Calofir Ionut

Homework 1

For this homework we had two main tasks based on the *Dog Breeds*:

- Classification
- Regression

1. Dataset

The dataset contains the following columns: *Weight(g), Height(cm), Energy Level, Attention Needs, Owner Name, Coat Length, Sex, Breed Name, Longevity(yrs).*

Steps for processing the dataset:

- I deleted the *Owner Name* column because this column does not bring any useful information for Classification and Regression tasks
- For numerical columns (*Weight(g), Height(cm)*):
    - I filled the missing values using the mean along each column
    - I created new polynomial features from those two columns
    - I scaled these new columns by removing the mean value of each feature, and by dividing features by their standard deviation. Scaling is important because some features may dominate and the estimator will be enable to learn properly.
- For categorical columns (*Energy Level, Attention Needs, Coat Length, Sex*):
    - I filled the missing values using the most frequent value along each column
    - I transformed every column into one hot vector

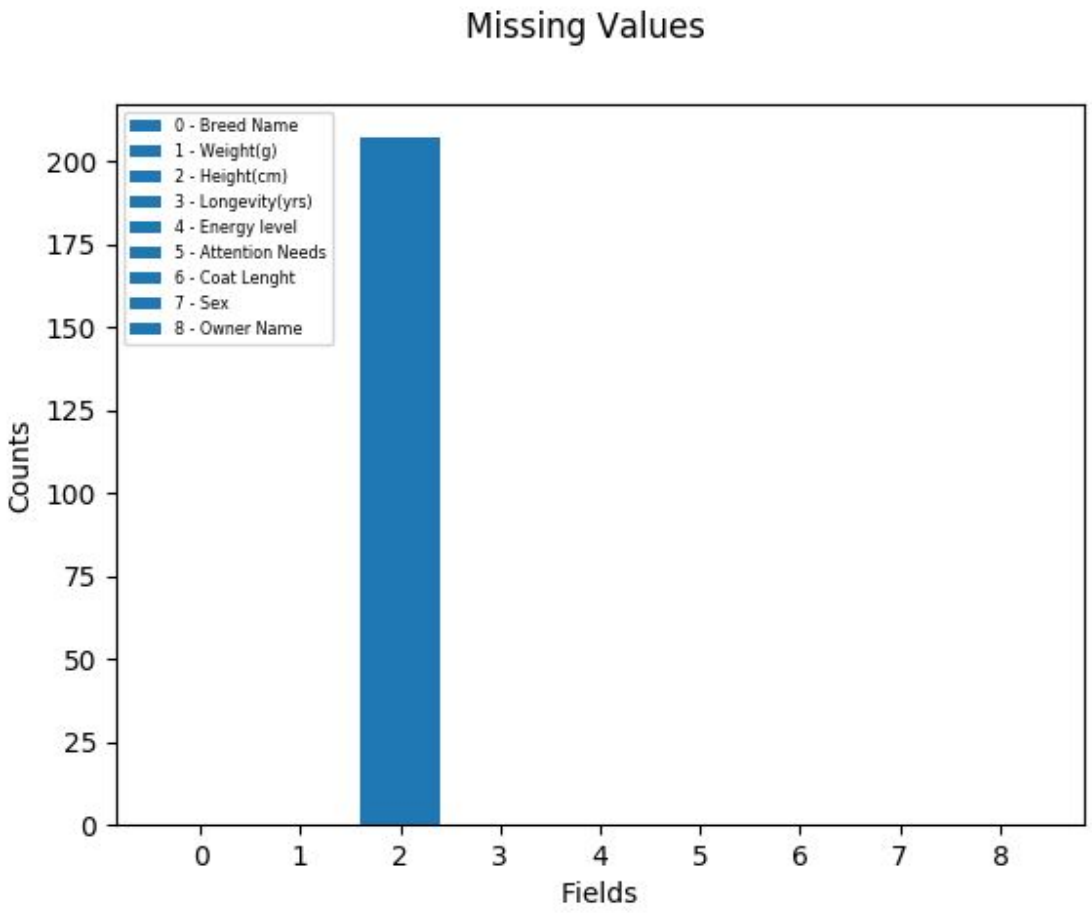
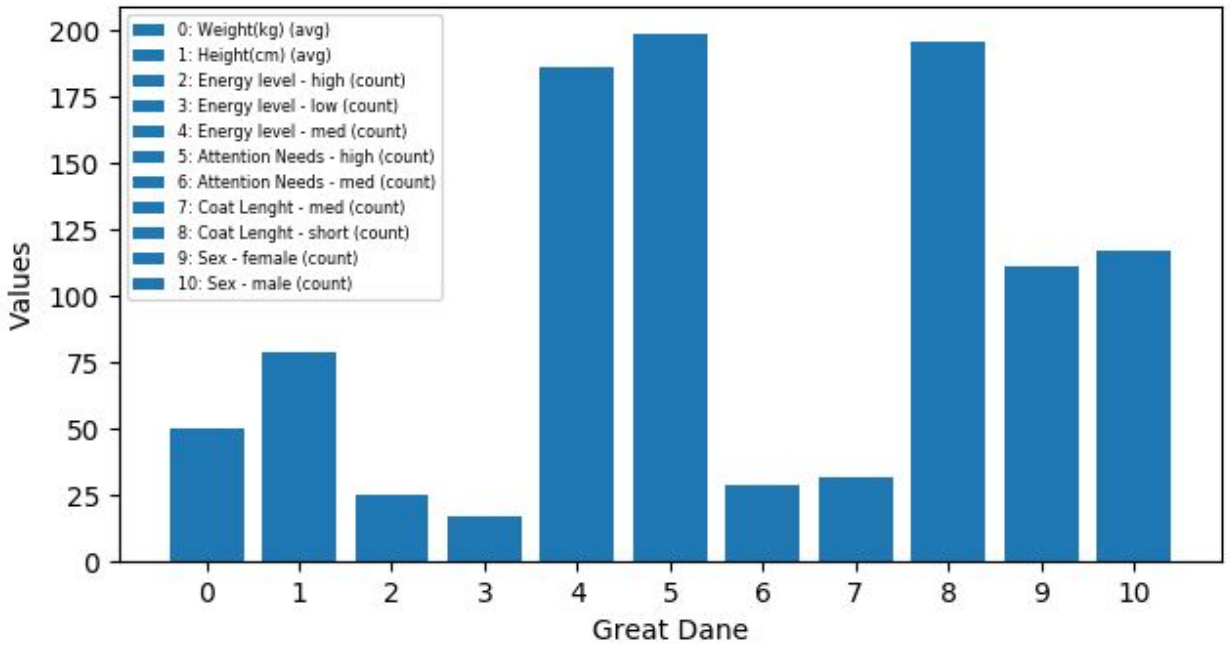


