UGotTheJob

Artificial Intelligence DOCUMENT

"Dipartimento di Informatica anno 2022/2023"

"Professore: Fabio Palomba"

Autori	Matricola
Giulio Incoronato	0512111363
Antonio Mazzarella	0512112830

Contents

1 Introduzione

Quante volte hai avuto l'ansia di essere preso o pure no in uno specifico lavoro? Quante volte ti sei domandato se fossi giusto tu per quel lavoro? Con la fine del proprio percorso di studio ci si pongono tante domande e dubbi se si viene presi in un determinato lavoro oppure no.

Tutto questo sorge perchè dopo diversi anni di studio si vuole avere la sicurezza di essere presi per il lavoro dei propri sogni. Sarebbe utile avere un tool in grado di prevedere, attraverso dei dati, quantà probabilità hai di avere il lavoro.

Il nostro team mira a combattere tutte queste ansie creando un tool chiamato **"UGotTheJob"** che integrerà un modello di machine learning supervisionato che andrà a prevedere la possibilità di essere piazzato.

1.1 Link Utili

1. Questo è il link alla repository ufficiale di **UGotTheJob**: Link
2. Questo è il link dove abbiamo preso i dataset: Link
3. Qui è dove è stata presa l'icona del nostro tool: Link

2 Specifiche P.E.A.S.

Performance	Capacità dell'agente di prevedere se l'utente sarà preso o meno per un lavoro.
Enviroment	L'ambiente in cui l'agente opera rappresentato da un form di cui l'utente scriverà i dati necessari.
Actuators	Interfaccia utente dell'applicazione dove uscirà il valore predetto.
Sensors	Form nell'interfaccia utente.

Table 1: Tabella PEAS

2.1 Proprietà dell'Ambiente

L'ambiente possiede le seguenti proprietà:

- **Completamente osservabile:** l'agente ha accesso completo a tutte le informazioni fornite dall'utente.
- **Deterministico:** lo stato dell'ambiente dipende dall'azione intrapresa dall'agente.
- **Sequenziale:** le decisioni dell'agente dipendono dagli input dell'utente.
- **Statico:** nel momento in cui l'agente sta elaborando la sua previsione l'utente non può modificare il form dato in partenza.
- **Discreto:** le previsioni dell'agente dipendono soprattutto dagli input inseriti dall'utente, oltretutto c'è un numero limitato e preciso di informazioni che l'utente può inserire.
- **Singolo-agente:** esiste solo un agente che opera nell'ambiente.

3 Machine Learning

Il machine learning (apprendimento automatico) è una tecnologia dell'intelligenza artificiale che consente alle macchine di imparare dai dati, senza essere esplicitamente programmate. In altre parole, il machine learning si basa sulla costruzione di algoritmi che possono imparare da un insieme di dati e migliorare la loro capacità di risolvere compiti specifici con l'esperienza.

Ci sono tre tipi principali di apprendimento automatico:

- **Apprendimento supervisionato:** in questo tipo di apprendimento, il modello è addestrato su un insieme di dati che includono sia le caratteristiche di input che le relative etichette di output. Il modello usa queste etichette per adattarsi ai dati di input e fare previsioni su dati simili.
- **Apprendimento non supervisionato:** in questo tipo di apprendimento, il modello è addestrato su un insieme di dati senza etichette di output. Il modello cerca di scoprire pattern o strutture nei dati di input.
- **Apprendimento per rinforzo:** in questo tipo di apprendimento, il modello impara attraverso l'interazione con un ambiente dinamico. Il modello prende decisioni in base allo stato attuale dell'ambiente e riceve feedback sulle sue azioni.

Il machine learning viene utilizzato in molte applicazioni, tra cui la classificazione di immagini, la traduzione automatica, la diagnosi medica, la rilevazione di frodi e molto altro ancora. Per il nostro tool abbiamo utilizzato un algoritmo di machine learning ad apprendimento supervisionato perché andremo a risolvere un problema di **classificazione**.

4 CRISP-DM

Per progettare una soluzione basata su machine learning bisogna avere un approccio **data and software engeneering**. Per la creazione di tale software abbiamo utilizzato il modello *CRISP-DM* (CRISP-DM è l'acronimo di Cross-Industry Standard Process for Data Mining.), che rappresenta il ciclo di vita di progetti basati su intelligenza artificiale e data science.

