DAT405: INTRODUCTION TO DATA SCIENCE AND AI

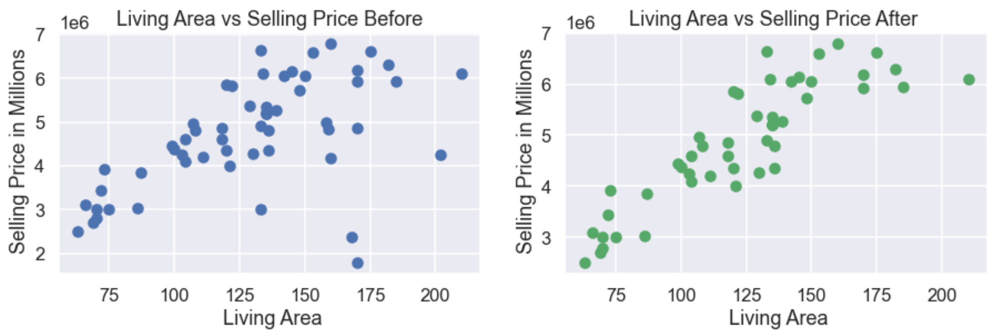
MODULE2

BROU BONI Joël

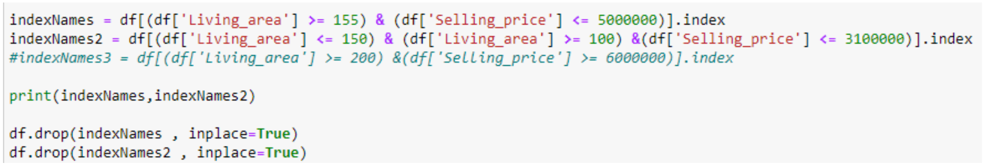
SANOGO Ibrahim Bechir

Part 1

a- Before coding the linear regression, we decided to make some data cleaning. Indeed, we decided to take away the points that were too far away from the main group of point as you can see in the following figure. These points not respecting at all the linearity between the selling price and the living area may make our prediction less accurate for the rest of the houses since they will be taken into account by our machine learning algorithm to calculate the slope and the intercept that define the equation of our regression line. After calculating the R-squared before and after cleaning the dataset we saw that it went from 0.42 to 0.81 which proves the efficiency of the cleaning and show how the 7 points that we removed were impacting badly our model. But the main limitation of our approach is its adaptability. Even if we obtain a good linear regression with this, here we make the hypothesis that any point that does not respect the linearity between the selling price and the living area is an outlier, which is not correct because the price of a house can depend on other factors. Therefore we will compare the performances of our 2 models (with data cleaning and without) through this report (especially in question d).



To do so we used theses codes lines made with the Pandas Package.

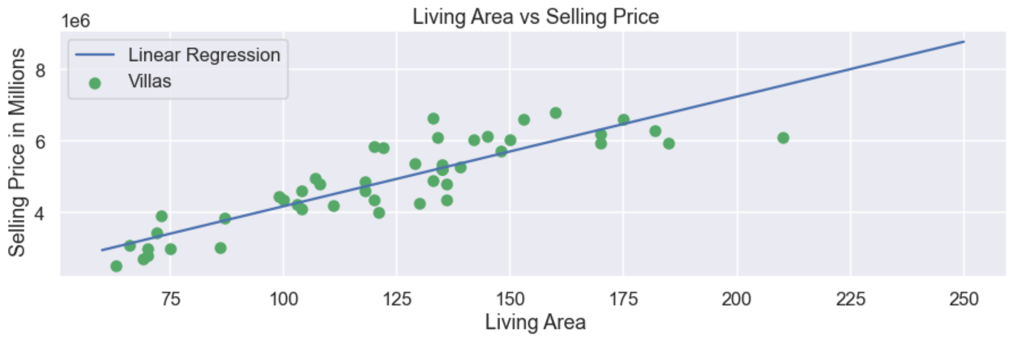


```
indexNames = df[(df['Living_area'] >= 155) & (df['Selling_price'] <= 5000000)].index
indexNames2 = df[(df['Living_area'] <= 150) & (df['Living_area'] >= 100) &(df['Selling_price'] <= 3100000)].index
#indexNames3 = df[(df['Living_area'] >= 200) &(df['Selling_price'] >= 6000000)].index

print(indexNames,indexNames2)

df.drop(indexNames , inplace=True)
df.drop(indexNames2 , inplace=True)
```

