Food Inspections Chicago
Relazione Caso di Studio

Gruppo di lavoro:
Giuseppe Ventrella, 716909, g.ventrella21@studenti.uniba.it
Lorenzo Scazzari, 758461, l.scazzari@studenti.uniba.it

https://github.com/giux2001/ProgettoICON

A.A. 2023-2024

1. Introduzione

Nel seguente caso di studio ci si è concentrati sulle ispezioni alimentari di strutture locate a Chicago. I dataset sono stati attinti dal Chicago Data Portal poichè si è rivelato un portale interessante e contente grandi quantità di dati reali. Nella fattispecie si è attinto ai dati relativi alle ispezioni alimentari registrate tra il 2010 e il 20 maggio 2024. A questo dataset si è deciso di affiancarne un altro relativo alle Community Area della città di Chicago, che rappresentano le aree amministrative in cui è suddivisa la città. Si è deciso di utilizzare questo dataset per arricchire il contesto relativo alle ispezioni e alle aree in cui le strutture sono locate.

L'obiettivo è stato principalmente quello di effettuare task di classificazione per prevedere il risultato delle ispezioni, sfruttando la conoscenza che si è potuta inferire tramite la Knowledge Base. Inoltre si sono analizzate le probabilità di interesse grazie ad una rete bayesiana.

Nel progetto si è utilizzato il linguaggio Python.

Di seguito si elencano gli argomenti di interesse presenti all'interno del caso di studio:

- Rappresentazione e ragionamento relazionale. A partire dai due dataset sopracitati si è creata tramite Prolog una Knowledge Base che è risultata utile per inferire nuove features non presenti nei dataset originali, che saranno sfruttate nel task di apprendimento supervisionato.
- Apprendimento Supervisionato. Si sono utilizzati diversi classificatori quali Regressione Logistica, Regressione Logistica Multinomiale, Random Forest, Gradient Boosting, Decision Tree e Naive Bayes.
- Modelli di conoscenza incerta. Utilizzo di una rete Bayesiana per effettuare inferenza probabilistica e previsioni.

2. Dataset e dominio di interesse

Come già detto sopra, si è attinto da due dataset:

- Food Inspections | City of Chicago | Data Portal, il dataset relativo alle ispezioni alimentari delle strutture della città di Chicago;
- Public Health Statistics - Selected public health indicators by Chicago community area - Historical | City of Chicago | Data Portal, il dataset relativo ai dati socio-sanitari delle Community Area di Chicago.

Come già anticipato, si è deciso di combinare i due dataset per definire un contesto migliore intorno ad ogni singola ispezione. Difatti tra gli obiettivi c'è stato quello di verificare la presenza di correlazioni o pattern tra i risultati delle ispezioni di strutture e le Community Area in cui esse si trovano. In questo modo la predizione del risultato non terrà

conto unicamente dei dati relativi all'ispezione stessa ma anche del quartiere e delle sue condizioni socioeconomiche.

Ora si procede a descrivere più accuratamente i dataset utilizzati all'interno del caso di studio.

Riguardo il dataset delle Food Inspections, i dati presenti sono:

- Inspection ID, un codice univoco che identifica una singola ispezione;
- DBA Name e AKA Name, rispettivamente il nome ufficiale della struttura e il nome con cui è conosciuta al pubblico;
- License, numero di licenza della struttura;
- Facility Type, la tipologia della struttura ispezionata (Ristoranti, Supermercati, Mense scolastiche, ecc.)
- Risk, valore da 1 a 3 che indica il grado con il quale la struttura può inficiare sulla salute pubblica;
- Address/City/State/Zip/Latitude/Longitude, dati relativi all'ubicazione della struttura;
- Inspection Date, data in cui l'ispezione è stata eseguita;
- Inspection Type, che rappresenta il motivo per cui l'ispezione è stata effettuata;
- Results, indicante il risultato dell'ispezione;
- Violations, le violazioni che sono state riscontrate durante l'ispezione della struttura.

All'interno del dataset relativo ai dati sociosanitari sono presenti:

- Nome e codice della Community Area;
- Dati relativi alle condizioni sociosanitarie della community area;
- Dati relativi alla criminalità della community area;
- Dati relativi alla condizione economica della community area;

3. Preprocessing dei dati

Si è effettuato il preprocessing su entrambi i dataset utilizzati per poi effettuare un join tra di essi sulla base delle community area.

3.1 Preprocessing Ispezioni

In particolare, per quanto riguarda il dataset relativo alle ispezioni, sono state eliminate le seguenti colonne in quanto poco interessanti nella fase di apprendimento:

- License,
- AKA Name,
- Address/City/State/Zip, eliminate in quanto il join tra i due dataset è avvenuto sulla base delle community area, come verrà illustrato meglio in seguito;
- Inspection Type

Si è deciso inoltre di filtrare il dataset lavorando unicamente sulle ispezioni che vanno dal 2019 al 20 maggio 2024 per due motivi in particolare:

- ridurre la dimensione del dataset che ammontava a 200 mila valori, considerati eccessivi
- la feature riguardante le violazioni è assai importante nel contesto del task di classificazione. Si è osservato che le normative concernenti le violazioni con i rispettivi codici fossero cambiate dal 1 luglio 2018 (https://www.chicago.gov/city/en/depts/cdph/provdrs/food_safety/svcs/understand_healthcoderequirementsforfoodestablishments.html). Onde evitare la presenza di dati aventi codici di violazioni anacronistici, si è effettuato il filtraggio.

Sempre riguardo le violazioni, i valori relativi alla colonna erano presenti prima del processing nella seguente forma:

2.CITY OF CHICAGO FOOD SERVICE SANITATION CERTIFICATE - Comments: OBSERVED NO CERTIFIED FOOD SERVICE SANITATION MANAGER WITH CERTIFICATE ON-SITE AT TIME OF INSPECTION. INSTRUCTED PERSON-IN-CHARGE TO OBTAIN CITY OF CHICAGO CERTIFIED FOOD SERVICE SANITATION MANAGER CERTIFICATE AND MAINTAIN POSTED AT ALL TIMES. PRIORITY FOUNDATION VIOLATION. 7-38-012 CITATION ISSUED. | 3. MANAGEMENT, FOOD EMPLOYEE AND CONDITIONAL EMPLOYEE; KNOWLEDGE, RESPONSIBILITIES AND REPORTING - Comments: OBSERVED NO SIGNED EMPLOYEE HEALTH POLICY COPIES ON-SITE AT TIME OF INSPECTION. INSTRUCTED TO PROVIDE EMPLOYEE HEALTH POLICY SIGNED BY ALL EMPLOYEES AND MAINTAIN ON-SITE AT ALL TIMES. PRIORITY FOUNDATION VIOLATION. 7-38-010 CITATION ISSUED...

Per ogni ispezione potevano dunque esserci più violazioni ognuna identificata da un codice numerico ed un relativo commento dell'ispettore. Per gestire tale problema si sarebbe potuto eliminare la colonna ma si è presupposto che le violazioni potessero essere utili sia nella predizione che nell'inferenza probabilistica della rete bayesiana. Dunque, per effettuare il preprocessing della violazioni in primis si sono estratti i codici delle violazioni mediante un'espressione regolare. Ora, dato il grande numero di violazioni, 63, non si è ritenuto opportuno effettuare una codifica one hot encoding su ogni singola violazione. Si è piuttosto optato per un raggruppamento nelle seguenti classi, che tenessero conto anche della suddivisione esplicitata nel documento sopra linkato

- Violations on Management and Supervision
- Violations on Hygiene and Food Security
- Violations on Temperature and Special Procedures
- Violations on Food Safety and Quality
- Violations on Instrument Storage and Maintenance
- Violations on Facilties and Regulations
- No Violations: una colonna aggiunta avvalorata ad 1 qualora non fossero riportate violazioni

Nella suddivisione si è tenuto conto della serietà delle violazioni, in quanto queste potevano essere relative sia a fattori di rischio su cui intervenire immediatamente e sia a buone pratiche da seguire. Da ciò ne consegue che le violazioni non abbiano tutte lo stesso peso. Le prime tre classi di violazioni sono violazioni serie, il che significa che se riscontrate hanno un importante peso sul risultato dell'ispezione, mentre le altre tre tipologie di violazioni sono meno impattanti.

Per quanto riguarda invece la colonna Results (la futura variabile target negli algoritmi di apprendimento supervisionato), essa conteneva i seguenti valori

- Pass
- Fail
- Pass with conditions: durante l'ispezione sono state riscontrate delle violazioni serie ma esse sono state risolte durante l'ispezione stessa e dunque la struttura risulta idonea
- No entry
- Out of Business
- Business Not Located
- Not Ready

Si è deciso di conservare i primi 3 valori (Pass, Fail, Pass with Conditions), eliminando le righe che non li contenessero come valori di Results. Essi, infatti, sono quelli più interessanti nel task di classificazione.

Un altro fattore interessante riguardo il preprocessing di tale dataset è il seguente: ad una singola struttura sono associate più ispezioni avvenute in date diverse (la relazione è dunque di uno a molti). Si è deciso di eliminare i duplicati presenti nella colonna DBA Name: in tal modo abbiamo come risultato un dataset contenente una singola ispezione per ristorante (la relazione diventa ora uno ad uno). Si è scelto di fare ciò per le seguenti ragioni

- riduzione della dimensionalità del dataset
- le ispezioni multiple possono introdurre rumore nei dati, specialmente se ci sono variazioni significative nei risultati delle ispezioni nel tempo.
- si è interessati a prevedere il risultato di un locale dati vari fattori relativi all'ispezione e alla community area in cui si trova; dunque, non è necessario tenere conto dei risultati delle ispezioni passate. L'obiettivo è difatti il seguente: dato un locale caratterizzato da certi fattori, si vuole prevedere l'esito dell'ispezione

3.2 Preprocessing Community Area

All'interno di questo dataset erano presenti molte colonne rappresentanti diversi dati sulle condizioni socioeconomico-sanitare della community area. Si è ritenuto che queste colonne, prese singolarmente, sarebbero state poco interessanti per gli obiettivi che ci si

era preposti di raggiungere. Di conseguenza abbiamo rienuto opportuno raggrupparle in due diverse macrocategorie:

- Health Index, raggruppando le colonne relative a diverse malattie, tumori, nascite premature, ecc;
- Crime Index, tramite le colonne relative agli omicidi e all'uso di armi da fuoco;

contenenti come valori la media dei valori presenti nelle colonne dalle quali sono stati ottenuti.

