PYTHON

DATA MANIPULATION FRAMEWORKS

-Numpy (Scipy è più completo)

-Pandas (data manipulation)

-pythonmathplot (data visualization)

MACHINE LEARNING PREDICTION MODEL FRAMEWORKS

-Scikit-learn

-Keras

-Tensorflow (2.0 with keras)

**CONCETTI CHIAVE SUPERVISED LEARNING**

**-Batch**: you can’t pass the entire dataset into the neural net at once. So, you **divide dataset into Number of Batches or sets or parts.**

**-Epochs**: A term that is often used in the context of machine learning. An epoch is one complete presentation of the *data set to be learned* to a learning machine.

**-Gradient descent (stochastic vs batch)**: variazione dell’accuratezza (minore numero errori in percentuale di successo) dopo cicli di backpropagation training

**-Selezionare features**: Per cercare un set di caratteristiche adeguato per creare un modello predittivo, è necessario sperimentare e conoscere approfonditamente il problema che si desidera risolvere. Alcune caratteristiche sono infatti migliori di altre per le stime. Alcune caratteristiche sono strettamente correlate ad altre e possono essere rimosse. Ad esempio, city-mpg e highway-mpg sono strettamente correlate ed è possibile rimuoverne una senza influire in modo significativo sulla stima.

**-Kernel**: a **function**that maps the data to a higher dimension where the data is separable. It projects the data up by k dimensions such that the data points are now separable in the higher dimensional plane. f(x,y) 🡪 (x,y,z). 2 tipi i più usati: **Gaussian Kernel** e **Polinomial Kernel**

**-Supervised Learning, steps:**

1. **Determine the type of training examples**, and their feature set
2. **Gather a training set**. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered
3. **Determine the input feature representation of the learned function**. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a [feature vector](https://en.wikipedia.org/wiki/Feature_vector), which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the [**curse of dimensionality**](https://en.wikipedia.org/wiki/Curse_of_dimensionality). To determine the type and number of features, **Data Mining tecniques** a**re used to analyze the raw data**.
4. **Determine the structure** of the learned function and **corresponding learning algorithm.**
5. Complete the design. **Run the learning algorithm on the gathered training set**. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via [cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)).
6. **Evaluate the accuracy of the learned function**. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set. **Some metrics for classification problems are: accuracy, precision and recall**

**-Bias**: (definition in bias-variance tradeoff) A **learning algorithm is biased** for a particular input x{\displaystyle x} if, when trained on each of these data sets, it is systematically incorrect when predicting the correct output for x {\displaystyle x}. It is a **source of error** in your model that causes it to **over-generalize and underfit your data**

**-Variance**: (definition in bias-variance tradeoff) A **learning algorithm has high variance** for a particular input x{\displaystyle x} if it predicts different output values for x when trained on different training sets. It **is sensitivity to noise** in the data that **causes your model to overfit.**

**-Bias Variance tradeoff**:

* **bias error**: An overly **simplified** model has high bias error.
* **variance error:** An overly**complex** model has low bias error but high **variance error** because a complex model will capture both real and random effects in the dataset.
* Its good to always find a compromise between**bias and variance.** A **good bias-variance ration means that the model is not overfitted nor underfitted. Variance and Bias can be (ideally) calculated on the Y variable calculated by the learning algorithm.**

**-Noise**:Samples in the training data that makes no sense, and does not follow the common principle behind the function the model needs to learn. I**f the desired output values are often incorrect** (because of human error or sensor errors), then **the learning algorithm should not attempt to find a function that exactly matches the training examples,** that would lead to **overfitting** in the trained model.

**-Cross Validation**: Cross-validation **combines (averages) measures of fitness in prediction to derive a more accurate estimate of model prediction performance**. One round involves [partitioning](https://en.wikipedia.org/wiki/Partition_of_a_set) a [sample](https://en.wikipedia.org/wiki/Statistical_sample) of [data](https://en.wikipedia.org/wiki/Data) into [complementary](https://en.wikipedia.org/wiki/Complement_(set_theory)) subsets, performing the analysis on one subset (called the **training set**), and validating the analysis on the other subset (called the **validation set** or testing set). To reduce [variability](https://en.wikipedia.org/wiki/Variance), in most methods **multiple rounds of cross-validation are performed using different partitions**, and the **validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance. Stratified K-Fold** keeps the same class rappresentation in the samples for each fold (for example each fold might have 25% of saples of the first class and 75% of the other). Instead in **standard K-Fold** each fold contains the same number of samples (as the stratified version) but each one is made from random samples.

