



Politecnico di Milano
School of Industrial and Information Engineering

Data Mining and Text Mining

January 18, 2020

NAME _____

CODICE PERSONA/ID _____

GENERAL INSTRUCTIONS

- Answers must be clearly written inside the answer box designated for each problem.
- All the answers must be adequately motivated.
- Pencils are not allowed. The exam consists of 6 sheets of paper. It must be returned with all the 6 sheets. No any other sheet can be added. No sheet can be removed.
- This is a closed-book/closed-notes exam.
- Only non-programmable calculators are allowed.
- Notes/books/mobile phones are not allowed.
- If you are caught using forbidden material, the exam will immediately end and an RP grade will be recorded; then, your Data Mining exam will consist of an oral examination from then on.

COURSE PROJECT SCORE _____

FINAL TIME _____

GRADES

1	2	3
4	5	6

SCORING

- A problem left unsolved will amount to zero points.
- A completely wrong solution will amount to -3 points

**STUDENTS WHO DID THE COURSE
PROJECT HAVE 1:40h TO SOLVE
PROBLEMS 1, 2, 3, AND 4**

**ALL THE OTHER STUDENTS HAVE 2:20h
TO SOLVE ALL THE SIX PROBLEMS**

Problem 1 (6 points). Consider a dataset containing the information about a bike sharing service which records the following ten attributes:

- timestamp: the day of the year and the hour of the day
- cnt: number of bikes
- t1: actual temperature
- t2: temperature as it feels
- hum: humidity %
- wind_speed: km/h
- weather_code: a number describing a weather condition (no other information provided)
- is_holiday: 1 if it is a holiday, 0 otherwise
- is_weekend: 1 if it is a weekend day, 0 otherwise
- season: a number encoding the season

Consider the following example rows and the attribute description printed by describe command. Attribute **cnt** is your target variable. The final objective is to build a model to predict the number of bikes (attribute **cnt**) used in a given hour of a given day.

```
df = pd.read_csv("London_merged.csv")
df.head(5)
```

	timestamp	cnt	t1	t2	hum	wind_speed	weather_code	is_holiday	is_weekend	season
0	2015-01-04 00:00:00	182	3.0	2.0	93.0	6.0	3.0	0.0	1.0	3.0
1	2015-01-04 01:00:00	138	3.0	2.5	93.0	5.0	1.0	0.0	1.0	3.0
2	2015-01-04 02:00:00	134	2.5	2.5	96.5	0.0	1.0	0.0	1.0	3.0
3	2015-01-04 03:00:00	72	2.0	2.0	100.0	0.0	1.0	0.0	1.0	3.0
4	2015-01-04 04:00:00	47	2.0	0.0	93.0	6.5	1.0	0.0	1.0	3.0

```
df.describe()
```

	cnt	t1	t2	hum	wind_speed	weather_code	is_holiday	is_weekend	season
count	17414.000000	17414.000000	17414.000000	17414.000000	17414.000000	17414.000000	17414.000000	17414.000000	17414.000000
mean	1143.101642	12.468091	11.520836	72.324954	15.913063	2.722752	0.022051	0.285403	1.492075
std	1085.108068	5.571818	6.615145	14.313186	7.894570	2.341163	0.146854	0.451619	1.118911
min	0.000000	-1.500000	-6.000000	20.500000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	257.000000	8.000000	6.000000	63.000000	10.000000	1.000000	0.000000	0.000000	0.000000
50%	844.000000	12.500000	12.500000	74.500000	15.000000	2.000000	0.000000	0.000000	1.000000
75%	1671.750000	16.000000	16.000000	83.000000	20.500000	3.000000	0.000000	1.000000	2.000000
max	7860.000000	34.000000	34.000000	100.000000	56.500000	26.000000	1.000000	1.000000	3.000000

As the very first step, you are asked to preprocess the data and build the best **four new features** (computed using the ten existing ones) that can help building a better model for **cnt**.

Question #1: For each new feature specify (1) how the new feature is computed from the existing attributes, (2) how it improves the available information, (3) why it might help building a better the predictive model for **cnt**.

Question #2: After the new features have been computed, would you still keep all the original features? If yes, why? If not, which variables would you eliminate?

Problem 1 (continued).

Feature #1

How is the new feature computed from the existing attributes?

How does it improve the available information?

Why might it help building a better the predictive model for cnt?

Feature #2

How is the new feature computed from the existing attributes?

How does it improve the available information?