In aggiunta a questi si sono mantenuti gli indici relativi alla disoccupazione, la soglia di povertà, e il reddito annuo medio, in quanto risultavano gli unici che avvesse senso considerare separatamente.

3.3 Join dei dataset

Effettuare il join dei due dataset non è stato immediato dal momento che non erano presenti colonne relative al nome della community area nei dati relativi alle ispezioni. Erano presenti però colonne relative alla latitudine e longitudine del locale ispezionato. Dunque, sulla base di ciò, mediante la libreria geopandas si è risalito dalla latitudine e longitudine alle community area in cui si trovavano i locali ispezionati. Si è poi dunque proceduto ad effettuare il join.

4. Knowledge Base

Si è creata una knowledge base in Prolog al fine di effettuare ragionamento e specialmente inferire nuove feature utili al task di apprendimento.

Per la creazione della KB si è utilizzata la libreria pyswip.

4.1 Individui

Nell'analisi del problema si sono determinati seguenti individui denotati da opportuni simboli di funzione:

- inspection_id(I), che denota la singola ispezione di una data struttura
- community_area(C), che denota una community area

Si è scelto di non introdurre il terzo individuo facility, che avrebbe dovuto denotare la struttura ispezionata, in quanto come già detto, la relazione adottata è uno ad uno e di conseguenza introdurre un terzo individuo sarebbe stato inutile.

4.2 Relazioni

Dal momento che gli individui denotati sono due, si è individuata una singola relazione tra questi:

inspection_in_community_area(inspection_id(I), community_area(C))

Tale relazione permette di collegare un'ispezione alla community area in cui la struttura si trova.

4.3 Fatti

Sulla base del dataset risultante dal join precedente, la Knowledge Base è stata popolata con dei fatti riguardanti le ispezioni e le community area.

Si procede ora ad illustrare le proprietà caratterizzanti gli individui.

Proprietà per ispezione:

- facility_name(inspection_id(I), N)
- facility_type(inspection_id(I), T)
- risk(inspection_id(I), R)
- results(inspection_id(I), RS)
- community_area(inspection_id(I), CA)
- no_violations(inspection_id(I), NV)
- violations_on_management_and_supervision(inspection_id(I), VM)
- violations_on_hygiene_and_food_security(inspection_id(I), VH)
- violations_on_temperature_and_special_procedures(inspection_id(I), VT)
- violations_on_food_safety_and_quality(inspection_id(I), VF)
- violations_on_instrument_storage_and_maintenence(inspection_id(I), VI)
- violations_on_facilities_and_regulations(inspection_id(I), VR)

Proprietà per community area:

- crime_index(community_area(C), CI)
- health_index(community_area(C), HI)
- below_poverty_level(community_area(C), PL)
- per_capita_income(community_area(C), IN)
- unemployement(community_area(C), UN)

4.4 Clausole

Le clausole definite sono state utili per effettuare ragionamento e inferire nuove feature utilizzate poi nel contesto dell'apprendimento supervisionato. Non tutte le clausole definite diverranno feature nel dataset risultante ma alcune di esse vengono semplicemente richiamate da altre clausole che saranno poi effettivamente delle features.

Di seguito elenchiamo le clausole definite:

- *total_inspections_in_area(community_area(C), Count) :-findall(inspection_id(I), inspection_in_community_area(inspection_id(I), community_area(C)), Inspections),length(Inspections, Count)*

  il cui risultato sarà il numero di ispezioni in una determinata community area

- *failed_inspections_in_area(community_area(C), Count) :-findall(inspection_id(I), (inspection_in_community_area(inspection_id(I), community_area(C)), results(inspection_id(I), 0)), FailedInspections),length(FailedInspections, Count)*

  il cui risultato sarà il numero di ispezione fallite in una determinata community area

- *percentage_failed_inspections_in_area(community_area(C), Percentage) :- total_inspections_in_area(community_area(C), TotalCount),failed_inspections_in_area(community_area(C), FailedCount),TotalCount > 0, Percentage is (FailedCount / TotalCount) * 100*

  il cui risultato sarà la percentuale di ispezioni fallite in un'area

- *passed_inspections_in_area(community_area(C), Count) :-findall(inspection_id(I), (inspection_in_community_area(inspection_id(I), community_area(C)), results(inspection_id(I), 1)), PassedInspections),length(PassedInspections, Count)*

  il cui risultato sarà il numero di ispezione superate in un'area

- *percentage_passed_inspections_in_area(community_area(C), Percentage) :- total_inspections_in_area(community_area(C), TotalCount),passed_inspections_in_area(community_area(C), PassedCount),TotalCount > 0, Percentage is (PassedCount / TotalCount) * 100*

  il cui risultato sarà la percentuale di ispezione passate in un'area

- *passed_with_condition_inspections_in_area(community_area(C), Count) :- findall(inspection_id(I), (inspection_in_community_area(inspection_id(I), community_area(C)), results(inspection_id(I), 2)), PassedWithConditionInspections),length(PassedWithConditionInspections, Count)*

  il cui risultato sarà il numero di ispezione passate con condizione

- *percentage_passed_with_condition_inspections_in_area(community_area(C), Percentage) :- total_inspections_in_area(community_area(C), TotalCount),passed_with_condition_inspections_in_area(community_area(C), PassedWithConditionCount),TotalCount > 0, Percentage is (PassedWithConditionCount / TotalCount) * 100*

  il cui risultato sarà la percentuale di ispezioni passate con condizioni in un'area

- *passed_no_violations(inspection_id(I)) :-results(inspection_id(I), 1),no_violations(inspection_id(I), 1)*

  il cui risultato sarà un booleano che determina se l'ispezione è passata e non ha violazioni

- *serious_violations(inspection_id(I)) :- violations_on_management_and_supervision(inspection_id(I), Count),Count > 0;violations_on_hygiene_and_food_security(inspection_id(I), Count),Count > 0;violations_on_temperature_and_special_procedures(inspection_id(I), Count),Count > 0")*

  il cui risultato sarà un booleano che determina se sono state riscontrate violazioni serie durante una ispezione

- *serious_violations_in_area(community_area(C), Count) :-findall(inspection_id(I), (inspection_in_community_area(inspection_id(I), community_area(C)),serious_violations(inspection_id(I)),results(inspection_id(I), 0)), SeriousViolations),length(SeriousViolations, Count)*

  il cui risultato sarà il numero di ispezioni con violazioni serie in un'area

- *percentage_serious_violations_in_area(community_area(C), Percentage) :- total_inspections_in_area(community_area(C), TotalCount),serious_violations_in_area(community_area(C), SeriousViolationsCount),TotalCount > 0, Percentage is (SeriousViolationsCount / TotalCount) * 100*

  il cui risultato sarà la percentuale di ispezioni con violazioni serie in un'area

- *average_crime_index(Average)*

  il cui risultato sarà la media del crime index delle aree

- *std_dev_crime_index(StdDev)*

  il cui risultato è il calcolo della deviazione standard per il crime index

- *high_crime_area(community_area(C)) :-crime_index(community_area(C), CrimeIndex),average_crime_index(Average),std_dev_crime_index(StdDev),CrimeIndex > Average + StdDev*

  il cui risultato è un booleano che determina se l'area ha un alto rischio di crimine

- *average_health_index(Average)*

  il cui risultato è la media dell' health index delle aree

- *std_dev_health_index(StdDev)*

  il cui risultato è la deviazione standard per l'health index

- *low_health_area(community_area(C)) :-health_index(community_area(C), HealthIndex),average_health_index(Average),std_dev_health_index(StdDev),HealthIndex < Average - StdDev*

  il cui risultato è un booleano che determina se l'area ha un basso livello sanitario

- *average_below_poverty_level(Average)*

  il cui risultato è la media delle soglie di povertà

- *std_dev_below_poverty_level(StdDev)*

  il cui risultato è la deviazione standard per le soglie di povertà

- *high_below_poverty_level(community_area(C)) :-below_poverty_level(community_area(C), BelowPovertyLevel),average_below_poverty_level(Average),std_dev_below_poverty_level(StdDev),BelowPovertyLevel > Average + StdDev*

  il cui risultato è un booleano che determina se l'area ha un alto livello di popolazione sotto la soglia di povertà

- *average_per_capita_income(Average)*

  il cui risultato è la media del reddito annuo delle aree

- *std_dev_per_capita_income(StdDev)*

  il cui risultato è la deviazione standard per il reddito annuo

- *low_per_capita_income(community_area(C)) :-per_capita_income(community_area(C), PerCapitaIncome),average_per_capita_income(Average),std_dev_per_capita_income(StdDev),PerCapitaIncome < Average - StdDev*

  il cui risultato è un booleano per determinare se un'area è a basso reddito

- *average_unemployment_rate(Average)*

  il cui risultato è la media dell'indice di disoccupazione delle aree

- *std_dev_unemployment_rate(StdDev)*

il cui risultato è la deviazione standard per la disoccupazione

- *high_unemployment_rate(community_area(C)) :-unemployment(community_area(C), UnemploymentRate),average_unemployment_rate(Average),std_dev_unemployment_rate(StdDev),UnemploymentRate > Average + StdDev*

  il cui risultato è un booleano per determinare se un'area è ad alto tasso di disoccupazione

Come è possibile notare alcune delle query effettuate sulle clausole restituiranno valori booleani. Saranno infatti inferite, oltre a features numeriche, anche feature booleane che indicano le condizioni di una data community area (se essa ha un alto tasso di disoccupazione, se è un'area a basso reddito...). Per determinare se i valori di tali features relativi ad una community area siano 0 o 1 (falsi o veri), si effettua il controllo rispetto ad una determinata soglia, che riguarda la media e la sua deviazione standard. Non si è scelto di utilizzare come soglia la sola media dal momento che si sarebbe falsata la condizione di una determinata community area: ad esempio se una data area avesse avuto un valore anche leggermente superiore rispetto alla media del crime index, sarebbe stata classificata erroneamente come area ad alto rischio di crimine. La deviazione standard serve dunque ad evitare ciò.

4.5 Query

Per ottenere il dataset di lavoro sul quale vengono lanciati gli algoritmi di apprendimento supervisionato sono state eseguite opportune query sulle features di interesse. In particolare, per ottenere nuove feature non presenti nel dataset originale sono state eseguite query su alcune delle clausole sopracitate.