**-Regularization:** Introduce a **penality function during the model training** to avoid overfitting and making the model more generalize, adding too much penality could cause underfitting, making the model to simple and unable to learn complex relations between the features. Some common method to achieve Regularization are the **L1 NORM**, **L2 NORM** or completely different method like the **droput in neural networks**

**LINEAR CLASSIFIER CONCETTI**

**-Support vector machine** (tipo di algoritmo LC): trova l’iperpiano che ha la massima distanza e orientamento parallelo ai valori limite appartenenti alle 2 classi.

**-logistic regression con sigmoid o thanx** (tipo di algoritmo LC): utilizzato più per la regressione che per la classificazione. Utilizza funzioni derivabili con backpropagation per diminuire la loss.

**-Kernel trick**: modo di rendere dati linearmente separabili utilizzando funzioni kernel

**-Decision tree classifier**: mappano un set di features in una scala discreta tramite un albero decisionale in cui le foglie rappresentano le classi (valori discreti) mentre gli archi rappresentano congiunzioni di features che portano al riconoscimento della classe di appartenenza.

**SUPPORT VECTOR MACHINE PER K CLASSI**

2 approcci:

1. **One vs all classifier**: viene generato ed allenato un classificatore binario per ogni classe presente.
2. **Weston and Watkins SVM classifier**: metodo più complesso ma che generalmente genera meno SVM per risolvere la classificazione con K classi distinte.

**LINEAR REGRESSION**

- SGNIFICATO DELLA BIAS COLUMN OF 1’s ADDED TO DATA 🡪 Y = M\*X + 1\*C: This explains the bias term, 1, for each instance.

**DECISION TREE**

2 algoritmi per costruirli:

* **CART**= Usa **Gini Index** (classificazione) come metrica per discriminare le transizioni tra nodi (ogni nodo è una combinazione di valori di features che contengono una certa informazione sull’appartenenza a una o l’altra classe)
* **ID3**= Usa la **Funzione di Entropia** e la correlata **Funzione di Information Gain**

**RANDOM FOREST**: Algoritmo di classificazione e regressione che utilizza molteplici decision tree.

**ENSAMBLE MODELS (utilizzare contemporaneamente diversi modelli):**

**-Bagging:** Differenti modelli addestrati su differenti resample del training set, previsioni fatte tramite voto 🡪 riduce di molto la varianza e aumento di poco il bias (errore)

**-Boosting:**L’obiettivo è passare da un weak learner a uno strong learner, questo viene fatto facendo addestrare nuovi classificatori o predittori non sui dati originali,come nel bagging, ma su una forma modificata di tali dati in modo che un classificatore si concentri di più sulla correttezza di uno o più specifici esempi o aspetti di esempi (ad esempio i residui tra valori reali e predizioni fatte dal predittore precedente) rispetto ad un altro classificatore. Il classificatore finale sarà C1+C2+C3+…+CN

**-Stacking:** Gli output di modelli di classificazione separati allenati sullo stesso training set diventano gli input di un classificatore di più alto livello che impara il modo migliore di combinare i loro risultati per ottenere la classificazione corretta dell’esempio.

**METRICHE PER LA MISURAZIONE DELL’ERRORE NELLA REGRESSIONE E CLASSIFICAZIONE:**

-CONTROLLARE PAGINA 5 METRICS TO MEASURE LOSS IN REGRESSION

**EVITARE OVERFITTING E UNDERFITTING**

* **Cross-validation**: train my models on subsets of my data, and then choose the model that performs best on the reserved portion of data
* **Regulariazion**: train my models on subsets of my data, and then choose the model that performs best on the reserved portion of data
* **Sampling:** undersampling, oversampling, smote sampling

**TIPI DI FEATURES**

1. **Nominal**: Sono dei semplici dati label, non rappresentano un valore quantitativo, possono assumere n valori discreti, se letterali si possono trasformare in interi.
2. **Ordinal**: Sono dei dati di valore discreto per cui è importante l’ordine. Il valore che assumono è inserito all’interno di una scala di importanza (es 1= basso 5=alto)
3. **Continous**: Sono dei dati dal valore continuo, possono generalmente assumere infiniti valori numerici.