After cleaning the dataset, we coded the regression line using sk-learn and plotted the regression line with the scatter plot as you can see down below:



b- We got the values down below for the slope and the intercept.

```
Entrée [12]: print("The slope is equal to", reg.coef_.tolist())

             The slope is equal to [30701.643235557178]

Entrée [13]: print("The intercept is equal to", reg.intercept_.tolist())

             The intercept is equal to 1106034.7625163412
```

Which means that the equation of our regression line is $30701.64x + 1106034.76$.

c- We get the following predictions for an living area of 100,150,200 m^2 for a clean (left) and non-cleaned (right) dataset.

	Living Area	Predicted Price
0	100.0	4.176e+06
1	150.0	5.711e+06
2	200.0	7.246e+06

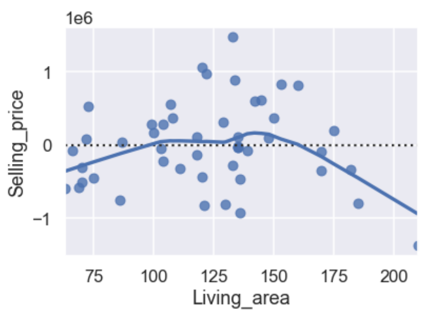
	Living Area	Predicted Price
0	100.0	4.227e+06
1	150.0	5.071e+06
2	200.0	5.915e+06

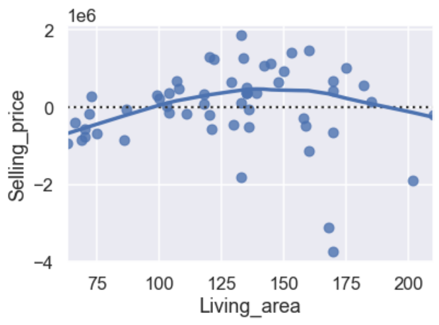
We can see that the results for 100 and 150 are pretty close to the excepted price from theses villas but the prediction for a 200 m2 is far from the real price. According to the result that we got and the figure in the question a we think that other features should be take in count to predict the price of a villa. With the non-cleared dataset, The results is pretty much the same but we can see that the estimations are less accurate for 150 m^2 square but the prediction is good for 200m2

d- After using the seaborne package, we got the residual plot and the lowess curve downwards for a clean (left and non-cleared right) datasetAfter using the seaborne package, we got the residual plot and the lowess curve downwards for a clean (left and non-cleaned right) dataset

```
Entrée [11]: sns.residplot(x=df['Living_area'], y=df['Selling_price'],lowess=True)

   Out[11]: <AxesSubplot:xlabel='Living_area', ylabel='Selling_price'>
```





A residual plot is a graph that shows the residuals (Observed value - Predicted value) on the y-axis and the independent variable on the horizontal axis. Based on the formula of the residual, the more the points in a residual plot are randomly dispersed around the horizontal axis, the more a linear regression model is appropriate for the data. A LOWESS curve is a tool used in regression analysis that creates a smooth line scatter plot in order to help to see a relationship between variables and foresee trends. A LOWESS curve will then help us to have better visualization and to analyses better the dataset. The residual and the curve help us to confirm what we said in the previous question. Indeed, we can see that our model with a cleaned dataset seems to be mode accurate for 100 to 150 m2. Indeed, the lowess curve for this model is closer to zero than the other one while it's the opposite for

