



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

AI-work

**Intelligenza artificiale per la gestione del lavoro
ed il benessere del lavoratore**

Gruppo formato da:

- Vito Nicola Losavio, 735622, v.losavio5@studenti.uniba.it
- Raffaella Ricciardi, 739074, r.ricciardi15@studenti.uniba.it
- Antonio D'Amelio, 697806, a.damelio9@studenti.uniba.it
- Maria Grazia Miccoli, 735076, m.miccoli45@studenti.uniba.it

L'intero progetto è disponibile al seguente link di [Github](#)

SOMMARIO

<i>Introduzione</i>	2
RAPPRESENTAZIONE	3
Analisi del dataset trovato e modellazione	3
Normalizzazione	5
RAGIONAMENTO	8
APPRENDIMENTO	9
Apprendimento supervisionato.	9
Albero regressione	9
Support vector machine	10
KNN	11
Rete neurale	12
Rete neurale bayesiana	14
Random forest	15
Regressione lineare	16
Regressione logistica	16
Apprendimento non supervisionato	17
Clustering	17
<i>Conclusioni</i>	17

Introduzione

In questo documento verranno illustrate le varie fasi per la realizzazione del progetto relativo al corso di Ingegneria della Conoscenza a.a. 2022/2023.

Il progetto AI-work ha come protagonista il mondo del lavoro e la conseguente gestione delle assunzioni e del benessere del lavoratore una volta assunto. Di fatto, ci siamo voluti concentrare sui seguenti due task:

- **Task 1:** *Valutazione delle caratteristiche lavorative di un candidato in base alle esigenze specifiche dell'azienda.*

Un'azienda può avere delle esigenze lavorative molto specifiche, soprattutto se è alla ricerca di figure tecniche. Di conseguenza, il nostro obiettivo è quello di proporre ai datori di lavoro dei profili che potrebbero soddisfare le loro esigenze.

- **Task 2:** *Valutazione del benessere del lavoratore ed assegnazione delle mansioni in base al suo livello di stress.*

Al giorno d'oggi, molti datori di lavoro concentrano la loro attenzione più al risultato che vogliono raggiungere che al benessere di chi produce quel risultato, ovvero i dipendenti, rischiando di mandare quest'ultimi nel cosiddetto *burnout lavorativo* ed andando a diminuire sia la produttività che la qualità del prodotto stesso. Di conseguenza, il nostro secondo obiettivo è quello di regolare l'assegnazione delle mansioni giornaliere al singolo dipendente in base al suo livello di benessere, valutato in base ad alcuni parametri.

Per fare ciò, abbiamo diviso la realizzazione nelle tre fasi principali, ovvero *rappresentazione, ragionamento e apprendimento*, che verranno descritte nel dettaglio in seguito.

RAPPRESENTAZIONE

In questo primo step, si va a rappresentare il mondo di riferimento seguendo un modello comprensibile dalla macchina in modo da permettere l'elaborazione dei dati che andremo ad introdurre in essa. Nel nostro caso, gli elementi che descrivono il nostro dominio vengono descritti tramite features, ovvero variabili, molto spesso indipendenti, che utilizzano dei simboli per denotare le caratteristiche del mondo.

Nel nostro caso, le features con cui abbiamo lavorato per soddisfare i nostri task ci sono state fornite dal dataset [IBM - Human Resources Data Analysis](#) trovato sulla nota piattaforma Kaggle. Da qui abbiamo quindi dato inizio al preprocessing dei dati così suddiviso:

Analisi del dataset trovato e modellazione

Il dataset trovato rappresenta il settore delle risorse umane del mondo lavorativo tramite:

- **Features discrete:** sono variabili che hanno il dominio definito in un certo intervallo numerico. Nel nostro dataset abbiamo:
 - Age: rappresenta l'età del lavoratore che può andare dai 18 ai 60.
 - DailyRate: rappresenta la tariffa giornaliera che un lavoratore può guadagnare. Può assumere un valore compreso tra 102 e 1499.
 - DistanceFromHome: rappresenta la distanza che il lavoratore deve coprire per raggiungere la sede lavorativa. Può assumere valori che vanno da 1 a 29.
 - Education: rappresenta il grado di istruzione raggiunto dal dipendente. Può assumere valori che vanno da 1 a 5.
 - EnviromentSatisfaction: rappresenta il livello di benessere che il lavoratore prova lavorando nell'ambiente di quell'azienda. Esso è espresso in un intervallo che va da 1 a 4
 - HourlyRate: rappresenta la tariffa oraria di un lavoratore. Esso può assumere un valore che va da 30 a 100.
 - JobInvolvement: rappresenta il livello di coinvolgimento del lavoratore nell'attività che svolge quotidianamente. Esso può assumere un valore che va da 1 a 4.
 - JobLevel: rappresenta il livello-lavoratore raggiunto dal dipendente. Esso può andare da 1 a 5.
 - JobSatisfaction: rappresenta il livello di soddisfazione del lavoratore nei confronti del suo lavoro. Esso può assumere un valore che va da 1 a 4.
 - MonthlyInCome: rappresenta lo stipendio mensile. Esso può assumere un valore che va da 2094 a 27k.
 - NumCompaniesWorked: rappresenta il numero di aziende cambiate dal lavoratore. Questo valore può andare da 0 a 9.
 - PercentSalaryHike: rappresenta l'aumento stipendiale ricevuto dal lavoratore. Esso può assumere valori che vanno da 11 a 25.
 - PerformanceRating: rappresenta quanto rende il lavoratore. Esso può assumere un valore che va da 3 a 4.

