University of Insubria
Department of Theoretical and Applied Sciences
2022-2023 Academic Year

Master's Degree in
Computer Science

Intelligent Systems Project Report

Project of:
Fabio Cirelli
Registration number: 740482

Contents

1 Regression **1**
1.1 Introduction 1
1.2 Theoretical concepts 1
1.3 Methods 2
1.4 Results 7

2 Classification **8**
2.1 Introduction 8
2.2 Theoretical concepts 8
2.3 Methods 9
2.4 Results 11

Chapter 1

Regression

1.1 Introduction

This first part of the project aims at creating two linear regression models (i.e. Ordinary Least Squares regression and Least Median Squares regression) based on data about housing prices in India. Once the models have been trained it will be possible to predict the renting price of a house and it will be possible to measure the models' performance.

1.2 Theoretical concepts

Ordinary Least Squares (OLS) regression is a linear regression model that fits the data by finding the straight line that is closest to all the datapoints. Formally, the distance between the line and a datapoint is a residual, thus $res_i = y_i - \hat{y}_i$, and the regression minimises the *sum of the squared residuals*, thus it also minimises the *mean of the squared residuals*:

$$\frac{\sum_{i=1}^{n} res_i^2}{n}$$

OLS is particularly sensitive to *outliers* (i.e. datapoints that are very different from the majority of the datapoints) because the mean is not a robust indicator. Thus, those points should be removed from the dataset before training an OLS model.

Least Median Squares (LMS) regression on the other hand is a regression model that is significantly less sensitive to outliers because, instead of minimising the mean of the squared residuals, it minimises the *median of the squared residuals*: $median\{res_i^2\}$. The smaller sensitivity to outliers comes

from the use of the median indicator instead of the mean indicator, as the former has a robustness of 50%.

1.3 Methods

The dataset has the following features:

- **Posted.On**: creation date of the advertisement
- **BHK**: sum of the number of bedrooms, halls and kitchens of the house
- **Rent**: renting price of the house (in Indian Rupees per month)
- **Size**: size of the house (in square feet)
- **Floor**: what floor is the house located on and total number of floors (e.g. Ground out of 2, 3 out of 5)
- **Area.Type**: method used to calculate the size of the house (super area, carpet area or build area)
- **Area.Locality**: area of the city in which the house is located
- **City**: city in which the house is located
- **Furnishing status**: furnishing status of the house (furnished, semi-furnished or unfurnished)
- **Tenant.Preferred**: type of tenant preferred by the owner or agent
- **Bathroom**: number of bathrooms
- **Point.of.Contact**: person to be contacted for more information regarding the advertisement