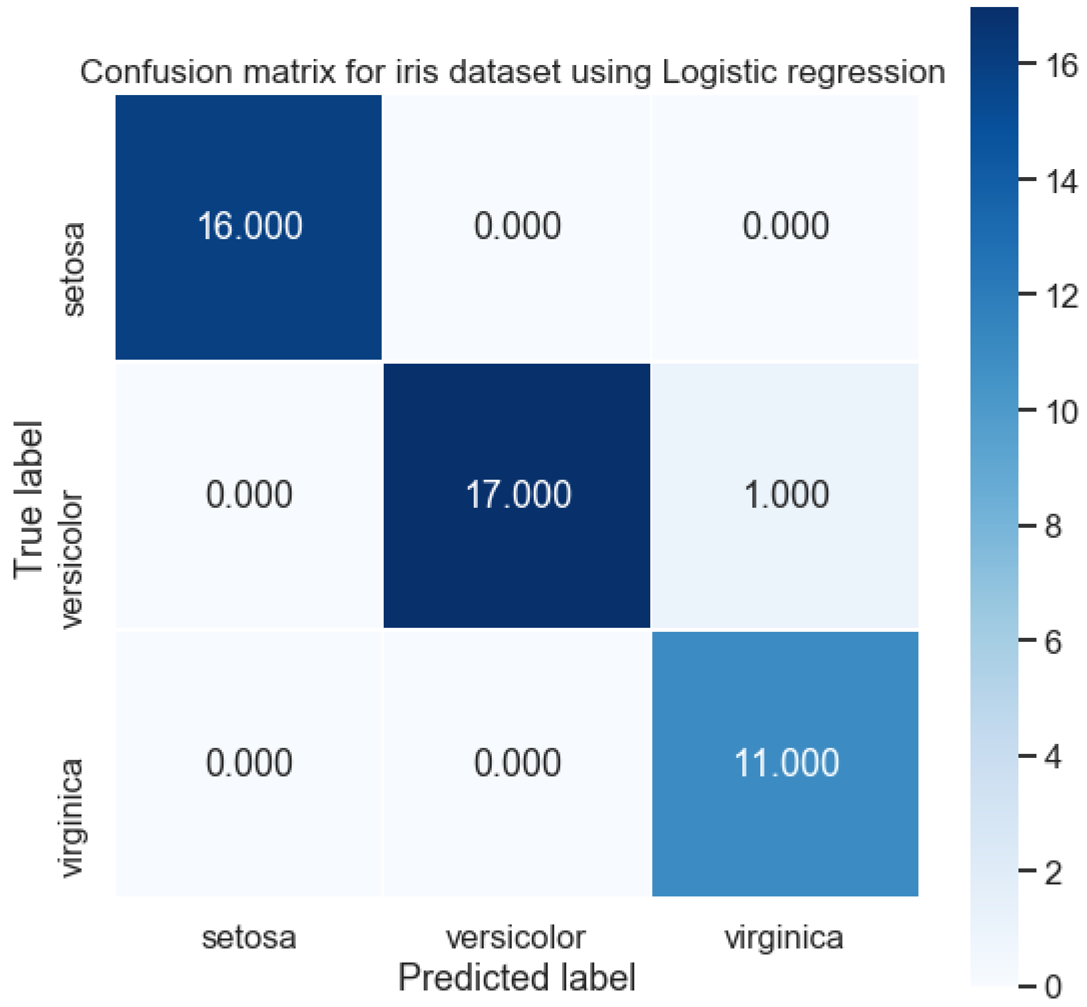
200 m2. But for both models, we can see that the residuals form a random pattern which means that linear regression is not the best model to estimate the price of a house. This confirms our suspicions in question 1 indeed the price of a house should not depend only on the living area.

e- In order to have a better estimation of the selling prices, we have to take in count the other features in the dataset. The price of a house should not depend on many things such as the date of construction the total land size and more important the localization. A house in a big city doesn't cost the same as a house in a smaller one for example and the price can even fluctuate depending on the neighborhoods. The address could for example be add in the dataset regarding what we said. One solution could be to use linear regression for different categories of houses and not for all of them at the same time. By using a clustering algorithm we could split the scatter plot into different clusters of houses with the same characteristics and then make a linear regression for each cluster. But to doing this will require to scale the data between 0 and for 1 for example so all the features have the same order of magnitude and this can be easily done with SK-learn.

