Immagine che contiene testo, moneta

Descrizione generata automaticamente

|  |
| --- |
| Immagine che contiene testo  Descrizione generata automaticamente  Scuola Politecnica e delle Scienze di Base  Corso di Laurea Magistrale in Ingegneria Informatica |

|  |
| --- |
| **Big Data Engineering**  Anno Accademico 2021/2022 |

Professore:

**Sperlì Giancarlo**

Studenti:

**Lamboglia Anna M63001219**

**Perrot Raffaella M63001135**

**Ricchiuti Fausto M63001144**

Sommario

[Big Data Engineering I](#_Toc105402950)

[Capitolo 1: Introduzione 3](#_Toc105402951)

[1.1 Descrizione della challenge 3](#_Toc105402952)

[1.2 Dataset utilizzato 3](#_Toc105402953)

[1.3 Strumenti Utilizzati 5](#_Toc105402954)

[1.3.1 MongoDB 5](#_Toc105402955)

[1.3.2 Python 5](#_Toc105402956)

[1.3.3 Spark 6](#_Toc105402957)

[1.3.4 Excel 7](#_Toc105402958)

[Capitolo 2: CREAZIONE DEL DATABASE 8](#_Toc105402959)

[CAPITOLO 3 : DATA ANALYSIS 9](#_Toc105402960)

[3.1 Trend Carriere Universitarie 10](#_Toc105402961)

[3.1.1 Iscritti per Anno Accademico 10](#_Toc105402962)

[3.1.2 Iscritti per Settore 12](#_Toc105402963)

[3.1.3 Immatricolati per Anno Accademico 13](#_Toc105402964)

[3.1.4 Immatricolati per Settore 15](#_Toc105402965)

[3.2 Trend Studenti Fuorisede 16](#_Toc105402966)

[3.2.1 Immatricolati Fuorisede Totali 16](#_Toc105402967)

[3.2.2 Fuorisede per Area di Provenienza 18](#_Toc105402968)

[3.1 Stampa dei risultati 20](#_Toc105402969)

[Capitolo 4: MODELLO PREDITTIVO 23](#_Toc105402970)

[4.1 Creazione del Dataset 23](#_Toc105402971)

[4.2 Modello Regressivo 27](#_Toc105402972)

[4.3 Dataset di Test 30](#_Toc105402973)

[4.4 Confronto con SciKit-Learn (SKLearn) 32](#_Toc105402974)

[Conclusioni 34](#_Toc105402975)

# Capitolo 1: Introduzione

### Descrizione della challenge

L’obiettivo della challenge è quello di effettuare uno studio sui trend nelle Università in Italia.

In particolare, ci si è focalizzati sui seguenti tre punti:

* Analizzare e descrivere i *Trend delle Carriere Universitarie*. Sono state estratte informazioni significative e interessanti utilizzando dati demografici, ambiti di studio, background personale e opportunità di lavoro;
* Analizzare e descrivere i *Trend del Decentramento degli Studenti* attraverso le scelte delle sedi universitarie. Sono state analizzate le scelte degli Studenti Fuorisede;
* *Prevedere i* *futuri Trend di Immatricolazione* sulla base di dati storici e fattori come l’istruzione o la demografia.

### Dataset utilizzato

Al fine di risolvere i tre punti sopraelencati, è stata svolta un’accurata ricerca per riuscire a trovare dati che fossero quanto più utili per l’analisi da effettuare.

In particolare, sono stati scaricati quattro file *.csv* messi a disposizione dal MIUR ([*ustat.miur.it/opendata*](http://ustat.miur.it/opendata)) e dall’ISTAT ([*dati.istat.it/popolazione*](http://dati.istat.it/Index.aspx?DataSetCode=DCIS_RICPOPRES2011)).

I quattro file scelti, disponibili inoltre nella Repository Github al link:

([*https://github.com/annalamboglia/KPMG-Challenge*](https://github.com/annalamboglia/KPMG-Challenge)) e sono strutturati come segue:

* *Immatricolati\_Anno\_Gruppo.csv* composto dalle seguenti cinque colonne*:*

1. *AnnoA,* l’Anno Accademico;
2. *GruppoNOME*, nella qualesono definite le tipologie di Università;
3. *Sesso;*
4. *Imm*, nella quale sono indicati il numero di Immatricolati;
5. *Note,* per definire delle note particolari.

A partire dai dati in esso contenuti è possibile ottenere il numero di Immatricolati per Anno Accademico, Settore Universitario e Sesso degli Studenti. Il file ha una dimensione pari a 96 Kb e presenta 360 istanze.

* *Iscritti\_Anno\_Gruppo.csv* composto dalle seguenti cinque colonne*:*

1. *\_id;*
2. *AnnoA,* l’Anno Accademico;
3. *GruppoNOME*, nella quale sono definite le tipologie di Università;
4. *Sesso;*
5. *Isc*, nella quale sono indicati il numero di Iscritti.

A partire dai dati in esso contenuti è possibile ottenere il numero di Iscritti per Anno Accademico, Settore Universitario e Sesso degli Studenti. Il file ha una dimensione pari a 89 Kb e presenta 330 istanze.

* *Immatricolati\_Residenza\_Ateneo.sv*. Il file è composto da otto colonne*:*

1. *\_id;*
2. *AnnoA,* l’Anno Accademico;
3. *AteneoCOD,* il codice associato ad un determinato Ateneo;
4. *SedeP*, ilcodice relativo ad una determinata Provincia;
5. *ResidenzaR,* la Regione di residenza;
6. *ResidenzaP,* la Provincia di residenza;
7. *Imm*, nella quale sono indicatiil numero di Immatricolati.

A partire dai dati in esso contenuti è possibile ottenere il numero di Immatricolati sulla base della Provincia di Residenza per ogni possibile coppia Anno Accademico – Ateneo. Il file ha una dimensione pari a 1.805 Kb e presenta 32000 istanze.

* *Popolazione Province.csv*. Il file è composto da tredici colonne:

1. *ITTER107*, id del Territorio;
2. *Territorio*, la Provincia o la Regione;
3. *TIPO\_DATO15*, il codice del periodo di riferimento;
4. *Tipo dato*, Popolazione del primo Gennaio;
5. *ETA1*, classe di Età (totale);
6. *Classe di età*, Totale;
7. *SEXISTAT1*, il codice per identificare il Sesso;
8. *Sesso*;
9. *CITTADINANZA*, la sigla della nazionalità;
10. *Cittadinanza*, la sigla della nazionalità;
11. *TIME*, l’anno di riferimento;
12. *Seleziona periodo*, l’anno di riferimento;
13. *Value*, la popolazione.

A partire dai dati in esso contenuti è possibile ottenere la Popolazione corrispondente a qualsiasi coppia Regione/Provincia – Anno. Il file ha una dimensione pari a 502 Kb e presenta 4051 istanze.