Le nuove features ottenute sono le seguenti:

- has_insp_serious_violations
- num_insp_area
- perc_ins_failed_area
- perc_ins_passed_area
- perc_ins_passed_cond_area
- perc_serious_violations_failed_area
- is_high_crime_area
- is_low_per_capita_income
- is_high_below_poverty_level
- is_high_unemployement_rate

Le altre features presenti nel dataset di lavoro sono ottenute a partire da query compiute su fatti già presenti nella KB.

4.6 Conclusioni

Come già detto la Knowledge Base si è rivelata un importante strumento per inferire nuove features che aiutassero le previsioni dei modelli di apprendimento supervisionato.

Inoltre, la KB si è rivelata utile anche per svolgere un ulteriore preprocessing sul dataset. Consideriamo infatti la feature num_insp_area, non presente nel dataset originale ma inferita tramite la KB, grazie alla quale ci si è accorti che nel dataset di lavoro si trovassero delle aree con una singola ispezione. Questo poteva comportare problemi poiché queste aree di poco interesse avrebbero poi falsato i dati generali comportandosi appunto da outliers. In particolare, vi era una community area con dati socioeconomici-sanitari molto bassi che però conteneva una sola ispezione passata. Si è deciso di eliminare dunque le community area di questo tipo.

5. Apprendimento Supervisionato

Per i task di classificazione l'obiettivo è stato quello di prevedere la feature target Results, che corrispondeva appunto ai risultati delle singole ispezioni. Per effettuare ciò si è effettuato l'apprendimento sulla gran parte delle features del dataset di lavoro escludendo le features che non fossero numeriche.

Per quanto riguarda la feature faciliy_type inizialmente si era pensato di effettuare un label encoding, tuttavia tra i valori di facility_type non era definita una relazione d'ordine, si è deciso pertanto di escludere tale feature durante la fase di apprendimento.

5.1 Analisi della target

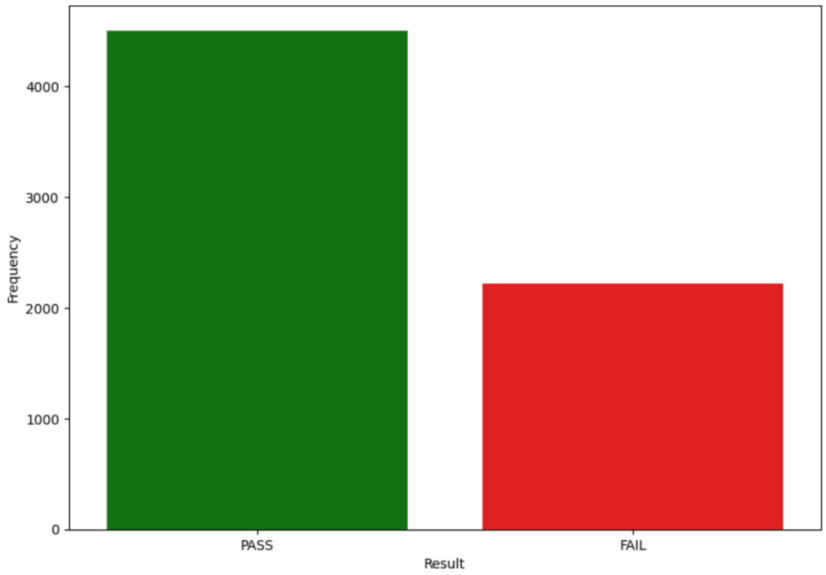
Per quanto riguarda la target da prevedere, Results, si ricorda che questa, a seguito del preprocessing effettuato, poteva assumere tre valori: Pass, Pass with Conditions, Fail.

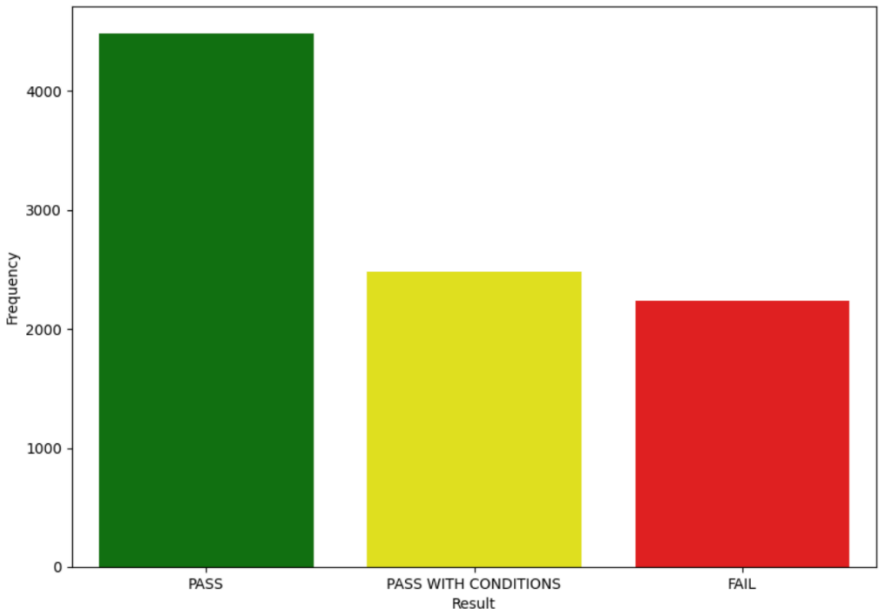
Durante la fase di apprendimento si è deciso di sperimentare dunque due approcci differenti.

1. Nel primo caso l'apprendimento è stato effettuato prendendo in riferimento come target i tre possibili valori di Results. Questo approccio risulta sicuramente il più realistico e il più corretto da eseguire poichè si prevedono quelli che sono i possibili risultati in un contesto reale.
2. Una seconda sperimentazione ha previsto l'eliminazione dal dataset di lavoro le righe i cui risultati fossero Pass with conditions. Si è deciso di sperimentare anche questo approccio poiché passare un’ispezione con condizione, in un contesto reale, è un risultato condizionato da fattori non completamente prevedibili. Come già detto infatti, i Pass with conditions rispecchiano quelle situazioni in cui, a seguito di una ispezione in cui risultano delle violazioni gravi, la struttura è in grado di rimediare sul momento e quindi passare comunque l'ispezione. È dunque chiaro che in fase di apprendimento questo possibile risultato non sia sufficientemente

distintivo come gli altri e che quindi abbassi le performance del modello. In seguito, si mostreranno i risultati ottenuti da entrambi gli approcci.

Inoltre, è interessante notare la distribuzione dei dati della target mostrata qui sotto:





Come è possibile notare la distribuzione dei Results è piuttosto sbilanciata, nell' approccio a due classi i Pass sono oltre il doppio dei Fail, mentre nel secondo approccio i Pass with conditions superano le ispezioni fallite ma rimangono comunque circa la metà delle ispezioni passate.

Vedendo questi dati si è dunque scelto di lavorare seguendo un ulteriore doppio approccio, addestrando i modelli sui campioni originali del dataset e di lavorare a seguito

di un piccolo bilanciamento che non falsasse la distribuzione reale. È infatti chiaro che in un contesto reale le ispezioni passate siano comunque maggiori di quelle fallite.

Di seguito mostriamo l'apprendimento dei modelli sui dati non bilanciati.

5.2 Scelta dei modelli e tuning degli iperparametri

Dato il grande numero di features la gran parte dei modelli utilizzati si sono utilizzati modelli basati su alberi quali: Decision Tree, Random Forest e Gradient Boosting. Questi, infatti, sono in grado di compiere automaticamente feature selection.

Si è sperimentato inoltre l'utilizzo della regressione logistica e della sua variante multinomiale, poiché, si ricorda, che si è effettuato l'apprendimento sia per l'approccio con target binaria, che con target a tre classi.

Si è inoltre utilizzato il Naive Bayes allenato unicamente sulle features categoriche del dataset in quanto l'assunzione di indipendenza condizionale tra le features è più facilmente soddisfatta con dati categorici.

Per tutti questi modelli è stato effettuato il tuning degli iperparametri ad eccezione del Naive Bayes, in quanto, basandosi questo su semplici calcoli probabilistici, non richiedeva l'ottimizzazione di parametri. Il Naive Bayes si basa infatti su assunzioni di indipendenza condizionale tra le features ed esse sono intrinseche al modello e non possono essere modificate tramite tuning.

Per gli altri modelli il tuning degli iperparametri è stato effettuato mediante una GridSearch con Cross Validation. La GridSearch esplora un insieme predefinito di combinazioni di iperparametri, da noi definito per ogni modello. Quando combinata con la Cross Validation per ogni combinazione di iperparametri viene eseguita la Cross Validation per valutare le prestazioni del modello.

Il tuning è stato effettuato sui seguenti iperparametri:

```
if model_name == 'RandomForest':
    model = RandomForestClassifier()
    hyperparameters = {
        'model__criterion': ['gini', 'entropy', 'log_loss'],
        'model__min_samples_split': [2, 5, 10],
        'model__min_samples_leaf': [1, 2, 4],
    }
elif model_name == 'LogisticRegression':
    model = LogisticRegression()
    hyperparameters = {
        'model__penalty': ['l1', 'l2'],
        'model__C': [0.1, 1, 10],
        'model__solver': ['liblinear', 'saga'],
        'model__max_iter': [1000,5000,10000]
    }
elif model_name == 'DecisionTree':
    model = DecisionTreeClassifier()
    hyperparameters = {
        'model__criterion': ['gini', 'entropy'],
        'model__min_samples_split': [2, 5, 10],
        'model__min_samples_leaf': [1, 2, 4],
    }
elif model_name == 'GradientBoosting':
    model = GradientBoostingClassifier()
    hyperparameters = {
        'model__loss': ['log_loss'],
        'model__learning_rate': [0.1, 0.01, 0.001],
        'model__min_samples_split': [2, 5, 10],
        'model__min_samples_leaf': [1, 2, 4],
    }
```

Fissati i migliori iperparametri trovati con la GridSearch, per quanto riguarda i modelli basati su alberi quali Random Forest, Decision Tree e Gradient Boosting si è studiato l'andamento delle prestazioni al variare di alcuni iperparametri:

- per il Decision Tree si sono studiate le prestazioni al variare della profondità massima
- per la Random Forest e il Gradient Boosting si sono valutate invece le prestazioni al variare della profondità massima e del numero degli stimatori