Part 2

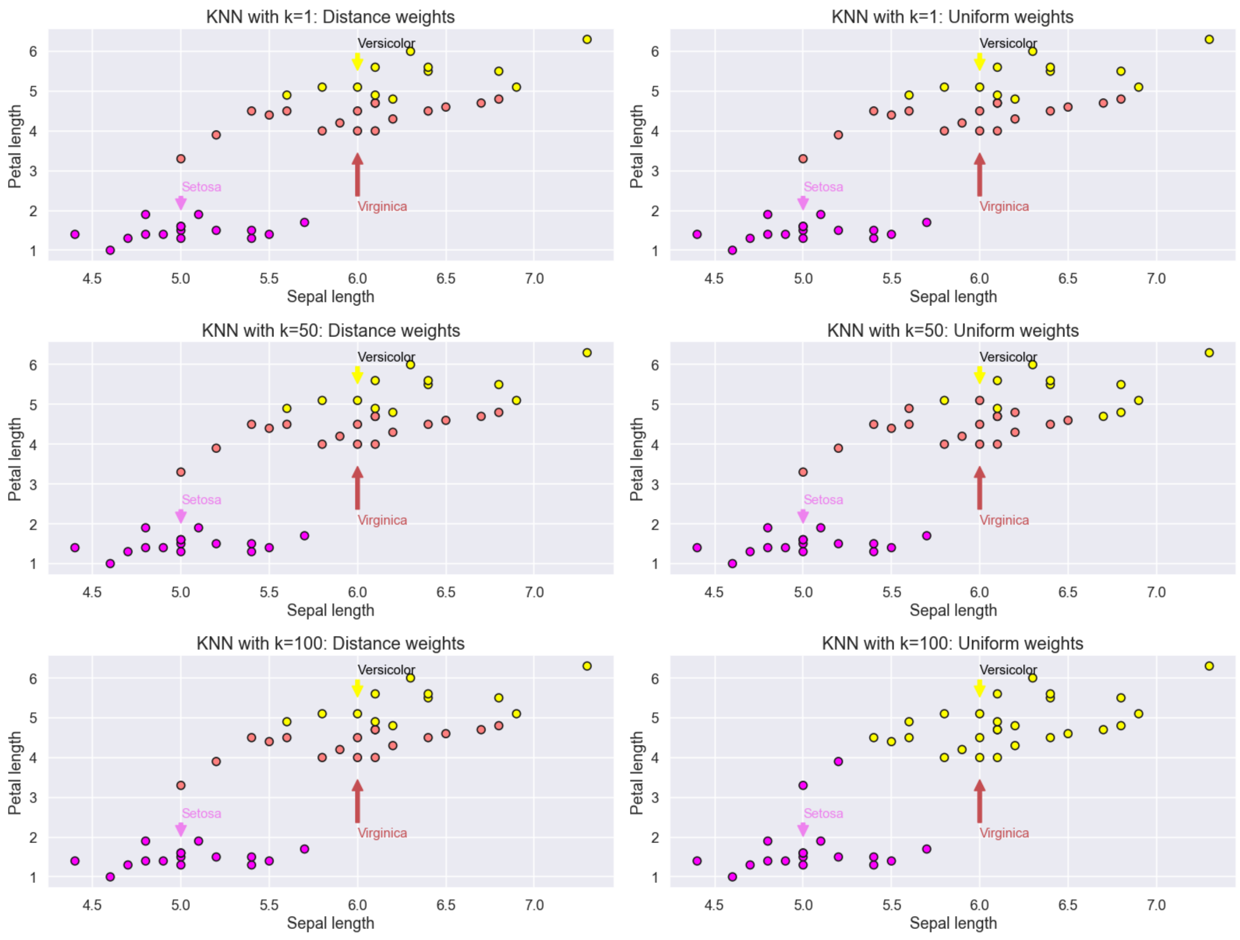
a- Before using the logistic regression, we split the dataset between training data (70%) and test data (30%). We trained our model with the logistic regression model and then applied it to our test dataset.
We got an accuracy of **97.8%** with the logistic regression by comparing the true labels of the iris test data to the predicted labels of our regression.The logistic regression is working efficiently to classify the iris between *Setosa, Versicolor and Virginica*.

We use a confusion matrix to evaluate the use of logistic regression to classify the iris data set.



b|c - We use k-nearest neighbors to classify the iris data set with some different values for k, and with uniform and distance-based weights.

We use k=1,50 and 100 points for the uniform and distance-based weights and we compared the results.



We compiled the observed results in the following array.

