IMBALANCE LEARNING

WHEN THE

MACHINE LEARNS THE HARD WAY

<https://github.com/kozen88/ProgettoIcon2023>

Diego Miccoli

Matricola 738735

<mailto:d.miccoli13@studenti.uniba.it>

Anno 2022-23

INTRODUZIONE

Per il Progetto di ingegneria della conoscenza si è scelto di affrontare come principale tematica dello studio il machine learning è in particolare l’attenzione è ricaduta sull’apprendimento supervisionato in contesti ardui. L’idea scaturisce dal data set utilizzato contenete informazioni sui dipendenti di una azienda operante nel settore tecnologico e dalla sua peculiarità di contenere due classi su cui abbiamo basato l’apprendimento dei modelli di classificazione, che sono la classe dei dipendenti che richiedono le dimissioni e la classe dei dipendenti che continuano a lavorare con l’azienda. La problematica del turn-over dei dipendenti è un problema che inevitabilmente colpisce qualsiasi azienda, indipendentemente dal contesto di lavoro, dal settore specifico e dalle dimensioni e importanze dell’azienda stessa, per cui riuscire ad avere una stima di quanti dipendenti in un dato momento possano lasciare l’azienda è un ottima base di partenza per le aziende che dovranno gestire le loro risorse umane e fare fronte alle carenze di staff, inoltre permetterebbe di organizzare campagne di assunzione nei periodi in cui si prevede una perdita eccessiva del personale tale da compromettere il lavoro e le attività commerciali.

Da questo punto di partenza quindi ci è deciso di affrontare un classico problema del machine learning che è di attinenza quotidiana dato che un gran numero di problematiche oggi giorno affrontabili con l’aiuto dell’intelligenza artificiale presentano un problema di squilibrio dei dati tra le classi di appartenenza e questo problema è l’imbalance learning.

IMBALANCE LEARNING

L'apprendimento con sbilanciamento di classi, più conosciuto in ambito informatico con il nome inglese di "imbalance learning" o "class imbalance learning," è un'area di ricerca nell'apprendimento automatico la quale si occupa del problema della gestione dell’apprendimento da parte di un agente intelligente delle classi di un dataset le quali non sono distribuite in modo uniforme, cioè alcune classi hanno un numero molto maggiore di esempi rispetto ad altre. Questo sbilanciamento rende difficile l'addestramento di modelli di machine learning in quanto il modello tende a favorire la classe più numerosa a scapito delle classi minoritarie, questo perché durante la fase di training il modello vedrà e imparerà a riconosce meglio e gestire le caratteristiche della classe maggioritaria poiché di essa vede più esempi comparati a quella minoritaria.

Nonostante queste difficoltà il problema è attualissimo in numerosi contesti in cui la capacità da parte di un agente intelligente di riuscire a discernere la classe minoritaria da quella maggioritaria con una buona percentuale di successo permetterebbe di risolvere molte delle sfide con le quali oggi l’intelligenza artificiale si confronta e che affliggono l’uomo e la società. Alcuni esempi di tale problematica sono:

* Frodi delle transazioni bancarie
* Previsione di attacchi hacker
* Diagnosi medica
* Sorveglianza dei processi di produzione aziendale
* Classificazione di mail spam
* Previsione di terremoti
* Rivelamento di anomalie nella rete

Questi sono solo alcuni delle problematiche che sono caratterizzate da uno sbilanciamento dei dati, ovvero per affrontare questi problemi i dati esistono e si raccolgono, ma il problema e che gli eventi per così dire positivi verso cui siamo interessati al riconoscimento sono raccolti in quantità inferiori poiché nella realtà quotidiana questi eventi si verificano sporadicamente rispetto alle situazioni opposte viste come di normalità.