### Strumenti Utilizzati

Durante le Fasi di Sviluppo sono stati utilizzati i seguenti strumenti:

* *Mongo DB;*
* *Python;*
* *Pyspark;*
* *Excel.*

#### 1.3.1 MongoDB

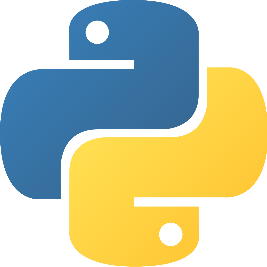
Per memorizzare i dati sopra descritti si è deciso di utilizzare MongoDB, un DBMS non relazionale sviluppato in C++, open-source, document-oriented e scalabile, caratterizzato da alte prestazioni sia in lettura che in scrittura. Tra i vantaggi derivanti dal suo utilizzo si osserva che le letture più consistenti possono essere distribuite su più server replicati; inoltre le interrogazioni sono più semplici e veloci grazie all’approccio ai documenti che rende possibile la rappresentazione di relazioni gerarchiche complesse mediante l’utilizzo di documenti nidificati e array. Le caratteristiche principali di MongoDB sono le seguenti:

* **Database document-oriented:** i dati vengono archiviati sotto forma di documenti in formato JSON;
* **Supporto completo agli indici**: indicizzazione di qualsiasi attributo;
* **Replicazione**: facilità nella replicazione dei dati attraverso la rete e alta scalabilità;
* **Sharding**: scalabilità orizzontale senza compromettere nessuna funzionalità.



#### 1.3.2 Python

Python è un linguaggio di programmazione di "alto livello", orientato agli oggetti, adatto, tra gli altri usi, a sviluppare applicazioni distribuite, scripting, computazione numerica e system testing.



#### 1.3.3 Spark

Per poter lavorare sui dati ed applicare tecniche di Machine Learning è stato adoperato *Pyspark*, una libreria attraverso la quale è possibile l’utilizzo di Apache Spark in Python.

*Apache Spark* è un framework di elaborazione parallela open source, che supporta l’elaborazione in memoria al fine di migliorare le prestazioni delle applicazioni che analizzano Big Data.

Le caratteristiche principali di Spark sono le seguenti:

* **Fast processing:** Spark contiene Resilient Distributed Dataset (RDD), che consentono di risparmiare tempo necessario per le operazioni di lettura e scrittura, permettendo di ottenere un tempo di elaborazione più veloce rispetto ad Hadoop, di uno o due ordini di grandezza;
* **In-Memory computing:** in Spark i dati sono salvati in RAM (Random Access Memory), in questo modo è possibile accedere a quest’ultimi in modo più veloce, incrementando le prestazioni del sistema nella fase di analytics;
* **Flexible**: Spark supporta diversi linguaggi e consente agli sviluppatori di realizzare applicazioni in Java, Scala, R o Python;
* **Fault tolerance.** Spark contiene Resilient Distributed Dataset (RDD), progettati per gestire il guasto di un qualsiasi nodo worker del cluster. In questo modo, si garantisce che la perdita di dati sia pressocché pari a zero;
* **Better Analytics**: Spark ha a disposizione una ricca serie di funzionalità, come la possibilità di effettuare query SQL oppure applicare algoritmi di Machine Learning, che consentono di migliorare le performance in fase di analisi.

#### 1.3.4 Excel

*Microsoft Excel* è un programma prodotto da Microsoft, dedicato alla produzione ed alla gestione di fogli elettronici. È parte della suite di software di produttività personale Microsoft Office, ed è disponibile per i sistemi operativi Windows e MacOS. È il programma per la produzione e gestione di fogli elettronici più utilizzato.



# Capitolo 2: CREAZIONE DEL DATABASE

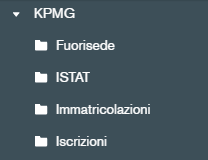
****La prima fase di lavoro è stata incentrata sulla progettazione e realizzazione della Base di Dati. In particolare, si è scelto di creare su MongoDB un nuovo Database dal nome *KPMG* costituito di quattro Collection, ad ognuna delle quali è stato assegnato uno dei file *.csv* descritti nel precedente capitolo. Il Database è stato popolato come segue:

Figure 1: Import su Compass

* **Iscrizioni**: *Iscritti\_Anno\_Gruppo.csv;*
* **Immatricolazioni**: *Immatricolati\_Anno\_Gruppo.csv;*
* **Fuorisede**: *Immatricolati\_Residenza\_Ateneo.csv*;
* **ISTAT**: *Popolazione\_Province.csv.*

CAPITOLO 3 : DATA ANALYSIS

Nel seguente capitolo sono mostrate le tecniche utilizzate durante la fase di *Data Analysis* affiché fosse possibile visualizzare i risultati desiderati a partire dai dati presenti nel Database. In particolare, l’elaborazione è stata effettuata mediante script Python, descritti in dettaglio nel proseguo del capitolo.

Il primo passo elaborativo è descritto dal seguente codice, il cui effetto è quello di importare le API di MongoDB lato Client e le altre librerie necessarie per gli script successivi, ed instaurare una connessione con il Database *KPMG* e le sue collezioni.

Immagine che contiene testo

Descrizione generata automaticamente

### *Trend Carriere Universitarie*

Nel corrente paragrafo sono illustrate nel dettaglio le procedure elaborative realizzate per descrivere gli *Attuali Trend di Carriera Universitaria.*

#### 3.1.1 Iscritti per Anno Accademico

Il codice che segue consente di associare ad ogni coppia *Anno\_Accademico – Sesso* il corrispondente numero di *Iscritti*. A tal fine è necessario effettuare un raggruppamento dei documenti presenti nella Collection *Iscritti* sugli attributi *AnnoA* e *SESSO*, sommando tra loro i valori del campo *Isc*. Il risultato della query è poi memorizzato nella struttura dati locale *IscrizioniTotali*.

Immagine che contiene testo

Descrizione generata automaticamente

Dai risultati sono stati ottenuti i seguenti grafici:



È possibile notare come i primi anni della passata decade siano caratterizzati da un trend fortemente decrescente del numero totale di Iscritti, invertito poi negli anni più recenti.



Inoltre dai diagrammi a torta è possibile notare come ci sia una significativa prevalenza di Studenti di Sesso femminile tra gli Iscritti delle Università italiane.

