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?

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 t1-t2 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.

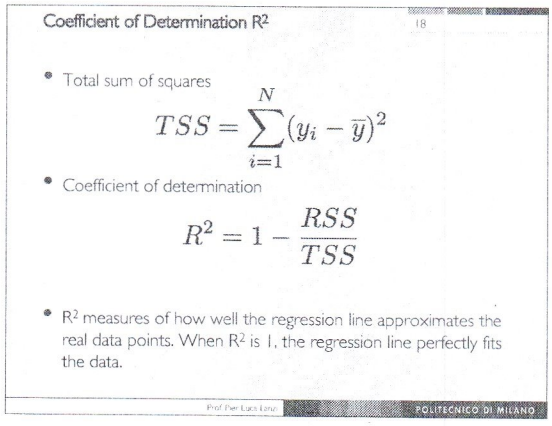
Problem 3 (6 points). Suppose the bike sharing data discussed in the first problem have been adequately preprocessed. It is now time to build some models. For this purpose, you apply ten-fold crossvalidation on the training data (the only data you have) to evaluate the performance of four regression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



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 5 (6 points). You and your boss attended a presentation of a data scientist who applied several classification methods to a data set containing 20000 data points. The data scientist first applied hold out splitting the data in train and test set. Then applied the 7 classification methods listed below and computed performance as the Mean Absolute Error (MAE) on the test set. The presenter concluded that "Bayesian Ridge" was the best method of the seven considered since it reached the lowest MAE value. At the end your boss wants to know whether you liked the presentation or would apply a different procedure to compare the seven approaches.

	Regressor	MAE
0	linear	7.248779
1	lasso	7.264249
2	ridge	7.248766
3	elastic_net	7.262760
4	AdaBoost	15.213487
5	bayesian ridge	7.247135
6	Lasso LARS	30.632625
7	xgb	12.283350

It would be better to apply k-fold crossvalidation instead of just comparing algorithms based on one single value, especially considering that the dataset contains a rather limited number of data points.

We can compare crossvalidation results using t-test and since we are making several comparisons we would be better off using Bonferroni adjustment to make the comparison more robust.

Problem 6 (3 points). You are participating to a round table where several experts discuss the techniques based on decision trees. Prof. JohnTree states that the ID3/C4.5 decision tree algorithms (the ones seen during the course) are guaranteed to find an optimal tree (that is, a tree that best classifies the training tuples over all possible trees). MartinBoost begs to differ and states that the only approach that guarantees optimality are boosting algorithms like xgboost. RudyForest joins the discussion and declares that Random Forests are the only possible mean to guarantee optimality. Comment the three statements below and explain which one in your opinion is the most correct one.

JohnTree

MartinBoost

RudyForest

Additional Comments

Problem 3 (6 points). Consider the following code from a python notebook that applies three visualization techniques to a data set containing some biking sharing data.

```
import pandas as pd
import numpy as np
import seaborn as sns
sns.set(style="white", color_codes=True)
data = pd.read_csv('london.csv')
input_variables = data.columns[data.columns!='cnt']
target_variable = 'cnt'
```

```
data.head()
```

	cnt	t1	t2	hum	wind_speed	weather_code	is_holiday	is_weekend	season
0	182	3.0	2.0	93.0	6.0	3.0	0.0	1.0	3.0
1	138	3.0	2.5	93.0	5.0	1.0	0.0	1.0	3.0
2	134	2.5	2.5	96.5	0.0	1.0	0.0	1.0	3.0
3	72	2.0	2.0	100.0	0.0	1.0	0.0	1.0	3.0
4	47	2.0	0.0	93.0	6.5	1.0	0.0	1.0	3.0

```
data.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

```
cov=data[input_variables].corr()
```

Visualization #1

```
sns.heatmap(cov,square=True,annot=True,cmap="Blues");
```

Visualization #2

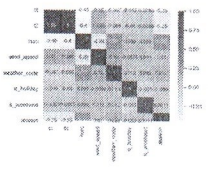
```
sns.clustermap(cov, square=True, annot=True, cmap="Blues");
```

Visualization #3