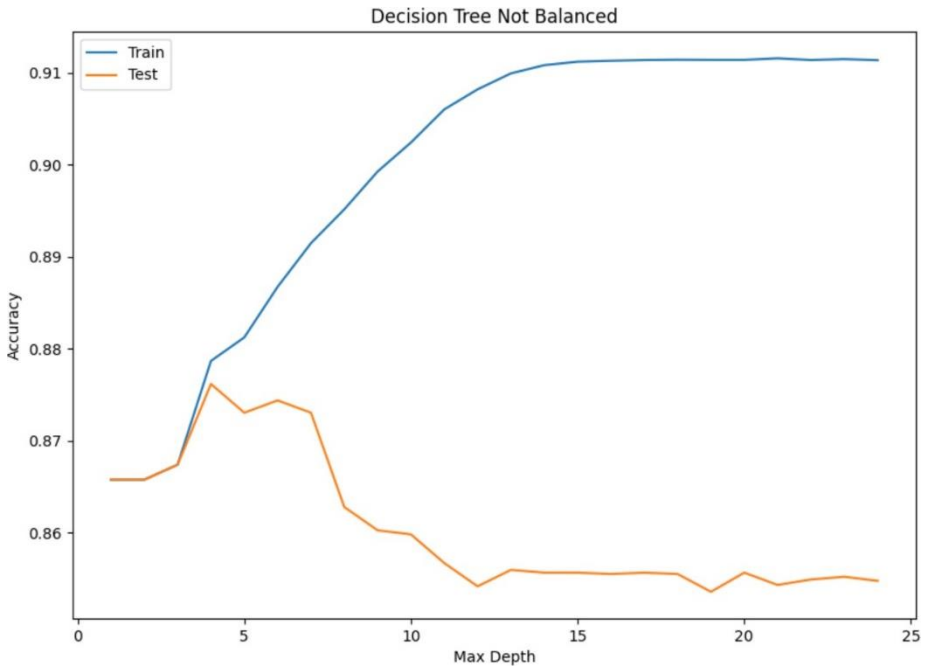
Per ogni iperparametro che è stato fatto variare, si è effettuata la KFold (con k = 10) calcolando la media delle prestazioni. Di seguito, ottenuto l'iperparametro che fatto variare ha permesso di ottenere l'accuracy più alta, si è riaddestrato il modello con quell'iperparametro e sono stati riportati i risultati sui dati di test in termini di

- Accuratezza
- Precisione
- Richiamo
- F1

5.2.1 Decision Tree

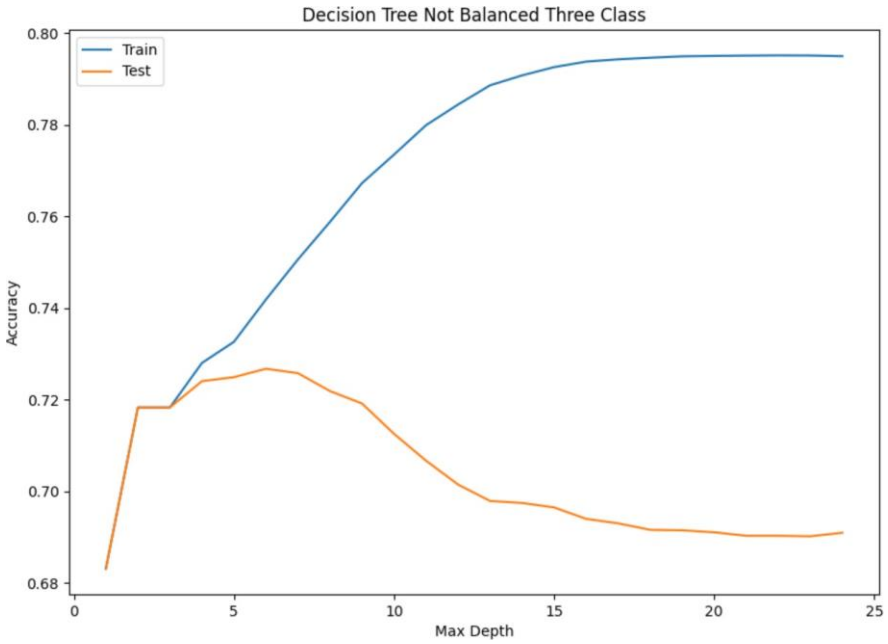
Si mostrano dunque ora i grafici che rappresentano l'accuratezza del modello al variare della massima profondità distinguendo l'accuratezza sui dati di test e sui dati di training. Si riportano inoltre i risultati del modello a seguito della scelta del miglior iperparametro. Come già detto in precedenza distinguiamo le prestazioni sui due approcci con target binaria e target a tre valori.

I risultati per l'approccio a due classi sono i seguenti:




```
Accuracy: 0.882282996432818
F1 Score: 0.912311780336581
Precision: 0.8910034602076125
Recall: 0.9346642468239564
```

Mentre per l'approccio con tre classi i risultati sono i seguenti:




```
Accuracy: 0.7286148501953973
F1 Score: 0.6795599162929475
Precision: 0.6828927157202825
Recall: 0.6812262105919208
```

Come era prevedivile, per quanto riguarda i risultati, la differenza tra i due approcci è molto evidente, poiché come già detto, eliminare i pass with conditions migliora molto le prestazioni del modello in quanto l'incertezza dovuta a questo risultato viene meno.

Come possiamo notare pare esserci anche un leggero overfitting all'aumentare della profondità dell'albero. Questo aspetto verrà rivalutato successivamente a seguito del bilanciamento

5.2.2 Logistic Regression

Per la regressione logistica si riportano solo i risultati sui dati di test dopo che i migliori iperparametri sono stati trovati mediante GridSearch.

I seguenti sono i risultati per l'approccio a due classi

```
Accuracy: 0.8810939357907254
F1 Score: 0.9122807017543859
Precision: 0.8828522920203735
Recall: 0.9437386569872959
```

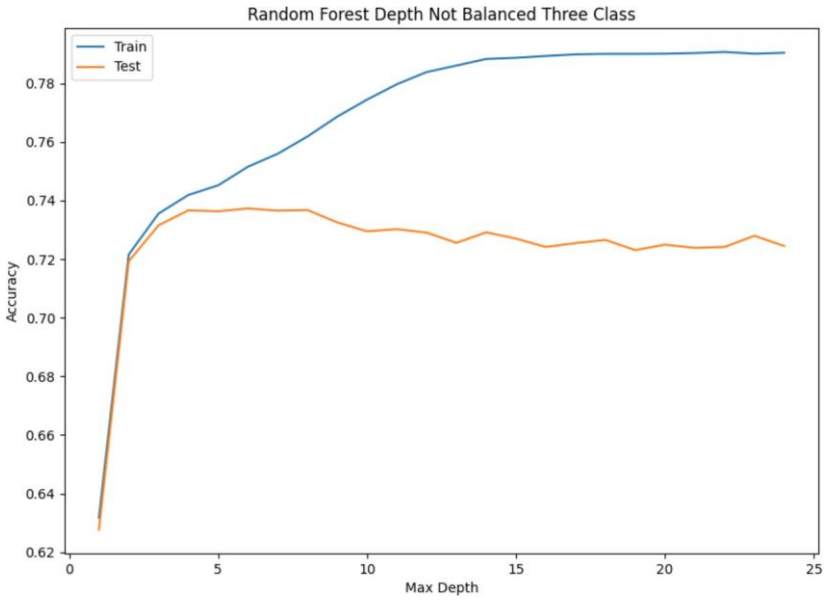
Di seguito invece si riportano i risultati della regressione logistica multinomiale:

```
Accuracy: 0.739036039947894
F1 Score: 0.6880844556817739
Precision: 0.6942929500584708
Recall: 0.6874638683981197
```

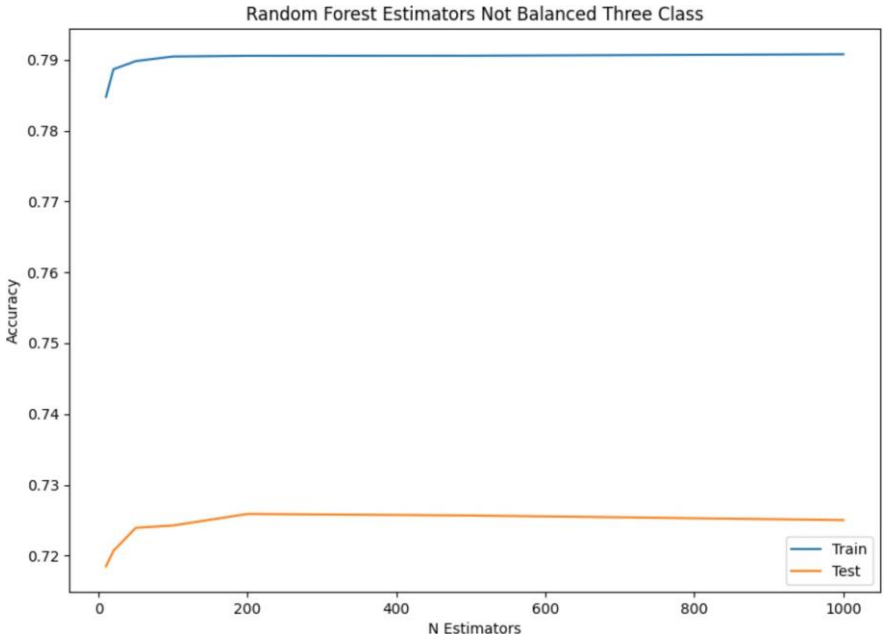
5.2.3 Random Forest

D'ora in poi per non appesantire la documentazione si riportano unicamente i risultati riguardanti l'approccio a tre classi, in quanto pur avendo prestazioni inferiori risulta, come già detto, rimane l'approccio che rispecchia di più un contesto verosimile.