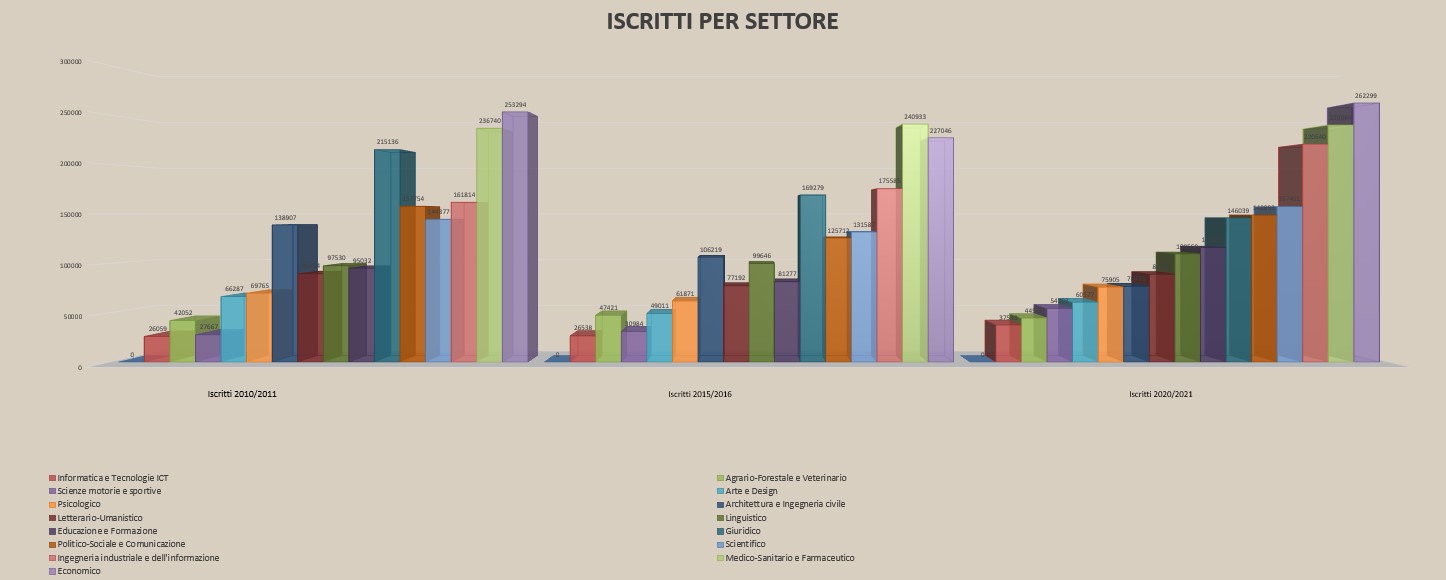
#### 3.1.2 Iscritti per Settore

Il seguente codice consente invece di associare ad ogni coppia *Anno\_Accademico – Settore* il corrispondente numero di *Iscritti*. A tal fine è necessario effettuare un raggruppamento dei documenti presenti nella Collection *Iscritti* sugli attributi *AnnoA* e *GruppoNOME*, sommando tra loro i valori del campo *Isc*. Il risultato della query è poi memorizzato nella struttura dati locale *IscrizioniPerSettore*.

Immagine che contiene testo

Descrizione generata automaticamente

Dai risultati è stato ottenuto il seguente grafico:



Dal grafico è possibile notare come il Settore *Giuridico* abbia perso un considerevole numero di Iscritti negli anni, al contrario del Settore dell’*Ingegneria Industriale e dell’Informazione*. I Settori più numerosi nell’Anno Accademico 2020/2021 sono: *Economico*, *Medico - Sanitario e Farmaceutico* e *Ingegneria Industriale e dell’Informazione*.

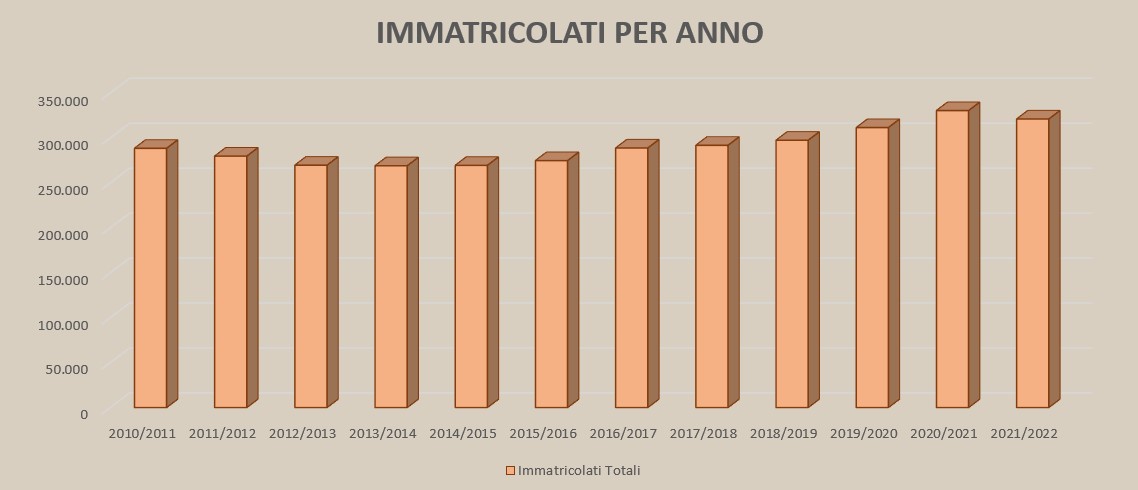
#### 3.1.3 Immatricolati per Anno Accademico

Il codice proposto consente di associare ad ogni coppia *Anno\_Accademico – Sesso* il corrispondente numero di *Immatricolati*. A tal fine è necessario effettuare un raggruppamento dei documenti presenti nella Collection *Immatricolati* sugli attributi *AnnoA* e *SESSO*, sommando tra loro i valori del campo *Imm*. Il risultato della query è poi memorizzato nella struttura dati locale *ImmatricolazioniTotali*.

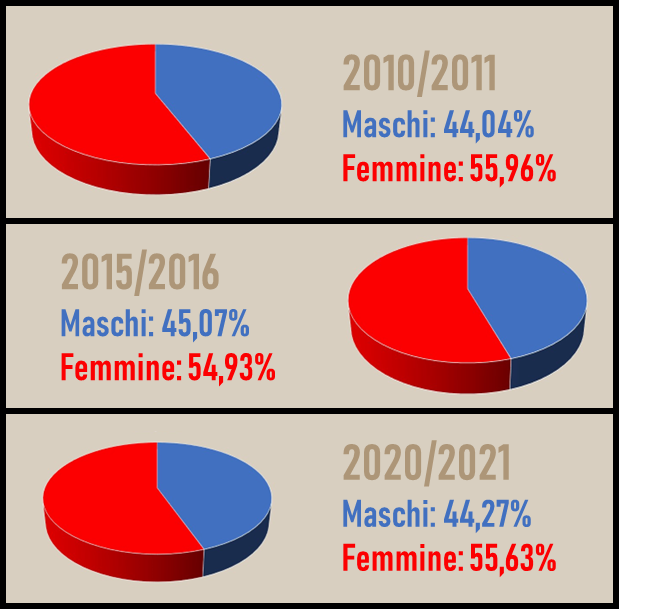
Immagine che contiene testo

Descrizione generata automaticamente

Dai risultati sono stati ottenuti i seguenti grafici:



È possibile notare come i primi anni della passata decade siano caratterizzati da un trend leggermente decrescente del numero totale di nuovi Immatricolati, invertito poi negli anni più recenti.