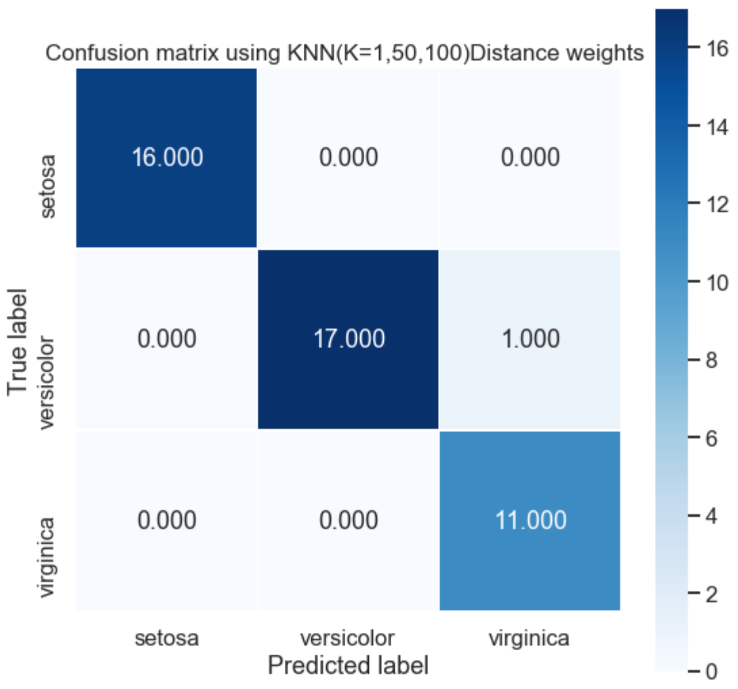
Accuracy	Distance-based weights	Uniform weights
K=1	97.8%	97.8%
K=50	97.8%	91.8%
K=100	97.8%	60%

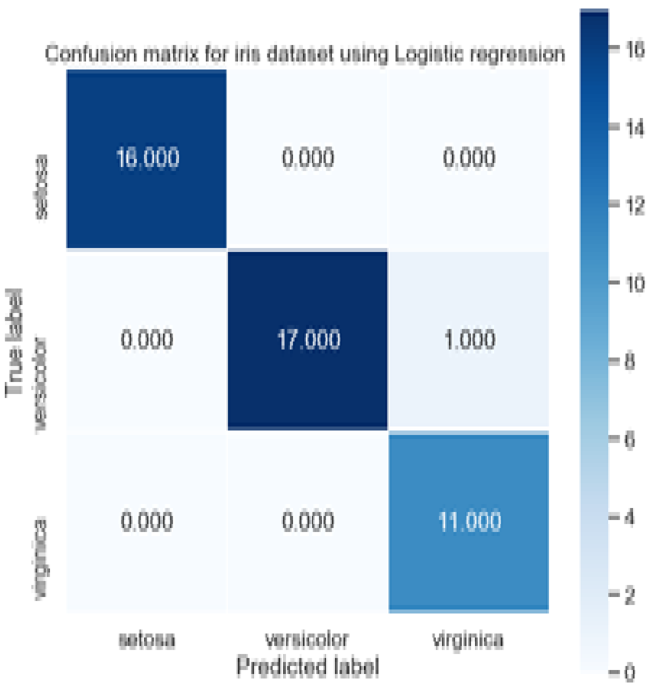
The choice of K has major effects on the classifier created, especially when k grows larger for the different cases.

