Assessing the agreement between common feature selection methods on feature importance for different regression tasks

Advanced Applied Statistics and Multivariate Calculus
KSAASMC1KU

János Máté
janma@itu.dk

Jorge del Pozo Lérida
jord@itu.dk

Mateusz Lemański
mmle@itu.dk

Nicolás Obregon
niob@itu.dk

***Abstract*—Feature selection methods are routinely used in Machine Learning to reduce computational complexity and improve interpretability by selecting a set of features based on their order of importance. However, the assumption that different methods yield similar results when ranking the features is often made and selection of a certain method is normally based on specific requirements of the modeling task. This report tries to elucidate this assumption by testing if five different models commonly used for feature selection agree on the order of importance of the features for 5 different datasets. We have selected 5 different models — Linear and Ridge Regression, Elastic Net, CART, Random Forests — and compared their feature importance rank lists using Kendall's W as well as performed a hypothesis test. Based on the study, we conclude that there is sufficient evidence to say that there is some agreement between the methods. We then argue that the level of agreement is quite strong, specially for the first few most important features, based on the value of the top-k overlap and W.**

***Index Terms*—feature selection, feature ranking, feature importance, rank correlation, linear regression, elastic net, CART, random forest, kendall W, hypothesis testing**

I. Introduction

In the last decade, Machine Learning (ML) has become an extremely interesting subject, garnering interest worldwide among thousands of organizations and companies. As interesting as it's capabilities may be, exactly how much of it works under the hood remains a mystery to most people, and most individuals focus on getting results rather than why something works. As future data scientists, our team is well aware that there is much more to machine learning than implementing a function from the scikit-learn library and calling it a day. On the contrary, data scientists require deep understanding of the statistical and algorithmic methods that are occurring behind the scenes in the methods they are implementing.

One of the cardinal steps in ML is feature selection, which has become very relevant in the past years, since the domain of features in ML has expanded from a just a few to hundreds of features. It can be defined as the process of selecting a subset of input variables or features which can efficiently describe the data while reducing effects from noise or irrelevant variables and still provide good prediction results, improving interpretability, generalization and computation efficiency. A very related concept is feature ranking, which is the attempt to see what is the order of importance of the features from a dataset in relation to predicting a certain value.

There are several techniques or methods for feature selection, and the choice between them will depend on the specific requirements of your modeling task or your final goal. However, to what extend do these different methods yield similar results? Do they agree on what features to select? To answer this question, this project aims to see if different feature selection methods agree on what features are more important for a given dataset and thus give somewhat similar results as to the feature ranks. This is some actually some assumption that data scientists always make that we will try to prove statistically.

When trying to perform a regression task in ML for a given dataset, we believe that some features describe the target variable better than others, as if due to intrinsic characteristics of the dataset there was a natural order of importance of its features. If this was the case, we would expect that common feature selection methods will learn this phenomenon and therefore will assign similar rankings of importance (even if using different approaches) to features when predicting a target variable. Assuming this, we ask ourselves the following research question: *Do different feature selection methods agree on the ranking of importance of our features?*. To investigate this, we will test the null hypothesis that *there is no agreement between different feature selection methods when assigning rankings to features* and try to claim the alternative hypothesis that *different feature selection methods agree when assigning rankings to features*. This way, we will first assume that they all assign different order of importance to the features and completely disagree to later prove that they actually agree. Later on, we also want to further argue how strong this agreement is, if any.

II. Datasets

We have first performed all analyses on one dataset, *USA_houseprices_2014* [1], which we will also call "main dataset" or *USA_houseprices* from now on. The similiar analysis and prepossessing was then repeated on four more datasets. Sections II-A to II-B describe in more detail the process.