Inoltre, dai grafici a torta è possibile notare come ci sia una significativa prevalenza di Studenti di Sesso femminile tra i nuovi Immmatricolati delle Università italiane.

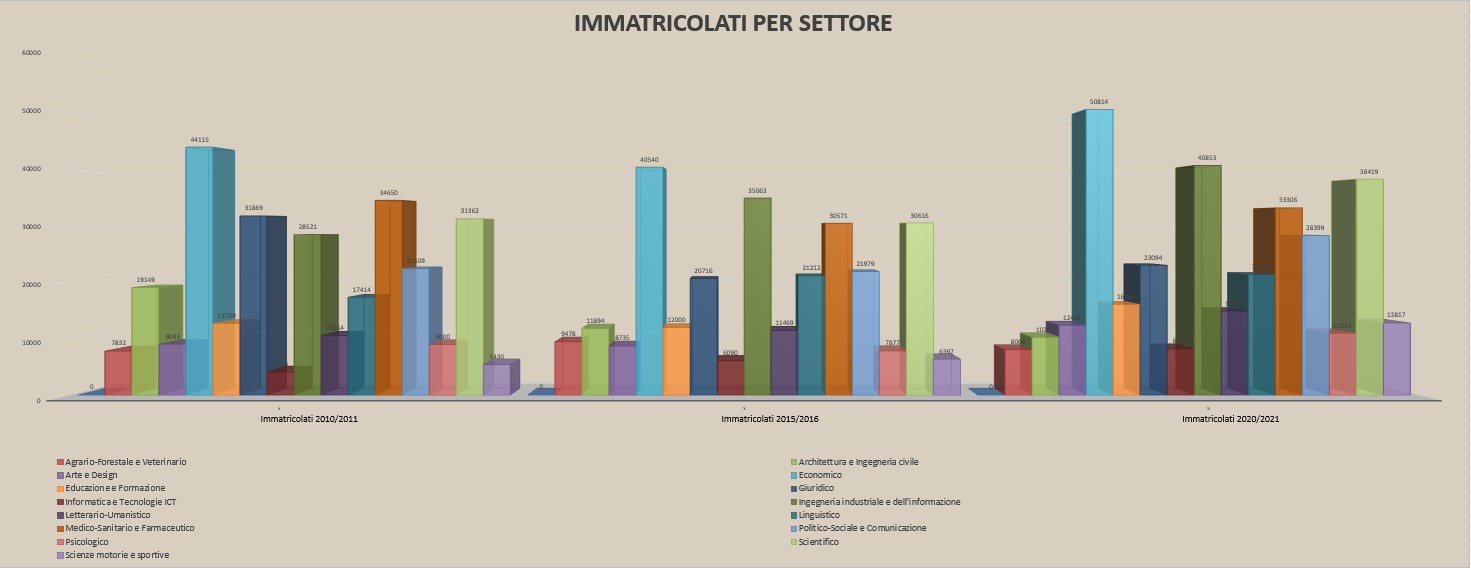
#### 3.1.4 Immatricolati per Settore

Il seguente codice consente invece di associare ad ogni coppia *Anno\_Accademico – Settore* il corrispondente numero di *Immatricolati*. A tal fine è necessario effettuare un raggruppamento dei documenti presenti nella Collection *Immatricolati* sugli attributi *AnnoA* e *GruppoNOME*, sommando tra loro i valori del campo *Imm*. Il risultato della query è poi memorizzato nella struttura dati locale *ImmatricolazioniPerSettore*.

Immagine che contiene testo

Descrizione generata automaticamente

Dai risultati è stato ottenuto il seguente grafico:



Dal grafico è possibile notare come i Settore dell’*Ingegneria Industriale e dell’Informazione* e dell’*Architettura e Ingegneria Civile* siano caratterizzati da un considerevole incremento del numero di immatricolati negli Anni. I Settori con il maggior numero di nuovi Immatricolati nell’Anno Accademico 2020/2021 sono: *Economico*, *Ingegneria Industriale e dell’Informazione* e *Architettura e Ingegneria Civile*.

### *Trend Studenti Fuorisede*

Nel corrente paragrafo sono illustrate nel dettaglio le procedure elaborative realizzate per descrivere i *Trend di Immatricolazione degli Studenti Fuorisede.*

#### 3.2.1 Immatricolati Fuorisede Totali

Il seguente codice consente di filtrare i dati della collezione *Fuorisede*, eliminando tutte le entries in cui:

* il nome della *Regione* o della *Provincia* è incluso nel nome dell’*Ateneo*;
* nel nome dell’Ateneo è presente l’attributo “*telematica*” o “*stranieri*”.

Il risultato dello script sono le seguenti due liste :

* *FuorisedeTotali*, che ad ogni Anno Accademico associa il relativo numero di Studenti Fuorisede;
* *FuorisedePerAteneo*, che ad ogni coppia Anno\_Accademico – Ateneo associa il relativo numero di Studenti Fuorisede.