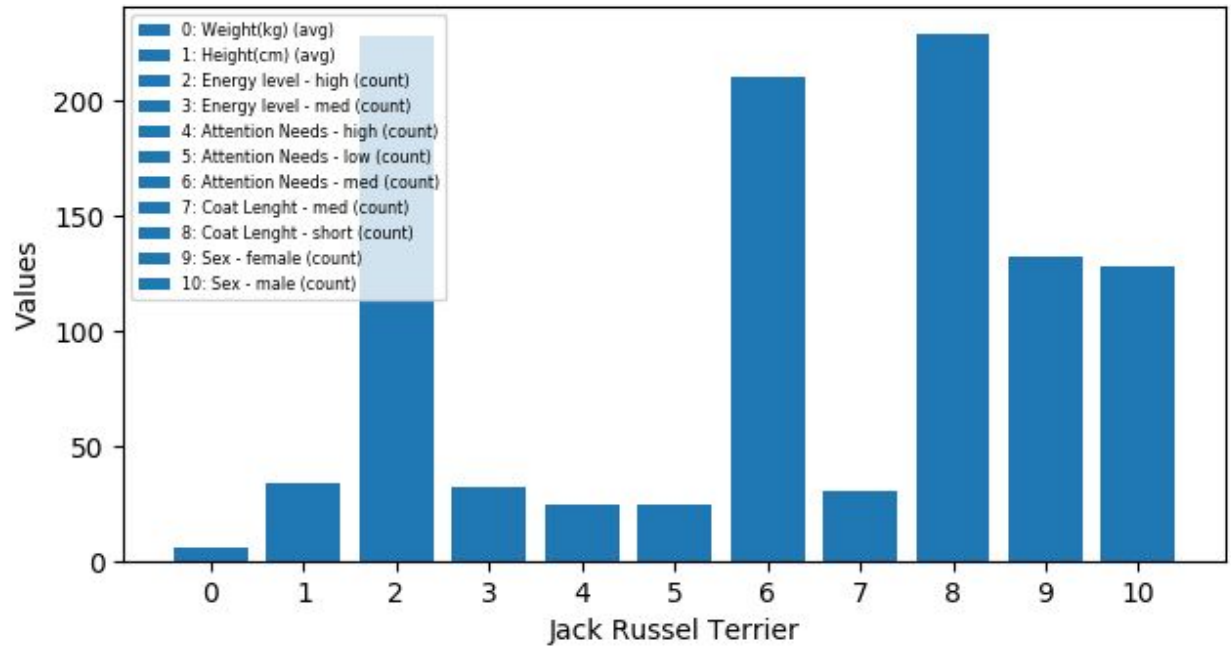
Fig - number of missing values for every feature