A. Features description

The initial dataset originates from Kaggle and is named *USA_houseprices*, containing 17 features, and a target value: *price*. There were numerical and categorical values in the data set. The dataset was modified to have more meaningful features, aiming to achieve better performance for the feature ranking by adding or removing features. Feature descriptions can be seen in Table I. It must be noted from the table that *sqft_living* includes both house and garden, *floors* half value refers to the loft, attic space and that *year_since_1st_renovation* was created by ourselves based on other features: construction year, renovation year.

feature	description
price	House's selling price in $
bedrooms	Number of bedrooms
bathrooms	It contains the number of bathrooms
sqft_living	Size of the house in square feet
sqft_lot	Size of the property in square feet
floors	Number of floors
waterfront	The house is next to the water (yes/no)
view	How good the house view is (scale: 0-4)
condition	How good the house condition is (scale: 0-1)
sqft_basement	The size of the basement in square feet
yr_since_1st_renovation	Years since the last modification
city	Which city the house is
statezip	The zip code of the state where the house is
have_basement	The house has a basement or not

TABLE I
FEATURE DESCRIPTIONS FOR MAIN DATASET

B. Data Exploration and Pre-processing

During data pre-processing, we cover various steps to ensure better results for the feature ranking methods. This was done for all datasets. We applied the following techniques: feature scaling, distribution checking, checking the normality of the target value, encoding, and outlier detection.

1) Null Values: NaN/null value detection in data is a significant task as it can generate bias in our models. When the Nan value supposed to be 0 , than we replaced it with 0. In the other cases, we deleted the whole observation or replaced it with the feature mean value.

2) Feature Removal: Having data that can properly describe our target is a key component in statistics, we therefore decided to commit significant changes to some features. *yr_built* and *yr_renovated* were combined into one new feature. There were two features called *sqft_above* and *date*, whose information was not considered relevant to our target feature, as well as a third one, *street*, for which nearly every value was unique. a fourth variable, *country* stored the same value for every record. Therefore, these 4 variables were dropped.

3) Feature Creation: In some cases, It made sense to create further features based on the existing ones to get more meaningful features. With feature creation we could improve the performance of feature selection. In the main dataset we created a feature: *yr_since_1st_renovation* to create this feature we used the following features: *yr_renovation*, *yr_built*, *last_modification(max(yr_renovation, yr_built))*.

4) Encoding: As many feature ranking methods are only available with numerical values, we introduced encoding to convert the categorical values into numerical values. For our particular datasets, Label Encoding was used, which assigns a unique integer based on the feature's alphabetic order, which ranks the categorical values. Although One Hot Encoding usually achieves better performance as it will not rank categorical values, we opted not to use it as it would generate massive amounts of new features. This can lead our dataset having the curse of dimensionality [2]. As this research project concerns itself with feature ranking methods agreeing with each other rather than dimensionality reduction, we instead used label encoding for simplicity.

statezip	city
WA 98133	Shoreline
WA 98119	Seattle
WA 98042	Kent
WA 98008	Bellevue
WA 98052	Redmond

TABLE II
VALUES FOR CATEGORICAL FEATURES

5) Distributions, values range and scaling: Some of the feature selection techniques can be only properly used for feature ranking when the features are correctly scaled (e.g.: Linear regression therefore we checked the distribution and range of the features and modified them in the necessary cases. The difference between ranges of features would corrupt our analysis.

As we choose to remove outliers, we decided to use Min-max normalization for scaling the data.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

We run Shapiro-Wilk normality test [3] for all of the features to determine whether they follow normal distribution or not.

6) Outliers detection: Outliers can distort the outcome of feature selection, so the detection and reduction of the outliers had significant importance. In our project, we mostly used Interquartile Range to detect outliers (IQR) with a threshold of 1.5, which is one of the most widely used techniques. IQR is the difference between the first and the third quartile, and it calculates thresholds in the following way [4]:

$$Tmin = Q1 - c \times IQR \tag{2}$$

$$Tmax = Q3 + c \times IQR \tag{3}$$

$$IQR = Q3 - Q1 \tag{4}$$

Q1: first quartile; Q3: second quartile; c: threshold value

In the case of the main dataset we handled outliers by checking the distribution and range of the *price* target variable. Based on that, we removed outlying values.