Possiamo paragonare il modello *CRISP-DM* ad un modello a cascata con feedback utilizzato per lo sviluppo di sistemi software tradizionali. Presenta anche un modello **non sequenziale** in cui le diverse fasi possono essere eseguite un numero illimitato di volte. Esistono diverse fasi raffigurate nell'immagine di seguito (Immagine 1):

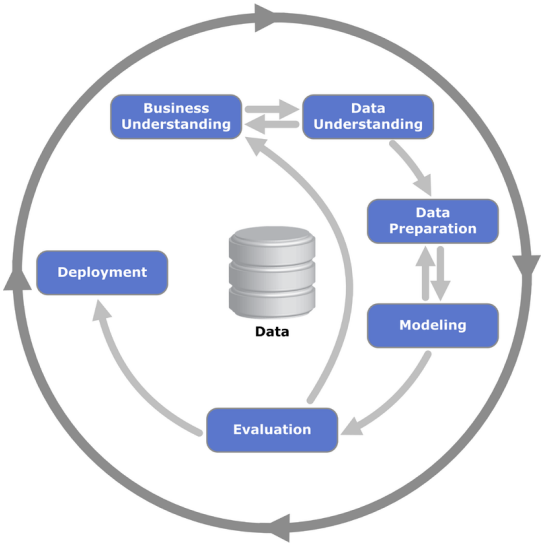


Immagine 1

4.1 Business Understanding

In questa fase si raccolgono e definiscono gli obiettivi di Business che si vogliono raggiungere, oltre a determinare la disponibilità delle risorse, stimare i rischi, indicare tecnologie e gli strumenti utilizzati per raggiungere gli obiettibi prefissati.

- **Obiettivi di Business:** L'obiettivo principale di **UGotTheJob** è la realizzazione di un tool con cui l'utente interagisce inserendo dei dati richiesti in partenza sul suo percorso di studi, il tutto verrà analizzato e processato per poi dare in output la probabilità di essere piazzati o non.
- **Disponibilità delle risorse:** La risorsa che utilizzeremo per il nostro software sarà un dataset che conterrà le informazioni sui collocamenti in base ai vari percorsi di studio e esperienze pregresse. Per reperire questo dataset utilizzeremo una piattaforma importante che è Kaggle.
- **Stima dei rischi:** I rischi che incontreremo saranno di tipo perlopiù Etico/Morale in quanto il dataset non fornisce una bilanciata percentuale di dati ad esempio tra persone di sesso differente.
- **Tecnologie e Strumenti:** Per analizzare, acquisire e modellare il dataset utilizzeremo il linguaggio *Python* che presenta alcune librerie come **Pandas**, **sklearn**, **seaborn** ed etc.

4.2 Data Understanding

Come già discusso nella *Stima dei rischi (parag. 4.1 Business Understanding)* il problema da noi riscontrato è stata la poca imparzialità che l'agente potesse avere con il dataset da noi utilizzato. Il dataset avendo un discreto bilanciamento dei dati relativi al gender *(vedi Figure 1)*, avrebbe portato al nostro agente una poco corretta previsione del piazzamento di una persona, rischiando quindi di cadere in una discriminazione di tipo Etico/Morale di gender.

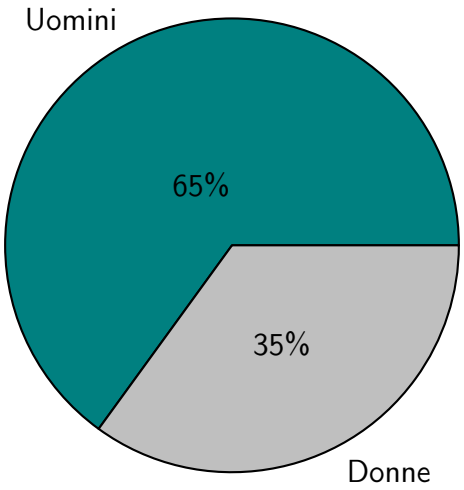


Figure 1: Gender Dataset

Il dataset inoltre presenta dati, come voti o specializzazioni, che non sono inerenti all'ambiente italiano. Possiamo vedere una lista con le descrizioni delle singole feature presenti:

- **Gender**: Indica appunto il *sesso* della persona (M o F).
- **SSC Percentage**: Si riferisce generalmente all'esame di fine anno in India o in altri paesi. Indica una percentuale di voti ottenuti dallo studente in quel determinato esame. Sarebbe l'equivalente di un esame di **scuola media**.

- **HSC Percentage**: Si riferisce generalmente all'esame di fine anno in India o in altri paesi. Indica una percentuale di voti ottenuti dallo studente in quel determinato esame. Sarebbe l'equivalente di un esame di **scuola superiore**.

- **SSC Board**: Si riferisce generalmente al consiglio o all'ente che organizza l'esame di fine anno delle **scuole medie** in India.

  Col valore *Central* si riferisce al CBSE (Central Board of Secondary Education), che sarebbe un organismo nazionale che organizza esami standardizzati per scuole pubbliche e private in India.

  Con Other si riferisce a consigli Statali o regionali.

- **HSC Board**: Si riferisce generalmente al consiglio o all'ente che organizza l'esame di fine anno delle **scuole superiori** in India.

  Col valore *Central* si riferisce al CBSE (Central Board of Secondary Education), che sarebbe un organismo nazionale che organizza esami standardizzati per scuole pubbliche e private in India.