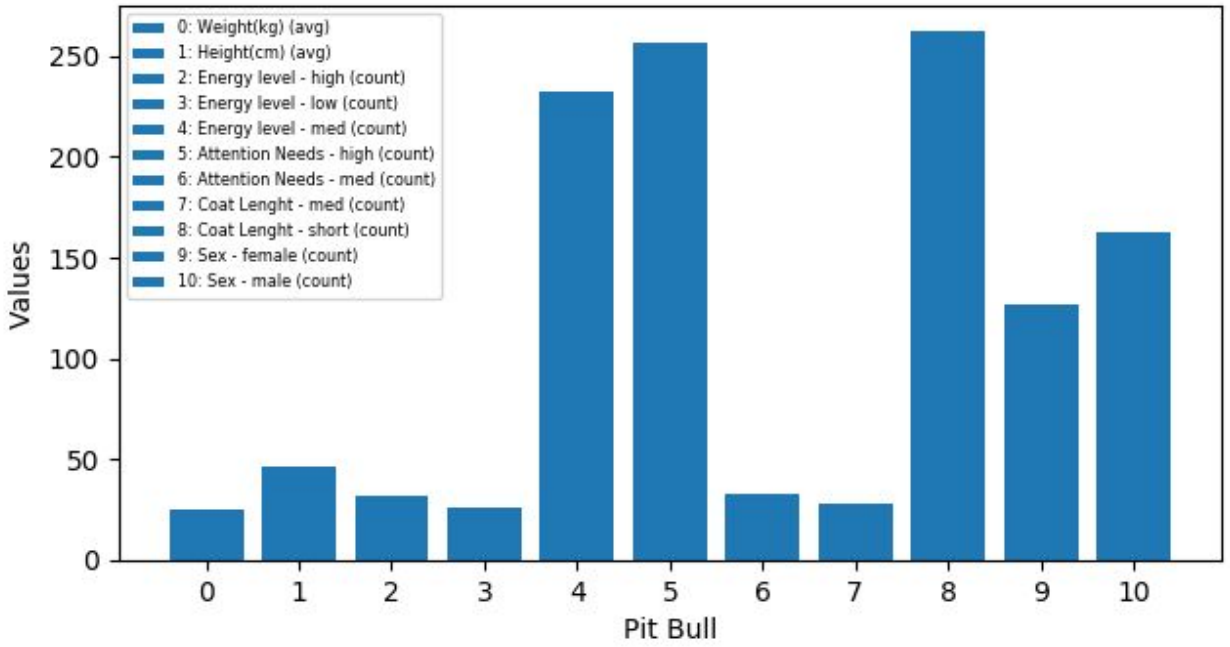
2. Classification

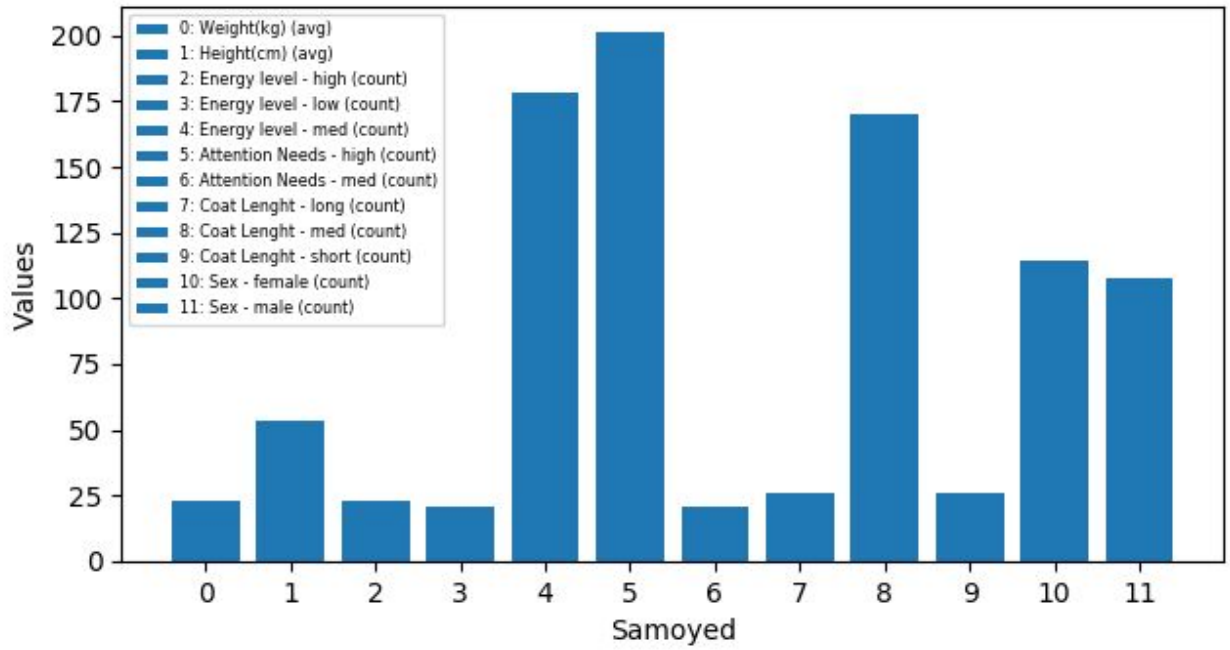
For this problem our task was to predict the *Breed Name*. My approach was to try three different models (*Logistic Regression, Random Forests and KNN*) and compare them.

In the following pictures I plotted for every *dog* some information to try to understand the dataset. From these pictures we can see that some features appear more often for some dogs, so my intuition is that these classes are well separable.

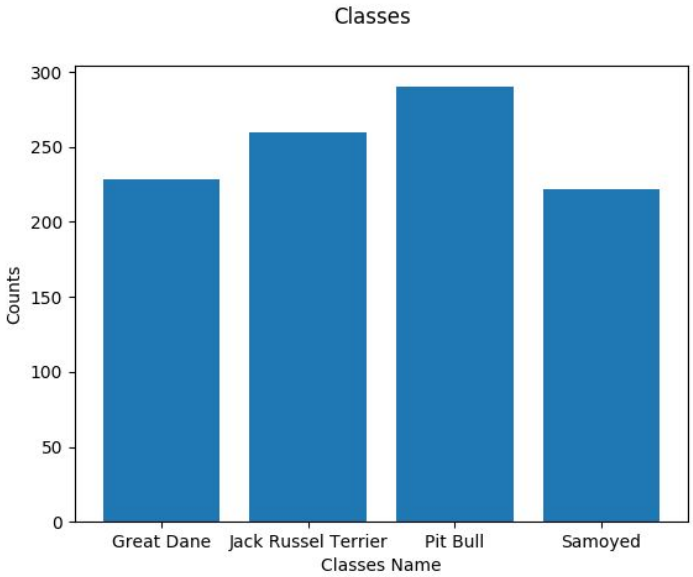

0: Weight(kg) (avg)
1: Height(cm) (avg)
2: Energy level - high (count)
3: Energy level - low (count)
4: Energy level - med (count)
5: Attention Needs - high (count)
6: Attention Needs - med (count)
7: Coat Lenght - med (count)
8: Coat Lenght - short (count)
9: Sex - female (count)
10: Sex - male (count)
Values
200
175
150
125
100
75
50
25
0
0 1 2 3 4 5 6 7 8 9 10
Great Dane


0: Weight(kg) (avg)
1: Height(cm) (avg)
2: Energy level - high (count)
3: Energy level - med (count)
4: Attention Needs - high (count)
5: Attention Needs - low (count)
6: Attention Needs - med (count)
7: Coat Lenght - med (count)
8: Coat Lenght - short (count)
9: Sex - female (count)
10: Sex - male (count)
Values
200
150
100
50
0
0 1 2 3 4 5 6 7 8 9 10
Jack Russel Terrier