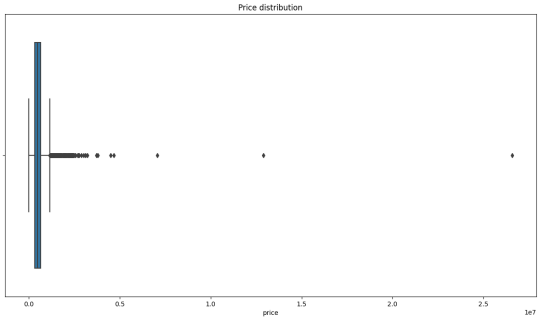

Fig. 1. *Price* distribution with outliers

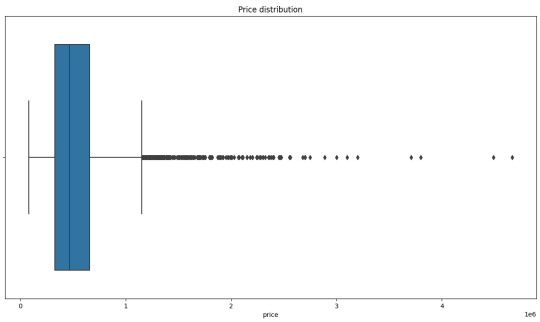

Fig. 2. *Price* distribution without outliers

C. Extra datasets

Four other datasets were used, with the condition that they all be datasets where a regression task can be done and that they contain a sufficient number of features. To see their original source see B.

1) Austin House Prices: This dataset is similar to the base one, holding 45 features total, with one being the price of the house. This was once again chosen as the target variable, therefore the feature ranking methods will aim to show which features are the most important to predict the price of a house [5].

2) Airbnb Price: The source data includes 74411 house listings and 29 columns - including *log_price*, which is the target we want to predict - that contain characteristics of such houses. There was no information about this data so we assume that since all the listings are in the US and that target variable is the price for one night. After pre-processing we end up with 18 features [6].

3) Bike sharing: Dataset contains the hourly rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information. Bike-sharing rental process is highly correlated to the environmental and seasonal settings. For instance, weather conditions, precipitation, day of week, season, hour of the day, etc. can affect the rental behaviors. Preprocessing shrinked initial features number to 12. [7].

4) Cars: This dataset is also similar to the initial one as it has different features regarding the car and the target variable is the price of that car (MRSP). Therefore, the feature ranking methods show the importance to predict car prices (MRSP). The original dataset had 56 features but a lot of them were similar or the same so we removed these features and in the end, we worked with 35 features [8].

III. Methods

We have selected five different feature selection methods due to their relevance and presence in literature, starting with the base case of linear regressions to more sophisticated ones like ensembled models such as Random Forests. We have also decided on several evaluation metrics that provide us with a framework for hypothesis testing and further interpretation. Finally, we have also introduced several techniques for noise reduction. All implementation has been done in the Python programming language [9] and can be found in the following Github repository: https://github.com/jorgedelpozolerida/KDS_Statistics_GroupProject.git

A. Feature selection methods

1) Linear Regression family: Because we wanted to include some base case that could be interpreted easily, we decided to include Multiple Linear Regression [10] as the first feature selection method. The simplicity of linear models along with the fact that they often provide an adequate and interpretable description of how the inputs affect the output [11], which makes them very suitable for this project.

After fitting the processed dataset to the model, we will use the coefficients of regression in absolute value, to rank our features according to their value or "importance". The underlying idea is that if all features are on the same scale, uncorrelated features with the target should have very low values for their coefficients and the most important features very high ones, thus giving us an importance criteria for ranking them.

However, it is known that when there are linearly correlated features present in the dataset, this approach becomes less suited since it becomes unstable and coefficients vary a lot with small changes in data, making interpretation harder and thus probably affecting our final ranks assigned. For this reason, we also used Ridge Regression, which is a regularized version that imposes a penalty on the l_2-norm in the loss function used for obtaining the regression coefficients β:

$$\beta_{ridge} = argmin \left(\sum_{i=1}^{n} (y_i - p_i)^2 + \lambda \|\beta\|_2^2 \right) \quad (5)$$

where p_i is the predicted value, y_i the real value and λ is a Lagrange multiplier. It must be noted that Lasso Regression was not considered due to its inherent tendency to yield 0 coefficients or assigning similar coefficients to correlated variables, which would make ranking very hard. On the other hand, in Ridge Regression a predictive or important feature will get a non-zero coefficient which is ideal for ranking our features and allows for better interpretation.

2) Elastic Net: Elastic Net is another linear model. We chose it as it uses the penalties from both Lasso and Ridge, aiming to combine the strengths of these two methods. Linear regression methods can become overtly sensitive and unstable when we have data that has the *big-p, little-n* problem, where there are more predictors than samples.