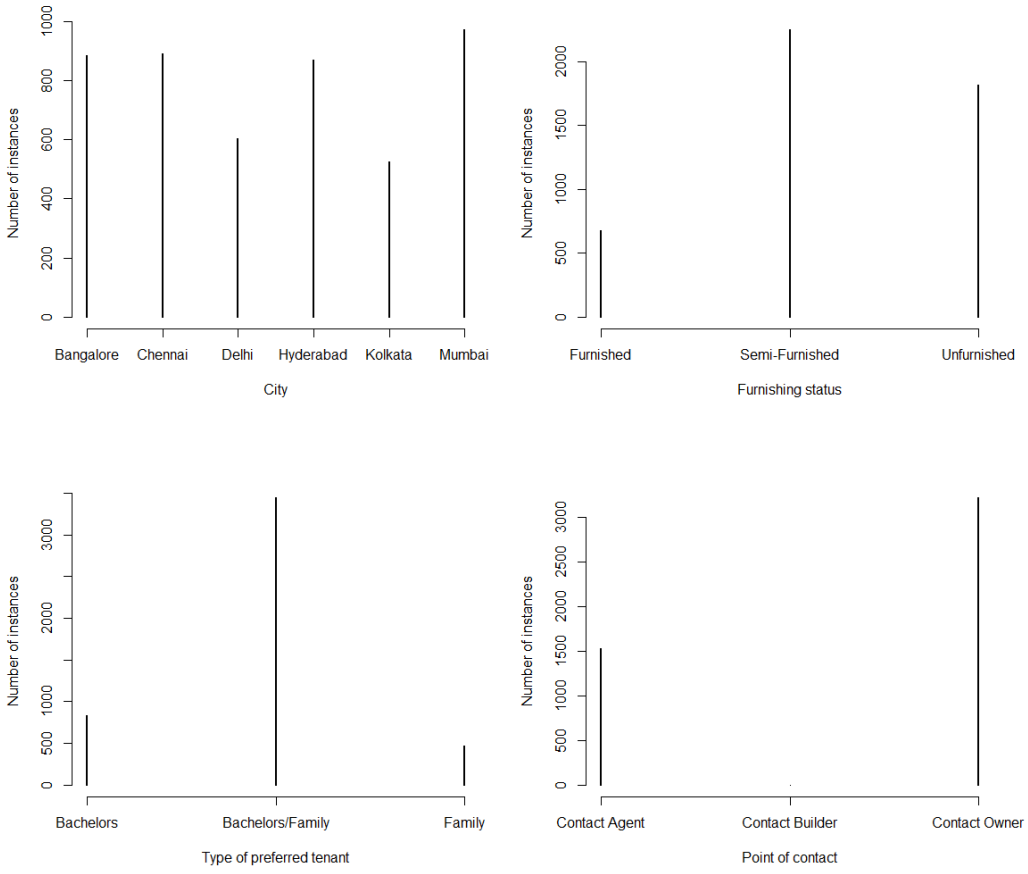

Figure 1.1: Plots of some features

These plots show the distributions of the categorical features. One thing to note is that there is only one instance where the point of contact is "contact builder", which is best to be removed as probably it is not a significant sample.

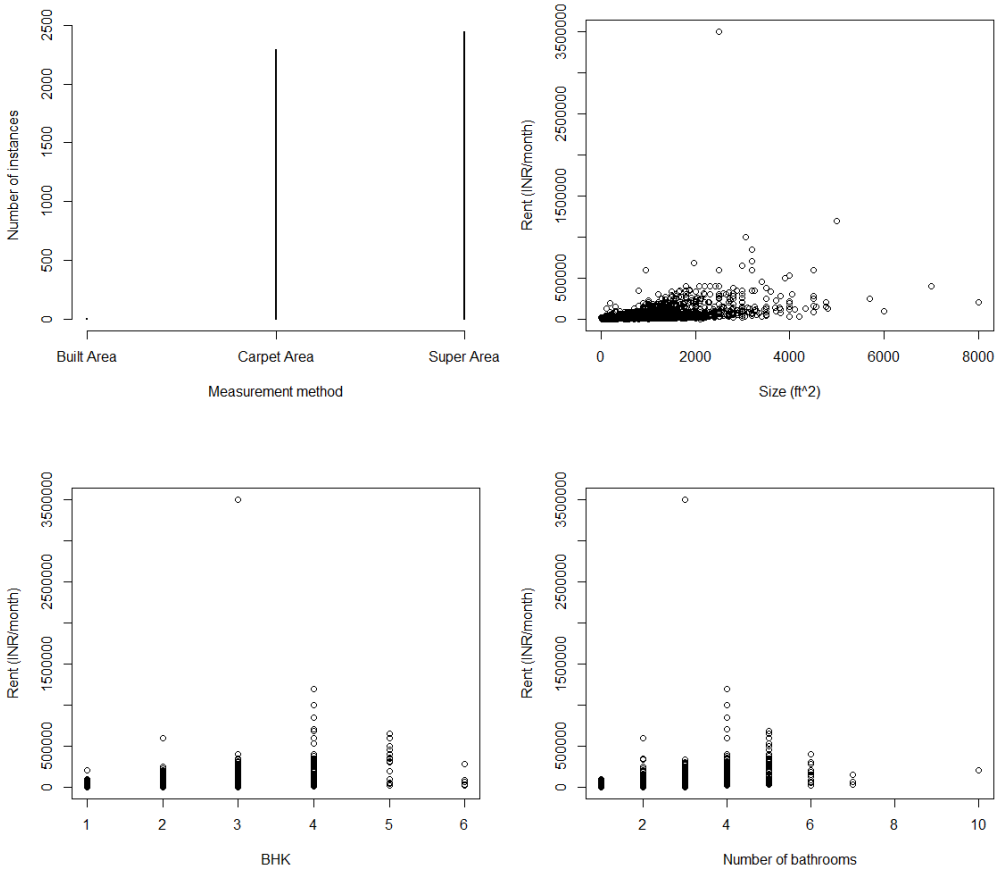

Figure 1.2: Plots of further features

In a similar way, the top-left scatterplot in figure 1.2 shows that there are very few instances where the house's size was measured according to "built area". Once again, probably it is better to remove those instances as they are not representative. The top-right plot in figure 1.2 show that there is a small positive correlation between the size of the house and its renting price. Overall, these plots also show that there may be some outliers.