Immagine che contiene testo

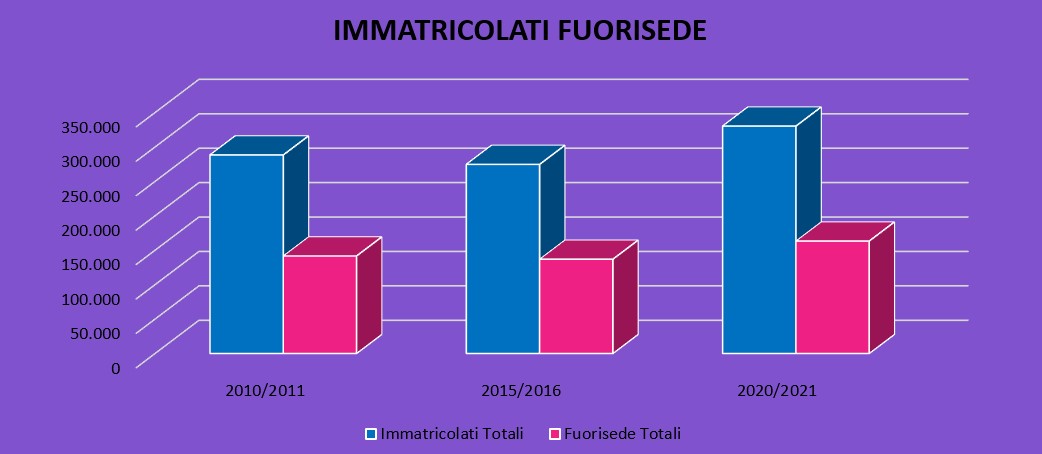
Descrizione generata automaticamente

Immagine che contiene testo

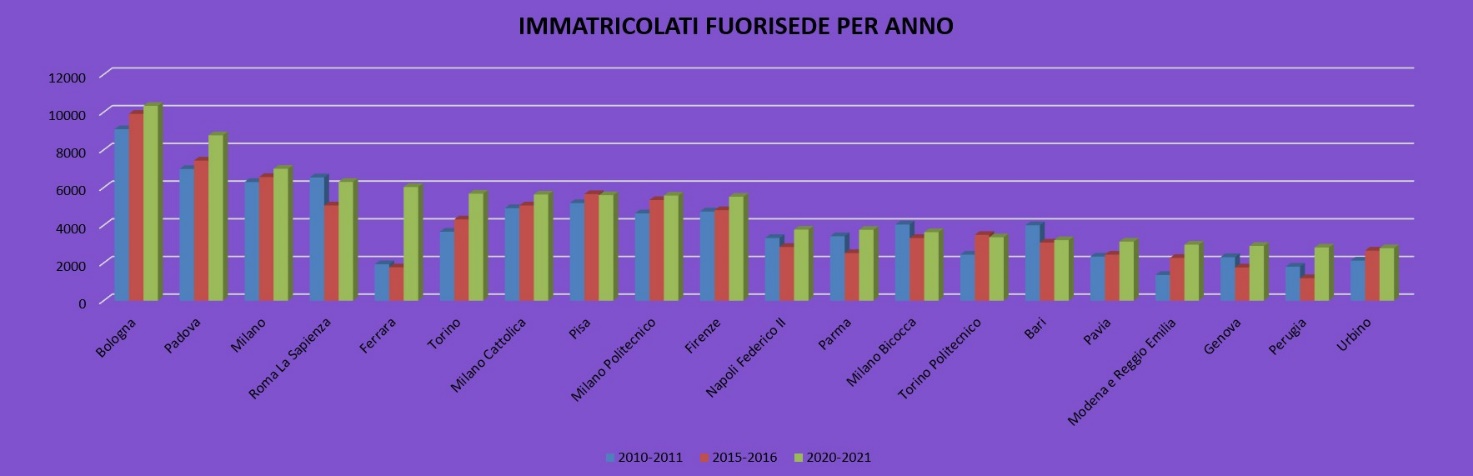
Descrizione generata automaticamente

Per la graficare i risultati si è scelto di prendere come riferimento i seguenti tre Anni Accademici: 2010/2011, 2015/2016 e 2020/2021.

Di seguito si riporta l’istogramma che mostra il numero di *Immatricolati fuorisede* rispetto agli *Immatricolati totali* negli anni:



Come si evince dal grafico, circa la metà degli Immatricolati totali, per tutti e tre gli anni considerati, sono Studenti Fuorisede.

Si riporta di seguito l’istogramma che mostra il numero di Immatricolati Fuorisede divisi per Ateneo:

Come si evince dal grafico, la facoltà di Bologna è quella con il numero maggiore di Immatricolati Fuorisede per tutti e tre gli anni di riferimento.

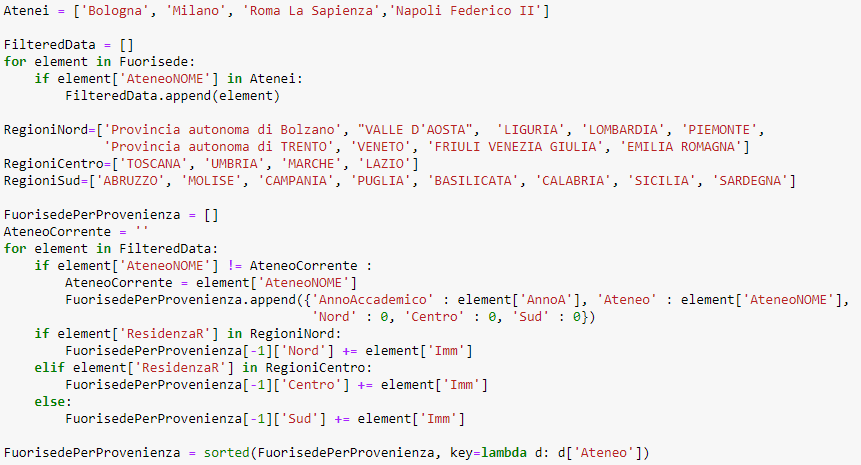
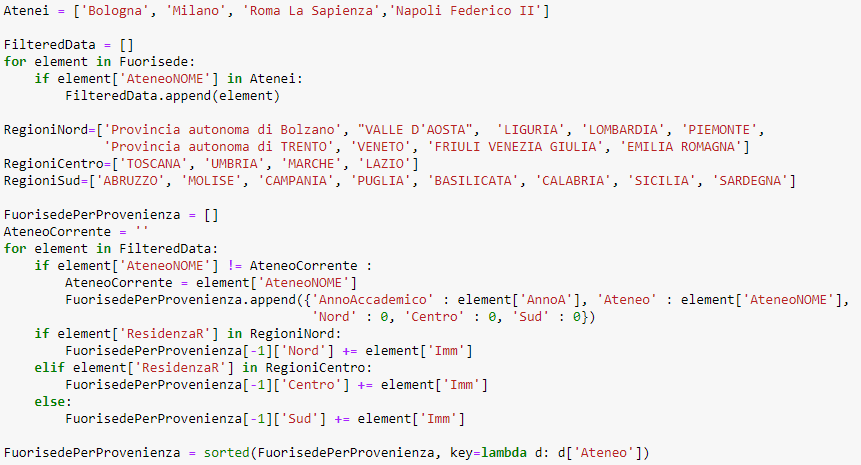
#### 3.2.2 Fuorisede per Area di Provenienza

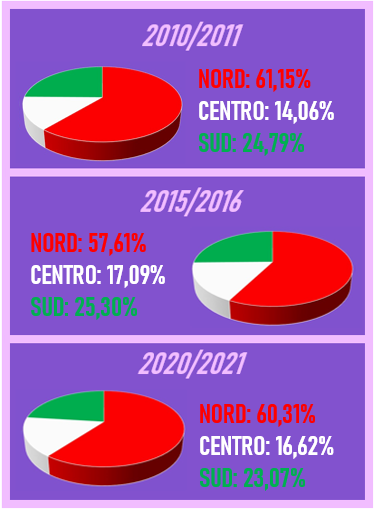
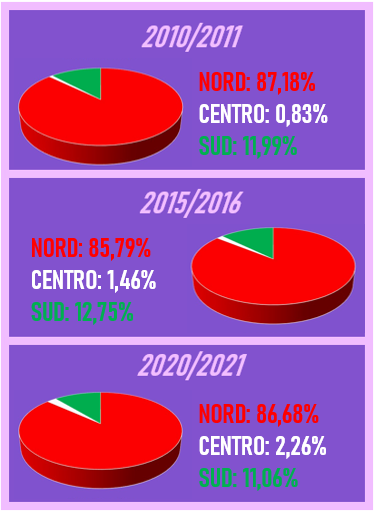
Sono stati analizzati, inoltre, il numero di Studenti Fuorisede per area di provenienza.