  Con Other si riferisce a consigli Statali o regionali.

- **HSC Subject**: Si riferisce alle materie che gli studenti devono studiare e superare per completare l'esame di fine anno dell'ultimo anno di **scuole superiori**.

  Questa variabile presenta tre tipi: *Commerce, Science e Arts*.

- **Degree Percentage**: Indica la percentuale di punteggio ottenuta dagli studenti in un programma di Laurea.

- **Undergrad Degree**: È un titolo di studio che gli studenti ottengono dopo aver completato un programma di studi Universitari.

  Ci sono tre valori: *Sci&Tech, Comm&Mgmt e Others*.

- **Work Experience**: Questa variabile, banalmente, rappresenta se il sottoscritto ha avuto o meno esperienza lavorativa pregressa.

- **Employee Test %**: Rappresenta la percentuale del test di idoneità per una posizione di lavoro effettuato presso l'azienda in cui il candidato ha fatto domanda.

- **Specialization**: Rappresenta di che tipo di Specializzazione il candidato è in possesso.

  Questo dataset presenta 2 opzioni: *Mkt&Fin e Mkt&HR*. Rispettivamente sono *Mercato e Finanza* e *Mercato e Risorse Umane* (Human Resource).

- **MBA Percentage**: Indica la percentuale di punteggio calcolata come media di voti di tutto il percorso di studi post-Laurea che si concentra sull'Amministrazione aziendale e sulla Gestione.

- **Status**: Indica, banalmente, se il candidato è piazzato o meno.

Questo ci porta a lavorare per un modello che non potrà essere utilizzato in una realtà italiana.

4.3 Data Preparation

In questa sezione, tratteremo le tecniche adottate per preparare i dati acquisiti in modo che il nostro machine learner non darà problemi e sarà quanto più efficiente possibile.

Il data preparation si articola nei seguenti quattro passaggi:

1. **Data cleaning;**
2. **Feature scaling;**
3. **Feature selection;**
4. **Data balancing;**

4.3.1 Data cleaning

Il *Data Cleaning*, definito come *"Pulizia dei dati"*, si occupa di rimediare a problemi quando ci sono righe di dati mancanti ma più in generale ha come obiettivo quello di fornire un dataset dotato di una qualità adeguata. Nel nostro dataset non sono presenti dati mancanti e di conseguenza non abbiamo effettuato la fase di *Data Imputation*.

4.3.2 Feature scaling

Il *Feature Scaling* è l'insieme di tecniche che consentono di normalizzare o scalare un insieme di valori di una caratteristica. Questa tecnica viene eseguita quando abbiamo dei valori estremamente diversi di una determinata caratteristica rispetto ad un altra. Nel nostro caso non abbiamo avuto bisogno di normalizzare o scalare i valori del nostro dataset, in quanto non sono particolarmente diversi tra di loro.

4.3.3 Feature selection

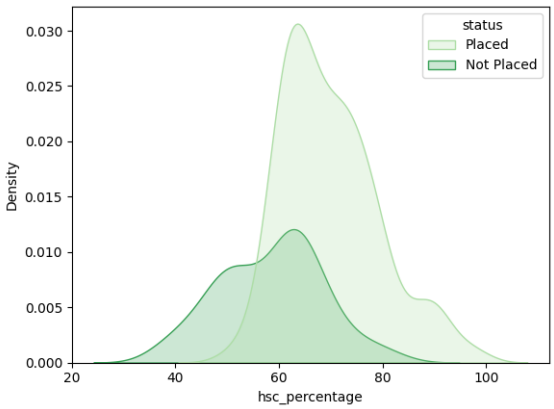
La *Feature selection* rientra nell'ambito del feature engeneering, che sarebbe il processo nel quale il progettista utilizza la propria conoscenza del dominio per determinare le caratteristiche (feature) dai dati grezzi estraibile tramite tecniche di data mining.

Nel nostro caso abbiamo pensato di rimuovere colonne che non erano adeguate per il nostro obiettivo ovvero, quello di creare un software che preveda un piazzamento nel mondo del lavoro quanto più etico possibile.

	gender	ssc_percentage	ssc_board	hsc_percentage	hsc_board	hsc_subject	degree_percentage	undergrad_degree	work_experience	emp_test_percentage	specialisation	mba_percent	status
0	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed
1	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed
2	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed
3	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed
4	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed

Table 2: Esempio dataset.csv

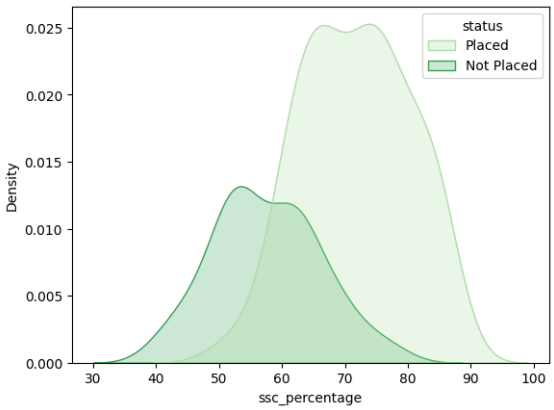
Si può vedere dalla *Table 2* il dataset che abbiamo scelto per il nostro progetto. Analizziamo l'influenza di ogni singola feature di questo dataset:



HSC Percentage

Da questo grafico possiamo vedere che la variabile *HSC Percentage* influisce nel dataset, perchè all'aumentare del valore (voto) può incidere sulla previsione del modello.