The floor feature comes in the form of "X out of Y", where X is the floor number the house is located at and Y is the total number of floors. X can be an integer number or "ground", "lower basement" or "upper basement". The Spearman's Rho test shows that there is a weak positive correlation (Rho = 0.0455, p-value = 0.0017) between this feature and the Rent feature.

Though, the number of floors and the number of total floors may still have an impact on the dependent variable, thus, the floor feature was split in two different features: one expressing the floor on which the house is located and the other expressing the number of total floors. During this process, entries such as "upper basement" and "lower basement" were considered equal and encoded to -1, while "ground" was encoded to 0. One of the instances was removed during the process because of its invalid value.

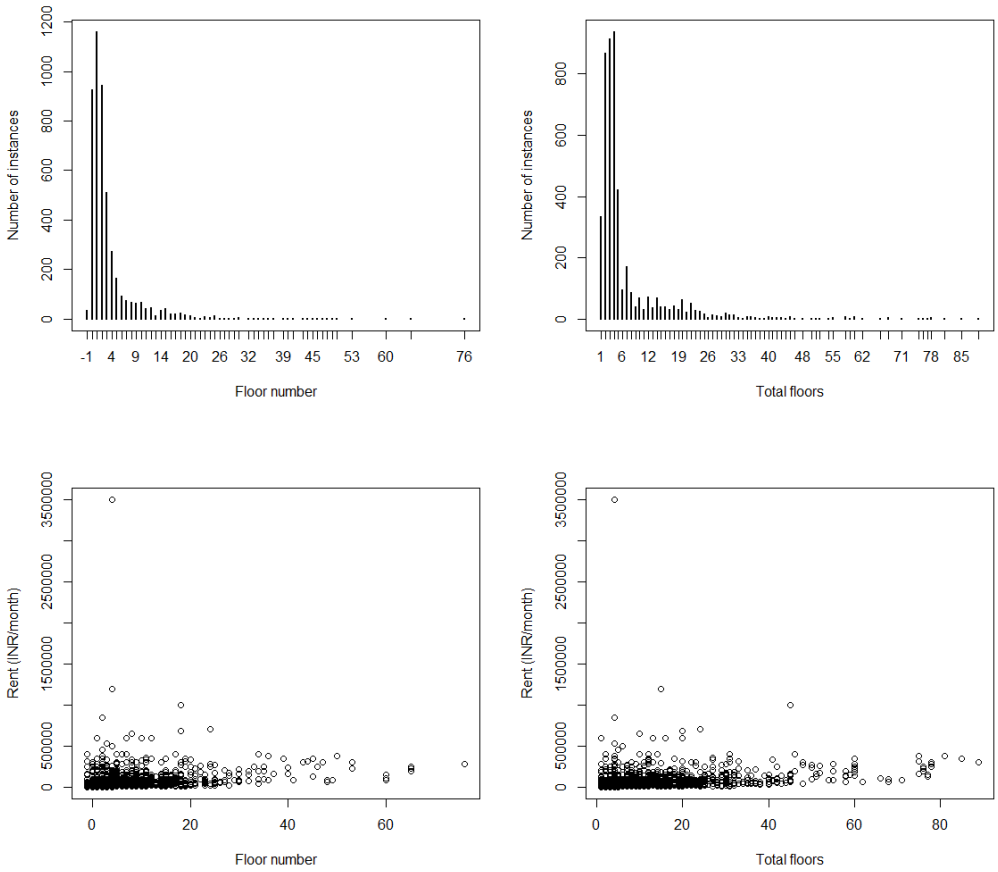

Figure 1.3: Plots of the new floor features

The plots shown in figure 1.3 show that there is a small positive correlation with the new floor features.