0: Weight(kg) (avg)
1: Height(cm) (avg)
2: Energy level - high (count)
3: Energy level - low (count)
4: Energy level - med (count)
5: Attention Needs - high (count)
6: Attention Needs - med (count)
7: Coat Lenght - med (count)
8: Coat Lenght - short (count)
9: Sex - female (count)
10: Sex - male (count)
Values
0
50
100
150
200
250
0 1 2 3 4 5 6 7 8 9 10
Pit Bull


0: Weight(kg) (avg)
1: Height(cm) (avg)
2: Energy level - high (count)
3: Energy level - low (count)
4: Energy level - med (count)
5: Attention Needs - high (count)
6: Attention Needs - med (count)
7: Coat Lenght - long (count)
8: Coat Lenght - med (count)
9: Coat Lenght - short (count)
10: Sex - female (count)
11: Sex - male (count)
Values
0
25
50
75
100
125
150
175
200
0 1 2 3 4 5 6 7 8 9 10 11
Samoyed

I also plotted the number of every class to see if they are balanced.



Training

To decide which model is the best I compared the three models with their best hyperparameters. To choose the best hyperparameters I used grid search with k-fold cross validation on a train set for every model.

I kept a separate test set to evaluate the best model at the end. For cross validation, **k** = 10.

To evaluate every model I looked at the F1 score. I used F1 weighted to take label imbalance into account. In context of cross validation, I selected the mean F1 score over the test splits.

Here are some results:

- KNN - the best model has a F1 score around 96%
- Logistic Regression - the best model has a F1 score around 98%
- Random Forests - the best model has a F1 score around 98%

More results can be found in the ***grid_search folder***.

I chose Logistic Regression as the best model with the following hyperparameters: **C = 30, class_weight = 'balanced', multi_class = 'multinomial', solver = 'lbfgs', polynomial_grade = 2**.

F1 score on the test set with the above model: 98%.