Why might it help building a better the predictive model for cnt?

Feature #3

How is the new feature computed from the existing attributes?

How does it improve the available information?

Why might it help building a better the predictive model for cnt?

Feature #4

How is the new feature computed from the existing attributes?

How does it improve the available information?

Why might it help building a better the predictive model for cnt?

Solution Problem #1:

Question #1: We know very few things about the data. We know how they look like (from the head command, and we know some stats from the describe command. Nothing else. We don't know what regression methods will be used. We don't know correlation properties; we don't know to what the encoding of season and weather_code represents. Maybe there is a meaningful order so there is not much we can do without additional information. Surely, the timestamp contains a lot of information that is useless as it is. The column is a primary key so it is a source of overfitting and cannot be used by methods that cannot deal with categorical variables.

We can split the timestamp in several variables

1. We can create a column year since maybe the behavior changes from year to year based on the number of bikes available. Surely it might be useful and improves the data by extracting information that was present but previously useless.
2. We can create a column month that improves over an encoding of season. Different months in the same season might have different behavior. It might be useful and improves the data by extracting information that was present but previously useless.
3. We can create a column with the hour since bike rental might change according to the time of the day. Again, it improves the data by extracting information that was present but previously useless.
4. We can create a column with the day of the week (Monday=1, 2, 3, 4, ..., Sunday=7) since the rental behavior might be different between the single working days. In addition, we have the information about weekends, but the behavior might change before Saturdays and Sundays. Again, it improves the data by extracting information that was present but previously useless.
5. We can create a column $t_1 - t_2$ since the difference in perceived temperature might be important. Note that we use the raw difference which is positive if the reported temperature is higher than the perceived temperature and negative otherwise.

Question #2: Timestamp must be eliminated. It is a primary key and the source of overfitting and cannot be used by methods that cannot deal with categorical attribute and obviously one-hot-encoding cannot be employed.

Problem 2 (6 points).

- (1) Write the pseudocode of agglomerative hierarchical clustering given the data D containing N data points described by d attributes
- (2) Compare Hierarchical Clustering and k-Means in terms of their computational complexity

Notes: Write one instruction per line. The function definition is specified using Python notation just for your convenience.

ALGORITHM 14.1. Agglomerative Hierarchical Clustering Algorithm

AGGLOMERATIVECLUSTERING(\mathbf{D}, k):

- 1 $\mathcal{C} \leftarrow \{C_i = \{\mathbf{x}_i\} \mid \mathbf{x}_i \in \mathbf{D}\}$ // Each point in separate cluster
 - 2 $\Delta \leftarrow \{\delta(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i, \mathbf{x}_j \in \mathbf{D}\}$ // Compute distance matrix
 - 3 **repeat**
 - 4 Find the closest pair of clusters $C_i, C_j \in \mathcal{C}$
 - 5 $C_{ij} \leftarrow C_i \cup C_j$ // Merge the clusters
 - 6 $\mathcal{C} \leftarrow (\mathcal{C} \setminus \{C_i, C_j\}) \cup \{C_{ij}\}$ // Update the clustering
 - 7 Update distance matrix Δ to reflect new clustering
 - 8 **until** $|\mathcal{C}| = k$
-

Hierarchical Clustering:
Time and Space Requirements

15

- $O(N^2)$ space since it uses the proximity matrix.
 - N is the number of points.
- $O(N^3)$ time in many cases
 - There are N steps and at each step the size, N^2 , proximity matrix must be updated and searched
 - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Problem 2 (Continued).

For k-means we have:

Computational Complexity	15
<ul style="list-style-type: none">• Cluster assignment takes $O(nkd)$ time since, for each of the n points, it computes its distance to each of the k clusters, which takes d operations in d dimensions• The centroid re-computation step takes $O(nd)$ time because it adds at total of n d-dimensional points• Assuming that there are t iterations, the total time for K-means is given as $O(tnkd)$.• In terms of the I/O cost it requires $O(t)$ full database scans, because we have to read the entire database in each iteration.	

Problem 3 (6 points). Suppose the bike sharing data discussed in the first problem have been adequately preprocessed. It is now time to build some models. For this purpose, you apply ten-fold crossvalidation on the training data (the only data you have) to evaluate the performance of four regression approaches:

- Multivariate linear regression
- Lasso regression
- K-Nearest Neighbor regression with a k equal to 5
- Random forests with 100 trees

You apply ten-fold crossvalidation and record the following results:

Linear Regression	R2 Average=0.44 Std Dev=0.02
Lasso Regression	R2 Average=0.44 Std Dev=0.02
KNN Regression	R2 Average=0.82 Std Dev=0.01
Random Forest Regression	R2 Average=0.96 Std Dev=0.00

Question #1 (1 point): What is R2? How is it computed? What does it indicate?