Furthermore, the following features were removed:

- the date of the announcement was removed because of the poor positive correlation with the renting price
- the floor feature was removed because of the new features that were created
- the area of the city in which the house is located was removed because of the very high p-value (0.92) in the correlation test, which indicated a very poor statistical significance of the test itself
- the type of tenant preferred by the owner was removed because of the high p-value (0.41) in the correlation test

Subsequently, feature selection was carried out. In particular, the best features were chosen via a filter method that used *Sequential Backwards Selection*, aiming at maximising the R^2 figure of merit:

	Spearman's Rho	P value
BHK	0.568	<2.2e-16
Size	0.521	<2.2e-16
Area Type	-0.368	<2.2e-16
City	0.305	<2.2e-16
Furnishing.Status	-0.298	<2.2e-16
Bathroom	0.662	<2.2e-16
Point.of.Contact	-0.596	<2.2e-16
Floor	0.494	<2.2e-16
Total.Floors	0.581	<2.2e-16

Table 1.1: Correlation of features with the Rent feature

All the features have survived except for the total number of floors.

Furthermore, a process of outlier removal was carried out using a heuristic algorithm based on *Cook's distance*: a model was built at every iteration and the farthest datapoint (in terms of Cook's distance) was removed. The algorithm stopped when the farthest datapoint's distance was less than $\frac{4}{L}$, with L being the number of datapoints, or when the total number of removed datapoints exceeded 10% of the total datapoints. 475 datapoints were removed. Finally, the dataset without outliers was split in a train set and a test set with an 80-20 proportion respectively. The OLS model was trained

and tested on the split without outliers, while the LMS model was trained and tested on the full dataset (i.e. the dataset that included the outliers). The split was done with the same proportions (80% of the datapoints for the train set, 20% for the test set).

1.4 Results

The predictions on the test set resulted in the following metrics:

	RMSE	MAE	R^2
OLS	7955.742	5704.983	74.60%
LMS	37367.71	13947.14	38.57%

Table 1.2: Metrics of the models

These results show that the OLS model expresses a good amount of the variance of the (training) data, with low enough error metrics. On the other hand, the LMS model doesn't express enough variance and it has quite high error metrics. These results are not surprising considering that the OLS model was trained on a dataset without outliers.

Chapter 2

Classification

2.1 Introduction

This second and last part of the project aims at creating two classification models, one based on Logistic Regression and the other one based on a Neural Network, in order to classify patients as diabetic or non diabetic. Furthermore, the models will be evaluated based on various performance metrics.

2.2 Theoretical concepts

Logistic Regression is a regression method that estimates the probability that a nominal variabile is true. It produces an S-shaped curve with the following formula:

$$\frac{e^{\omega x+\beta}}{1+e^{\omega x+\beta}}$$

The curve's steepness is directly proportional with the impact of a feature on the dependent variable, i.e. a very steep curve means that the independent variable has a great impact on the value of the dependent variable, and viceversa, a very flat curve means that the independent variable has a tiny impact on the value of the dependent variable.

A *Neural Network* is a set of artificial neurons organised as follows:

- an input layer with as many nodes as there are features to be considered
- a set of hidden layers that are fully connected to the input layer
- an output level with as many nodes as there are classes

Each artificial neuron calculates a weighted sum of its input x: $z = wx + b$, where w is a vector of weights and b is a bias. Subsequently, the result of an activation *function function* of the value z is passed to the neurons in the next layer. The network has to estimate the best weights and biases to fit the data: this is done through *backpropagation*, which is a process that iteratively updates those values by minimising the loss function via gradient descent.

2.3 Methods

The dataset has the following features:

- **Pregnancies**: number of pregnancies of the patient
- **Glucose**: glucose level in the patient's blood
- **BloodPressure**: blood pressure of the patient
- **SkinThickness**: thickness of the patient's skin
- **Insulin**: insulin level in the patient's blood
- **BMI**: Body Mass Index of the patient
- **DiabetesPedigreeFunction**: likelihood of diabetes based on family history
- **Age**: age of the patient
- **Outcome**: whether the patient has diabetes or not

One thing to note is that all the instances in this dataset represent female patients at least 21 years old, thus, the models trained on this dataset should only be used to predict whether a patient has diabetes or not only if the patient meets those characteristics.

