le complesse trasformazioni di dati e l'iperpiano separatore risultante sono molto difficili da interpretare. Negli alberi decisionali, invece, è molto facile capire esattamente cosa stiano facendo a scapito però delle prestazioni.

Nonostante la difficile interpretazione, è stato scelto di utilizzare SVM per poter usufruire delle alte performance che è in grado di offrire.

Naive Bayes

Naive Bayes è un classificatore di apprendimento automatico semplice, ma efficace e comunemente usato. Può anche essere rappresentato utilizzando una rete bayesiana molto semplice. La dimensione del modello di Naive Bayes è bassa e abbastanza costante rispetto ai dati, inoltre la velocità di apprendimento è molto elevata.
Naive Bayes è un buon algoritmo per lavorare con la classificazione del testo. Quando si ha a che fare con il testo, è molto comune trattare ogni singola parola come una feature, e dal momento che il vocabolario è composto da molte migliaia di parole, questo rappresenta un gran numero di feature. La relativa semplicità dell'algoritmo e l'assunzione delle feature condizionalmente indipendenti di Naive Bayes ne fanno un ottimo interprete per la classificazione dei testi. Inoltre, Naive Bayes risulta essere utile quando si ha a che fare con train set di medie/grandi dimensioni.

Per queste ragioni è stato scelto di utilizzare Naive Bayes.

Esperimenti

10-fold cross validation

La cross validation è una tecnica statistica utilizzabile in presenza di una buona numerosità del training set. In particolare, la k-fold cross validation consiste nella suddivisione del dataset totale in k parti di uguale numerosità e, ad ogni passo, la k-esima parte del dataset viene considerata come test set, mentre la restante parte costituisce il training set. Così, per ognuna delle k parti si allena il modello, evitando quindi problemi di overfitting, ma anche di campionamento asimmetrico del training set, tipico della suddivisione del dataset in due sole parti (training set e test set). In altre parole, si suddivide il campione osservato in gruppi di egual numerosità, si esclude iterativamente un gruppo alla volta e lo si cerca di predire con i gruppi non esclusi. Questo procedimento viene effettuato al fine di verificare la bontà del modello di predizione utilizzato.

Questa tecnica è stata utilizzata per entrambi i modelli sopra elencati settando il valore di k=10.

Stima delle misure di performance

Verranno effettuate delle stime di alcune delle misure di performance sia per SVM che per Naive Bayes. In particolare, verranno stimate:

- Precision: la quale, è misurata tramite il rapporto tra i veri positivi e i positivi totali. Nel nostro caso indicherà la probabilità che il modello predica correttamente che una persona guadagna meno di 50k dollari annui.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall: la quale, è misurata tramite il rapporto tra i veri positivi e il numero di tutti i test che dovrebbero essere risultati positivi. Nel nostro caso indicherà la probabilità che il modello predica correttamente tutte le persone che guadagnano meno di 50k dollari annui.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Accuracy: la quale, è misurata tramite il rapporto tra il numero di predizioni corrette e il numero totale di predizioni. Nel nostro caso indicherà la probabilità con la quale il modello predice correttamente.

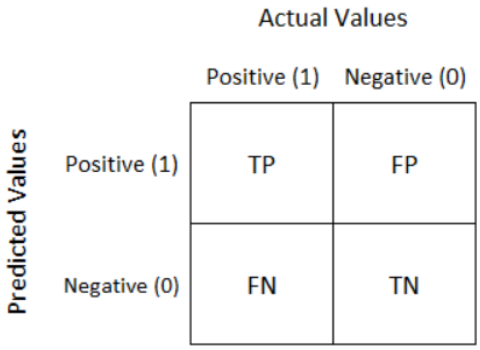
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- F-measure: la quale, rappresenta la media armonica tra precision e recall.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- AUC (Area Under the Curve): la quale, rappresenta la probabilità che il modello distingua correttamente i positivi dai negativi.

Queste misure vengono calcolate tramite l'utilizzo di una matrice di confusione, che restituisce una rappresentazione dell'accuratezza della classificazione.