Per quanto riguarda il Random Forest si mostrano i grafici e i risultati al variare della massima profondità e del numero degli stimatori.




```
Accuracy: 0.7412071211463309
F1 Score: 0.6898696325154866
Precision: 0.7001365043379172
Recall: 0.6905712032070611
```

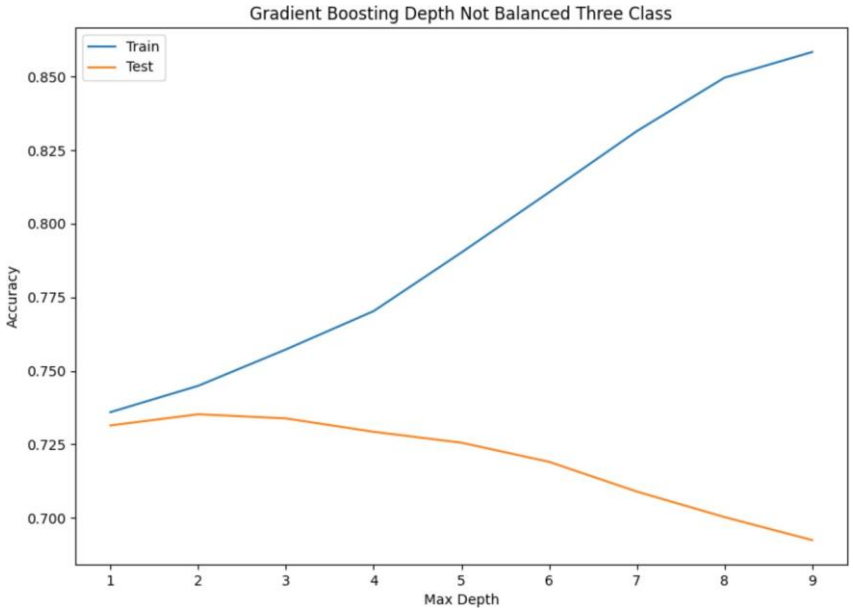



```
Accuracy: 0.7277464177160226
F1 Score: 0.6745592490750623
Precision: 0.6801448246090631
Recall: 0.6751999384254089
```

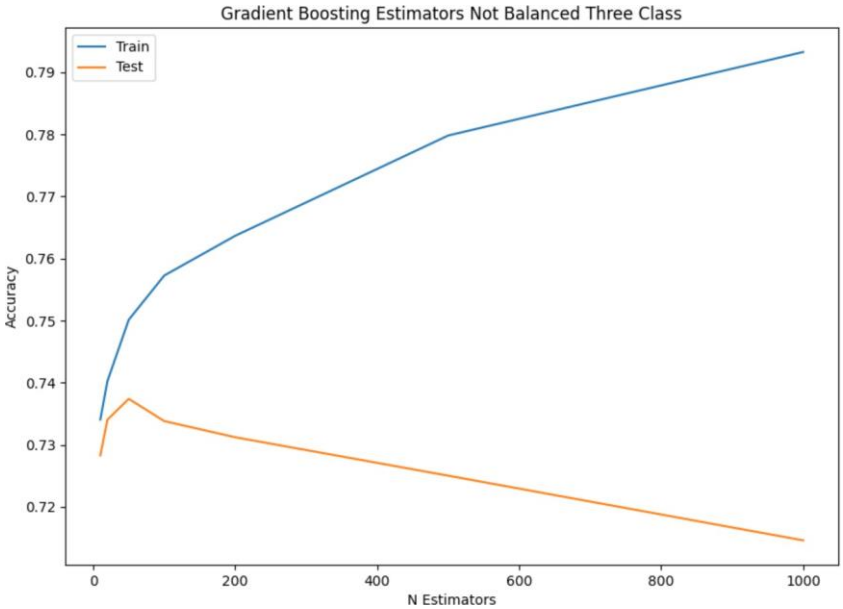
In questo caso come è possibile notare l'accuratezza si stabilizza al crescere della massima profondità mentre rimane pressoché invariata al variare degli stimatori che si rivelano dunque essere un iperparametro non impattante nel nostro caso.

5.2.4 Gradient Boosting

Anche in questo caso, si sono valutate le prestazioni al variare della profondità e degli stimatori, i risultati sono i seguenti:



```
Accuracy: 0.737733391228832
F1 Score: 0.6870041070859397
Precision: 0.6932404078183066
Recall: 0.6862786598014066
```




```
Accuracy: 0.7359965262700825
F1 Score: 0.6846576544737969
Precision: 0.6906238588348593
Recall: 0.6842617254250557
```

Anche in questo caso, come per il Decision Tree, si nota un leggero overfitting all'aumentare della profondità e del numero degli stimatori per il Gradient Boosting.

5.2.5 Naive Bayes

Si riportano unicamente i risultati a seguito del training.

```
Accuracy: 0.7242726877985237
F1 Score: 0.6750937581554114
Precision: 0.6770762857636209
Recall: 0.6744548210927409
```

5.2.6 Conclusioni

A seguito di questo primo ciclo di apprendimento i risultati si effettuano le seguenti considerazioni:

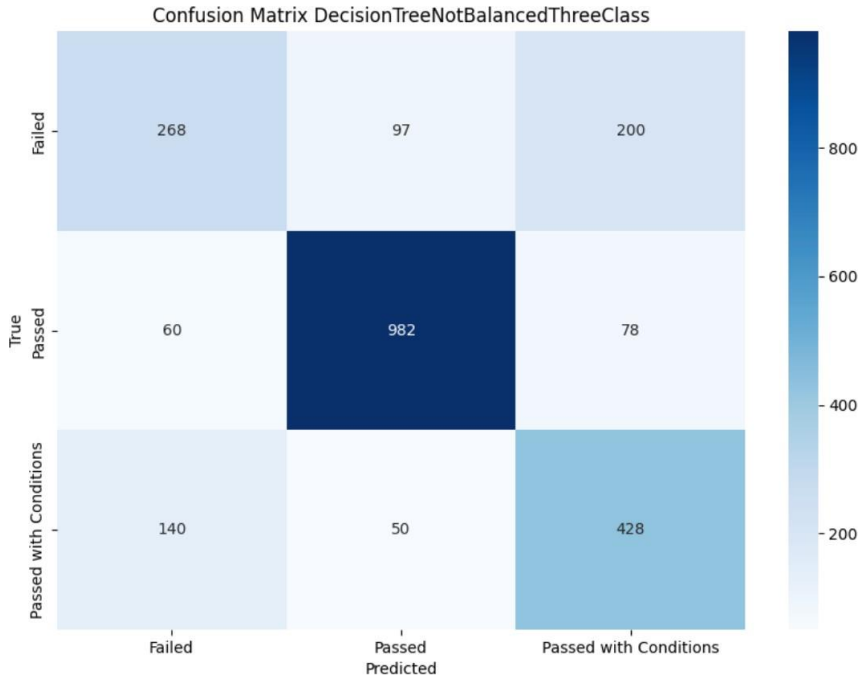
- il miglior modello risulta essere il random forest con la migliore max depth trovata
- in alcuni casi si osserva del leggero overfitting

5.3 Bilanciamento dei dati

Si è già potuto osservare che i dati risultano essere sbilanciati. Ciò può portare a due conseguenze:

- la presenza di overfitting
- Il favorimento da parte dei modelli della classe maggioritaria (in questo caso la classe delle ispezioni con esito Pass), sfavorendo invece le altre due classi presenti in maniera minoritaria: ciò porta il modello a predire la classe maggioritaria per la maggior parte degli esempi, ottenendo un'alta accuratezza superficiale ma ignorando completamente le classi minoritarie.

Si presenta come esempio la matrice di confusione del modello Decision Tree allenato su tre classi:

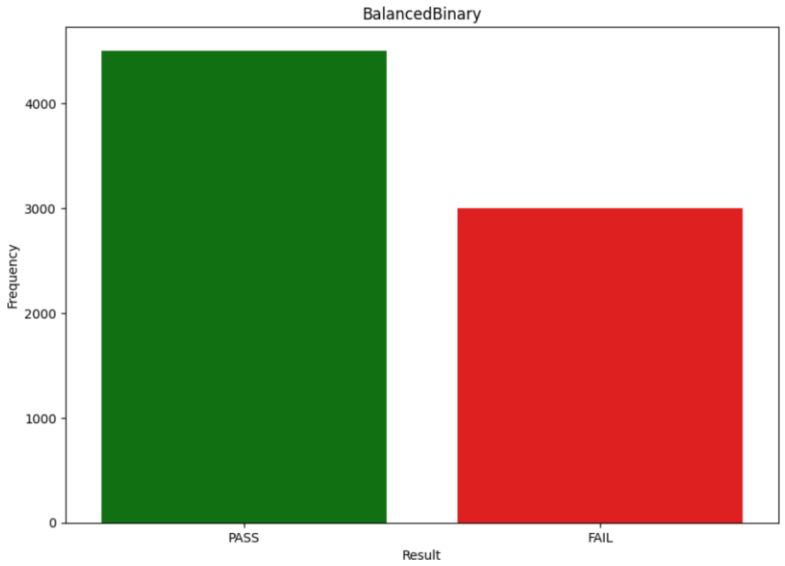


Come è possibile osservare il modello tende a predire molto più facilmente la classe dei Pass rispetto alle classi minoritarie. Infatti, è possibile osservare che per quanto riguarda le ispezioni non passate, il numero di esempi predetti correttamente è inferiore al numero di esempi predetti in maniera scorretta.

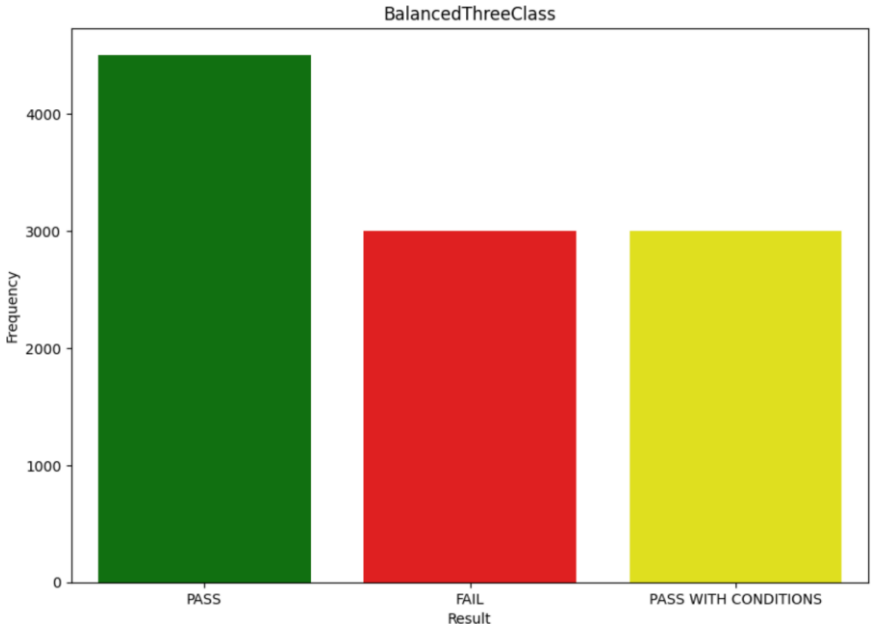
Si è deciso dunque di effettuare un lieve bilanciamente dei dati mediante la tecnica SMOTE effettuando un oversampling delle classi minoritarie. Il numero di esempi delle classi minoritarie non è però stato portato ad eguagliare il numero di esempi della classe maggioritaria dal momento che, come già prima specificato e come si è potuto osservare dalla distribuzione dei dati presenti del dataset, una struttura è più portata a passare un'ispezione rispetto al non passarla.

