**HARD DEADLINE: 16 February 2021**

Submit by sending an email to: [dinunzio@dei.unipd.it](mailto:dinunzio@dei.unipd.it) and [silvello@dei.unipd.it](mailto:silvello@dei.unipd.it)

**Evaluation 0-4 pt**

Project are sufficient if the mark is >= 2

Those who do not submit on time cannot do the project afterward. There will be an oral exam in the place of the project.

**Project 1**

ID: FIFA19Analytics

Data: http://www.dei.unipd.it/~silvello/FIS2020/fifa19.csv

Description: The data includes the FIFA 2019 players attributes like Age, Nationality, Overall, Potential, Club, Value, Wage, Preferred Foot, International Reputation, Weak Foot, Skill Moves, etc.

Output, a Jupyter notebook producing:

* an interactive application that takes two-player names as input and produces a visualization to compare their attributes. The visualization can be interactive.
* An application that takes the name of a team as input and produces a visualization comparing the players in terms of an attribute of choice. The attribute can be an input or can be selected interactively.
* A visualization (possibly interactive) that compares the top 20 clubs in terms of the average overall value grouped by player position (goalkeeper, defensor, etc.).

**Project 2**

ID: TripAdvisor

Data: http://www.dei.unipd.it/~silvello/FIS2020/tripadvisor\_hotel\_reviews.csv

Description: The dataset comprehends textual reviews for a bunch of hotels

Output, a Jupyter notebook:

* Using the review text, employ a classification model to predict the rating associated with that review. Suggested Metrics: Root Mean Square Error
* An application that given a review returns the predicted sentiment (employ some sentiment analysis existing model).

**Project 3**

ID: Covid19Tweets

Data: http://www.dei.unipd.it/~silvello/FIS2020/covid19Tweets.zip

Description: These tweets are collected using Twitter API and a Python script. A query for this high-frequency hashtag (#covid19) is run daily for a certain time period, to collect a larger number of tweets samples.

Output, a Jupyter notebook:

* that with the available data, analyzes the trends of COVID-19 subject in the collected tweets. Follow both the time and spatial distribution. Use graphs and preferably animation to show the trends in time/space. Maps are welcomed.
* sentiment analysis from Covid-19 tweets.

**Project 4**

ID: Solar Power Generation Data

Data: http://www.dei.unipd.it/~silvello/FIS2020/Solar\_Power.zip

Description: This data has been gathered at two solar power plants in India over a 34 day period. It has two pairs of files - each pair has one power generation dataset and one sensor readings dataset. The power generation datasets are gathered at the inverter level - each inverter has multiple lines of solar panels attached to it. The sensor data is gathered at a plant level - a single array of sensors optimally placed at the plant.

Output:

* A Python notebook answering these questions:
  + What is the mean value of daily yield? What is the total irradiation per day? What is the max ambient and module temperature? How many inverters are there for each plant? What is the maximum/minimum amount of DC/AC Power generated in a time interval/day? Which inverter (source\_key) has produced maximum DC/AC power? Rank the inverters based on the DC/AC power they produce? Is there any missing data?
* Graphs that explain the patterns for attributes independent of other variables. These will usually be tracked as changes of attributes against DATETIME, DATE, or TIME. Examples - how is DC or AC Power changing as time goes by? how is irradiation changing as time goes by? how are ambient and module temperature changing as time goes by? how does yield change as time goes by? Explore plotting variables against different granularities of DATETIME and which is the best option for each variable.

**Project 5**

ID: CoronaVirus2019Dataset

Data: http://www.dei.unipd.it/~silvello/FIS2020/Covid19\_data.zip

Description: Looking at China's growth rate over time, we can clearly see that they were able to curb the growth rate of the virus. Over the last few weeks, different countries have performed different forms of mitigation to do the same: ban large gatherings, close schools, stop incoming flights, put cities in lockdown, etc.

The task is to evaluate the effectiveness of mitigation by trying to see if a correlation can be discovered between the different types of mitigation and the growth rate of confirmed cases. What measures seem to work and which not? Which ones are the most effective?

Keep in mind there are numerous factors that might affect the growth rate (e.g. country's general hygiene, population density, how much time they had to prepare) so not all countries can be compared easily. China being an obvious outlier since it was country 0.

Output:

* A Python notebook reporting the conducted analyses and a summary of the conclusions achieved by the conducted correlation analysis.

**Project 6**

ID: WineReviews

Data: http://www.dei.unipd.it/~silvello/FIS2020/Wine\_reviews.zip

Description: The dataset was scraped from WineEnthusiast (https://www.winemag.com) in 2017 and contains three files: two csv files with about 280k reviews described by 10 features, and one json file with about 7k nodes of wine reviews.

The task is to design and implement a system that predicts the points of a wine based on the features extracted from the data.

Output:

* A Python notebook reporting all the steps conducted to implement the system: from the cleaning of data, to the design of the features, the training and validation of the system, and the analysis of the results.

**Project 7**

ID: CreditCardFraud

Data: http://www.dei.unipd.it/~silvello/FIS2020/CC\_frauds.zip

Description: The dataset contains transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

The task is to design and build a system that automatically recognizes a fraudulent transaction.

* A Python notebook reporting all the steps conducted to implement the system: from the cleaning of data, to the design of the features, the training and validation of the system, and the analysis of the results.

**Project 8**

ID: VideoGamesSales

Data: http://www.dei.unipd.it/~silvello/FIS2020/Video\_games.zip

Description: This dataset contains a list of video games with sales greater than 100,000 copies (http://www.vgchartz.com). The fields of the dataset include, for example, the ranking of overall sales, the platform of the games release (i.e. PS4, Xbox, etc.), the genre of the game.

The task is to explore the data and produce an interactive data visualization that helps the reader to answer questions like “what are the most successful video games in history” (<https://medium.com/@tomasborrella/what-is-the-most-successful-video-game-platform-in-history-a18dfab20a41>).

* A Python notebook that describes the analysis and tells the reader the story of these videogames with the use of interactive Matplotlib pots.

**Project 9**

ID: NetflixMoviesTVShows

Data: http://www.dei.unipd.it/~silvello/FIS2020/Netflix.zip

Description: This dataset consists of tv shows and movies available on Netflix as of 2019. The dataset was collected from Flixable (https://www.flixable.com).

The task is to link the information contained in this dataset with the Internet Movie Database IMDB (<https://www.imdb.com>) and produce an analysis of the correlation between the Netflix ratings and the IMDB ratings.

* A Python notebook that reports the steps that are needed to produce a single dataset containing both Netflix and IMDB information, and an analysis of the ratings given by Netflix users and IMDB users.

**Project 10**

ID: SpotifyDataset

Data: http://www.dei.unipd.it/~silvello/FIS2020/Spotify.zip

Description: The dataset contains more than 160.000 songs collected from Spotify Web API, grouped by artist, year, or genre in the data section. The description of the features of the dataset can be found (https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/) and (<https://developer.spotify.com/documentation/web-api/reference/tracks/get-track/>).

The task consists in building a recommender system that suggests songs based on the profile of a user.

* A Python notebook that describes the choices of the features you want to include in the recommender system, the procedure you used to train and validate the system, an example of song recommendation based on a profile of a user.