TECNICHE PER AFFRONTARE

L’IMBALANCE LEARNING

Per aiutare un agente intelligente ad apprendere nel migliore dei modi evitando di apprendere solo la classificazione della classe minoritaria sono state sviluppate diverse tecniche le quali sono state affinate con il passare degli anni e con l’arrivo di tecnologie sempre più performanti. Nonostante questo, le principali tecniche prevedono:

* COST-SENSITIVE LEARNING
* UNDER SAMPLING
* OVER SAMPLING
* APPROCCIO MISTO DI CAMPIONAMENTO
* ENSEMBLE LEARNIG

Ogni uno delle precedenti tecniche affronta il problema in modo diverso, ma tutte hanno il medesimo obiettivo, ottenere modelli di machine learning più equilibrati e capaci di riconoscere correttamente anche le classi minoritarie.

SCOPO DELLO STUDIO

Il nostro studio si focalizzerà sul mettere in atto tali tecniche e valutare criticamente i risultati ottenuti, essendo conscenti che con gli esperimenti che andremo a condurre non si ha l’obiettivo di andare a creare un nuovo modello commerciale di intelligenza artificiale che sia capace di risolvere il problema citato precedentemente e che quindi possa essere distribuito sul mercato. Il focus dello studio sarà invece capire come applicare nella maniera più corretta possibile in base alle conoscenze attuali nel campo del machine learning le tecniche con le quali affrontare il problema dell’imbalance learning e valutare in modo critico e ponderato i risultati che si otterranno.

IL DATASET UTILIZZATO

Il dataset che abbiamo utilizzato per condurre lo studio è stato preso da un noto sito di intelligenza artificiale <https://www.intelligenzaartificialeitalia.net/> ed è caratterizzato dall’avere un totale di dieci features descrittive di cui:

* Features discrete: numero progetti, ore medie mensili, tempo speso in azienda, incidenti sul lavoro, promozione negli ultimi 5 anni e richiesta di licenziamento.
* Features continue: livello di soddisfacimento e ultima valutazione.
* Features categoriche: salario e dipartimento.

La feature target è richiesta di licenziamento. Una volta analizzato e capito a cosa le variabili si riferissero e le scale di misurazione utilizzate per raccogliere tali dati si è cercato di andare più nel dettaglio delle caratteristiche del dataset in cerca di problematiche da risolvere prima di affrontare l’apprendimento dei modelli.

DATA EXPLORING & VISUALIZATION

Le prime operazioni effettuate sono state utilizzate a comprendere meglio la natura e la distribuzione dei dati con i quali stiamo lavorano e hanno portato ai seguenti risultati:

A graph with numbers and a bar

Description automatically generated

A pie chart with numbers and a number of percentages

Description automatically generated

Da questi immagini capiamo che il nostro data set soffre di uno sbilanciamento che ha all’incirca un rapporto 3:1 a sfavore della classe minoritario ovvero gli impiegati che lasciano il posto di lavoro.

A screenshot of a computer screen

Description automatically generated

Dallo studio del pairplot ovvero la distribuzione degli esempi visti su di uno spazio cartesiano che prende in considerazione solo due features per volta per visualizzare come in base a tali features gli esempi si distribuiscano nello spazio cartesiano abbiamo appreso che ci sono feature che riescono a distribuire gli esempi in determinate zone dello spazio delle features e per tali ragioni crediamo che ci siano buone possibilità per riuscire a classificare gli esempi in base alle due classi di appartenenza.

A screenshot of a graph

Description automatically generated

Usando il coefficiente di correlazione lineare di Pearson abbiamo dato conferma di quanto presupposto in precedenza ovvero che il problema di classificazione sia effettivamente affrontabile poiché ci sono features che riescono a clusterizzare gli esempi comuni alla stessa classe. Tali features sono quelle che mostrano una maggiore correlazione lineare nella matrice anche se c’è da dire che la correlazione e lieve-moderata.