Il bilanciamento non è stato eseguito casualmente ma sono stati effettuati diversi bilanciamenti di prova per cercare il miglior compromesso tra risultati e consistenza.

Si presenta il bilanciamento su due classi



e il bilanciamento su tre classi

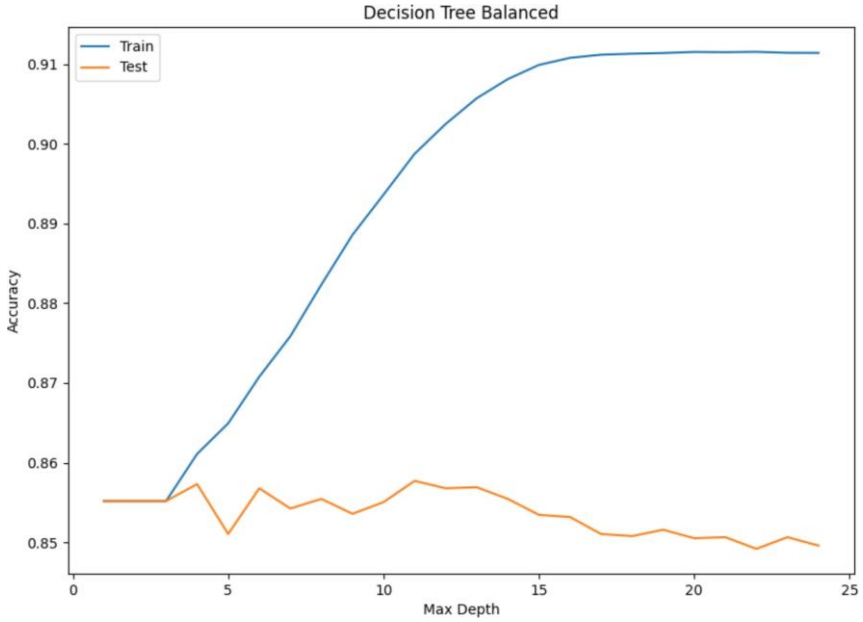


Fatto ciò, ora si ripresentano i risultati ottenuti con l'addestramento dei modelli sui dati bilanciati.

5.3.1 Decision Tree

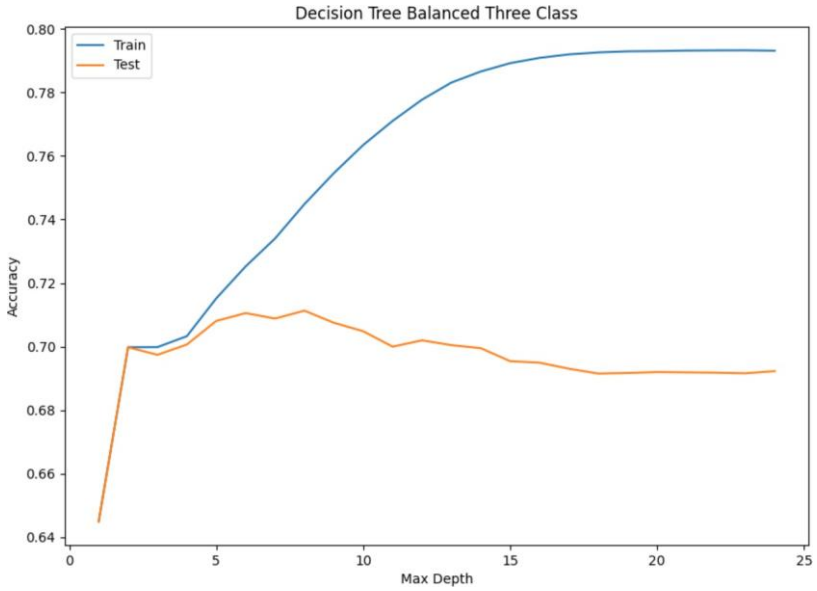
In questo caso mostriamo i risultati e i grafici basati sui dati bilanciati per entrambi gli approcci a due e a tre classi.

Per quanto riguarda la target binaria:




```
Accuracy: 0.861474435196195
F1 Score: 0.8949977467327626
Precision: 0.8889883616830797
Recall: 0.9010889292196007
```

Per la target a tre classi:



```
Accuracy: 0.7268779852366478
F1 Score: 0.6830376996076767
Precision: 0.6834662074118961
Recall: 0.6833583245710576
```

Come è possibile notare l'accuratezza diminuisce molto lievemente ma rispetto ai risultati sui dati sbilanciati non si riscontra lo stesso overfitting.

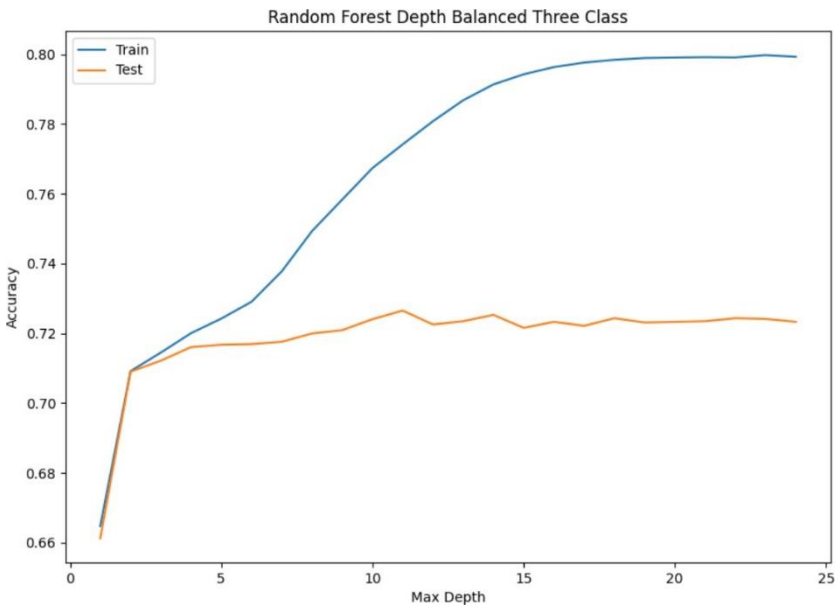
Per il resto dei risultati ci si concentra nuovamente sull'approccio a tre classi, come già spiegato precedentemente.

5.3.2 Logistic Regression

```
Accuracy: 0.732957012592271
F1 Score: 0.6850049444428711
Precision: 0.6868871694927537
Recall: 0.6838368402892847
```

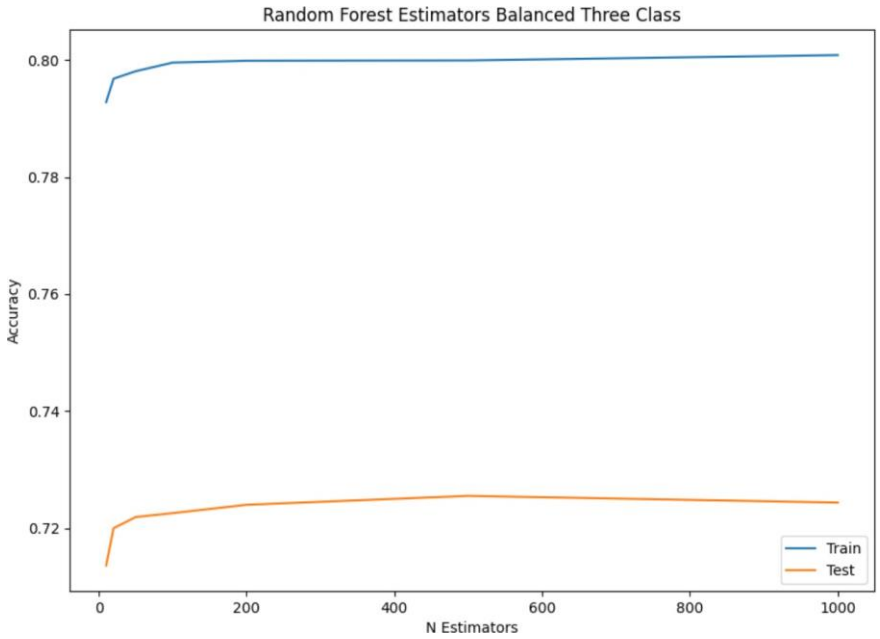
5.3.3 Random Forest

Al variare della massima profondità:




```
Accuracy: 0.7264437689969605
F1 Score: 0.6802996424132246
Precision: 0.6808218191562917
Recall: 0.6799684830830118
```

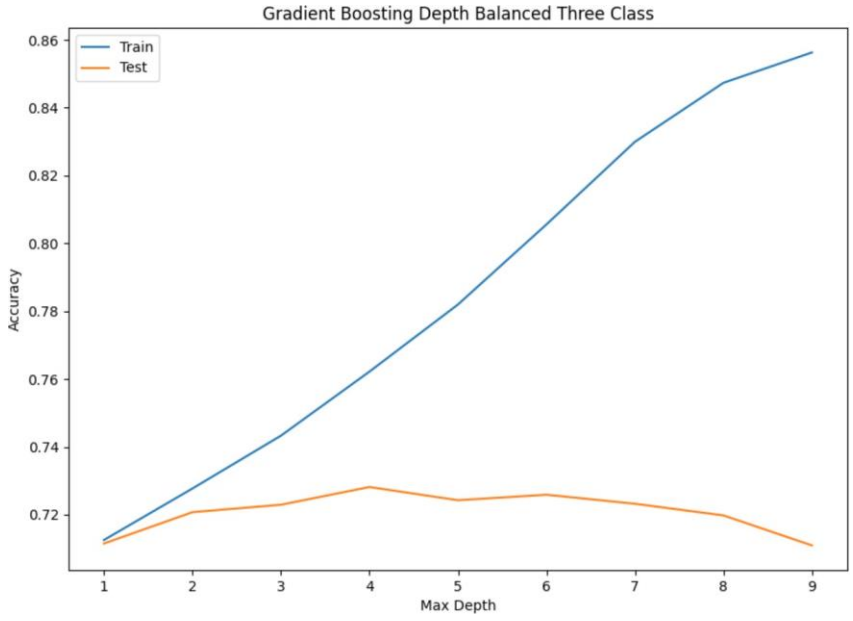
Al variare degli stimatori:




```
Accuracy: 0.7177594442032132
F1 Score: 0.668817527352522
Precision: 0.6695175945289421
Recall: 0.668332087180893
```