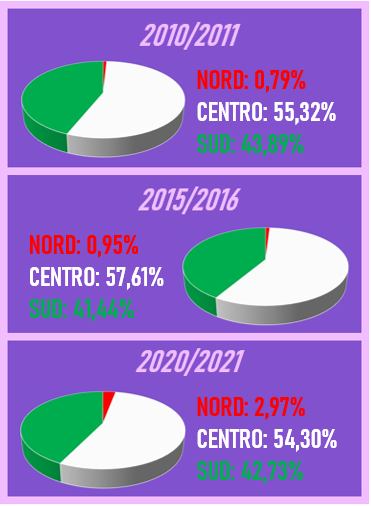
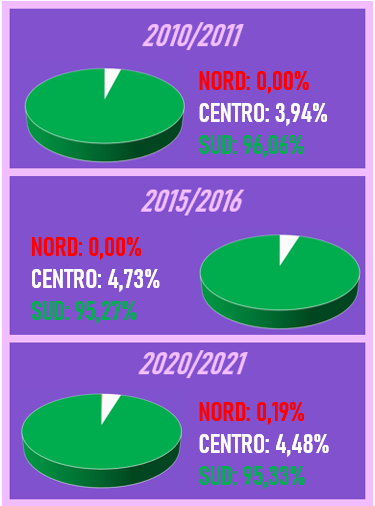
Sono state selezionate quatto Università di interesse:

* *Bologna*, in quanto Università con il più alto numero di Studenti Fuorisede;
* *Milano*, come ‘rappresentante’ delle Università del Nord;
* *Roma La Sapienza*, come ‘rappresentante’ delle Università del Centro;
* *Napoli Federico II*, come ‘rappresentante’ delle Università del Sud.

Per ognuna di esse sono state analizzate le percentuali di studenti Fuorisede provenienti dal Nord, dal Centro e dal Sud. Si riporta il codice prodotto per effettuare l’attività elaborativa appena descritta:



Bologna: Milano:

 Roma: Napoli:

### Stampa dei risultati

Per esportare i risultati ottenuti e realizzare i grafici mostrati nei paragrafi precedenti, si è scelto di creare una *Cartella di Lavoro Excel* costituita di sette fogli, ognuno dei quali contenente una delle strutture dati sviluppate durante l’elaborazione mostrata in precedenza. Segue una descrizione dettagliata del contenuto di ciascun *Foglio* del file *Risultati.xlsx* generato:

* *Iscrizioni per A.A.* costituito dalle seguenti quattro colonne:

1. *AnnoAccedemico*;
2. *Iscritti*;
3. F;
4. M.

* *Iscrizioni per Settore* costituito dalle seguenti tre colonne:

1. *AnnoAccedemico*;
2. *Settore*;
3. *Iscritti*.

* *Immatricolazioni per A.A.* costituito dalle seguenti quattro colonne:

1. *AnnoAccademico*;
2. *Immatricolati*;
3. F;
4. M.

* *Immatricolazioni per Settore* costituito dalle seguenti tre colonne:

1. *AnnoAccedemico*;
2. *Settore*;
3. *Immatricolati*.

* *Fuorisede per A.A*. costituito dalle seguenti due colonne:

1. *AnnoAccedemico*;
2. *Fuorisede*.

* *Fuorisede per Ateneo* costituito dalle seguenti due colonne:

1. *AnnoAccedemico*;
2. *Ateneo*;
3. *Fuorisede*.

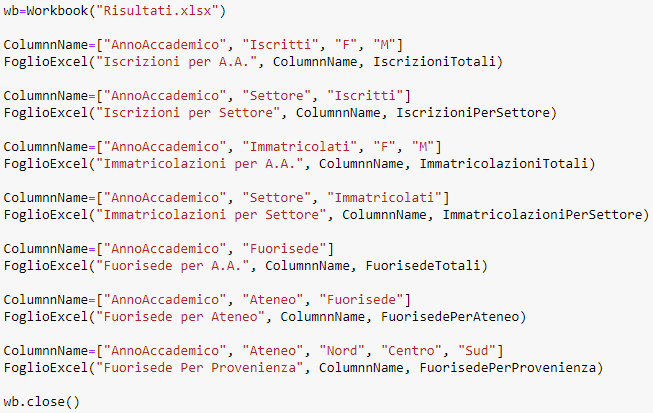
* *Fuorisede Per Provenienza* costituito dalle seguenti cinque colonne:

1. *AnnoAccademico*;
2. *Ateneo*;
3. *Nord*;
4. *Centro*;
5. *Sud*.

Segue, infine, il codice realizzato per generare il file Excel appena descritto :

Immagine che contiene testo

Descrizione generata automaticamente



# Capitolo 4: MODELLO PREDITTIVO

Nel seguente capitolo è mostrato l’approccio utilizzato per risolvere il task legato alla predizione dei Trend di Iscrizione degli Studenti Universitari nei prossimi anni. A tale scopo, si è ritenuto necessario l’utilizzo della liberia Python *Pyspark*, in modo da potersi avvalere del componente Mllib di Apache Spark, che permette di effettuare operazioni di Machine Learning.

Inoltre, è stato effettuato un confronto tra le performance della soluzione ottenuta con *Pyspark* e quelle dell’analogo processo svolto mediante l’utilizzo della libreria Python *Scikit-Learn*.

### 4.1 *Creazione del Dataset*

La prima fase necessaria per la risoluzione del task appena descritto è la *Creazione del Dataset*, il cui primo step è stato quello di comprendere quali potessero essere le *features* di interesse per l’analisi in oggetto.

Pertanto, in seguito ad un confronto tra gli autori, sono state valutate le *features* di maggiore interesse, di seguito riportate:

* *AnnoAccademico*: Anno Accademico oggetto di analisi;
* *Ateneo*: Ateneo a cui si riferiscono le elaborazioni;
* *PopolazioneP*: Numero di abitanti della Provincia di cui fa parte l’Ateneo analizzato;
* *AteneiP*: Numero di Atenei presenti nella Provincia dell’Ateneo di riferimento;
* *PopolazioneR*: Numero di abitanti della Regione di cui fa parte l’Ateneo analizzato;
* *AteneiR*: Numero di Atenei presenti nella Regione dell’Ateneo di riferimento;
* *Fuorisede*: Numero di Studenti dell’Ateneo di riferimento, residenti in una Provincia diversa da quella dell’Ateneo stesso, ottenuto grazie alle elaborazioni descritte nel precedente capitolo;
* *Immatricolati: N*umero di Studenti Iscritti al primo anno nell’Ateneo e nell’*AnnoAccademico* di riferimento.

Una volta definite le *features* del Dataset, bisogna comprendere come popolarlo.

È possibile notare che, grazie alle elaborazioni descritte nel capitolo precedente, si è già in possesso dei dati necessari per popolare le features *AnnoAccademico, Ateneo, Fuorisede* ed *Immatricolati.*

Per ottenere i dati sulla Popolazione è stato invece importato nel progetto Python il contenuto della Collection ISTAT del database, filtrando i soli dati relativi alle popolazioni di Province e Regioni negli anni che vanno dal 2010 al 2019.