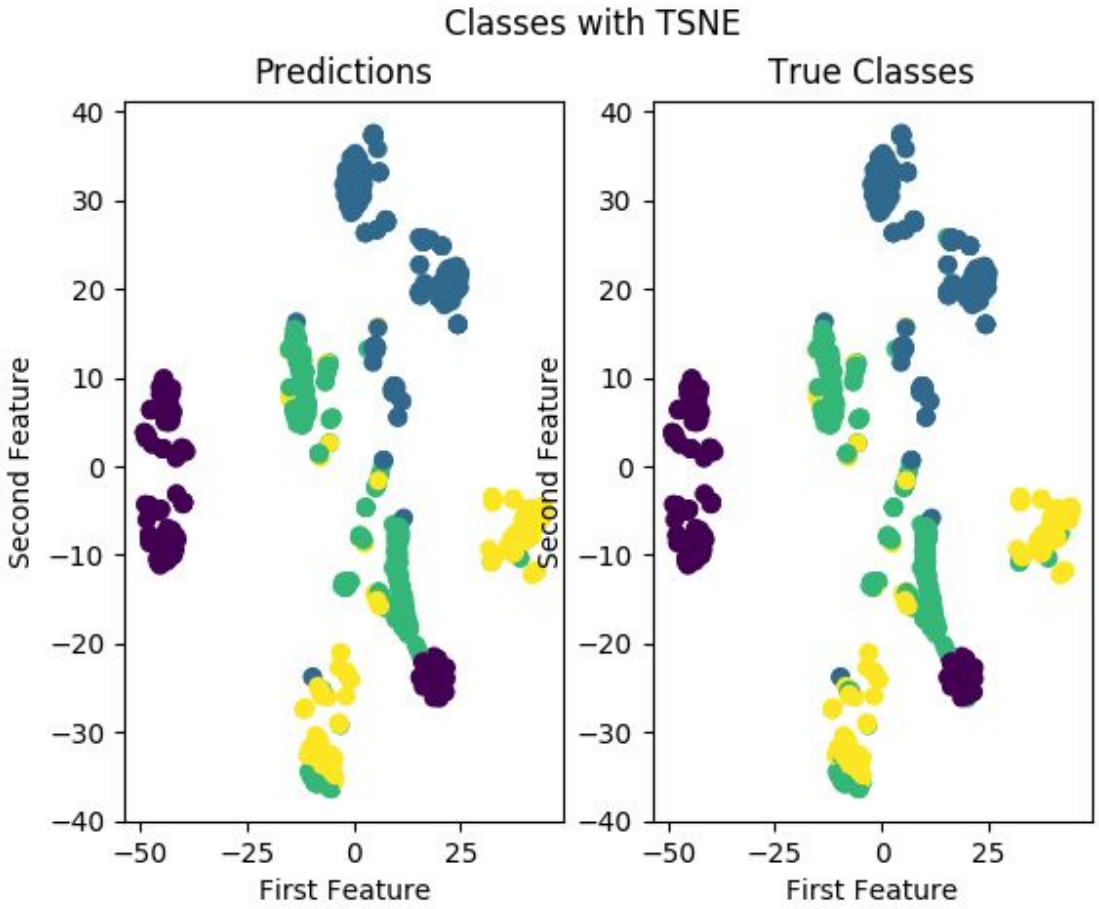
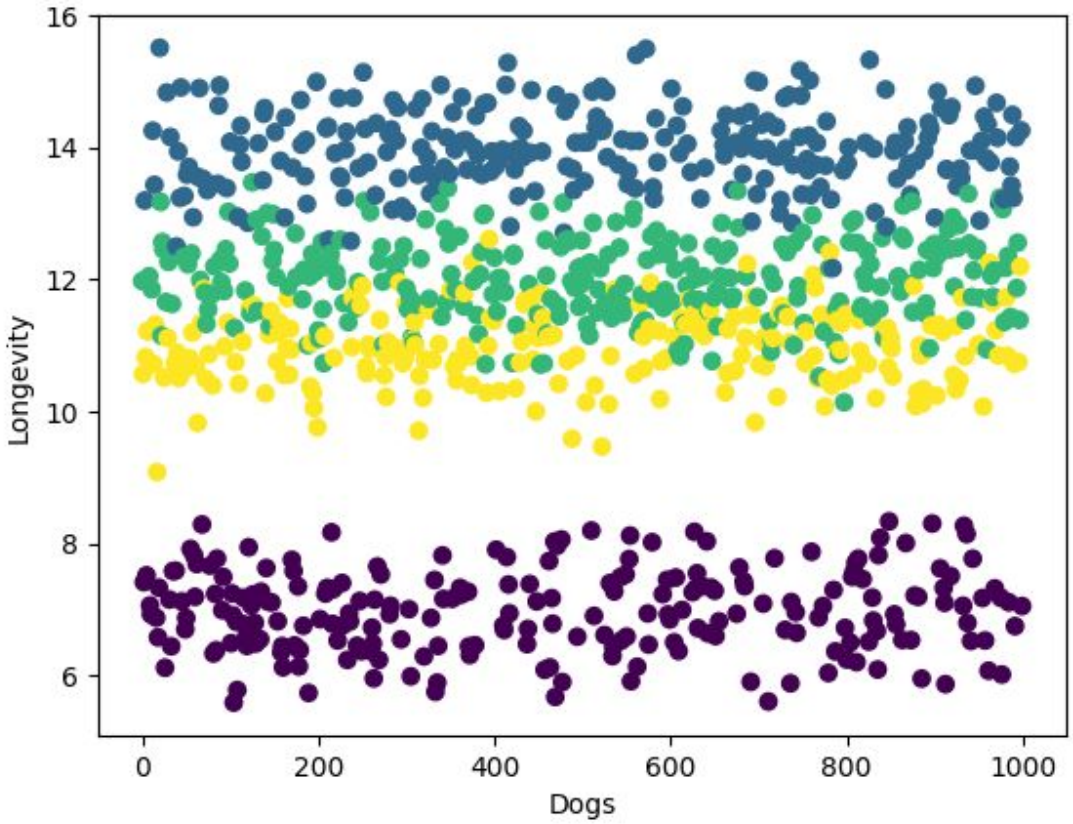


Fig - I used TSNE to obtain obtain only two features so I can plot the predicted classes from Logistic Regression and the true classes.

3. Regression

For this problem our task was to predict the *Longevity(yrs)*. My approach was to try three different models (*Ridge Regression, Lasso Regression and KNN*) and compare them.

In the picture below we can see that three classes are overlapping. The yellow and green classes are overlapping more than the blue and green classes. My intuition is that we will obtain a fairly good regression model.



Training

To decide which model is the best I compared the three models with their best hyperparameters. To choose the best hyperparameters I used grid search with k-fold cross validation on a train set for every model.

I kept a separate test set to evaluate the best model at the end. For cross validation, **k** = 10.

To evaluate every model I looked at the R2 score. In context of cross validation, I selected the mean R2 score over the test splits.