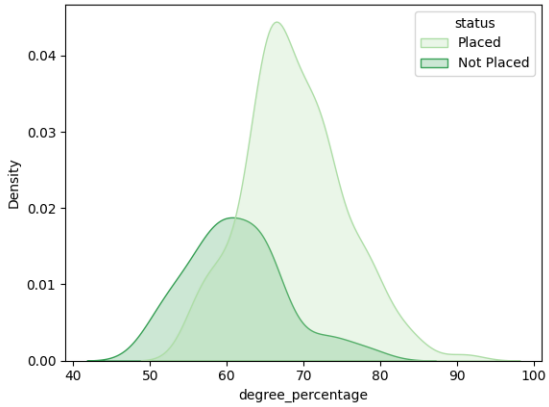
Da questa considerazione abbiamo deciso di non eliminare questa variabile.



SSC Percentage

Da questo grafico si può notare che la variabile *SSC Percentage* influisce notevolmente nel dataset perchè come abbiamo visto con la variabile *HSC Percentage*, all'aumentare del valore aumenta anche la possibilità di essere piazzati.

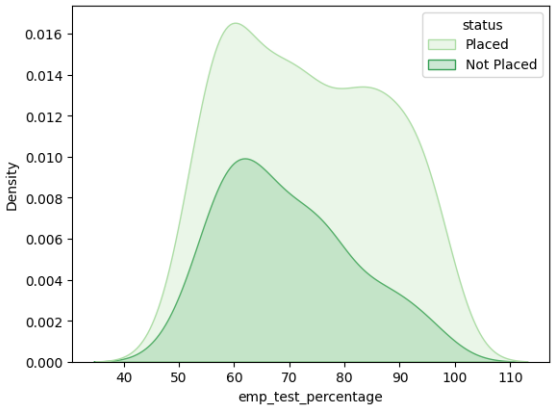
Da questa considerazione abbiamo deciso di non eliminare questa variabile.



Degree Percentage

Da questo grafico si può notare che la variabile *Degree Percentage* influisce nel dataset, perchè all'aumentare del valore aumenta anche la possibilità di essere piazzati.

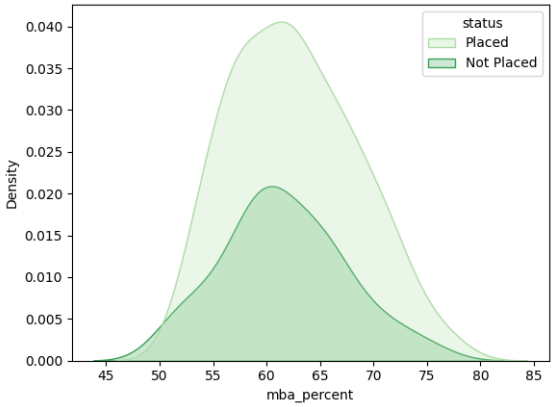
Da questa considerazione abbiamo deciso di non rimuovere questa variabile.



EmpTest Percentage

Come si nota da questo grafico la variabile *EmpTest Percentage* influisce di poco nel dataset, perchè all'aumentare del valore aumenta leggermente la possibilità di essere presi e non.

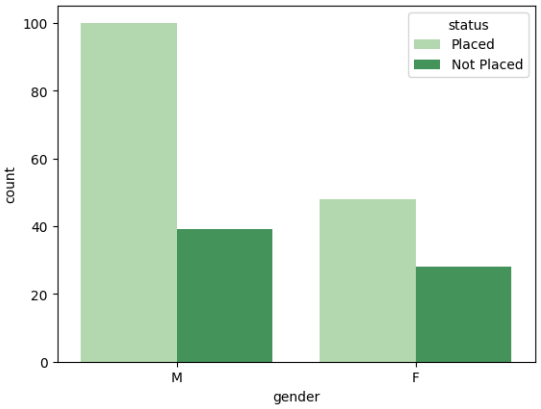
Da questa considerazione abbiamo deciso di rimuoverla.