Immagine che contiene testo

Descrizione generata automaticamente



Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Per ottenere i dati necessari a popolare le features *AteneiP* e *AteneiR*, sono state prodotte le funzioni *ContaAteneiProvincia* e *ContaAteneiRegione*, le quali, ricevuto in input rispettivamente il nome della Provincia e della Regione, restituiscono in output il numero di Atenei presenti nell’area di interesse.

Immagine che contiene testo

Descrizione generata automaticamente

Il codice che segue consente di effettuare in modo opportuno il *merge* dei dati per costruire correttamente Dataset ed esportarlo su un file *.xlsx* .

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Segue un’anteprima del Dataset prodotto:

Immagine che contiene tavolo

Descrizione generata automaticamente

### 4.2 *Modello Regressivo*

Una volta costruito il *Dataset*, è possibile realizzare un modello di *Machine Learning* utilizzando la libreria Python *Pyspark*.

In particolare, volendo ottenere una previsione dell’andamento degli immatricolati per gli anni futuri, è stato deciso di realizzare un *Modello Regressivo*. Tale approccio presuppone preliminarmente un *preprocessing* dei dati, in quanto sono presenti due colonne che con dati di tipo *string*, ossia *AnnoAccademico* e *Ateneo*.

Quindi, per ovviare a tale problematica, una volta caricato il dataset è stato applicato il processo di *One Hot Encoding*, trasformando preliminarmente il tipo *string* in un array all’interno delle colonne denominate *AnnoAccademico\_Array* e *Ateneo\_Array*. Segue il codice Python prodotto per realizzare le attività elaborative appena descritte:

Immagine che contiene testo

Descrizione generata automaticamente

Figura 8: Creazione delle colonne AnnoAccademico\_Array e Ateneo\_Array

Successivamente, è stato applicato il *One Hot Encoding* sulle colonne appena create:

Immagine che contiene testo

Descrizione generata automaticamente

Figura 9: One Hot Encoding

Infine, per terminare la fase di preprocessing, sono state rimosse le colonne *AnnoAccademico*, *Ateneo, AnnoAccademico\_Array* e *Ateneo\_Array.*

A questo punto, è necessario definire le colonne *features* e *label*:

Immagine che contiene testo

Descrizione generata automaticamente

Figura 10: Creazione del Dataframe

Ottenendo dunque:

Immagine che contiene tavolo

Descrizione generata automaticamente

Figura 11: Aggiunta delle colonne Features e Label

Una volta ottenute le colonne *features* e *label*, è stato effettuato lo split del dataset in *train*, *validation* e *test*:

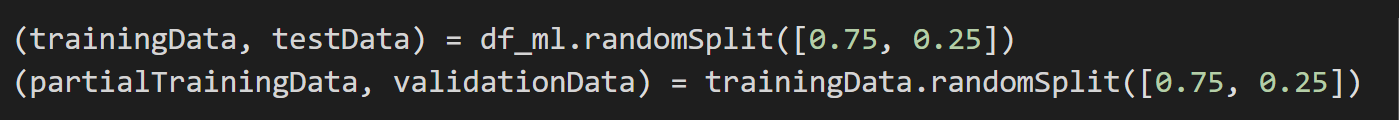


Figura 12: Split del dataset in Train, Validation e Test

In tal senso, è stato dunque suddiviso il dataset come segue:

* **75%**: *training*, di cui è stata effettuata la seguente ulteriore suddivisione per ottimizzare gli iperparametri:
* 75%: *training*;
* 25%: *validation*.
* **25%**: *test*.

Inoltre, sono stati applicati tre algoritmi di regressione differenti: *Linear Regressor*, *Random Forest Regressor* e *GBT Regressor*. In particolare, per il *Random Forest Regressor*, sono stati impostati un *numTrees* pari a 1000 ed una *maxDepth* pari a 3, mentre per il *GBT Regressor* sono stati impostati un *maxIter* pari a 100, ed una *stepSize* di 0.01, ottenendo i seguenti risultati:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ***Linear Regressor*** | | ***Random Forest Regressor*** | | ***GBT Regressor*** | |
| **MAE [Immatricolati]** | ***R 2*** | **MAE [Immatricolati]** | ***R 2*** | **MAE [Immatricolati]** | ***R 2*** | |
| **Validation** | *≃ 160* | ***-0.021*** | *≃ 139* | ***0,127*** | *≃ 104* | ***0,222*** | |
| **Test** | *≃ 161* | ***0.004*** | *≃ 143* | ***0.148*** | *≃ 109* | ***0.261*** | |
| **Re-Train** | *≃ 163* | ***-0.024*** | *≃ 144* | ***0.168*** | *≃ 101* | ***0.391*** | |

Figura 13: Tabella di confronto dei vari algoritmi

Come è possibile notare, i risultati migliori sono stati ottenuti applicando il *GBT Regressor*, dunque sarà quest’ultimo l’algoritmo utilizzato per la predizione finale.

### 4.3 *Dataset di Test*

Una volta creato il dataset e scelto il modello, la fase successiva è la definizione del *dataset di test*. Infatti, sebbene dati come *Anno Accademico* ed *Ateneo* possano essere semplici da definire, per quanto riguarda i dati della *Popolazione* e dei *Fuorisede*, risulta complesso definire come debbano essere popolate le corrispondenti colonne.

La soluzione che è stata utilizzata consiste nell’applicazione della seguente formula per calcolare l’incremento o decremento annuale della Popolazione e dei Fuorisede:

In figura, è riportato un esempio del calcolo che è stato effettuato:

Immagine che contiene testo, verde, segnale, puntamento

Descrizione generata automaticamente

Figura 14: Esempio di applicazione della formula

Di seguito, viene riportato il codice Python che è stato scritto per poter automatizzare questo calcolo:

Immagine che contiene testo

Descrizione generata automaticamente

Figura 15: Calcolo della media degli scarti annuali

A questo punto, è stato applicato l’algoritmo di predizione sulle istanze di test create, in modo da predire l’andamento degli Immatricolati dal 2021/2022, fino al 2025/2026 per ogni singolo Ateneo.

Nelle seguenti figure sono riportati alcuni dei risultati relativi ai valori di immatricolazione ottenuti dalla predizione[[1]](#footnote-1). In particolare, è stato deciso di mostrare *in primis* l’andamento nel tempo delle immatricolazioni totali in Italia (figura 16), dal quale è possibile notare come il trend risulti essere in crescita. Successivamente è stata replicata l’analisi distinguendo tra i diversi Atenei; per brevità sono stati riportati gli istogrammi relativi agli Atenei “campione” di *Bologna*, *Milano*, *Napoli Federico II* e *Roma La Sapienza* (figura 17)[[2]](#footnote-2), utilizzati come riferimento anche negli scorsi capitoli, denotando un andamento crescente in tutti e quattro i casi. E’ stato inoltre mostrato come tale andamento non sia identico per tutti gli Atenei, in quanto ad esempio l’Ateneo di *Bari* presenta un trend di immatricolazioni decrescente per i prossimi cinque Anni Accademici (figura 18).