```
sns.clustermap(data[input_variables])
```

Question #1: What will the first visualization (#1) plot?

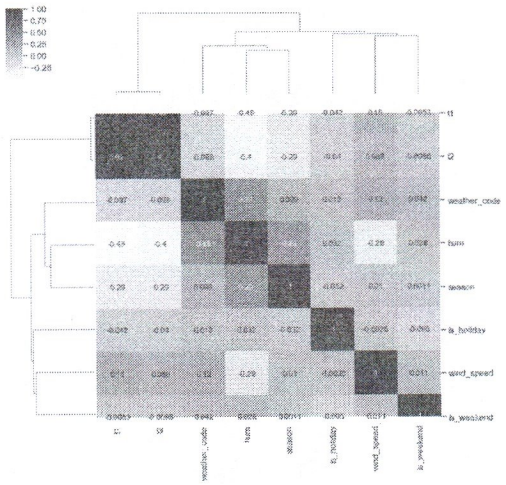
This plots the correlation matrix as a heatmap with darker blue colors representing higher correlation values.



Problem 3 (continued).

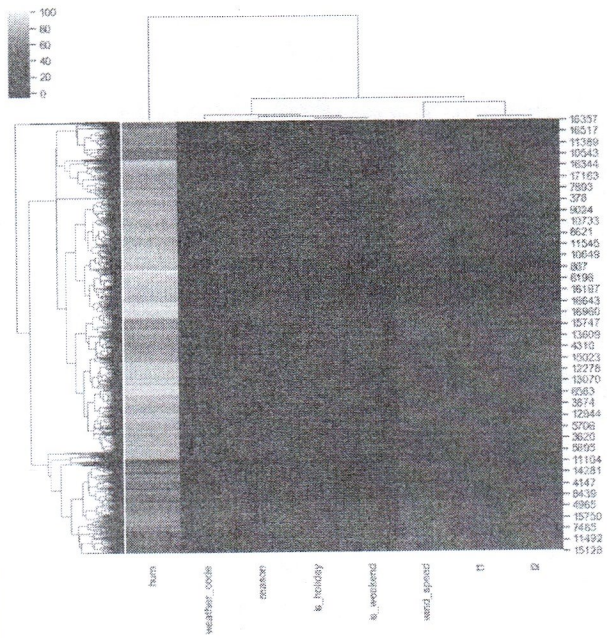
Question #2: What will the second visualization (#2) plot? How does it differ from #1?

This also performs a hierarchical clustering on the correlation matrix and also plots the dendrogram. It also sort the variables according to the dendrogram highlight clusters of variables with similar correlation values over the entire matrix row/column.



Question #3: What will the third visualization (#3) plot? What type of information does it give us?

By applying the clustermap over the entire dataset it plots the heatmap of all the data and also applies hierarchical clustering over the rows (as usual) and over the columns to search group of variables with similar behavior.



Problem 5 (6 points). Suppose you are applying bagging trees using 40 base classifiers to a binary classification problem. Each classifier has an error rate e=0.60; assume that classifiers are independent. Compute the probability that the ensemble classifier makes a wrong prediction and comment the result.

Why does it work?

- Suppose there are 25 base classifiers
- Each classifier has error rate, $\varepsilon = 0.35$
- Assume classifiers are independent
- The probability that the ensemble makes a wrong prediction is

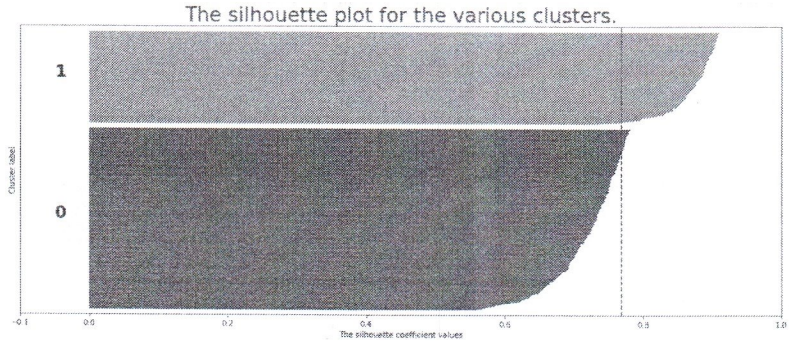
$$\sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1-\epsilon)^{25-i} = 0.06$$

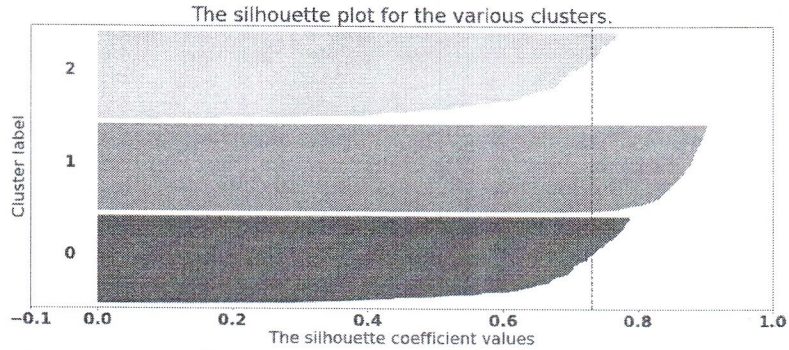
If we repeat the computation with 40 classifiers and an error of 0.6 we obtain an ensemble error of 0.87 that is much higher than the error of the single classifiers. This because the error on the binary classification problem is larger than 0.5. Thus it is not convenient to use an ensemble with these parameters unless we modify our interpretation of the classifiers' output.

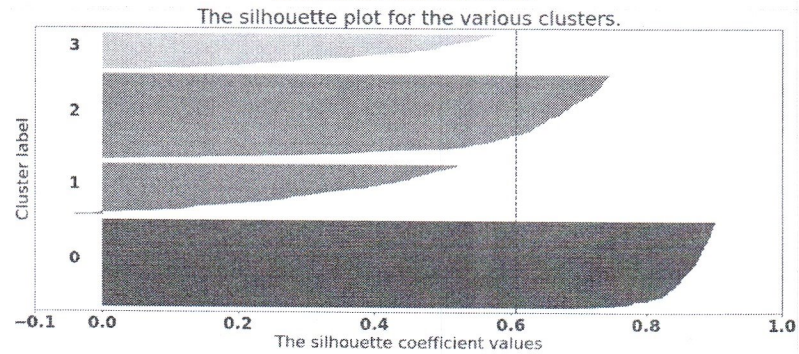
Problem 3 (6 points). You applied k-means with different values of k (2, 3, 4, 5) to a data set. To compare the solutions you first compute the following average silhouette score:

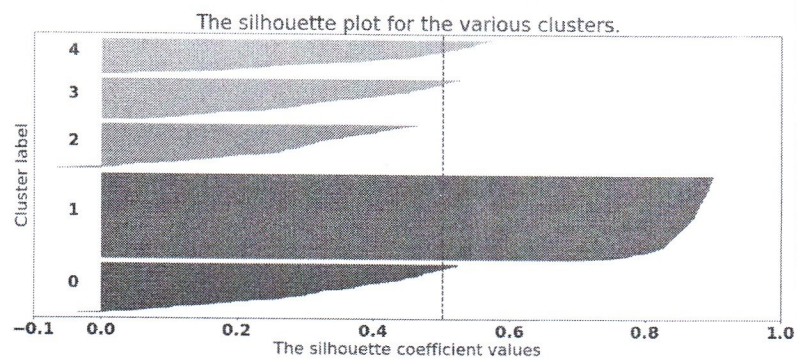
# Clusters	Average Silhouette Score
2	0.77
3	0.73
4	0.60
5	0.50

And then further analyzed the results creating the silhouette plots below.









Problem 3 (continued).

Question #1: Given the average silhouette score, what value of k would you choose?