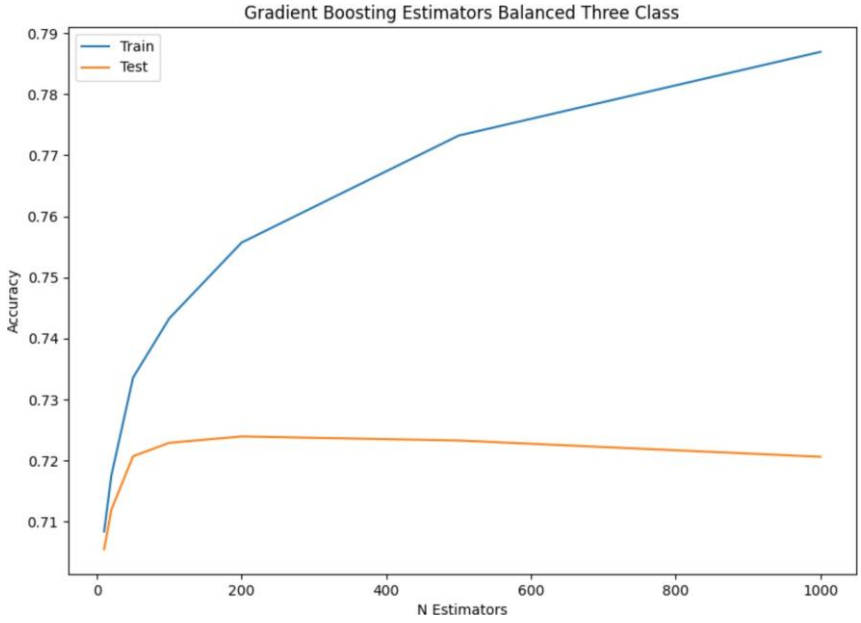
5.3.4 Gradient Boosting

Al variare della profondità:



```
Accuracy: 0.724706904038211
F1 Score: 0.6772267519614718
Precision: 0.6777186934863648
Recall: 0.6768327390669301
```

Al variare degli stimatori:



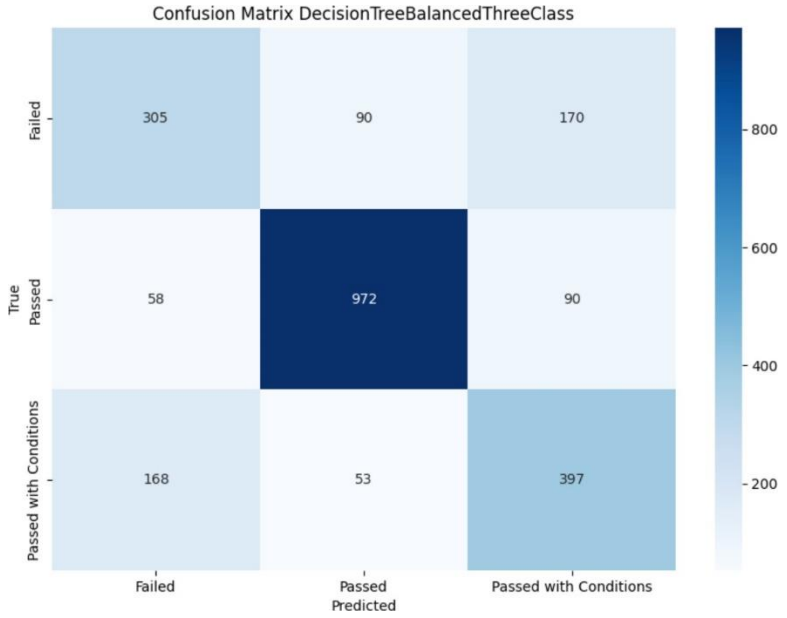

```
Accuracy: 0.7264437689969605
F1 Score: 0.678624708746987
Precision: 0.6799717548460343
Recall: 0.6779613934814123
```

5.3.5 Naive Bayes

```
Accuracy: 0.723404255319149
F1 Score: 0.6747945856741123
Precision: 0.6763682224974789
Recall: 0.674343093553829
```

5.3.6 Conclusioni

Come è possibile notare, a seguito del bilanciamento, l'overfitting risulta ridotto per il decision tree e il gradient boosting e praticamente assente per la random forest allenata al variare della profondità. Inoltre, si mostra la stessa matrice di confusione per il decision tree a seguito del bilanciamento



Come è possibile notare, in questo caso, gli esempi sulla classe fail, sono per la maggior parte corretti, al contrario della situazione precedente.

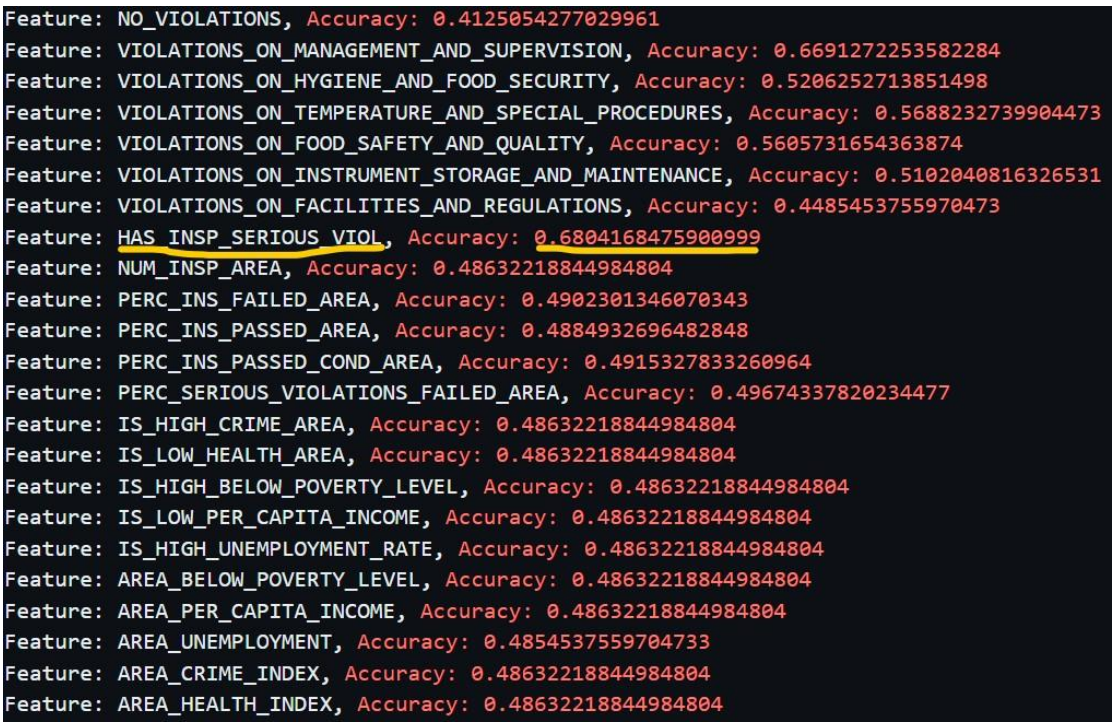
Si può quindi concludere che nonostante un leggero calo sui valori delle metriche, i modelli tendano ad avere migliori performance e a predire più correttamente i risultati.

5.4 Feature Importance

In seguito a questa fase di addestramento dei modelli, emerge che la regressione logistica multinomiale sia il modello le cui performance risultano migliori. Di conseguenza si è deciso di utilizzare tale modello per effettuare il seguente task: comprendere la feature col più alto potere predittivo.

Per fare ciò si è allenata la regressione logistica multinomiale sul dataset bilanciato considerando le tre classi.

I risultati emersi sono i seguenti:



Come è possibile notare la feature che risulta più importante è la feature has_insp_serious_viol, rappresentate la presenza di violazioni serie in una ispezione e inferita dalla knowledge base.

6. Rete Bayesiana

Oltre i task di apprendimento supervisionato si è strutturata una rete bayesiana con lo scopo di compiere inferenza probabilistica, ossia calcolare le probabilità di alcune variabili di interesse, date delle osservazioni. Difatti è sembrato di interesse osservare i legami in termini di probabilità tra i risultati di un'ispezione e le variabili che possano influenzarla, che siano indicanti violazioni o condizioni delle community area in cui è locata la struttura.

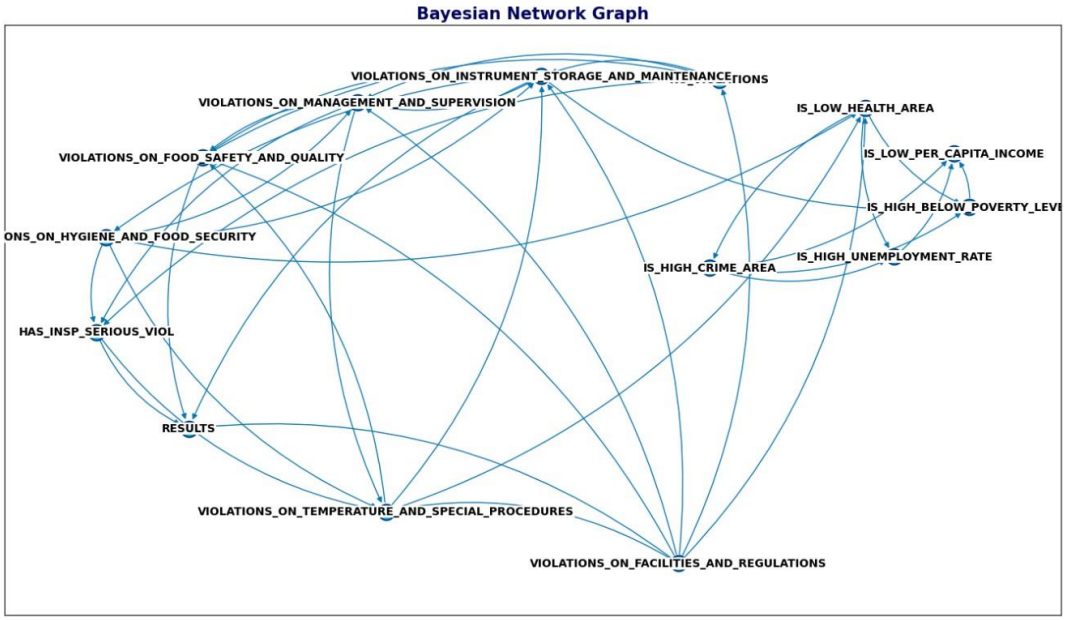
Per costruire la rete bayesiana sono state utilizzate features categoriche presenti all'interno del dataset, quali *'RESULTS', 'IS_HIGH_CRIME_AREA', 'IS_LOW_HEALTH_AREA', 'IS_HIGH_BELOW_POVERTY_LEVEL', 'IS_LOW_PER_CAPITA_INCOME', 'IS_HIGH_UNEMPLOYMENT_RATE','NO_VIOLATIONS','VIOLATIONS_ON_MANAGEMENT_AND_SUPERVISION','VIOLATIONS_ON_HYGIENE_AND_FOOD_SECURITY','VIOLATIONS_ON_TEMPERATURE_AND_SPECIAL_PROCEDURES','VIOLATIONS_ON_FOOD_SAFETY_AND_QUALITY','VIOLATIONS_ON_INSTRUMENT_STORAGE_AND_MAINTENANCE','VIOLATIONS_ON_FACILITIES_AND_REGULATIONS','HAS_INSP_SERIOUS_VIOL'*.