- RelationshipSatisfaction: indica il livello di soddisfazione del lavoratore nei confronti dei rapporti umani all'interno dell'azienda. Esso può andare da 1 a 4.
- StandardHours: indica le ore giornaliere del lavoratore.
- TotalWorkingYears: rappresenta le ore annuali del lavoratore. Esso può andare da 0 a 40.
- TrainingTimesLastYears: rappresenta il numero di ore di formazione fatti l'anno precedente dal lavoratore. Esso può andare da 0 a 6.
- YearsAtCompany: rappresenta gli anni passati in azienda. Esso può andare dai 0 ai 40.
- YearsInCurrentRole: rappresenta da quanti anni il lavoratore ricopre quel ruolo. Esso può andare dai 0 ai 18.
- YearsSinceLastPromotion: rappresenta gli anni passati dall'ultima promozione. Esso può andare dai 0 ai 15.
- YearsWithCurrManager: rappresenta gli anni passati con l'attuale manager. Esso può andare dai 0 ai 17.
- **Features categoriche:** sono variabili che possono assumere solo determinati valori appartenenti ad una classe di dominio.
 - BusinessTravel: rappresenta quanto il dipendente viaggia per lavoro. Esso può assumere i seguenti valori:
 - Travel_Rarely: se viaggia raramente.
 - Travel_Frequently: se viaggia spesso.
 - Non-Travel: se non viaggia
 - Department: rappresenta il settore in cui ha lavorato. Può assumere diversi valori, come Sales, Research & Development, Human Resources ecc.
 - EducationField: si riferisce al settore in cui una persona ha acquisito competenze e conoscenze specifiche. Esso può assumere diversi valori come Life Sciences, Medical ecc.
 - Gender: rappresenta il genere di un lavoratore che può essere Male o Female.
 - JobRole: rappresenta il ruolo del lavoratore all'interno dell'azienda. Questo può assumere diversi valori, come Laboratory Technician, Research Scientist ecc.
 - MaritalStatus: rappresenta se il lavoratore è single, sposato o divorziato.
 - Over18: indica se un lavoratore è maggiorenne (True) o no (False)
 - OverTime: indica se un lavoratore è disponibile per fare gli straordinari (True) o no (False)

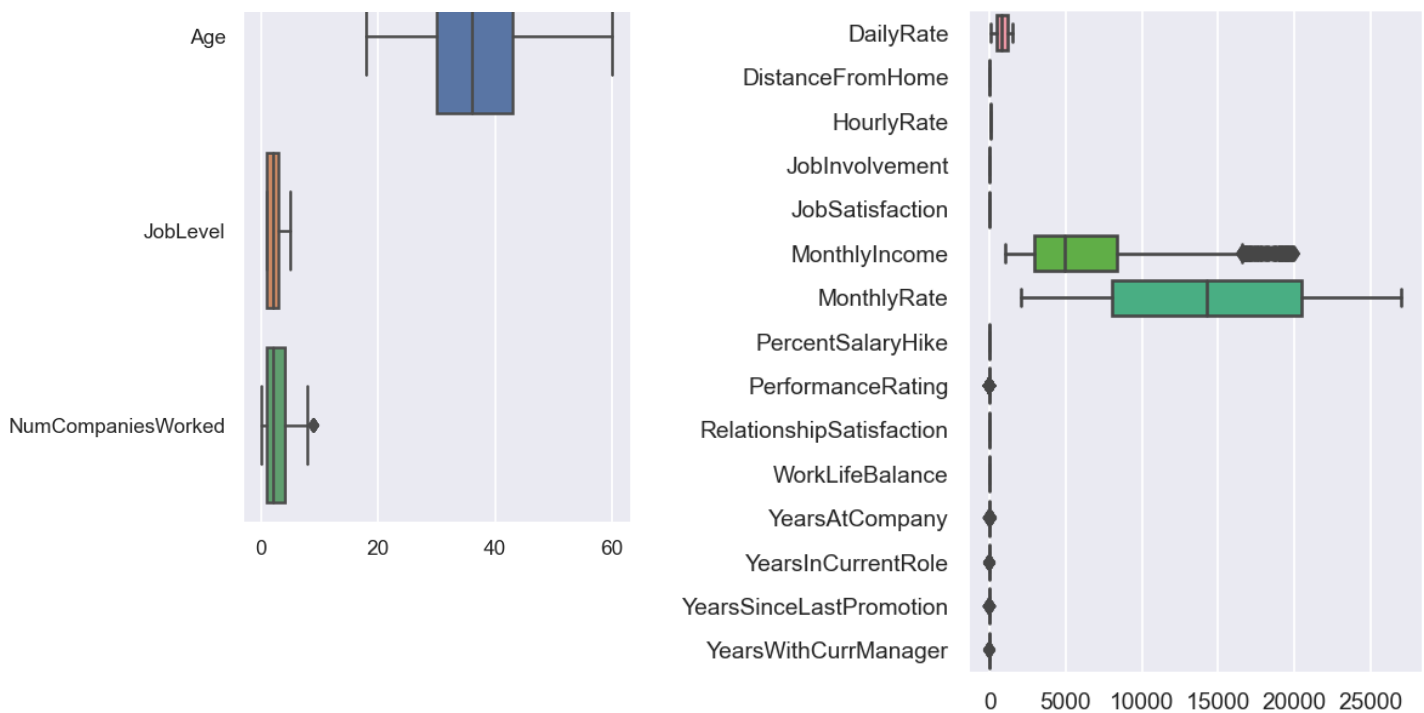
Partendo da questo dataset e considerando i nostri due task, abbiamo generato due datasets distinti mantenendo le features valide per raggiungere i nostri due obiettivi.