Andiamo a dare un occhiata più vicino alle statistiche per ogni una delle features che stiamo utilizzando:A group of graphs with different colored bars

Description automatically generated with medium confidence

Dalle quali notiamo che le promozioni in azienda sono più uniche che rare e che stranamente c’è un significativo anche se non enorme livello di infortuni sul lavoro. Capiamo anche che la maggior parte dei dipendenti dell’azienda lavora in pianta stabile da tre anni e che passati i cinque anni i dipendenti della azienda fissi diminuiscono drasticamente cosa che fa pensare ad un abbastanza rapido turn over dei dipendenti. Per la gestione del carico lavorativo sembra essere più o meno ripartito equamente dal numero di progetti che vengono assegnati ad ogni dipendente.

A graph of a box with a blue rectangle

Description automatically generatedA graph of a box

Description automatically generated

A graph of a box

Description automatically generated

I boxplot delle variabili continue ci hanno permesso di capire la distribuzione degli esempi del dataset e le seguenti solo le principali statistiche raccolte:

A screenshot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated A screen shot of a computer

Description automatically generated

Di seguito riportiamo le statistiche generali sulle variabili numeriche presenti nel dataset:

A screen shot of a computer

Description automatically generated

DATA CLEANNING

Dopo aver preso visione delle principali caratteristiche del dataset passiamo al verifica dell’integrità dei dati i quali non presentano eccessivi problemi, gli esempi contenenti troppi campi mancanti sono stati scartati. Un problema riscontrato riguardava l’attributo categorico del dipartimento il quale presentava valori che potevano essere accorpati tutti sotto lo stesso valore categorico per cui sono state apportate queste modifiche. Infine, è stata effettuata un analisi degli outliers ovvero degli elementi che potrebbero influenzare negativamente le fasi dell’apprendimento. Per rimuoverli è stato usato un algoritmo basato su IQR l'Interquartile Range il quale è una misura statistica utilizzata per identificare e gestire gli outlier nei dati. L'IQR è calcolato come la differenza tra il terzo quartile (Q3) e il primo quartile (Q1) di un insieme di dati. Permette di rimuovere dati eccedenti dai range delle distribuzioni di appartenenza.

A diagram of a project

Description automatically generated with medium confidence

DATA PREPROCESSING

Ai dati ripuliti si è applicato la trasformazione di alcune features tramite one hot encoding, il processo con il quale da una feature qualitativa né otteniamo una quantitativa e questo processo è stato apportato sia a dipartimento che a salario provocando l’aumento delle features per espansione. Una volta terminato questo processo poiché abbiamo scelto di lavorare con modelli lineari per supportarne l’apprendimento abbiamo standardizzato tutte le features di modo che fossero tutte tra di loro comparabile e che la magnitudine di una non avrebbe prevalso sul calcolo dei pesi durante le fasi di apprendimento degli algoritmi.

COST SENSITIVE VERSUS COST INSENSITIVE

Siamo giunti all’applicazione della prima tecnica per il trattamento dell’imbalance learning. Abbiamo deciso di prendere due modelli lineari e di confrontarli tra di loro la scelta è stata influenzata dalla libreria di scikit-lerarn con la quale abbiamo operato poiché non tutti i modelli di apprendimento supervisionati proposti da tale libreria offrono l’opzione dell’apprendimento tramite cost sensitive che consiste nell’associare un peso differente alle due classi per portare a termine l’addestramento. Per tali ragioni la nostra scelta è ricaduta sullo Stochastic Gradient Descent che sarà messo a confronto con il Percettrone per capire chi tra i due riesca meglio ad affrontare le problematiche dello sbilanciamento. Per effettuare lo studio prima si è posta una base di partenza data dall’apprendimento cost insensitive e successivamente si è valutato il cost sensitive.

COST INSENSITIVE LEARNING

L’obiettivo prevede lo studio dei modelli nelle condizioni peggiori ovvero senza alcun supporto per capire come si comportano e le difficoltà da essi riscontrate. Inizieremo valutando la qualità dell’apprendimento studiato con le curve di apprendimento in base a delle metriche poiché ci troviamo difronte ad un contesto sbilanciato l’accurancy poco si addice per la valutazione e per cui le nostre scelte sono state affidate principalmente al recall dato che ci interessa sapere in modo esaustivo quanti dipendenti decideranno di dare le dimissioni e in seguito abbiamo valutato anche la precision.