Ciò è stato fatto in quanto per apprendere la struttura della rete si è utilizzata la tecnica di HillClimbing. Usando le variabili categoriche le distribuzioni di probabilità potranno essere rappresentate usando tabelle di probabilità discreta (le CPT). Hill Climbing è stato usato

per trovare la struttura della rete che massimizzasse il BIC score (Bayesian Information Criterion).

La rete bayesiana è stata appresa a partire dai dati con i tre possibili esiti (Pass, Failed, Pass with Conditions).

Di seguito riportiamo il grafo della rete appresa mediante HillClimbing:



6.1 Query sulla rete Bayesiana

Sono state effettuate diverse query al fine di calcolare le probabilità condizionali di diverse variabili, date alcune evidenze.

In particolare, si presentano le seguenti:

```
Risultato della query che calcola la probabilità che una struttura abbia un basso reddito pro capite dato che ha passato l'ispezione:
+----------------------------+---------------------------------+
| IS_LOW_PER_CAPITA_INCOME   |   phi(IS_LOW_PER_CAPITA_INCOME) |
+============================+=================================+
| IS_LOW_PER_CAPITA_INCOME(0) |                          0.9464 |
+----------------------------+---------------------------------+
| IS_LOW_PER_CAPITA_INCOME(1) |                          0.0536 |
+----------------------------+---------------------------------+
```

```
Risultato della query che calcola la probabilità che una struttura abbia violazioni serie dato che ha passato l'ispezione:
+--------------------------+-----------------------------+
| HAS_INSP_SERIOUS_VIOL    |   phi(HAS_INSP_SERIOUS_VIOL) |
+==========================+=============================+
| HAS_INSP_SERIOUS_VIOL(0) |                      0.8900 |
+--------------------------+-----------------------------+
| HAS_INSP_SERIOUS_VIOL(1) |                      0.1100 |
+--------------------------+-----------------------------+
```

```
Risultato della query che calcola la probabilità dei risultato delle ispezioni pdato che una struttura ha violazioni serie:
+-------------------------------+----------------+
| RESULTS                       |   phi(RESULTS) |
+===============================+================+
| RESULTS(NOT PASS)             |         0.3975 |
+-------------------------------+----------------+
| RESULTS(PASS)                 |         0.1069 |
+-------------------------------+----------------+
| RESULTS(PASS WITH CONDITIONS) |         0.4956 |
+-------------------------------+----------------+
```

```
Risultato della query che calcola la probabilità che un'area abbia un alto tasso di disoccupazione dato che una struttura ha passato l'ispezione:
+-----------------------------+---------------------------------+
| IS_HIGH_UNEMPLOYMENT_RATE   |   phi(IS_HIGH_UNEMPLOYMENT_RATE) |
+=============================+=================================+
| IS_HIGH_UNEMPLOYMENT_RATE(0) |                          0.8118 |
+-----------------------------+---------------------------------+
| IS_HIGH_UNEMPLOYMENT_RATE(1) |                          0.1882 |
+-----------------------------+---------------------------------+
```

Le probabilità calcolate risultano essere coerenti con quanto ci si aspettava, difatti è più probabile che una struttura passi con condizioni o non passi l'ispezione se da questa emergono violazioni serie, oppure che data un'ispezione passata è molto più probabile che l'area della struttura sia un area in cui non vi è un alto tasso di disoccupazione.

6.2 Forward Sampling e probabilità a posteriori

Mediante il campionamento in avanti si è proceduto alla generazione di campioni che sono stati poi utilizzati come evidenza per il calcolo di probabilità a posteriori.

Di seguito mostriamo alcuni esempi:

```
Probabilità a posteriori di RESULTS data l'evidenza:
{'IS_LOW_HEALTH_AREA': 0, 'HAS_INSP_SERIOUS_VIOL': 0, 'VIOLATIONS_ON_INSTRUMENT_STORAGE_AND_MAINTENANCE': 0, 'IS_HIGH_BELOW_POVERTY_LEVEL': 0, 'IS_LOW_PER_CAPITA_INCOME'
: 0, 'VIOLATIONS_ON_MANAGEMENT_AND_SUPERVISION': 0, 'VIOLATIONS_ON_TEMPERATURE_AND_SPECIAL_PROCEDURES': 0, 'VIOLATIONS_ON_FOOD_SAFETY_AND_QUALITY': 0, 'NO_VIOLATIONS': 1
, 'VIOLATIONS_ON_FACILITIES_AND_REGULATIONS': 0, 'IS_HIGH_UNEMPLOYMENT_RATE': 0, 'IS_HIGH_CRIME_AREA': 0, 'VIOLATIONS_ON_HYGIENE_AND_FOOD_SECURITY': 0}
+-------------------------------+----------------+
| RESULTS                       |   phi(RESULTS) |
+===============================+================+
| RESULTS(NOT PASS)             |         0.0333 |
+-------------------------------+----------------+
| RESULTS(PASS)                 |         0.9360 |
+-------------------------------+----------------+
| RESULTS(PASS WITH CONDITIONS) |         0.0306 |
+-------------------------------+----------------+
Finding Elimination Order: : : 0it [00:00, ?it/s]
0it [00:00, ?it/s]
{'RESULTS': 'PASS'}
```

```
Probabilità a posteriori di RESULTS data l'evidenza:
{'IS_LOW_HEALTH_AREA': 0, 'HAS_INSP_SERIOUS_VIOL': 1, 'VIOLATIONS_ON_INSTRUMENT_STORAGE_AND_MAINTENANCE': 0, 'IS_HIGH_BELOW_POVERTY_LEVEL': 0, 'IS_LOW_PER_CAPITA_INCOME'
: 0, 'VIOLATIONS_ON_MANAGEMENT_AND_SUPERVISION': 1, 'VIOLATIONS_ON_TEMPERATURE_AND_SPECIAL_PROCEDURES': 0, 'VIOLATIONS_ON_FOOD_SAFETY_AND_QUALITY': 0, 'NO_VIOLATIONS': 0
, 'VIOLATIONS_ON_FACILITIES_AND_REGULATIONS': 1, 'IS_HIGH_UNEMPLOYMENT_RATE': 0, 'IS_HIGH_CRIME_AREA': 0, 'VIOLATIONS_ON_HYGIENE_AND_FOOD_SECURITY': 0}
+-------------------------------+----------------+
| RESULTS                       |   phi(RESULTS) |
+===============================+================+
| RESULTS(NOT PASS)             |         0.1323 |
+-------------------------------+----------------+
| RESULTS(PASS)                 |         0.2358 |
+-------------------------------+----------------+
| RESULTS(PASS WITH CONDITIONS) |         0.6319 |
+-------------------------------+----------------+
Finding Elimination Order: : : 0it [00:00, ?it/s]
0it [00:00, ?it/s]
{'RESULTS': 'PASS WITH CONDITIONS'}
```

```
Probabilità a posteriori di RESULTS data l'evidenza:
{'VIOLATIONS_ON_FACILITIES_AND_REGULATIONS': 1, 'IS_HIGH_UNEMPLOYMENT_RATE': 0, 'VIOLATIONS_ON_TEMPERATURE_AND_SPECIAL_PROCEDURES': 1, 'NO_VIOLATIONS': 0, 'VIOLATIONS_ON
_FOOD_SAFETY_AND_QUALITY': 1, 'IS_HIGH_CRIME_AREA': 0, 'HAS_INSP_SERIOUS_VIOL': 1, 'VIOLATIONS_ON_HYGIENE_AND_FOOD_SECURITY': 1, 'IS_LOW_PER_CAPITA_INCOME': 0, 'IS_HIGH_
BELOW_POVERTY_LEVEL': 0, 'VIOLATIONS_ON_INSTRUMENT_STORAGE_AND_MAINTENANCE': 1, 'VIOLATIONS_ON_MANAGEMENT_AND_SUPERVISION': 1, 'IS_LOW_HEALTH_AREA': 0}
+------------------------------+----------------+
| RESULTS                      |   phi(RESULTS) |
+==============================+================+
| RESULTS(NOT PASS)            |         0.7253 |
+------------------------------+----------------+
| RESULTS(PASS)                |         0.0006 |
+------------------------------+----------------+
| RESULTS(PASS WITH CONDITIONS) |         0.2741 |
+------------------------------+----------------+
Finding Elimination Order: : : 0it [00:00, ?it/s]
0it [00:00, ?it/s]
{'RESULTS': 'NOT PASS'}
```

È possibile osservare che la rete è riuscita a prevedere i risultati delle ispezioni sulla base del forward sampling.

7. Conclusioni

Il lavoro svolto potrebbe essere ulteriormente ampliato integrando nuovi dataset al fine di estendere la KB inferendo nuove features che possano risultare utili per migliorare la predizione dei modelli sulle tre classi.

La knowledge base può essere modificata considerando i risultati emersi durante il lavoro di feature importance.

Si potrebbe fare uso di altri modelli di apprendimento (come le reti neurali). Un'altra idea potrebbe essere l'applicazione di clustering su risorse online riguardanti le strutture come le recensioni degli utenti, al fine di osservare ulteriori informazioni distintive.