**ENCODING DI CATEGORICAL FEATURES 🡪 FEATURES CON VALORI DISCRETI**

* **Ordinal:** Converte le label in interi da 1 a k seguendo l’ordine di importanza (ORDINAL)
* **OneHot:** Creo una nuova feature che può valere o 0 o 1 per ogni valore discreto che può assumere il dato (ORDINAL, NOMINAL)
* **Binary:** Converto ogni intero a binario, per ogni bit credo una nuova feature. C’è un po di perdita informazioni rispetto a onehot ma ho meno dimensioni (ORDINAL)
* **BaseN:** …
* **Hashing:** Come OneHot ma con meno dimensioni, c’è perdita di informazioni data da collisioni di hash uguali. (ORDINAL, NOMINAL)

**PRECISION E RECALL**

Metriche per la classificazione, si possono visualizzare con 2 grafici:

-**Precision vs Recall curve**: da usare quando si ritiene più importante minimizzare i falsi positivi rispetto ai falsi negativi, il punto di incontro tra le 2 curve dovrebbe trovarsi più in alto possibile.

-**ROC curve (TPR/FPR):** da usare quando si ritiene più importante minimizzare i falsi negativi che positivi (questo perché più la curva si trova in alto, ovvero più saranno i TP rispetto ai FP più sarà sicuramente più basso il numero dei FALSI NEGATIVI, perché avrò appunto più TP che lascia meno spazio a falsi negativi)

Il **THRESHOLD** con cui discriminare gli input una volta calcolato il valore della decision\_function è un valore di default per ogni algoritmo, ad esempio nelle SVM è 0 (considera le label in automatico come -1 negativa e 1 positiva)

AZURE MACHINE LEARNING

KEY FEATURES

-completamente disponibile online

-parte di pulizia/visualizzazione csv inclusa

-pre-elaborazione dei dati (split training-test set, selezione features ecc)

-scelta di un algoritmo di apprendimento e apprendimento stesso inclusi

METRICHE DI ERRORE NELLA REGRESSIONE

* **Errore assoluto medio** (MAE): Media degli errori assoluti (un *errore* è la differenza tra il valore stimato e quello effettivo).
* **Radice dell'errore quadratico medio** (RMSE): Radice quadrata della media degli errori quadratici delle stime effettuate sul set di dati di test.
* **Errore assoluto relativo**: Media degli errori assoluti relativamente alla differenza assoluta tra i valori effettivi e la media di tutti i valori effettivi.
* **Errore quadratico relativo**: Media degli errori quadratici relativamente alla differenza quadratica tra i valori effettivi e la media di tutti i valori effettivi.
* **Coefficiente di determinazione**: Noto anche come **valore quadratico R**, è una metrica statistica che indica l'esattezza del modello rispetto ai dati.

| **Acroynm** | **Full Name** | **Residual Operation?** | **Robust To Outliers?** |
| --- | --- | --- | --- |
| MAE | Mean Absolute Error | Absolute Value | Yes |
| MSE | Mean Squared Error | Square | No |
| RMSE | Root Mean Squared Error | Square | No |
| MAPE | Mean Absolute Percentage Error | Absolute Value | Yes |
| MPE | Mean Percentage Error | N/A | Yes |

Mean absolute error è più robusto ad outliers perché mean squared error nel calcolo da’ più importanza alle variazioni più elevate dalla media (e quindi ai valori di training più anomali)

**DATA MINING**

-2 aspetti di **preparazione del dato**, una volta presenti in formati utilizzabili come .csv. Lo scopo sarà quello di essere sicuri di fornire all’algoritmo di apprendimento automatico solo le informazioni necessarie ad un apprendimento efficace, ovvero quelle contenente informazioni utili e non magari inutile o ripetuta da altre colonne già presenti. Questo viene fatto con 2 tecniche di preparazione: **Feature Selection** and **Feature Engeneering**. Uno strumento che punta a fornire un certo grado di automazione per queste metodologie è **SQL Server Data Mining**, che utilizza diversi algoritmi per calcolare un punteggio di utilità ad ogni colonna (feature) e cercare di minimizzare il numero di colonne a quelle che contengono informazione piuttosto che noise.