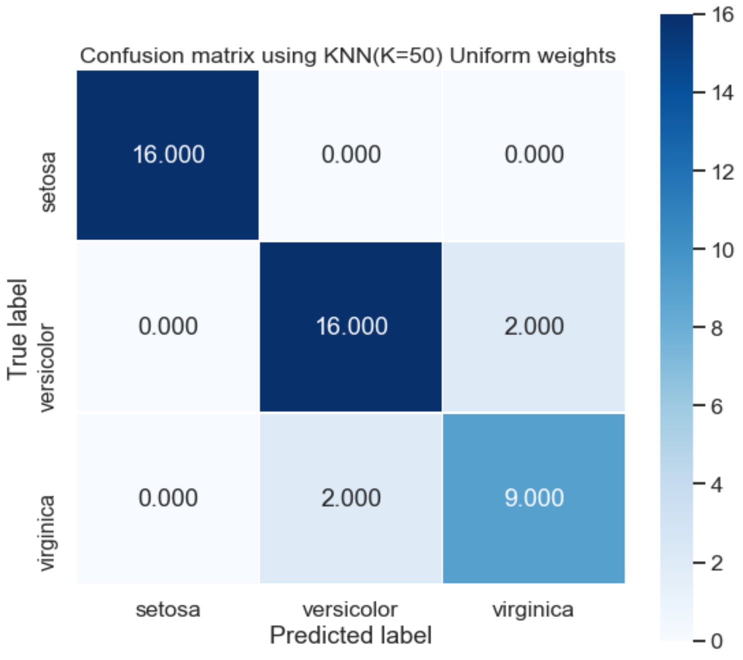
- The greater the K, more linear (high bias and low variance) the decision boundary. We can see it for the uniform case for K=50 and K=100. Since *Versicolor* and *Virginica* are close species, the classifier cannot separate correctly the two species and is completely biased for K=100 points.
- Uniform-based weights means that all neighbors get an equally weighted "vote" about an observation's class. It could be problematic to use it when there is not a clear distinction between two classes.
- Distance-based weights would tell the classifier to weigh each observation's "vote" by its distance from the observation we are classifying. We have an excellent constant score accuracy for it because for one species, all the points are really close, especially for *Setosa*.

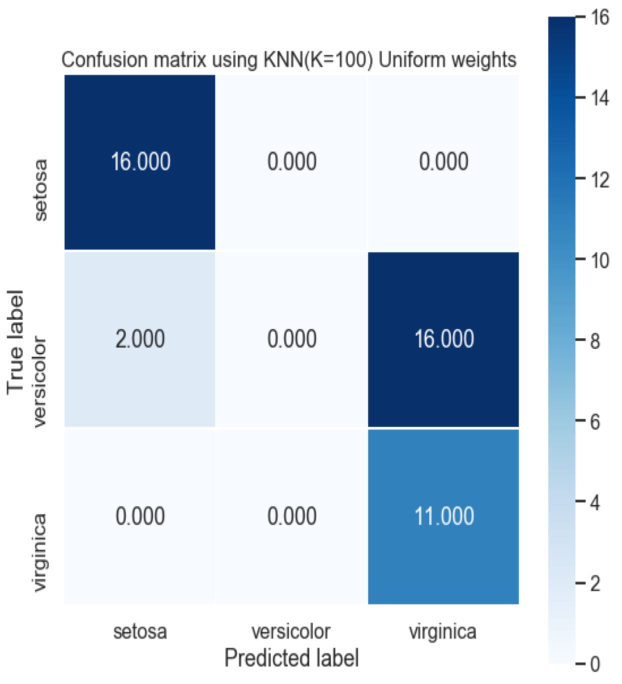
The classifier will not be biased because it will not give a strong importance to points situated far away even when K grows. However, it could not consider outliers. We do not have them in our dataset, but it exists some *Setosa* with higher petal/sepal length and our classifier do not take this into account.

c-









We calculated confusion matrices for these models and already
discussed the performance of the various models during the previous question. The confusion matrix totally represents what have been discussed earlier and explains the efficiency of the different models (excellent logistic regression and KNN distance-based weights, very good and poor KNN uniform-based weights when K grows)

Part 3

It is extremely important to use a separate test (and sometimes validation) set. The test set is used to provide an unbiased evaluation of a final model fit on the training dataset. It contains carefully sampled data that spans the various classes that the model would face.

Our test set proved us how biased our uniform-based weights was.

The validation set results help to update higher level hyperparameters. It provides an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set affects a model, but only indirectly.

Moreover, splitting the dataset helps us to avoid the overfitting problems. After we have trained our model, testing it on fresh sample helps us to estimate how well our model is performing given that it is facing new inputs. If the model memorized the pattern from the training dataset and performs bad with the new inputs we can see that there is an overfitting problem for example. This is why it its even better to split the dataset into 3 parts, the training set, cross validation set and test set. The validation set is used to tune the model.