Coefficient of Determination R²

• Total sum of squares

$$TSS = \sum_{i=1}^N (y_i - \bar{y})^2$$

• Coefficient of determination

$$R^2 = 1 - \frac{RSS}{TSS}$$

• R² measures of how well the regression line approximates the real data points. When R² is 1, the regression line perfectly fits the data.

Prof. Pier Luca Lanzi POLITECNICO DI MILANO

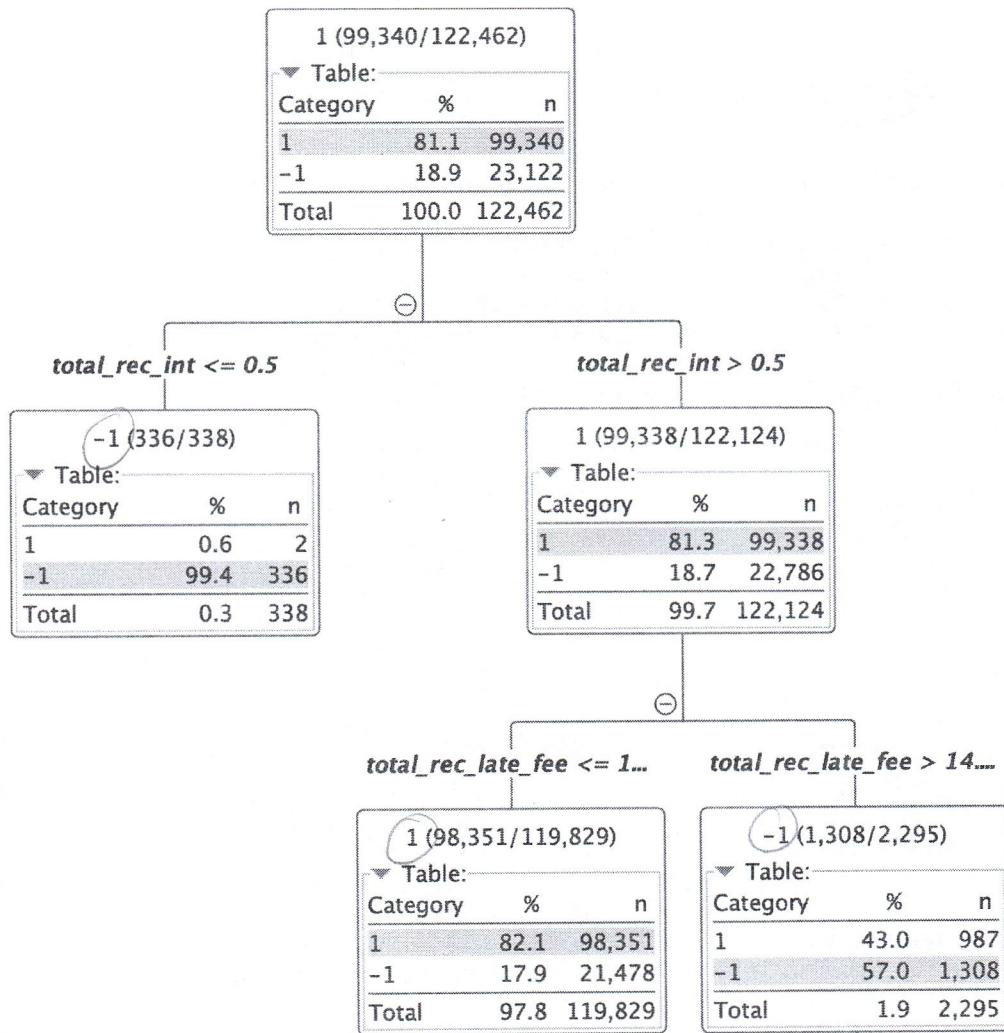
Question #2 (3 points): You are asked to comment the results above and to suggest some interesting findings about the nature of the problem at hand. What would you say about the problem given the performance of the regressors?