-**Feature engeneering** pertiene anche al lavorare sulle feature presenti per portare le informazioni nella forma più funzionale all’utilizzo per un algoritmo di apprendimento. Degli strumenti matematici/statistici che potrebbero aiutare a questo scopo sono:

* Scaling
* PCA
* Encoding
* Sampling

-La **Feature selection** si può effettuare seguendo 2 metodologie differenti: utilizzando Wrapper Method o **Filter Method**. La seconda (Filter) utilizza delle metriche standard per valutare e selezionare le feature che contengono la maggior quantità di informazione, queste sono:

* Chi-Squared
* Information Gain (misura la variazione di **entropia** quando la feature è presente e assente)
* F-Test
* Mutual Information
* Variance Threshold (If the feature doesn’t vary between the data points, it can be removed, as it is apparent that this variable doesn’t predict the target variable well.)

-**Paper** for metrics: <http://www.jmlr.org/papers/volume3/forman03a/forman03a_full.pdf>

-Per quanto riguarda la **Feature selection con Wrapper Method**, questa viene fatta utilzzando il training set per allenare diversi modelli, usando ogni volta un numero diverso di features, con uno specifico algoritmo di apprendimento, e infine valutando tutti i modelli sul test set. Verrà scelto quello più accurato con il minor numero di feature che, teoricamente, dovrebbero essere quelle contenenti le informazioni più rilevanti ai fini della classificazione.

-DATA VISUALIZATION CON HEATMAP: tipi di correlazioni tra features (ovvero per una feature rispetto ad un'altra il valore di correlazione è positivo se quando aumenta una aumenta anche l’altra e viceversa):

* Pearson (standard)
* Kendall
* Spearman

IMPUTATION DI VALORI MANCANTI

-Un idea può essere di fare regressione lineare sulla varaibile mancante e inserire valori a random tra il valore predetto il un range di +/- l’errore medio della regressione

QUESITI SENZA RISPOSTA

1. Per variabili categoriche non ordinali è preferibile un one hot encoding oppure l’encoding a numero discreto da 0 a n?
2. Encoding per latitudine e longitudine?
3. Scaling per variabili con label encoding?

FONTI

* Librerie python: <https://medium.com/edureka/python-libraries-for-data-science-and-machine-learning-1c502744f277>
* Dense neural network con tensorflow/pandas : <https://heartbeat.fritz.ai/classification-with-tensorflow-and-dense-neural-networks-8299327a818a>
* Gradient descent: <https://medium.com/@divakar_239/stochastic-vs-batch-gradient-descent-8820568eada1>
* Simple linear classifier model from raw data: <https://towardsdatascience.com/experimenting-with-twitter-data-using-tensorflow-ea88a8078fd>
* Corso generale: <https://newonlinecourses.science.psu.edu/stat508/lesson/10/10.2>
* Libro: pattern recognition and machine learning
* Librerie: <https://qph.fs.quoracdn.net/main-qimg-8240099ed4a397e9ff4f035bccf201e5.webp>
* Azure machine learning panoramica base: <https://docs.microsoft.com/it-it/azure/machine-learning/studio/create-experiment>
* come selezionare un algoritmo dato un dataset? 🡪 <https://www.researchgate.net/post/What_is_the_best_algorithm_for_classification_task#targetText=Approximate%20Statistical%20Tests%20for%20Comparing%20Supervised%20Classification%20Learning%20Algorithms.&targetText=If%20you%20want%20to%20predict,you%20can%20use%20is%20metalearning.>
* Feature engeneering vs feature selection : <https://blog.featurelabs.com/feature-engineering-vs-feature-selection/#targetText=Feature%20engineering%20is%20the%20process,make%20machine%20learning%20algorithms%20work.&targetText=We%20can%20then%20use%20this,not%20contain%20much%20relevant%20information.>
* <https://www.kaggle.com/joparga3/in-depth-skewed-data-classif-93-recall-acc-now>