Quindi abbiamo ottenuto:

- **First_Dataset.csv:** dataset relativo al task 1, in cui troviamo le seguenti features: Age, BusinessTravel, EducationField, Gender, JobLevel, JobRole, NumCompaniesWorked.
- **Second_Dataset.csv:** dataset relativo al task 2, in cui troviamo le seguenti features: BusinessTravel, DailyRate, DistanceFromHome, HourlyRate, JobInvolvement,

JobSatisfaction, MaritalStatus, MonthlyIncome, MonthlyRate, OverTime,
PercentSalaryHike, PerformanceRating, RelationshipSatisfaction, WorkLifeBalance,
YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion,
YearsWithCurrManager.

A questo punto, siamo passati con la valutazione dei valori delle features discrete fatta tramite grafici *boxplot* utilizzando la libreria *seaborn*. Abbiamo ottenuto i seguenti risultati:



Tramite questi due grafici è possibile visualizzare la distribuzione delle variabili numeriche rispettivamente del primo e del secondo dataset.

Notiamo che in *NumCompaniesWorked* e *MonthlyIncome* sono presenti degli outlier, ovvero dei valori che vanno al di fuori del limite massimo. Da una nostra valutazione, abbiamo deciso di lasciarli perchè nel primo caso, potrebbe capitare che un'azienda ricerchi un lavoratore con un tasso di esperienza e di serietà in più, valutando per l'appunto il numero di aziende in cui ha lavorato, mentre nel secondo caso, la distribuzione dei valori si divide in due intervalli distinti. Da qui si passa alla fase di normalizzazione.

Normalizzazione

In questa fase abbiamo effettuato la normalizzazione delle variabili sia categoriche che numeriche , un'operazione importante per garantire che i dati siano rappresentati in modo coerente.

La funzione '**categorical_var_normalization**' viene utilizzata per la normalizzazione delle variabili categoriche del dataset nel seguente modo:

- **Business Travel**

La variabile Business Travel è rappresentata attraverso tre categorie:

- *Travel_Rarely* è mappato con il valore 1.
- *Travel_Frequently* è mappato con il valore 2.
- *Non-Travel* è mappato con 0.

Durante il processo di normalizzazione, è stata osservata una discrepanza nei valori '*Travel_Rarely*', '*Travel_Frequently*', '*Non-Travel*'. Inizialmente erano scritti con un underscore tra le due parole. Tuttavia, a causa di problemi riscontrati nel Prolog per la presenza di spazi bianchi sono stati rimossi, scrivendoli come '*TravelRarely*', '*TravelFrequently*', '*NonTravel*'.

- **Gender**

La variabile Gender rappresenta il genere del soggetto ed è una variabile binaria , con due categorie : F (femmina) e M (maschio) , normalizzata nel seguente modo :

- *F* è mappato con il valore 1
- *M* è mappato con il valore 0

- **Education Field**

Per quanto riguarda la variabile Education Field considereremo tutti gli individui specializzati nel campo '*Life Science*' e '*Technical Degree*' , che sono mappati nel seguente modo:

- *Life Science* e *Technical Degree* sono mappati a 1.
- Tutti gli altri possibili valori sono mappati a 0.

- **Job Role**

La variabile Job Role considereremo tutti i ruoli che un'azienda ricerca cioè Research Scientist , Manager , Laboratory Technician .La mappatura sarà effettuata nel seguente modo:

- '*Research Scientist*' , '*Manager*' , '*Laboratory Technician*' sono mappati a 1
- Tutti gli altri ruoli sono mappati a 0.

- **Marital Status**

La variabile Marital Status è espressa attraverso tre categorie:

- *Married* è mappata a 0.
- *Divorced* è mappato a 1.
- *Single* è mappato a 2.

- **OverTime**

La variabile Over Time indica se un certo individuo svolge attività lavorative oltre l'orario di lavoro stabilito.

Essa è rappresentata come una variabile booleana e assume i seguenti valori:

- *Yes* , mappato con 1 se il soggetto lavora effettuando straordinari.
- *No* , mappato con 0 se il soggetto non fa straordinari.

- **Suitable**

La variabile Suitable indica se un certo individuo è idoneo a un determinato ruolo che deve svolgere all'interno dell'azienda. Essa è stata ricavata utilizzando le regole del Prolog e si tratta della *features target* sulla quale andremo ad eseguire la classificazione; è rappresentata come una variabile booleana ed assume i seguenti valori:

- *False*, mappato a 0 se il soggetto non è idoneo.
- *True*, mappato a 1 se il soggetto non è idoneo.

La funzione '**numeric_variables**' viene utilizzata per normalizzare le variabili numeriche presenti nel dataset .

Nel processo di normalizzazione, vengono utilizzati due approcci : la standardizzazione Z score e la normalizzazione min-max.