MBA Percentage

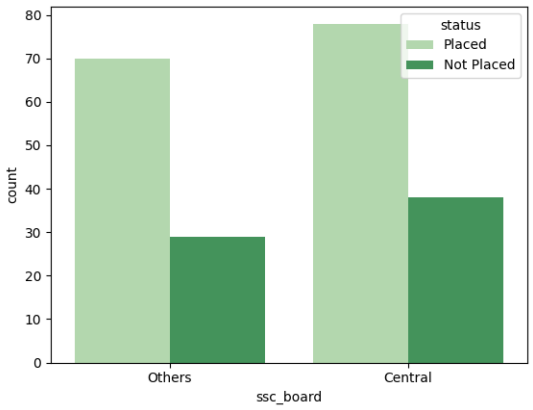
Come si nota da questo grafico la variabile *MBA Percentage* influisce di poco nel dataset, perchè all'aumentare del valore aumenta leggermente la possibilità di essere presi o non.

Da questa considerazione abbiamo deciso di rimuovere la variabile.



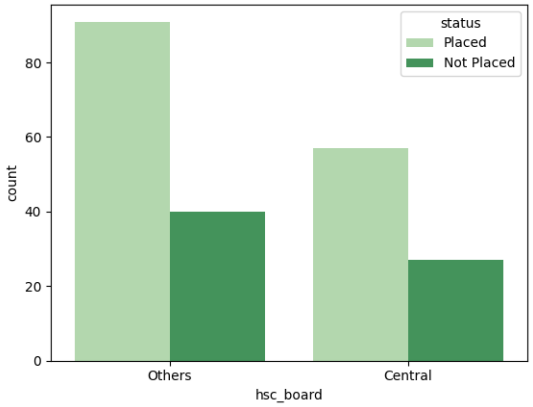
Gender

Come è evidente dal grafico la variabile *Gender* influisce notevolmente nel dataset, tanto che il numero di uomini piazzati è maggiore rispetto alle donne. Questo influenzerebbe il modello, portandolo anche a dare una previsione discriminatoria per un gender. Alla luce di questo abbiamo deciso di rimuovere la variabile *Gender* così da rendere il modello quanto più imparziare possibile da un punto di vista **Etico.**



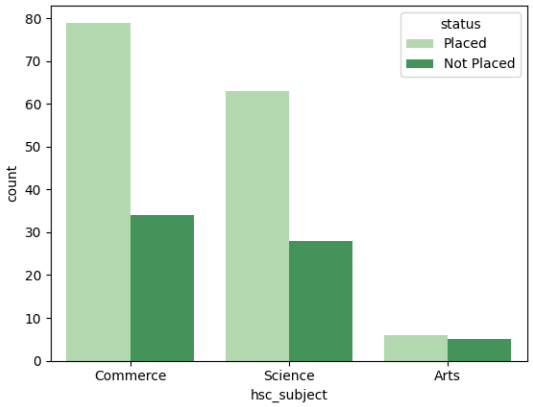
SSC Board

Come si può vedere da grafico, la variabile *SSC Board* influisce discretamente nel dataset, in quanto in base al valore della variabile, non c'è una differenza notevole tra le possibilità di essere preso o non. Quando la variabile ha come valore "Others", abbiamo una probabilità lievemente maggiore di essere piazzati rispetto al valore "Central". Nonostante questo, abbiamo deciso di non rimuovere la variabile.



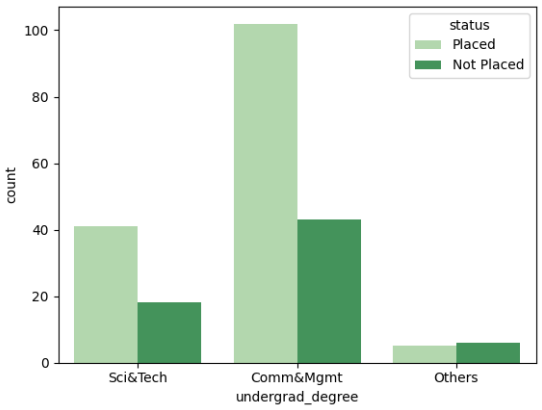
HSC Board

Come si può notare dal grafico, la variabile *HSC Board* influisce notevolmente nel dataset, in quanto in base al valore della variabile, la probabilità di essere piazzato è notevole. In conclusione abbiamo deciso di non rimuovere la variabile.



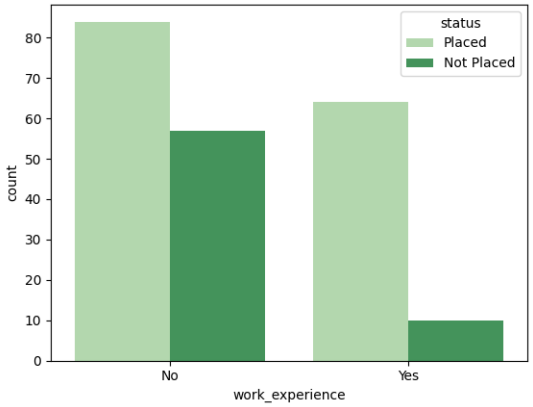
HSC Subject

Come si può vedere dal grafico di sopra, la variabile *HSC Subject* influisce nel dataset, in quanto in base al valore della variabile, si potrebbe avere più possibilità di essere presi o non. In questo caso il valore "Commerce" aumenterebbe le possibilità di essere presi per una persona che ha studiato in questo campo. Da questa considerazione abbiamo deciso di non rimuoverla.