There are some values of the Diabetes Pedigree Function that are greater than one, but this value is the result of a probability function, which is by definition a value between 0 and 1. Thus, those values were removed, resulting in a loss of 51 datapoints and the following statistics:

	Pregnancies	Glucose	Blood pressure	Skin thickness
Min	0	0	0	0
Max	17	198	122	99
Mean	3.848	119.908	68.714	20.099
Median	3	116	72	23
Standard dev	3.344	31.490	19.701	15.903
Variance	11.182	991.628	388.143	252.919

	Insulin	BMI	Diabetes Pedigree Function	Age
Min	0	0	0.078	21
Max	846	67.1	0.997	81
Mean	77.026	31.811	0.409	33.100
Median	22	32	0.344	29
Standard dev	112.080	7.778	0.222	11.812
Variance	12562.07	60.499	0.049	139.535

Table 2.1: Descriptive statistics

This dataset has a prevalence of 33.33% as there are 239 positives and 478 negatives. An *oversampling* method could be used to obtain a more balanced dataset, however, this unbalance may stem from the natural presence of the disease, that is, it could be reflecting diabetes' frequency in the population. Also, balancing the dataset would include an amount of bias, thus, the dataset was kept as unbalanced it was.

Feature selection was carried out with a filter method that used *Sequential Forward Selection* and aimed at minimising the *Gini Index*, which quantifies the probability of misclassification. The features which survived are the number of pregnancies, the glucose level and the result of the Diabetes Pedigree Function. Furthermore, the dataset was split in a training set and a test set with an 80-20 proportion respectively. The prevalence of the dataset was carried over in the splitted sets. Finally, the optimal threshold used to classify an instance as positive (i.e. unhealthy) with the logistic regression model was found using the *Receiving Operating Characteristic* (ROC) curve. The neural network's hyperparameters were optimised via cross validation in order to find the best tune in terms of number of nodes in the (only) hidden layer and in terms of decay, which is a regularisation hyperparameter used to prevent overfitting. The network trained until convergence was reached or until a predefined number of iterations was reached. Furthermore, the network used a *Sigmoid* activation function in each node.

2.4 Results

The models' performance were measured by comparing the actual health status of the instances in the test set and the predicted health status of the same instances, with the following results:

	Accuracy	Precision	Recall	True Negative Rate
Logistic Regression	78.87%	81.55%	88.42%	59.58%
Neural Network	80.28%	83.17%	88.42%	63.83%

Table 2.2: Evaluation metrics calculated on the test set

These metrics show that the neural network has overall better metrics than the logistic regression model, in particular it has slightly higher Accuracy, Precision and True Negative Rate, which means that it correctly identifies more negative (i.e. healthy) people, but also it is a (slightly) more believable model in general.

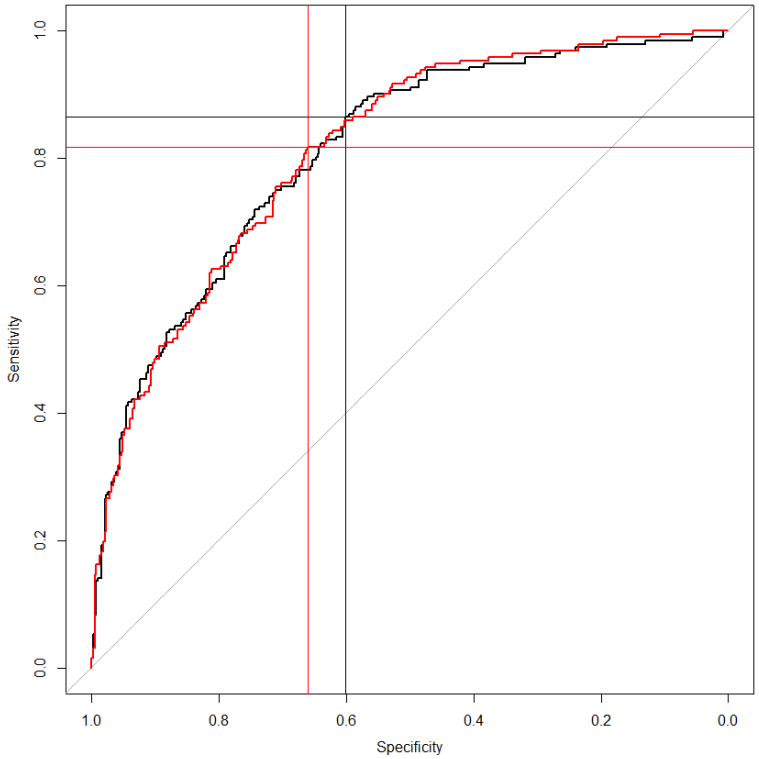

Figure 2.1: ROC curves of the models

Figure 2.1 shows the ROC curves for the logistic regression model (in black) and the neural network (in red), and the points marked in black and red are, respectively, the ideal logistic regression and neural network classifiers.