La *standardizzazione Z score* è un metodo efficace per ridurre le differenze di scala tra le variabili consentendo un confronto più accurato.

La *normalizzazione min-max* , invece, ridimensiona i valori delle variabili in un intervallo compreso tra 0 e 1 , basandosi sul valore minimo e massimo della variabile .

Questo metodo permette di confrontare variabili che hanno range di valori molto diversi.

I valori normalizzati vengono salvati nei nuovi file 'Normalized_FirstDataset.csv' e 'Normalized_SecondDataset.csv'.

RAGIONAMENTO

In questo secondo step, andremo ad applicare il RAGIONAMENTO particolarmente utile per effettuare ricerche , ottenere soluzioni ai problemi e verificare la veridicità di affermazioni logiche.

Nel caso di studio è stato utilizzato il linguaggio Prolog , in particolare la versione 8.4.3 dell'IDE SWI-Prolog attraverso cui è stato possibile scrivere, testare ed eseguire le regole per l'estrapolazione della features target.

Le regole e i fatti definiti a livello aziendale sono presenti e consultabili all'interno del file **'rule.pl'**.

I **fatti** sono rappresentati da predicati che sono i seguenti:

- **"role"** , definisce i ruoli possibili come 'ResearchScientist', 'Manager' e 'LaboratoryTechnician'.
- **"age"** ,il range d'età considera gli individui dai 21 fino ai 38 anni.
- **"educationField"** considera solo gli individui che hanno conoscenze e competenze nei seguenti campi : "TechnicalDegree" e "LifeSciences".
- **"numCompanies"** definisce il numero massimo di aziende per le quali i soggetti possono aver lavorato, nel seguente caso possono essere al massimo quattro.
- **"businessTravel"** definisce quanto il soggetto viaggia per lavoro considerando tutti e tre i casi : 'TravelRarely' , 'TravelFrequently' , 'NonTravel'.

La **regola** principale definita è **"suitable"**, che prende in considerazione i valori precedentemente definiti per determinare l'idoneità di una persona.

La funzione **"create_feature_target"**,pertanto serve per filtrare e selezionare tutti gli individui del dataset che soddisfano i criteri definiti.

APPRENDIMENTO

Una volta manipolati i due dataset e realizzata la base di conoscenza siamo passati all'utilizzo di algoritmi di Machine Learning per verificare quanto l'apprendimento automatico possa essere di aiuto nello svolgimento delle task da noi descritte e con quale accuratezza otteniamo i risultati.

Per i diversi modelli viene effettuata prima una Grid Search per definire quali sono i migliori parametri che il modello utilizzerà per effettuare le classificazioni o predizioni.

Una volta che otterremo i parametri migliori per i diversi modelli potremmo addestrarli e passare alla valutazione del modello andando a visualizzare a video i grafici e analizzarli.

La prima task che andremo a svolgere è una task di classificazione, ovvero andremo a definire se una persona è consona o meno per il lavoro ricercato dall'azienda.

Per quanto riguarda la seconda task considerata, trattasi di una task che può essere affrontata come classificazione o predizione, abbiamo sviluppato sia dei modelli di classificazione per definire se un lavoratore è stressato o meno dal lavoro ed abbiamo introdotto anche modelli di regressione per effettuare una predizione.

In seguito abbiamo anche effettuato una task di apprendimento non supervisionato effettuando il clustering delle feature.

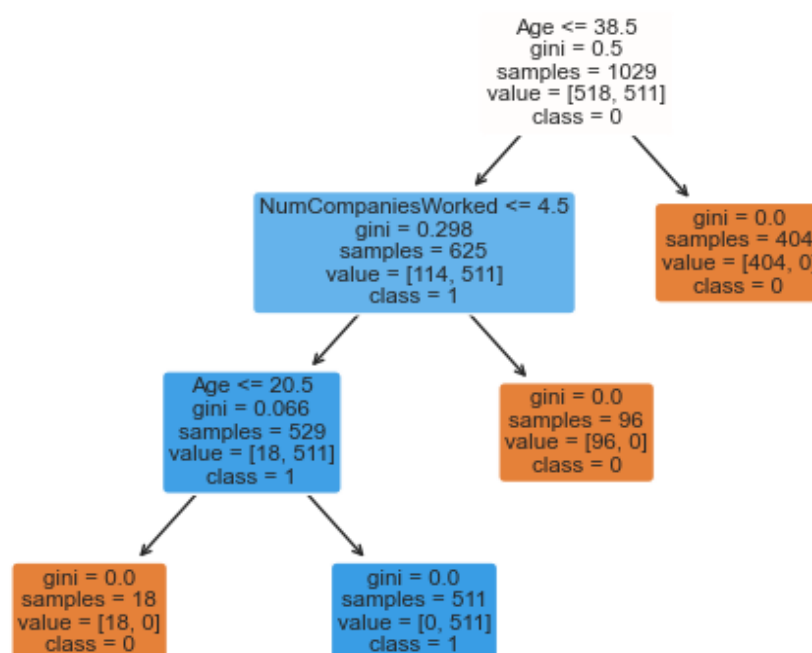
Inoltre per le due task andremo a visualizzare gli alberi di decisione realizzati.