Curve di apprendimento a confronto:

curva di recall dello SGDC:

A graph showing the number of numbers

Description automatically generated with medium confidence

Curva di recall del Percettrone

A graph showing the amount of data

Description automatically generated with medium confidence

Curva di precision dello SGDC

A graph showing the number of numbers

Description automatically generated with medium confidence

Curva di precision del Percettrone

A graph showing a number of numbers

Description automatically generated

Dalle forti oscillazioni possiamo notare come entrambi i modelli abbiano avuto difficoltà del apprendere i concetti delle due classi e riuscire in seguito a generalizzare sui futuri esempi e di fatto non riescono a stabilizzare ad un valore accurato nessuna delle due metriche con le quali si è valutato l’addestramento.

Vediamo adesso le loro performance valutate su di un quadro di classificazione che raccolga le principali metriche con le quali si valutano i classificatori:

A screenshot of a computer

Description automatically generated A screen shot of a graph

Description automatically generated

Le differenze non sono eccessive, ma lo SGDC risulta più performante, cerchiamo di prendere visione in maniera più chiara e facilitata.

A chart with red and white squares

Description automatically generated with medium confidence

A graph with red squares

Description automatically generated with medium confidence

Infine, andiamo a metterli a confronto su di un grafico che mostri il Receiver operator characteristic e ne misuri l’area sottostante:

A graph showing a number of different colored lines

Description automatically generated with medium confidence

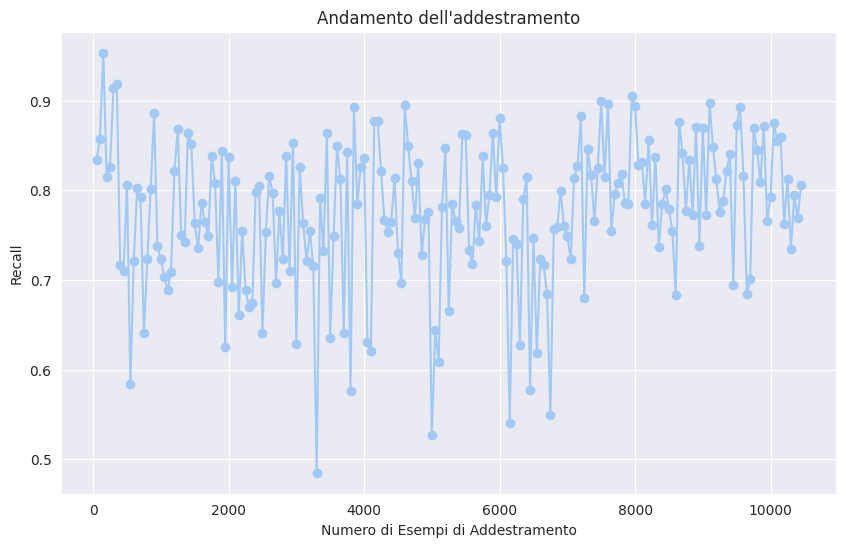
Da gli esperimenti condotti abbiamo capito che in assenza di un supporto esterno dato ai modelli essi hanno difficoltà a trattare lo sbilanciamento, ad ogni modo sono riusciti a completare l’addestramento anche se non nel modo ottimale e possiamo concludere che in tali condizioni lo SGDC è stato più performante del Percettrone.