The question asks to discuss findings about the problem at hand. So, what do the result suggest about the problem. To answer this, we should consider the type of models that all the methods build. As can be noted from the performance reported global linear models (multivariate linear regression and lasso regression) have a poor performance. This suggests that the number of bikes in a given day cannot be modeled as a linear relation involving all the data. On the other hand, locality appears to be important in this problem, in fact k-nearest neighbor has a much better performance than linear models. And this is also suggested by the results on random forests which are made of trees that isolates problem subspaces and build a simple local model for each subspace. **Extra Comment:** This is kind of make sense if you consider the problem, if I like to take a bike to work when it is sunny or I usually take a bike during the weekends, the days I will take a bike are very similar to each other so similarity between days is likely to be important. The results above support this.

Question #3 (2 points): How would you compute the final model to deploy in production?

Random forests clearly perform better and the results are quite robust. Per se, random forests don't overfit and in this case cross-validation was applied so we can trust the reported performance. Thus, we retrain the random forest with all the data available, in this case, the training set.

Problem 4 (6 points). Given the following decision tree, compute the confusion matrix, the accuracy, the precision, and the recall. For the computation consider 1 represents the positive class and -1 represents the negative class. Note that the comma "," is the thousands separator; for example: 1,234 represents 1234.



Question #1: Compute the confusion matrix on the data used to build the decision tree.

		Predicted Class	
		1	-1
		1	-1
True Class	1	98351	989
	-1	21478	1644

Problem 4 (Continued).

Question #2: Show the formula to compute accuracy and compute the accuracy from the confusion matrix above.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) = 99995 / 122462 = 0.82$$

Question #3: Show the formula to compute precision and compute the precision from the confusion matrix above.

$$\text{Precision}(p) = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Precision} = 98351 / (98351 + 21478) = 0.82$$

Question #4: Show the formula to compute recall and compute the recall from the confusion matrix above.

$$\text{Recall}(r) = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Recall} = 98351 / (98351 + 989) = 0.99$$

Problem 5 (6 points). In the context of clustering, answer the following questions.

Question #1: Define internal validation measures and provide an example.

Clustering Quality Measures 36

- Internal Validation Measures
 - Employ criteria that are derived from the data itself
 - For instance, intracenter and intercluster distances to measure cluster cohesion and separation
 - Cohesion evaluates how similar are the points in the same cluster
 - Separation, how far apart are the points in different clusters
- External Validation Measures
 - Use prior or expert-specified knowledge about the clusters
 - For example, we cluster the Iris data using the four input variables, then we evaluate the clustering using known class labels
 - Employs criteria that are not inherent to the dataset
- Relative Validation Measures
 - Aim to directly compare different solutions, usually those obtained via different parameter settings for the same algorithm.

Prof. Pier Luca Lanzi POLITECNICO DI MILANO

WSS and BSS are internal validation measures.

Question #2: Define external validation measures and provide an example.

An example of external validation measures is the validation against existing labels.

Question #3: What are cohesion and separation measures?

Internal Measures: Cohesion and Separation 37

- Cohesion measures how closely related are objects in a cluster
 - Within-cluster sum of squares
- Separation measures how well separated a cluster is from other clusters
 - Between-cluster sum of squares

$$WSS(C) = \sum_{i=1}^k \sum_{x_j \in C_i} d(x_j, \mu_i)^2$$

where μ_i is the centroid of cluster C_i (in case of Euclidean spaces)

$$BSS(C) = \sum_{i=1}^k |C_i| d(\mu, \mu_i)^2$$

where μ is the centroid of the whole dataset

Prof. Pier Luca Lanzi POLITECNICO DI MILANO

Problem 6 (3 points). Define autocorrelation in the context of data series.

Definition from the notebooks on data series.

The screenshot shows a Jupyter Notebook interface with three tabs: Untitled.ipynb, basics.ipynb, and modeling.ipynb. The basics.ipynb tab is active. The code cell contains the following text:

```
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Below the code, there is a section titled "Autocorrelation and Partial Autocorrelation" with two bullet points:

- Autocorrelation - The autocorrelation function (ACF) measures how a series is correlated with itself at different lags.
- Partial Autocorrelation - The partial autocorrelation function can be interpreted as a regression of the series against its past lags. The terms can be interpreted the same way as a standard linear regression, that is the contribution of a change in that particular lag while holding others constant.

Below this text is a table with two columns: "Series" and "Lagged Series". The data is as follows:

Series	Lagged Series
5	
10	5
15	10
20	15
25	20
⋮	⋮

A red arrow points from the value "5" in the "Series" column to the value "5" in the "Lagged Series" column, indicating the correlation between the current value and its first lag.