This problem is taken from the python notebook and has been discussed in class. Check the recordings for further discussions.

Given the average silhouette score, we select k=2 since it is the higher score. But as stated during the course, the score tells only one side of the story.

Question #2: Discuss each silhouette plot adequately commenting why the corresponding value of k should be chosen or discarded

Comment for Silhouette Plot k=2

We have to clusters one completely above the average silhouette score and another very large cluster with almost all the points below the average silhouette score. Good candidate.

Comment for Silhouette Plot k=3

We have three clusters with almost the same number of data points with data points above the average silhouette score. Better candidate than the previous one.

Comment for Silhouette Plot k=4

Two clusters are way below the average silhouette score, so not a good candidate.

Comment for Silhouette Plot k=5

Three clusters have very few points above the average which in this case is very low so it seems that these clusters are now well defined. Not a good candidate, probably the solution with k=3 is the best one at this point.

Problem 6 (3 points). You are an employee of ClusterThis! A company specialized in the clustering of massive amount of data, described by thousands of variables, using k-means clustering. While you are sitting at your desk your boss storms into the room and tells you that is worried since your competitor (the company ThisIsClustering) is applying a brand-new method that is very successful called Mean Shift Clustering. Your boss is worried and asks you whether, for what you are doing, k-means is still the best option. What would be the pros for switching to Mean Shift Clustering? What would be the cons? Elaborate an answer for your boss.

There are several pros and cons in replacing k-Means with Mean Shift Clustering. From the discussion, it appears there are no issues about k-Means, in fact our boss does not mention existing issues with k-Means. She is only asking whether we should switch. So, k-Means is working and the question is whether it would be a good idea to switch. The only thing we know about our application is that we apply k-Means to massive amounts of data and have thousands of variables and this is a problem for Mean Shift that we know has a higher computational complexity with respect to k-Means ($O(Tn^2)$ instead of $O(Tkn)$ see the slides). So, no, it would not be a good idea to switch.