Here are some results:

- KNN - the best model has a R2 score around 0.91
- Ridge Regression - the best model has a R2 score around 0.89
- Lasso Regression - the best model has a R2 score around 0.87

More results can be found in the ***grid_search folder***.

I chose KNN Regressor as the best model with the following hyperparameters: **leaf_size = 15, n_neighbors = 5, p = 1, weights = 'uniform', polynomial_grade = 2**.

R2 score on the test set with the above model: 0.93.

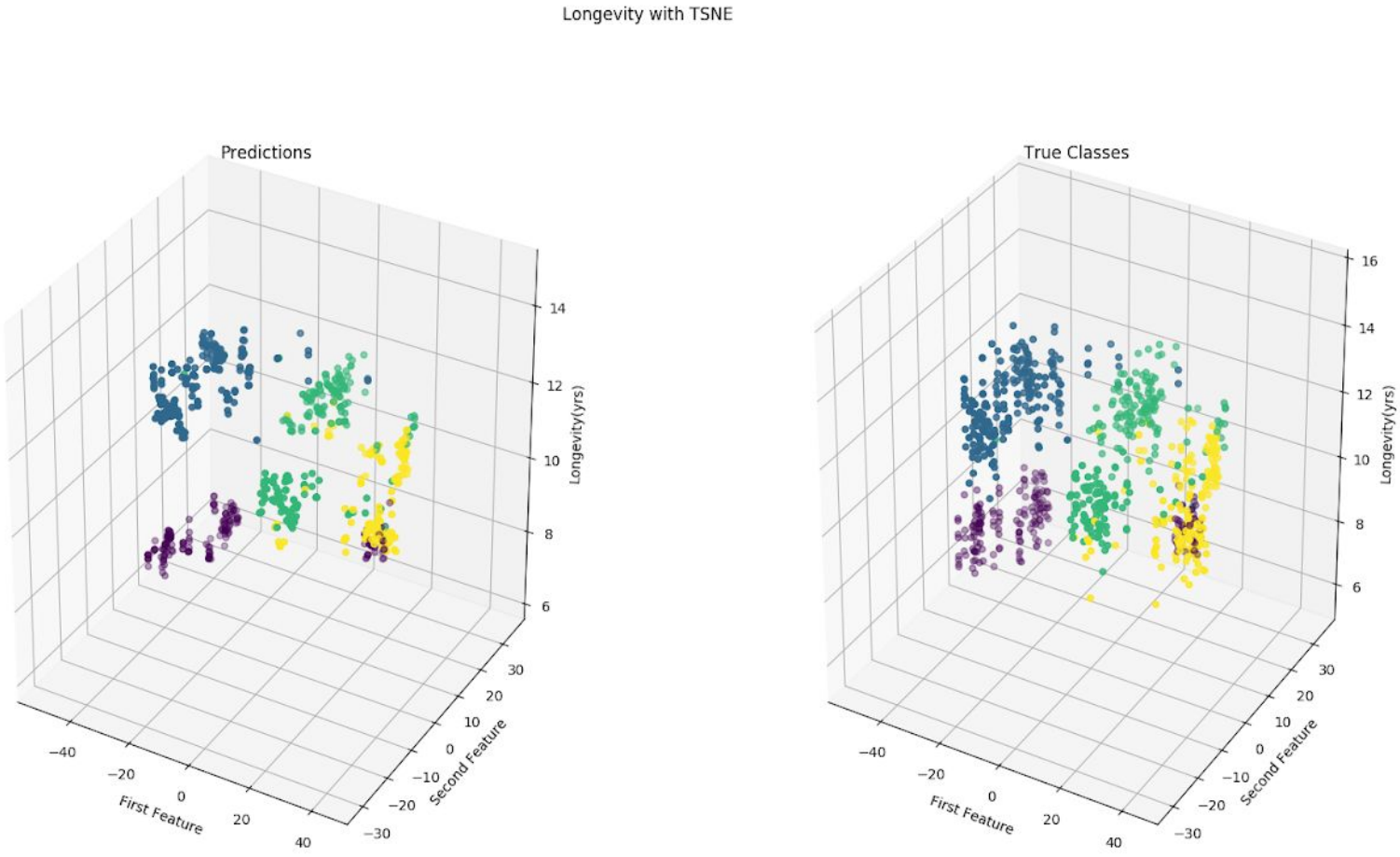


Fig - I used TSNE to obtain obtain only two features so I can plot the predicted values from KNN Regressor and the true values.