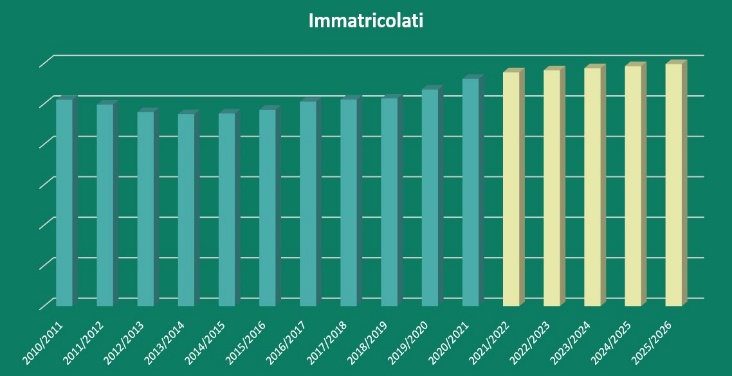


Figura 16: Trend delle immatricolazioni totali



Figura 17: Trend degli Atenei presi in considerazione

Immagine che contiene tavolo

Descrizione generata automaticamente

Figura 18: Trend di immatricolazioni per l’ateneo di Bari

### 4.4 *Confronto con SciKit-Learn (SKLearn)*

Nello scorso paragrafo, sono stati riportati i risultati delle performances fornite dai regressori testati con Pyspark, ossia *Linear Regressor, Random Forest Regressor* e *GBT Regressor*,ottenuti con i vari split in *Train*, *Validation* e *Test*.

Sebbene il *MAE* (*Mean Absolute Error*) ottenuto fosse abbastanza basso, ciò che ha messo in dubbio la bontà dei risultati ottenuti è stato l’indice R2, abbastanza vicino allo zero.

Per cui, al fine di ottenere dei risultati più soddisfacenti, è stato deciso di applicare le stesse considerazioni alla base dell’analisi precedente in *Pyspark*, utilizzando questa volta la libreria *Scikit-learn*. Nel dettaglio, è stata replicata la stessa fase di preprocessing dei dati ed è stato nuovamente suddiviso il dataset in *Train*, *Validation* e *Test*.

Di seguito sono riportati degli esempi dei risultati ottenuti, differenti tra loro, dove è stato utilizzato lo stesso algoritmo, oltre che un confronto tra i migliori algoritmi che si sono trovati in entrambi i casi.

Prima di tutto, si riporta il caso in cui è stato utilizzato lo stesso regressore con gli stessi iperparametri; in particolare, è stato provato il *Random Forest Regressor* con un *numTrees* pari a 1000 ed una *maxDepth* pari a 3, ottenendo i seguenti risulati e tempi di training:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ***PySpark (RFRegressor)*** | | | ***Sklearn (RFRegressor)*** | | |
| **MAE [Immatricolati]** | ***R 2*** | **Time[s]** | **MAE [Immatricolati]** | ***R 2*** | **Time[s]** |
| **Validation** | *≃ 139* | ***0,127*** | *≃ 2.843* | *≃ 882* | ***0,894*** | *≃ 1.54* |
| **Test** | *≃ 143* | ***0.148*** | *≃ 2.843* | *≃ 1063* | ***0.842*** | *≃ 1.54* |
| **Re-Train** | *≃ 144* | ***0.168*** | *≃ 4.897* | *≃ 1050* | ***0.840*** | *≃ 2.69* |

Come è possibile osservare, i risultati risultano abbastanza differenti in quanto, sebbene utilizzando Pyspark si ottengono dei valori di MAE più bassi, d’altro canto l’R2 risulta essere molto basso ed i tempi di training più alti rispetto all’utilizzo di SKLearn.

Il secondo caso preso in esame riguarda un confronto tra le migliori soluzioni trovate utilizzando entrambe le librerie. Nello specifico, sono stati confrontati il *GBTRegressor* utilizzato in *Pyspark* ed un *Extratrees* *Regressor* con 400 stimatori, ottenendo i seguenti risultati:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ***PySpark (GBTRegressor)*** | | | ***Sklearn (ExtraTreesRegressor)*** | | |
| **MAE [Immatricolati]** | ***R 2*** | **Time[s]** | **MAE [Immatricolati]** | ***R 2*** | **Time[s]** |
| **Validation** | *≃ 104* | ***0,222*** | *≃ 29.013* | *≃ 131* | ***0,996*** | *≃ 1.949* |
| **Test** | *≃ 109* | ***0.261*** | *≃ 33.16* | *≃ 181* | ***0.991*** | *≃ 1.* *949* |
| **Re-Train** | *≃ 101* | ***0.391*** | *≃ 29.799* | *≃ 156* | ***0.993*** | *≃ 2.01* |

In questo caso, si nota ancor di più la differenza, soprattutto a livello di tempo, nel training tra i due algoritmi regressivi, infatti il tempo per addestrare un *GBTRegressor* in *Spark* risulta di circa 30 secondi mentre, utilizzando un algoritmo che presenta performance abbastanza simili se non migliori, tenuto conto dell’R2, essi sono di circa 2 secondi.

Tuttavia, tale confronto non vuole far intendere che l’utilizzo di *SKLearn* sia una soluzione migliore rispetto a Pyspark, ma semplicemente che, in questo caso, avendo a disposizione una mole contenuta di dati e la possibilità di testare algoritmi che nella libreria Pyspark non sono ancora presenti, è stato deciso di scegliere i risultati forniti dall’*Extratrees Regressor* per mostrare l’andamento del trend di Immatricolazioni per i vari Atenei.

# Conclusioni

Uno dei punti di forza dell’elaborato è sicuramente l’automatizzazione delle attività di Data Manipulation.

Inoltre, le elaborazioni proposte per le Università di *Bologna, Milano, Roma La Sapienza* e *Napoli Federico II* possono essere estese a tutti gli Atenei, ed il codice mostrato è riusabile su futuri Dataset aggiornati.

In conclusione, si riportano i principali risultati riscontrati in questa Challenge:

* Prevalenza di Studenti Iscritti e Immatricolati di sesso femminile;
* Incremento del numero di Immatricolazioni nella passata decade;
* Previsione di un incremento nel numero di immatricolazioni nei prossimi cinque anni;
* Flusso di Studenti Fuorisede in prevalenza da Sud verso Centro e Nord.

1. I risultati riportati sono stati ottenuti mediante l’utilizzo dello strumento *Sklearn*, del quale si rimanda al paragrafo successivo. [↑](#footnote-ref-1)
2. Si precisa che gli atenei mostrati in figura sono stati scelti a fini esemplificativi, ma in ogni caso sono state generate le previsioni del trend degli Immatricolati per tutti gli Atenei. [↑](#footnote-ref-2)