Apprendimento supervisionato

Albero regressione

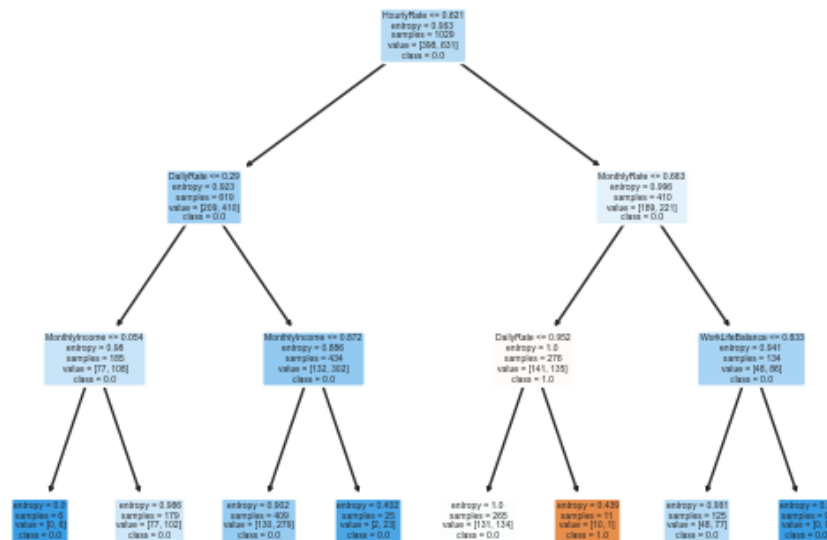
Prima task

Nella figura di seguito è presente un albero decisionale estrapolato dalla prima task, possiamo verificare che man mano che scendiamo nell'albero la distribuzione valuta gli anni del candidato ed il numero di compagnie per le quali ha lavorato.



Seconda task

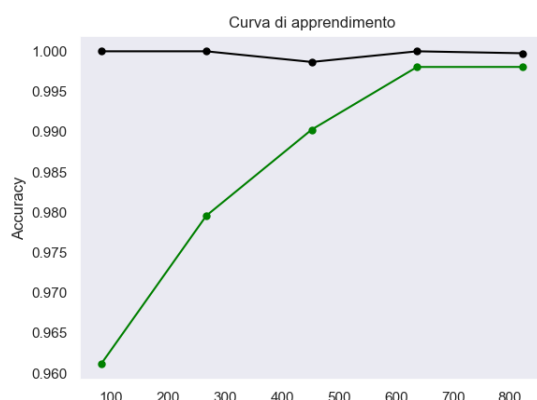
Nella figura di seguito è presente l'albero decisionale della seconda task. In questo caso la distribuzione risulta essere omogenea sui nodi e le feature che in questo caso andremo a valutare sono il guadagno all'ora del lavoratore, il guadagno giornaliero e mensile ed infine il bilancio tra la vita esterna ed il lavoro.



Support vector machine

Nelle due task abbiamo usato una particolare versione della support vector machine ovvero la SVC (support vector classifier) andando a classificare i due dataset.

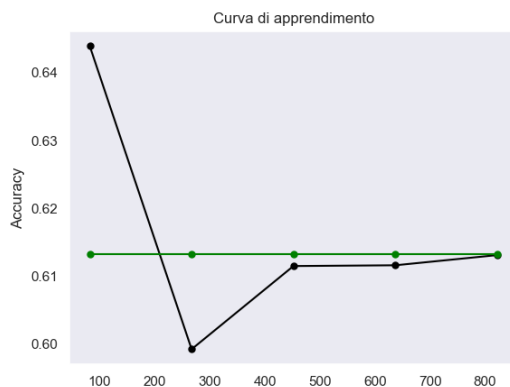
Risultati primo dataset



REPORT DEL MIGLIORE MODELLO SVM TROVATO				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	223
1	1.00	1.00	1.00	218
accuracy			1.00	441
macro avg	1.00	1.00	1.00	441
weighted avg	1.00	1.00	1.00	441

Analizzando la learning curve e valutando l'accuratezza del modello, possiamo riscontrare che il modello si comporta molto bene nello svolgimento della task avendo un'accuratezza molto elevata analizzando già i primi 650-700 esempi. Il report viene creato andando ad analizzare la SVM con dati che il modello non ha ancora classificato ottenendo un risultato perfetto.

Risultati secondo dataset

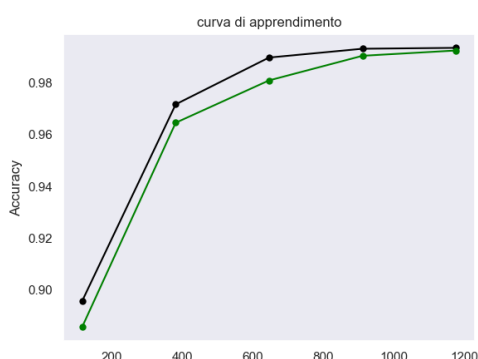


REPORT DEL MIGLIORE MODELLO SVM TROVATO					
		precision	recall	f1-score	support
	0.0	0.00	0.00	0.00	0
	1.0	1.00	0.50	0.66	441
	accuracy			0.50	441
	macro avg	0.50	0.25	0.33	441
	weighted avg	1.00	0.50	0.66	441

Nel caso della seconda task, considerato che la soddisfazione di un lavoratore può dipendere da diversi fattori che potrebbero non essere indicati all'interno del dataset, l'accuratezza della SVM risulta essere inferiore e dalla learning curve possiamo notare che l'accuratezza converge dopo aver dato alla SVM 800 esempi.