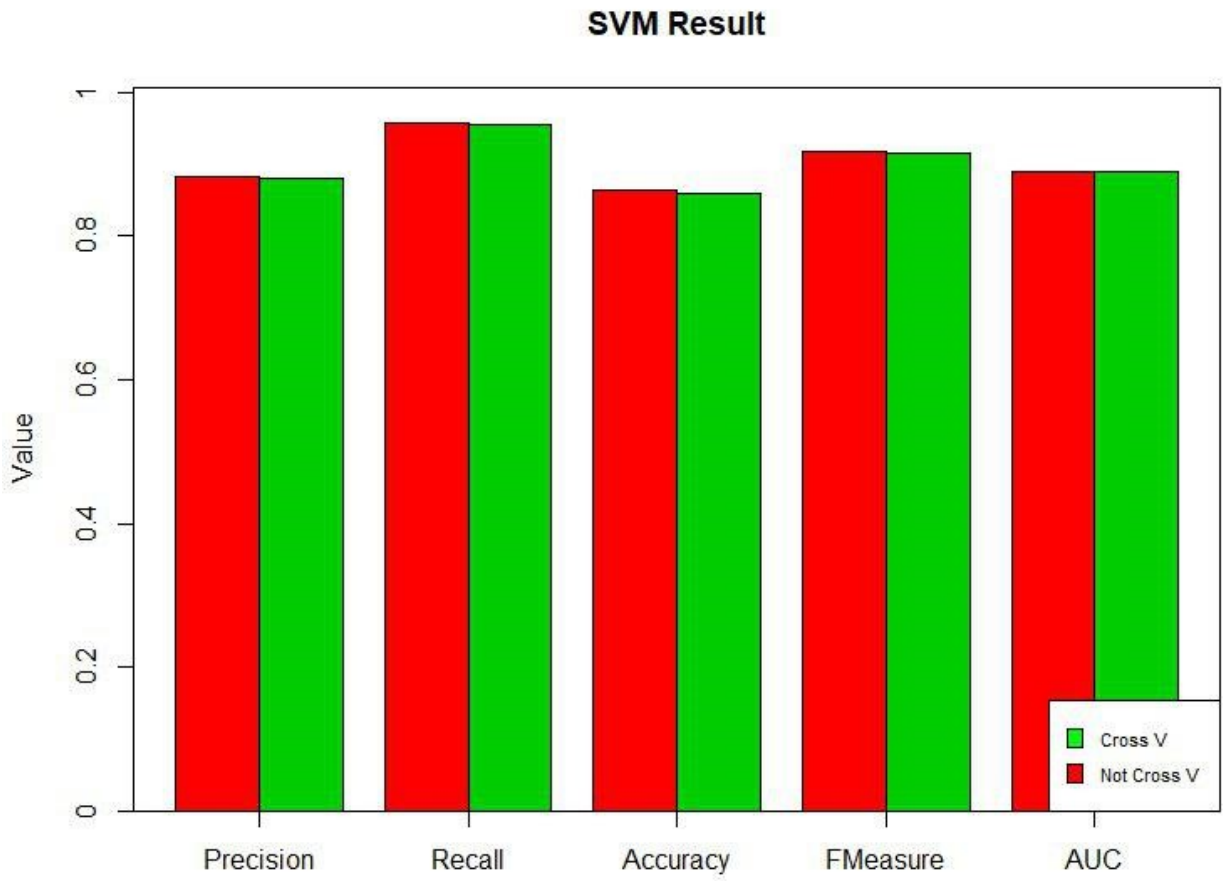


Stima delle misure di performance SVM

Dopo aver effettuato il training del dataset e predetto la variabile obiettivo sono state calcolate alcune misure di performance del modello selezionato. Successivamente all'utilizzo della tecnica di cross validation, sono state ricalcolate le misure di performance per poter effettuare dei confronti.