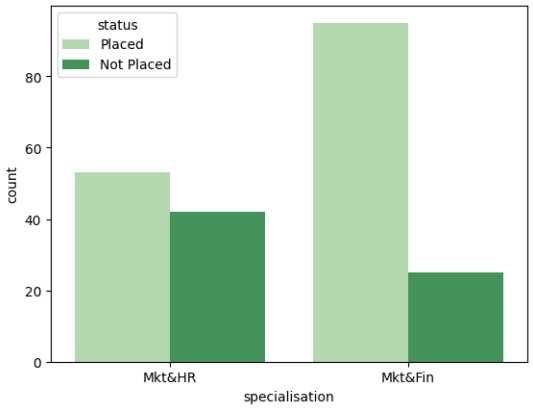
Undergrad Degree

Come si può vedere dal grafico di sopra, la variabile *Undergrad degree* influisce nel dataset, in quanto in base al valore della variabile, si avrà una probabilità maggiore di essere presi o non. Si può notare infatti che le persone che si sono laureate nel campo "Comm&Mgmt" hanno più possibilità di essere presi. Da questa analisi abbiamo deciso di non rimuovere la variabile.



Work Experience

Come si può vedere da questo grafico, la variabile *Work experience* influisce nel dataset. Chi non ha esperienze lavorative, ha più possibilità di essere preso rispetto a chi ne ha. In conclusione abbiamo deciso di non rimuoverla.



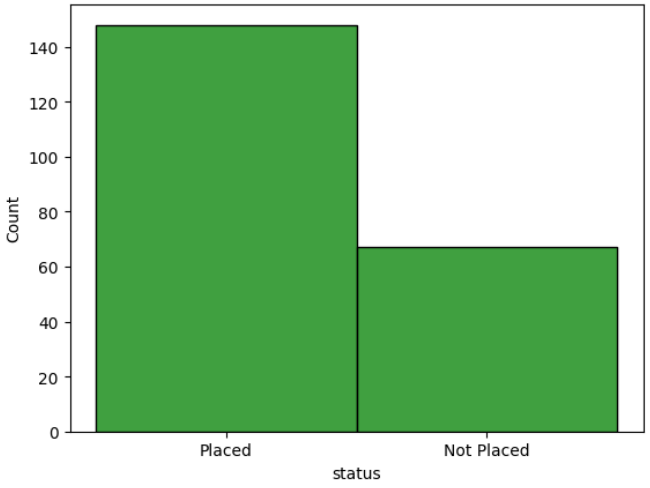
Specialisation

In questo grafico la variabile *Specialisation* influisce nel dataset. Chi ha un certo tipo di specializzazione, come "Mkt&Fin", ha più possibilità di essere preso. In conclusione abbiamo deciso di non rimuovere la variabile.

4.3.4 Data balancing

Il *Data Balancing* è l'insieme di tecniche per convertire un dataset sbilanciato in un dataset bilanciato. Questa è una delle fasi più importanti del **Data Preparation** perchè molti problemi reali sono sbilanciati e la maggior parte dei Machine Learning funzionano bene solo quando il numero di esempi di una certa classe è simile al numero di esempi di un'altra classe.

Questa per noi è una delle fasi più importanti per il nostro progetto avendo un numero di istanze di una classe estremamente diversa da un altra. Analizziamo ora com'è bilanciato il nostro dataset:

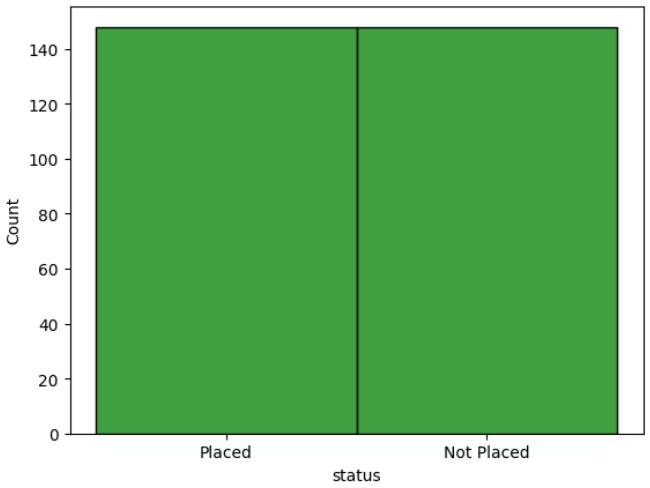


Dal grafico si evince che non abbiamo un numero di istanze della classe *Not Placed* uguale alla classe *Placed*, oltretutto è un numero largamente inferiore (Placed: 148, Not Placed: 67).