KNN

Risultati primo dataset

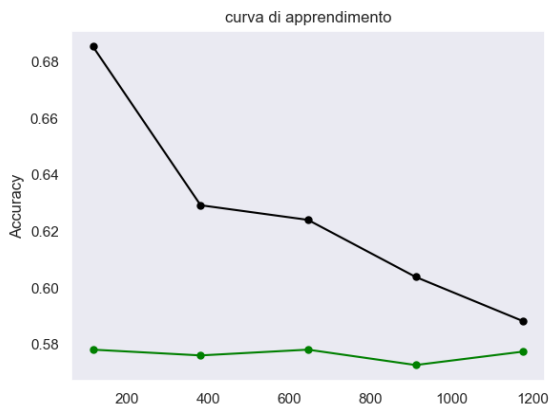


REPORT DEL MIGLIORE MODELLO KNN TROVATO				
	precision	recall	f1-score	support
0	0.97	1.00	0.98	215
1	1.00	0.97	0.98	226
accuracy			0.98	441
macro avg	0.98	0.98	0.98	441
weighted avg	0.98	0.98	0.98	441

Utilizzando il knn con un numero di figli e di parametri definiti durante la grid search possiamo verificare che il modello si comporta molto bene nello svolgimento della prima task

riscontrando una accuratezza molto elevata, il che renderebbe il modello uno dei migliori per lo svolgimento della task.

Risultati secondo dataset



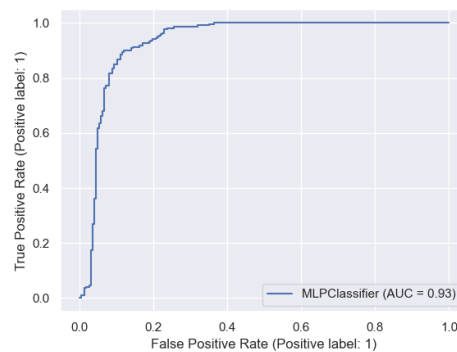
REPORT DEL MIGLIORE MODELLO KNN TROVATO

	precision	recall	f1-score	support
0.0	0.10	0.55	0.17	42
1.0	0.91	0.50	0.65	399
accuracy			0.51	441
macro avg	0.51	0.52	0.41	441
weighted avg	0.84	0.51	0.60	441

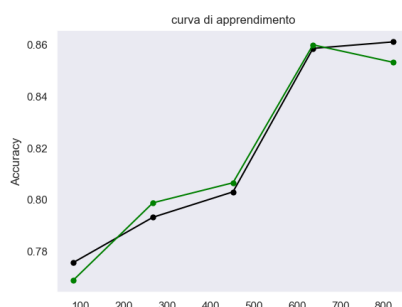
Per lo svolgimento della seconda task il modello knn risulta avere una accuratezza molto bassa, considerata la task ci aspettavamo già questo andamento.

Rete neurale

Risultati primo dataset



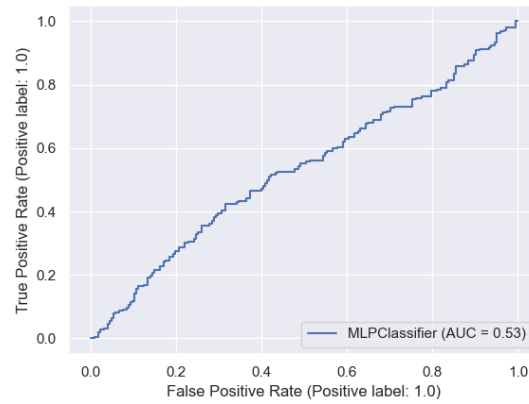
Analizzando la curva di Roc derivata dalla rete neurale notiamo che presenta un aumento esponenziale, segno di buon andamento del modello. Di seguito sono presenti i grafici della learning curve e il report del modello, evidenziando che quest'ultimo risulta stabile con una accuratezza massima sul training ottenuta valutando tra i 600 ed i 700 esempi.



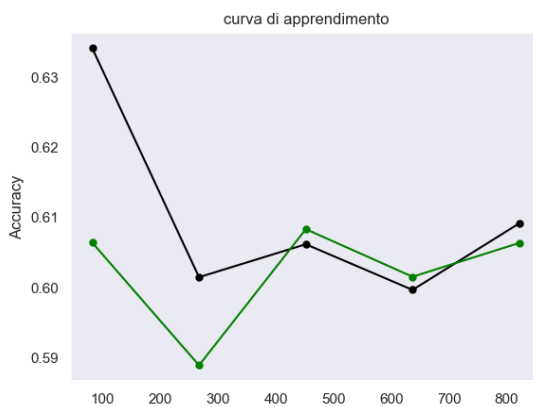
REPORT DELLA MIGLIORE RETE NEURALE TROVATA

	precision	recall	f1-score	support
0	0.88	0.89	0.89	219
1	0.89	0.88	0.89	222
accuracy			0.89	441
macro avg	0.89	0.89	0.89	441
weighted avg	0.89	0.89	0.89	441

Risultati secondo dataset



Nel caso del secondo dataset abbiamo una curva di Roc con un andamento lineare.
Di seguito il grafico della training curve ed il report della rete neurale.