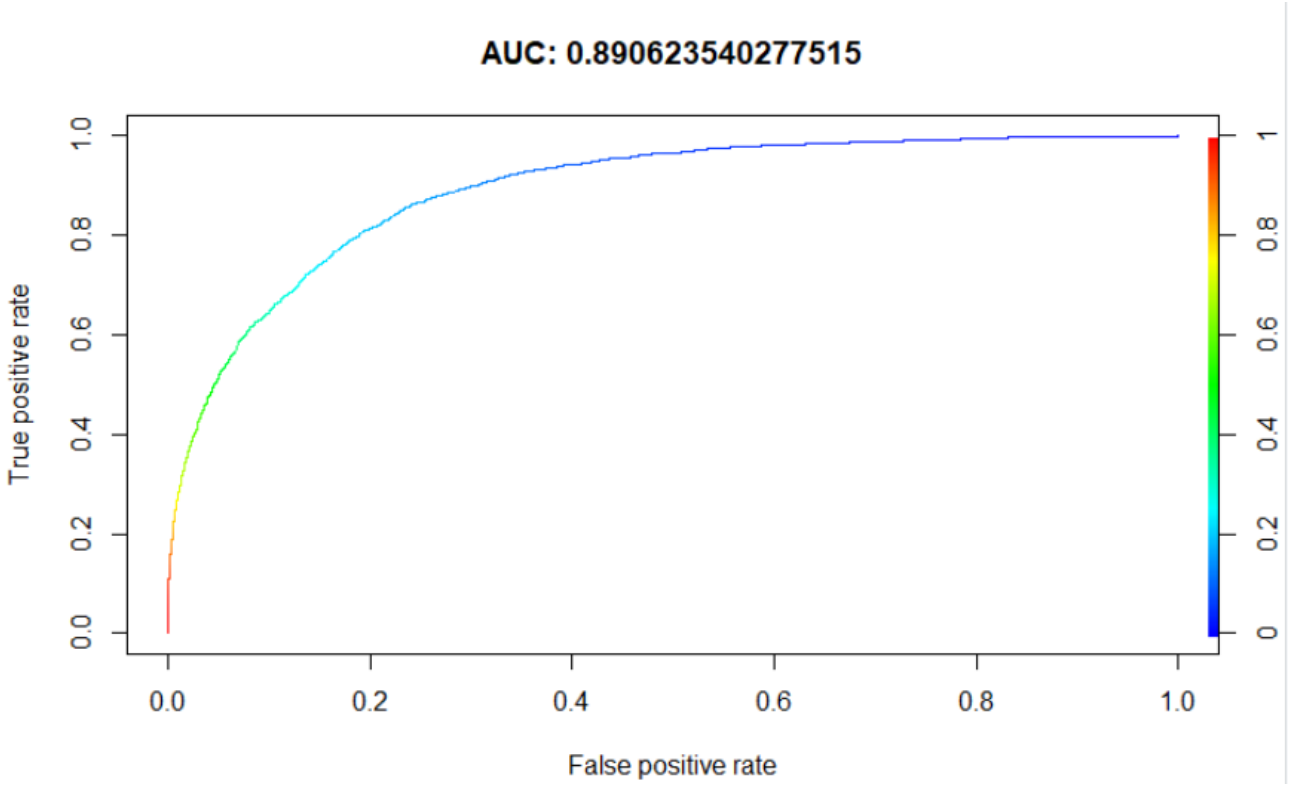
Di seguito sono riportati la matrice di confusione e i risultati ottenuti con il relativo istogramma:

```
svm.pred <=50k >50k
   <=50k  9855 1319
   >50k    428 1223
```



	Precision	Recall	Accuracy	F-Measure	AUC
No Cross Validation	0.8819581	0.9583779	0.8637817	0.9185813	0.8906235
Cross Validation	0.8802724	0.9561824	0.8606550	0.9166518	0.8889169

Viene mostrato inoltre il grafico dell'Area Under the Curve:

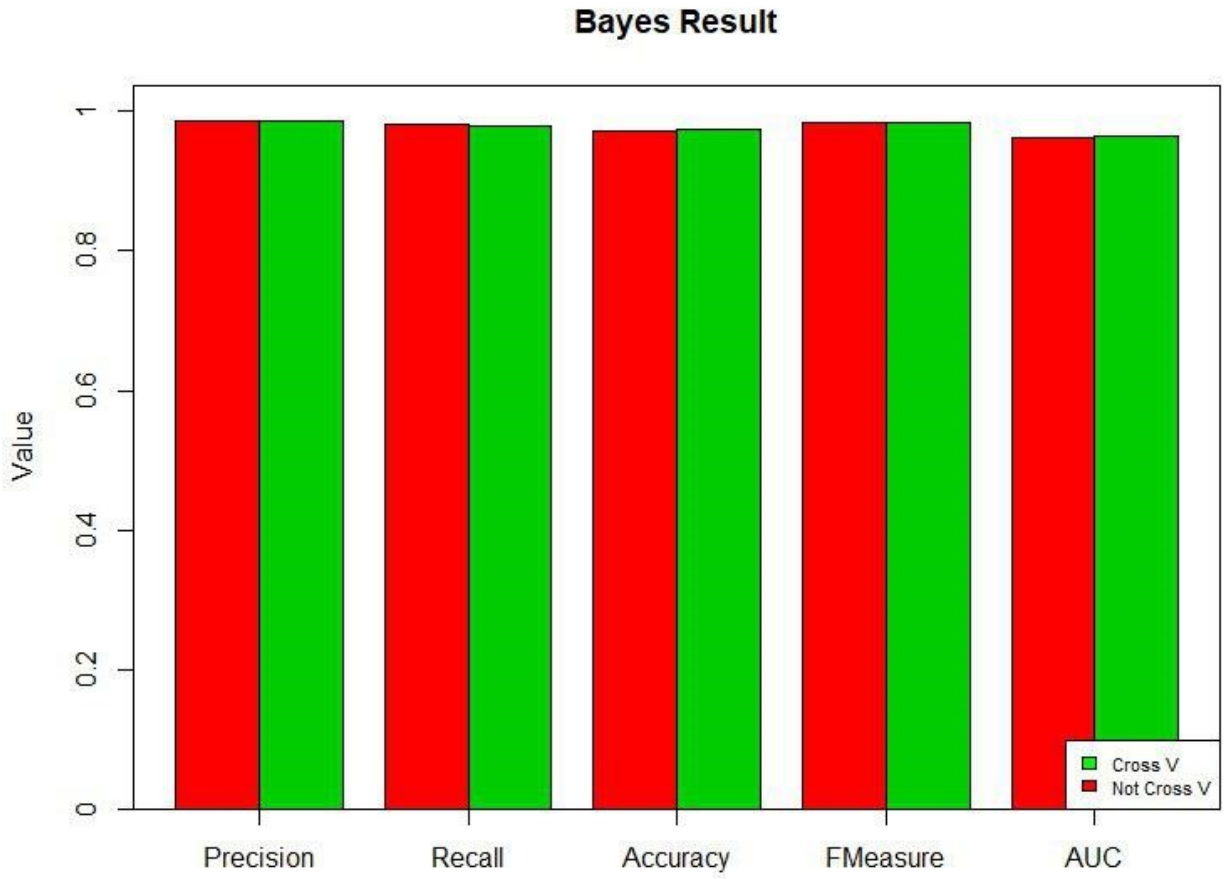


Stima delle misure di performance Naive Bayes

Dopo aver effettuato il training del dataset e predetto la variabile obiettivo sono state calcolate alcune misure di performance del modello selezionato. Successivamente all'utilizzo della tecnica di cross validation, sono state ricalcolate le misure di performance per poter effettuare dei confronti.