COST SENSITIVE LEARNING

Procediamo ad applicare l’apprendimento con pesi per la gestione dello sbilanciamento. L’idea è assegnare un bonus sugli esempi classificati correttamente delle classe minoritaria e un malus quando la classificazione è fallace, ad ogni modo questo processo è effettuato per entrambe le classi quello che cambia è il peso il quale sarà più importante e quindi porterà a maggiori cambiamenti, per la classe minoritaria, mentre sarà meno incisivo per la classe maggioritaria. Spiegato il funzionamento del cost sensitive learning ci soffermiamo sul fatto che la libreria scikit-learn offre differenti opzioni per la gestione del perso quali la specifica del peso assegnato quando si ha della conoscenza del dominio da parte dell’esperto da sfruttare oppure il calcolo del peso commisurato agli esempi appartenenti a ciascuna delle classi ovvero calcolato in base alla proporzione dello sbilanciamento più sarà marcato lo sbilanciamento e più il peso sarà inciso per la classe minoritaria e meno influente per la maggioritaria. Questo approccio è quello che ci è sembrato più sensato ed è stato quello scelto per l’addestramento con cost sensitive.

Andiamo a prendere visione delle curve di apprendimento dei modelli sottoposti ad apprendimento con cost sensitive:

curva di recall per lo SGDC



Curva di recall per il percettrone

A graph showing the number of numbers

Description automatically generated with medium confidence

Curva di precision per lo SGDC:

A graph showing a number of dots

Description automatically generated with medium confidence

Curva di precision per il percettrone:

A graph showing a number of numbers

Description automatically generated with medium confidence

Dalle curve notiamo che anche in questo caso ci sono stati dei problemi di apprendimento per i due modelli, ma le oscillazioni sono diventate meno ampie rispetto alle curve di apprendimento senza i pesi applicati, inoltre in particolare la precision per lo SGDC sembra stabilizzarsi abbastanza si dà subito.

Adesso andiamo a prendere atto del classification report che ci darà maggiori informazione sulle performance di questi due classificatori:

A screenshot of a computer

Description automatically generated A screen shot of a graph

Description automatically generated

Infine, andiamo a dare una prospettiva in termini di percentuali con le matrici di confusione e poi mostreremo il confronto con il roc score.

A red and white chart

Description automatically generated

A screenshot of a graph

Description automatically generated

A graph showing a line graph

Description automatically generated with medium confidence

Come mostrato dalle metriche prodotte dopo l’addestramento di questi due modelli, anche in questo caso lo SGDC è risultato essere il modello più performante che sovrasta il percettrone su ogni metrica. Quello che abbiamo notato ad ogni modo è un lieve incremento delle performance dovuto al cost sensitive learning per tale ragione anche se ci aspettavano un margine di miglioramento superiore possiamo dire che il cost sensitive ha sicuramente un impatto positivo sull’apprendimento di questi modelli come è possibile vedere nel sottostante grafico:

A graph of a graph showing the results of a long line

Description automatically generated with medium confidence

TECNICHE DI CAMPIONAMENTO

Adesso spostiamo la nostra attenzione sull’applicazione delle tecniche di campionamento esse sfruttano l’idea base della statistica nel campionare opportunamente gli esempi dal dataset sbilanciato con l’obiettivo di ribilanciare il più possibile le classi. Per cui quello che queste tecniche offrono ai nostri modelli è il passaggio da un problema di imbalance learning a un semplice problema di apprendimento per cui il loro obbiettivo è smussare le difficoltà di apprendimento equilibrando lo sbilanciamento e portandolo in condizioni standard. L’idea è allettante, ma questo supporto che offrono a seconda della tecnica che si sceglie ha un prezzo da pagare che chiaramente incide sull’apprendimento dei modelli. Le tecniche di campionamento si dividono in tre gruppi che sono under sampling, over sampling e approccio misto per ogni gruppo troviamo numerosi approcci che affrontano i problemi dello sbilanciamento. Per il nostro studio abbiamo scelto un approccio per ogni uno dei tre gruppi di tecniche e lo abbiamo fatto in modo da diversificare il più possibile i nostri risultati.

Nello studio delle tecniche di campionamento per gestire lo sbilanciamento il nostro interesse si sposta su ulteriori due modelli e questa volta andremo a confrontare le loro performance con le tre tecniche di campionamento e i modelli scelti sono la Regressione Logistica e il Gaussian Naive Bayes.

