# Report 1 di tesi:

## Analisi degli ambienti di programmazione PyCharm e Google Colab attraverso lo svolgimento di un task

### Motivazione:

PyCharm e Google Colab sono due ambienti di programmazione dalle caratteristiche diverse e ognuno offre determinate peculiarità. Può essere quindi interessante analizzarne pregi e difetti al fine di scegliere l’ambiente più adatto ai fini del progetto che si andrà ad affrontare.

### Problema:

Il task che è stato scelto per testare i due ambienti di programmazione è una semplice analisi da fare sul dataset di prova fornito che riguarda il traffico dei taxi nell’hotspot D di Manhattan.

Il dataset è organizzato secondo due documenti:

* Un foglio Excel che mantiene le informazioni relative al traffico orario (colonne ‘hN’) di ogni Cluster (colonna ‘Cluster’) e l’indicazione della presenza di **anomalie** (colonna ‘Anomalous’) per ogni giorno (Colonne ‘day’ e ‘month’)
* Un documento Word dove sono segnalate tutte le anomalie rilevate nel 2015, con la relativa spiegazione e divise per Cluster

L’analisi richiesta chiedeva di settare in modo opportuno i valori di ‘Anomalous’ ricorrendo agli eventi segnalati nel documento Word.

Per fare questo ho scelto di trasferire le tabelle da Word ad Excel, mettendo un Cluster per ogni sheet.

In questo modo è stato possibile aprire e manipolare in modo semplice il file Excel con le tabelle e ricavare le date degli eventi anomali con cui ho creato un dataframe.

Attraverso un ciclo ho confrontato i valori delle date presenti nel primo foglio Excel con quelle degli eventi anomali e settato in maniera opportuna il valore di **Anomalous**.

Il task richiedeva poi di eliminare le righe con Anomalous, salvando il risultato come file .csv.

Infine, dopo aver preparato i dati, era richiesta l’applicazione dell’algoritmo **Random Forest**, un algoritmo di classificazione usato per distinguere i cluster partendo dalle serie temporali.

Per usare il Random Forest occorre per prima cosa ricavare dai dati due dataframe che rappresentano la “X” (l’oggetto) e la “Y” (il target) della classificazione. La “X” è rappresentata dalle colonne che riguardano l’affluenza oraria e la “Y” rappresenta i Cluster.

Dopo ciò ho creato un oggetto del tipo **RandomForestClassifier** con cui è stato fatto il fit dei due set ottenuti in precedenza ed è stata infine ricavata la predizione.

Infine si è calcolata la precisione della predizione tramite la funzione **score**.

### Risultati:

**PyCharm** è un ambiente di sviluppo integrato professionale specificatamente pensato per la programmazione Python. Essendo un ambiente pensato per l’utilizzo professionale è molto completo e facilmente utilizzabile se si proviene da altri IDE.

**Google Colab** è un servizio cloud offerto da Google basato su Jupyter notebooks. Jupyter [Notebook](https://en.wikipedia.org/wiki/Notebook_interface) (o anche IPython Notebooks) è un ambiente di programmazione web-based usato per creare documenti Jupyter notebook. Come servizio Google è integrato con la suite (es. Google Drive) ed è dotato di una approfondita e completa guida iniziale.

Dopo l’utilizzo di entrambi gli ambienti di programmazione e una ricerca sul web sono arrivato a stilare una lista dei vantaggi e degli svantaggi di ognuno dei due:

**PyCharm:**

**Pro:**

* + Ambiente di programmazione professionale, stabile e ben documentato
  + Molto simile ai principali IDE di altri linguaggi
  + Assistenza alla programmazione con code completion e code analysis
  + Correzione guidata di errori sintattici e semantici
  + Integrazione con sistemi di version control
  + Presenza di numerosi plugin

**Contro:**

* + Esegue il codice Python su un virtual enviroment e questo lo rende lento nell’esecuzione
  + Configurazione all’avvio risulta lenta per la necessità di creare il virtual enviroment
  + Necessità di gestione manuale delle dipendenze

**Google Colab:**

**Pro:**

* + Servizio cloud che garantisce esecuzione veloce grazie ai server Google
  + Possibilità di scrivere ed eseguire separatamente porzioni di codice integrandolo con del testo
  + I notebook vengono automaticamente salvati su Google Drive
  + Pensato appositamente per Data Science
  + Gestisce in automatico l’installazione delle dipendenze

**Contro:**

* + Al primo avvio può risultare meno intuitivo se si proviene da un IDE “classico”
  + Gestione dei file non molto intuitiva

### Conclusioni:

In generale da un confronto tra i due ambienti di programmazione si può ricavare che **PyCharm** è sicuramente un ambiente più completo ed integrato con funzionalità che lo estendono in maniera molto utile, però risulta abbastanza lento nell’esecuzione. E’ molto adatto per la gestione di progetti di medio-grandi dimensioni.

**Google Colab** è invece pensato per la redazione di paper scientifici dove il testo deve andare di pari passo con il codice e permette una velocità di esecuzione quasi instantanea.

A livello personale, è risultato più semplice implementare il task richiesto con PyCharm proveniendo da ambienti di sviluppo C++ o Java più “classici” però devo dire che la guida di introduzione di Google Colab lo rende di facile fruizione e la possibilità di scrivere del testo e di formattarlo rende i commenti molto più chiari e leggibili consentendo di spiegare il codice passo dopo passo.

L’accuratezza ottenuta con i due ambienti sembra essere paragonabile, in quanto in entrambi i casi si ottengono valori che oscillano dal 40% al 48%, risultati non molto alti.

Ho registrato alcuni valori di precisione ottenuti variando due parametri dell’analisi:

* **max\_depth (default = none):** rappresenta la massima profondità dell’albero.
* **n\_estimators (default = 100):** rappresenta il numero di alberi nella foresta.

**PyCharm:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **max\_depth / n\_estimators** | **none (default)** | **2** | **10** |
| **100 (default)** | **0,447154472** | **0,43902439** | **0,452235772** |
| **200** | **0,44004065** | **0,430894309** | **0,445121951** |
| **500** | **0,43495935** | **0,444105691** | **0,422764228** |

Sono state provate 3 possibili valori di **max\_depth** e di **n\_estimators** partendo da quella di default che prevede **max\_depth = none** e **n\_estimators = 100**.

Il valore della precisione è stato ottenuto facendo la media di 3 valori di precisione registrati dall’esecuzione e la cella evidenziata è quella per cui si è ottenuta la precisione più alta.

La configurazione migliore in questo caso risulta quindi essere quella che prevede **max\_depth = 10** e **n\_estimators = 100** (il valore di default).

**Google Colab:**

|  |  |  |  |
| --- | --- | --- | --- |
| **max\_depth / n\_estimators** | **none (default)** | **2** | **10** |
| **100 (default)** | **0,431910569** | **0,448170732** | **0,466463415** |
| **200** | **0,442073171** | **0,452235772** | **0,408536585** |
| **500** | **0,423780488** | **0,458333333** | **0,43495935** |

La configurazione migliore risulta essere la stessa anche per Google Colab che però fornisce un risultato leggermente migliore.

Da queste semplici misurazioni si evince una sostanziale uguaglianza nella precisione di calcolo dei due ambienti a fronte di una diversa velocità di esecuzione. Infatti se su Google Colab l’esecuzione rimane quasi istantanea su PyCharm questa richiede più tempo, arrivando fino a **2.927 secondi.** (misura effettuata con CProfile)