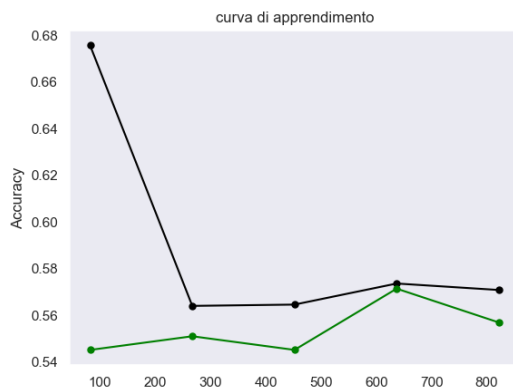


REPORT DELLA MIGLIORE RETE NEURALE TROVATA				
	precision	recall	f1-score	support
0.0	0.07	0.65	0.12	23
1.0	0.96	0.50	0.66	418
accuracy			0.51	441
macro avg	0.52	0.58	0.39	441
weighted avg	0.92	0.51	0.63	441

Rete neurale bayesiana

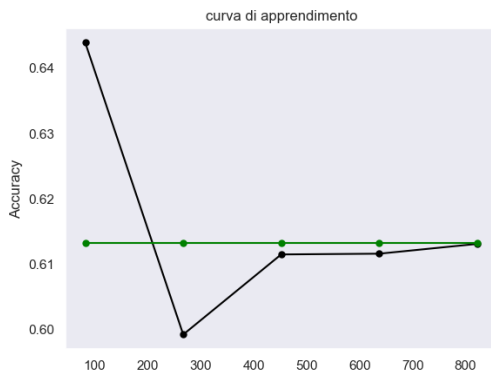
Risultati primo dataset

Nell'analisi della classificazione con la rete neurale bayesiana, notiamo che l'accuratezza massima riscontrata risulta essere ottenuta dando alla rete neurale bayesiana circa 650 esempi. Qui di seguito sono presenti il grafico della learning curve ed il report della rete neurale bayesiana.



REPORT DELLA MIGLIOR RETE NEURALE BAYESSIANA RITROVATA				
	precision	recall	f1-score	support
0	0.55	0.95	0.69	222
1	0.80	0.20	0.32	219
accuracy			0.58	441
macro avg	0.67	0.58	0.51	441
weighted avg	0.67	0.58	0.51	441

Risultati secondo dataset



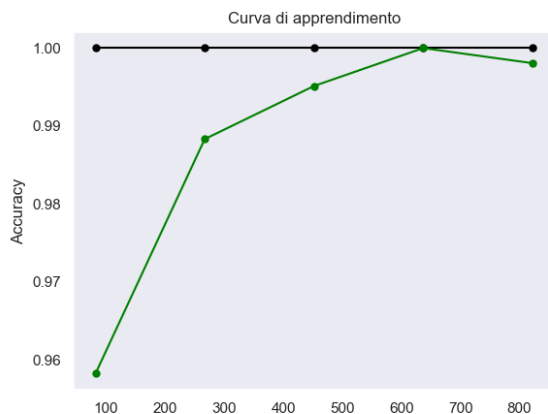
REPORT DELLA MIGLIOR RETE NEURALE BAYESSIANA RITROVATA				
	precision	recall	f1-score	support
0	0.00	0.00	0.00	222
1	0.50	1.00	0.66	219
accuracy			0.50	441
macro avg	0.25	0.50	0.33	441
weighted avg	0.25	0.50	0.33	441

Nel caso della seconda task il modello risulta convergere ad una accuratezza del 50% dopo aver analizzato 800 esempi del train set.

Random forest

Prima task

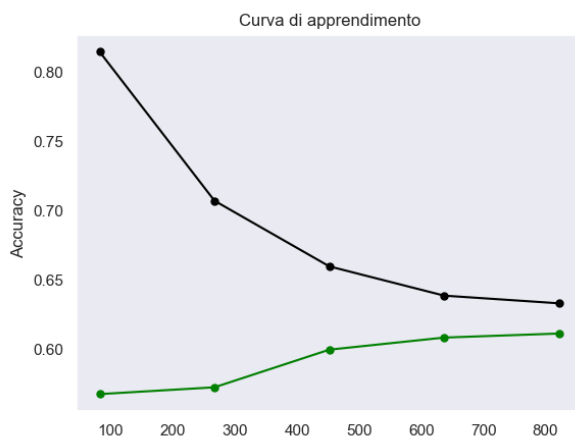
Nello svolgimento della prima task, abbiamo ricercato il numero di estimatori più consono per la random forest. Anche in questo caso andremo a valutare l'accuratezza del modello evidenziando che dopo i primi 600-700 esempi analizzati l'accuratezza risulta essere massima.



REPORT DELLA MIGLIOR RANDOM FOREST				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	221
1	1.00	1.00	1.00	220
accuracy			1.00	441
macro avg	1.00	1.00	1.00	441
weighted avg	1.00	1.00	1.00	441

Seconda task

Per quanto riguarda la seconda task invece il modello risulta avere una accuratezza nettamente inferiore ottenendo la learning curve seguente.