Dopo una serie di analisi siamo arrivati alla conclusione di utilizzare una tecnica di OverSampling, che sarebbe un metodo in cui vengono casualmente aggiunte un numero di istanze del dataset della classe di minoranza, che in questo caso è *Not Placed*.

Utilizzando la classe RandomOverSampler siamo riusciti a bilanciare il nostro dataset.

Di seguito viene mostrato il grafico una volta svolta questa operazione:



Tutto questo ci ha permesso quindi di migliorare le performance del nostro modello di Machine Learning, portandoci alla conclusione del nostro **Data Preparation**. Da adesso andremo a esporre il nostro modello di Machine Learning.

4.4 Modelling

In questa fase valuteremo la tecnica di Machine Learning da utilizzare.

Come abbiamo definito in precedenza utilizzeremo un modello di *Machine Learning* supervisionato che andrà a risolvere problemi di classificazione.

4.4.1 Classificazione

La classificazione è una task in cui l'obbiettivo è predire il valore di una variabile categorica, chiamata variabile dipendente tramite l'utilizzo di un training set, ovvero un insieme di osservazioni per cui la variabile dipendente è nota.

Esistono dei problemi di classificazione che possono essere risolti tramite l'utilizzo di un modello chiamato classificatore. Nel nostro caso abbiamo analizzato diversi modelli:

- **Decision Tree**.
- **Random Forest**.
- **Naive Bayes**.
- **K-Nearest Neighbors**.

Ogni classificatore è stato analizzato attraverso l'uso di metriche di valutazione che sono servite anche per aiutarci sulla scelta del Modello da utilizzare. Di seguito spiegheremo ogni modello e il suo obiettivo.

4.4.2 Decision Tree

Il Decision Tree (Albero Decisionale) è un modello di Machine Learning può essere utilizzato anche per problemi di classificazione. In questo modello, l'algoritmo cosrtuisce un albero di decisione che rappresenta la sequenza di decisioni che devono essere prese per classificare correttamente un'istanza. L'obiettivo di questo algoritmo è quello di predire il valore di una variabile target, apprendendo semplici regole di decisione inferite dai dati di training. La particolarità di questi alberi è la loro facilità di lettura, grazie al quale è possibile comprendere la ragione con la quale è stata fatta una determinata predizione.

4.4.3 Random Forest

Il Random Forest è un modello di Machine Learning utilizzato per la classificazione e va a combinare molteplici alberi decisionali in un unico modello. In poche parole vengono costruiti questi alberi decisionali utilizzando sottoinsiemi casuali del set di dati di training e selezionando casualmente le variabili da utilizzare in ciascun albero.

4.4.4 Naive Bayes

Il Naive Bayes è un modello di Machine Learning utilizzato per la classificazione che va a considerare le caratteristiche della nuova istanza da classificare e calcola la probabilità che queste facciano parte di una classe tramite l'applicazione del teorema di Bayes.

4.4.5 K-Nearest Neighbors

Il K-Nearest Neighbors è un modello di Machine Learning utilizzato per la classificazione che va a rappresentare delle istanze di training come punti nello spazio multidimensionale, dove ogni dimensione rappresenta una variabile del problema.

4.4.6 Training set e Test set

Come si è potuto notare dai paragrafi precedenti, la preparazione dei dati è molto importante perché poi quest'ultimi dovranno essere utilizzati della fase di addestramento per il nostro modello. Una cosa importante che bisogna ricordarsi è che non si deve far allenare il Machine Leaner con l'intero dataset perché può portare a risultati totalmente inaffidabili. Per questo motivo bisogna dividere il dataset in due insiemi: Training set e Test set. Il Training set è un insieme di dati che il nostro algoritmo utilizzerà per l'addestramento, mentre il Test Set verrà utilizzato dall'algoritmo addestrato per mettersi alla prova perché dovrù predire la classe di appartenenza di questi dati. Per il nostro modello abbiamo deciso di dividere il dataset in un buon 67% per il Training set e il restante 33% per il Test set.