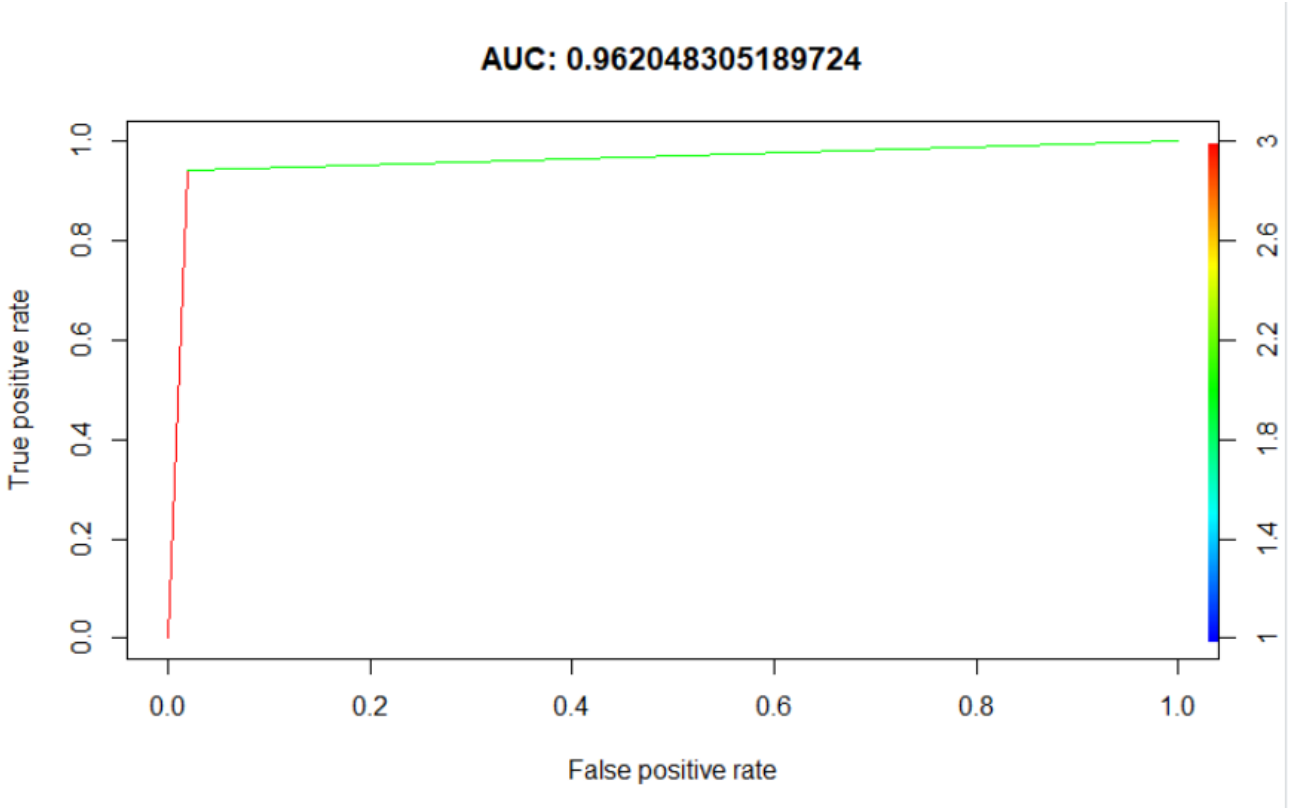
Di seguito sono riportati la matrice di confusione e i risultati ottenuti con il relativo istogramma:

bayes.pred	<=50k	>50k
<=50k	10085	144
>50k	198	2398



	Precision	Recall	Accuracy	F-Measure	AUC
No Cross Validation	0.9859224	0.9807449	0.9733333	0.9833268	0.9620483
Cross Validation	0.9873236	0.9796590	0.9736140	0.9834725	0.9644734

Viene mostrato inoltre il grafico dell'Area Under Curve:



Analisi dei risultati ottenuti

Analizzando i risultati ottenuti verifichiamo che il modello Naive Bayes risulta essere ampiamente più performante rispetto a SVM in tutte le sue misure di performance. Inoltre, dai risultati è possibile riscontrare una pressoché nulla differenza nelle misure di performance tra i modelli di apprendimento automatico e le loro rispettive 10-fold cross validation.

Nel modello Naive Bayes si sono verificati risultati sempre ben superiori al 95% di probabilità e molto simili tra loro; invece, nel modello SVM si sono verificati risultati relativamente bassi (mediamente inferiori al 90% di probabilità) tranne che nel valore di Recall ma ciò è dovuto al fatto che la formula di questa performance tiene conto solo dei falsi negativi (che sono in numero contenuto) senza considerare i falsi positivi.

Conclusioni

In conclusione, è possibile affermare che utilizzando il modello Naive Bayes sul nostro dataset si ottengono delle performance migliori sia in termini di qualità che di velocità di apprendimento, ciò, probabilmente, dovuto anche all'elevata numerosità di attributi nominali all'interno del dataset.

Osservando l'analisi in merito alle 10-fold cross validation è possibile affermare, inoltre, che i due modelli non risentono del problema dell'overfitting.