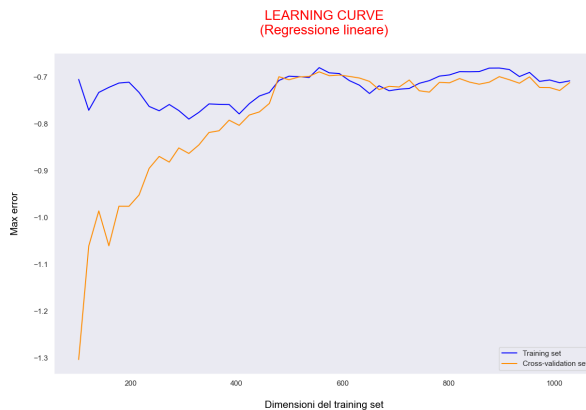
REPORT DELLA MIGLIOR RANDOM FOREST				
	precision	recall	f1-score	support
0.0	0.01	0.67	0.02	3
1.0	1.00	0.50	0.66	438
accuracy			0.50	441
macro avg	0.50	0.58	0.34	441
weighted avg	0.99	0.50	0.66	441

Regressione lineare

Premessa

Abbiamo svolto la regressione lineare soltanto per la seconda task in quanto abbiamo delle variabili continue sul quale poter svolgere la task di regressione.

Risultati secondo dataset



```
R2 Score della Regressione lineare : 0.021967999434606256  
MAE : 0.32772997777980767  
MSE : 0.14111484289373663  
RMSE: 0.3756525560857222
```

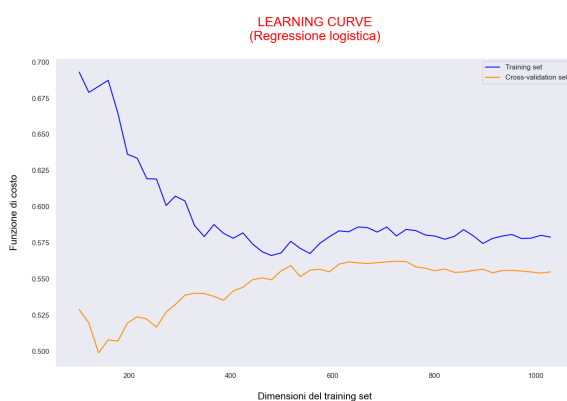
Andando a valutare il grafico e le diverse metriche di valutazione possiamo verificare che la regressione lineare va in underfitting in quanto l'errore del modello risulta alto e astrae dalle features.

Regressione logistica

Premessa

Abbiamo svolto la regressione logistica soltanto per la seconda task in quanto abbiamo delle variabili continue sul quale poter svolgere la task di regressione.

Risultati secondo dataset



```
R2 Score della Regressione logistica : 1.013698630136986  
MAE : 0.5034013605442177  
MSE : 0.5034013605442177  
RMSE: 0.7095078297976829
```

Analizzando il grafico della learning curve per la regressione logistica verifichiamo che il modello risulta essere corretto in quanto le due curve tendono a convergere in funzione della funzione di costo.

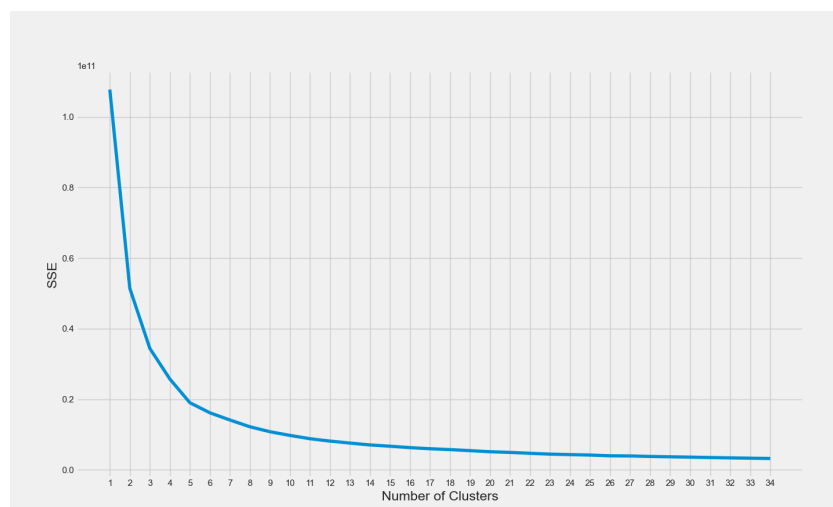
Apprendimento non supervisionato

Clustering

Per effettuare la task di apprendimento non supervisionato abbiamo considerato il dataset intero normalizzando le variabili.

Abbiamo effettuato il clustering del kmeans andando ad aumentare il numero di cluster creati e valutando il decremento della SSE. Incrementando il numero di cluster possiamo notare che il valore della SSE diminuisce fino a stabilizzarsi e, analizzando il grafico riportato di seguito, si nota che per avere un SSE relativamente basso potevamo fermarci ad un numero di cluster pari a 17.

Qui di seguito il grafico rappresentante l'andamento del grafico all'aumentare dei cluster.



Conclusioni

Nello sviluppo del sistema per lo svolgimento delle task designate abbiamo ottenuto dei buoni risultati nello svolgimento della prima task in quanto è stato possibile attraverso le regole designate individuare gli individui idonei alle richieste aziendali e di conseguenza ci ha reso possibile effettuare una classificazione binaria addestrando i diversi modelli e valutandoli di conseguenza.

Nello sviluppo della seconda task invece abbiamo ottenuto dei risultati meno accurati in quanto abbiamo designato una task meno tracciabile con dati, ma dipendenti dalla opinione soggettiva del lavoratore, di conseguenza non esiste un modello in grado di valutare in maniera corretta ed accurata questa task.