4.4.7 Validation

Da questo momento inizieremo a valutare i modelli precedentemente descritti per decidere quale utilizzare nel nostro progetto. Ovviamente avremo bisogno di qualche metrica da utilizzare per verificare le prestazioni del nostro modello nella fase di training. Di sotto sono elencate le metriche che andremo a utilizzare e valutare per ogni singolo modello:

- **Acuratezza**: indica il numero totale di predizioni corrette, sia della classe positiva che negativa.

$$\textit{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

- **Precisione**: indica il numero di predizioni corrette delle classe **True** rispetto tutte le predizioni fatte dal classificatore. In poche parole indica quanti errori ci saranno nella lista delle predizioni fatte dal classificatore.

$$\textit{Precision} = \frac{(TP)}{(TP + FP)}$$

- **Recall**: indica il numero di predizioni corrette per la classe **True** rispetto tutte le istanze positive di quella classe. In breve indica quante istanze positive nell'intero dataset il classificatore può determinare.

$$\textit{Recall} = \frac{(TP)}{(TP + FN)}$$

Vogliamo massimizzare ognuno di questi valori nella loro interezza. Elenchiamo adesso i vari modelli con i relativi valori delle metriche:

- **Decision Tree**
  - **acuratezza**: 0.85
  - **precisione**: Placed: 0.85, Not Placed: 0.84
  - **recall**: Placed: 0.80, Not Placed: 0.89
- **Random Forest**
  - **acuratezza**: 0.87
  - **precisione**: Placed: 0.84, Not Placed: 0.89

  - **recall**: Placed: 0.86, Not Placed: 0.87

- **Naive Bayes**
  - **acuratezza**: 0.68
  - **precisione**: Placed: 0.64, Not Placed: 0.72
  - **recall**: Placed: 0.66, Not Placed: 0.70

- **K-Nearest Neighbors**
  - **acuratezza**: 0.82
  - **precisione**: Placed: 0.76, Not Placed: 0.88
  - **recall**: Placed: 0.78, Not Placed: 0.86

Da questa analisi possiamo vedere che i modelli Decision Tree e Random Forest sono tra quelli più promettenti, con delle metriche abbastanza alte. La decisione del modello alla fine è ricaduta nel Random Forest, in quanto offriva dei valori piuttosto bilanciati e migliori rispetto al Decision Tree.

4.5 Evaluation

Questa è la fase in cui valuteremo i nostri risultati finali perché poi dovremo passare alla fase di **Deployment**. Da qui capiremo se i nostri obiettivi di business sono stati rispettati, se le nostre analisi sono state corrette e se il nostro progetto è consistente e solido. Durante il nostro lavoro, abbiamo sempre avuto dubbi riguardo il raggiungimento del nostro obiettivo, siamo dovuti ritornare indietro nei nostri passi in certi casi per errori in fase di analisi o perché ci sono stati problemi in fase di modelling. Precedentemente avevamo pensato di renderlo utilizzabile per studenti italiani, però quando abbiamo verificato che ciò non poteva realizzarsi per il dataset che avevamo deciso di utilizzare, siamo ritornati indietro rianalizzando i dati offerti e rifacendo la fase di **Data Preparation**. Oltrettutto la nostra preoccupazione era dovuta alla scelta di lasciare o meno una determinata feature **(Gender)**, che per noi era trascurabile però influenzava notevolmente la predizione del nostro Modello. Alla fine i risultati usciti sono soddisfacenti, coerenti e quanto più solidi possibili tanto da rendere possibile la fase di **Deployment**.

4.6 Deployment

In questa fase parleremo di come abbiamo implementato il nostro modello di **Intelligenza Artificiale**.

Come precedentemente avevamo descritto nella fase di *Business Understanding*, abbiamo utilizzato **Python** come linguaggio di programmazione del nostro modello che offriva una serie di librerie interessanti e utili come *Pandas, Sklearn, Seaborn, RandomOverSampler, CustomTkinter* e altri moduli utili per i modelli che abbiamo testato.

Per esempio abbiamo importato all'interno del nostro codice il modulo *RandomForestClassifier*, che appunto sfruttava l'algoritmo Random Forest. Com'è possibile vedere dalla nostra pagina **GitHub** *(ref. parag. 1.1 pg. 3)*, è stato suddiviso il nostro programma in directory dov'è possibile vedere il codice sorgente suddiviso in due parti:

1. **Business Logic**, che presenta la parte logica dell'applicativo e che integra il modello di Machine Learning.

2. **View**, che presenta la GUI del tool.

Alla fine siamo riusciti a integrare il modello di Machine Learning nel modo più completo possibile e terminare il nostro Tool.

5 Conclusione

Il progetto si può dire concluso, dopo il tanto lavoro e impegno nella fase di modellazione del tool e del modello presentato. Un grande scalino, come descritto nella fase di *Data Understanding* e un pò in tutto il documento è stato il discorso dell'**Etica**, il tutto anche dovuto alla difficoltà nella fase di ricerca del/dei dataset, in questo questo caso unico, in quanto molti dataset non tornavano utili per il nostro progetto. Con questo però non si pregiudica in futuro la possibilità di aggiungere altri dati e rendere il modello ancora più efficace e veritiero possibile e che sia in futuro utilizzabile da più studenti di Nazioni diverse.

In questa conclusione vogliamo ovviamente dire di essere enormemente soddisfatti del progetto che è stato creato e concluso, soprattutto per il nostro primo progetto che integra un modello di **Intelligenza Artificiale.**