RANDOM UNDER SAMPLING

Il random under sampling RUS è una tecnica di under sampling che si basa su un idea semplice e banale, ovvero andare a campionare dalla classe maggioritaria esempi casualmente per scartarli e andare a bilanciare il numero degli esempi tra le due classi. L’aspetto che viene ad essere garantito è il bilanciamento perfetto tra le due classi, ma ad un prezzo alto perché stiamo perdendo informazione utile, cioè, stiamo perdendo l’informazione contenuta nei rappresentanti della classe maggioritaria, inoltre stiamo buttando deliberatamente i dati raccolti con sacrifici e tempo. Per queste ragioni è una tecnica che non viene raccomandata se non in quei casi in cui abbiamo un quantità enorme di dati. Di contro abbiamo scelto questa tecnica anziché le più sofisticate ed evolute ENN edited Nearest Neighbours o il TomeK links, poiché queste tecniche sono usate negli approcci misti o, meglio, negli algoritmi più performanti degli approcci misti e siccome è nostra intensione andare ad usarne uno tra questi ci siamo limitati al RUS. La differenza tra il RUS e gli altri due è che anziché andare a rimuovere esempi casualmente sfruttano qualche inventiva in più per eliminare esempi della classe maggioritaria che non sono fortemente rappresentativi o che risultano essere outliers. Di seguito riportiamo i risultati ottenuti con l’addestramento dopo l’applicazione del RUS.

A graph with a number of data

Description automatically generated with medium confidence

A pie chart with numbers and text

Description automatically generated

Come precedentemente accennato il RUS è capace di riequilibrare la situazione.

Curva di accuracy della regressione logistica

A graph showing the growth of a stock market

Description automatically generated

Curva di accuracy del gaussian naive bayes

A graph showing a line

Description automatically generated

Curva di recall della regressione logistica

A graph showing the growth of a stock market

Description automatically generated

Curva di recall del gaussian naive bayes

A graph showing a line

Description automatically generated

Curva di precision della regressione logistica

A graph showing a line graph

Description automatically generated with medium confidence

Curva di precision del gaussian naive bayes

A graph showing the growth of a stock market

Description automatically generated

Anche in questo caso l’apprendimento mostra difficoltà le quali sono mostrate dalle oscillazioni che caratterizzano le curve, nonostante questo ci evince come il gaussian naive bayes risenta meno delle problematiche dovute all’apprendimento andando ad ottenere dei risultati di recall interessanti e soddisfacenti.

Andiamo a mettere a confronto i due classificatore sulle metriche generalmente usate per la classificazione.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Prendiamo visione delle matrici di confusione associate ai modelli:

A red squares with white text

Description automatically generated

A screenshot of a graph

Description automatically generated

Infine, mettiamo i classificatori a confronto su di un grafo con score ROC:

A graph of a number of data

Description automatically generated with medium confidence

Dal metriche calcolate il confronto non è semplice e lascia margine all’incertezza, tuttavia considerando il problema di partenza e visto che se fossimo a capo del direttivo delle risorse umane non vorremmo rimanere senza dipendenti si accetta di rischiare falsi allarmismi da parte del gaussian naive bayes sapendo di restare sul lato sicuro del rischio.

ADAPTIVE SYNTETHIC SAMPLING

Adesso il nostro studio si focalizza su una tecnica di over sampling ovvero l’ADASYN adaptive syntethic sampling. L’idea di questo algoritmo è quella di andare a creare dei dati fittizi ovvero finti che abbiano caratteristiche simili a quelle dei dati degli esempi della classe minoritaria per farlo si basa sul calcolo della densità delle classi nel dataset. La densità di una classe è una misura della distribuzione dei suoi esempi nello spazio delle feature e le classi che presentano una densità molto bassa indicano che sono classi minoritarie e potenzialmente soggette a problemi di sbilanciamento una volta che vengono individuate per ogni esempio nella classe minoritaria, ADASYN calcola il grado di disomogeneità rispetto alle classi circostanti. Il grado di disomogeneità misura quanto un esempio sia diverso dalle sue vicine, basandosi sulla densità delle classi. Una volta ottenute le misure di densità e di disomogeneità ADASYN si concentra sulla generazione di esempi sintetici per le istanze delle classi minoritarie che hanno un alto grado di disomogeneità ovvero che sono il più possibile differenti dagli esempi della classe maggioritaria per caratteristiche. Questi esempi sintetici vengono generati mediante interpolazione tra gli esempi originali, con un peso maggiore assegnato agli esempi che hanno un grado di disomogeneità più alto. Questa tecnica a differenza della precedente non riesce a garantire il perfetto ribilanciamento del dataset, ma ad ogni modo riesce quasi del tutto a risanarlo tranne nei casi in cui gli esempi della classe minoritaria sono sparsi e immersi in esempi di classe maggioritaria. Ancora una volta questo bilanciamento per riportare l’apprendimento in condizioni normali è pagato con l’immissione di rumore o, meglio, di informazione di fittizia non del tutto veritiera la quale andrà ad influenzare in maniera negativa l’apprendimento, ciononostante in letteratura scientifica risulta essere una tecnica da preferire al sotto campionamento.

Passiamo a vedere l’apprendimento dei modelli e le loro performance:

Curva di accuracy per regressione logistica

A graph showing a line

Description automatically generated

Curva di accuracy per gaussian naive bayes

A graph showing a line

Description automatically generated

Curva di recall per la regressione logistica

A graph showing the growth of a stock market

Description automatically generated

Curva di recall per il gaussian naive bayes

A graph showing a line

Description automatically generated

Curva di precision per la regressione logistica

A graph showing a line

Description automatically generated

Curva di precisione per il gaussian naive bayes

A graph showing a line

Description automatically generated

Andiamo a confrontare le restanti metriche e comparare il report di classificazione per i due modelli:

A screenshot of a computer screen

Description automatically generated A screen shot of a black screen

Description automatically generated

Riportiamo di seguito le matrici di confusione inerenti ai modelli:

A graph with red squares

Description automatically generated

A red squares with white text

Description automatically generated

Infine, andiamo a confrontare i due classificatore con score ROC su di un grafico:

A graph of a number of objects

Description automatically generated with medium confidence

A discapito di quanto notato dalle curve di apprendimento il gaussian naive bayes performa peggio della regressione logistica che risulta essere il modello vincente dopo aver trattato il dataset con ADASYN.

Synthetic Minority Over-sampling Technique Edited Nearest Neighbors

Q

Passiamo a vedere l’apprendimento dei modelli e le loro performance:

Curva di accuracy della regressione logistica

A graph showing a line

Description automatically generated

Curva da accuracy del gaussian naive bayes

A graph showing a line

Description automatically generated

Curva di recall della regressione logistica

A graph showing a line

Description automatically generated

Curva di recall del gaussian naive bayes

A graph showing the growth of a stock market

Description automatically generated

Curva di precision della regressione logistica

A graph showing the growth of a stock market

Description automatically generated

Curva di precision del gaussian naive bayes

A graph showing a line

Description automatically generated

Andiamo a confrontare le restanti metriche e comparare il report di classificazione per i due modelli:

A screenshot of a computer screen

Description automatically generated A screen shot of a computer

Description automatically generated

Riportiamo di seguito le matrici di confusione inerenti ai modelli:

A screenshot of a graph

Description automatically generated

A chart with red squares

Description automatically generated with medium confidence

Infine, andiamo a confrontare i due classificatore con score ROC su di un grafico:

A graph showing the number of indicators

Description automatically generated with medium confidence

Confronto finale sulle tre tecniche di campionamento utilizzate:

A graph showing different colored lines

Description automatically generated