The already mentioned Lasso's, or l_1 regularization can be limited if we have high-dimensional data, or in some cases it will select only one variable from the dataset if it finds it to be highly correlated with another one. Ridge, or l_2 on the other hand, as mentioned above minimizes the size of all coefficients.

Elastic Net applies both these penalties when used, and provides us with a parameter α which is a value between 0 and 1, denoting the weight of each penalization that will take place, which we can change. A α of 0 means all weight will be given to the l_2, an α of 1 on the other hand will give it to the l_1 penalty. Because Elastic Net represents one of the most complex linear methods we decided to include it in the analysis.

3) Classification And Regression Tree (CART): CART is a statistical technique, which was introduced by Breiman et al. [12] in 1984 and it can be used for classification and regression problems as well. It works by constructing a tree-like model of decisions, with each internal node representing a decision about the input data and each leaf node representing a prediction. For our purpose we used *DecisionTreeRegressor* model from **sklearn** package [13], where the default way to produce feature scores is calculating the impurity based on variance. To achieve that, algorithm minimize total sum of squares between the observation and the mean of each node:

$$impurity = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{6}$$

where y_i is the i^{th} item in the set, $\bar{y}$ is The mean of all items in the set and (y_i - $\bar{y}$) is the deviation of each item from the mean [14].

CART is widely used for feature ranking because it calculates the feature importance score, which shows the relative importance of each feature. The feature importance scores are ranked by their values; thus, we can select the best-performing features and remove the useless features, which is very useful for our project. Moreover, CART has plenty of advantages as a feature selection technique:

- Follows a non-parametric method: It works with every distribution, and the result does not depend on the probability distributions.
- As was mentioned CART can be used for feature selection as well. However, compared to other feature selection techniques, it is suitable for both categorical and numerical data as well.
- It can be used when the features have a non-linear relationship.
- It can cope with missing values and outliers

Overall, CART can be applied easier than other feature selection methods because it requires much less feature normalization and scaling. Furthermore, CART provides a clear ranking of the feature importance, which can be useful for understanding the relationships between the features and the target variable. On top of that, is the one of the earliest as well as the most straightforward decision tree algorithm, so it has high relevance to show its performance compare to other methods.

4) Random Forest: A random forest is an ensemble learning method that is used for classification, regression, and other tasks. It is a type of decision tree algorithm that is based on the idea of CART, but it is creating a forest of decision trees, which are trained on random subset of data, and then averaging the predictions of each tree to make a final prediction. For calculating features scores, we also used variance as impurity metric.

The main advantage of Random Forest over CART is that it can handle large and complex datasets and is resistant to over-fitting, so in result can improve the generalization of the model. It is also relatively easy to implement and can be trained in parallel, which makes it a popular choice for many machine learning tasks. Random Forest has higher predictive power and accuracy than a single CART model, but on the other hand computation is more consuming and less clear to interpret.

B. Calculation of Rankings

For each of the methods the same procedure for obtaining a list of rankings has been followed. Once you have the model fitted and certain weights or scores obtained for each feature, we convert them to absolute value. Then, min-max scaling is performed and ranks are assigned in descending order of value. In case of a tie occurring in the ranking list between some number of features, a mean of the different rankings associated is given to each of them. For instance, if we have the set $\{40, 50, 40, 40, 70\}$, the rank 4 ($\frac{3+4+5}{3}$) would be given to the three elements having 40.

C. Evaluation Metrics

To evaluate the degree of similarity or difference between the rankings assigned to the features by each of our methods, we measure how much they agree with each other with certain metrics. Because there are no good libraries for computing these metrics in Python, all calculations have been implemented from scratch.

1) Kendall's W: Kendall's coefficient of concordance (W) is a non-parametric statistic that can be used to measure extent of agreement among multiple rank lists given by different raters. Suppose that object i is given the rank $r_{i,j}$ by rater number j, where there are in total n objects and m raters. Then Kendall's W is defined as:

$$W = \frac{12 \sum_{i=1}^{n} (R_i - \overline{R})^2}{m^2 (n^3 - n)} \tag{7}$$

where R_i is total rank given to feature i, $\overline{R}$ is the mean value of these total ranks. In our case, the m raters are the

feature selection methods and the n objects are the features of our dataset ranked by the previous. Its values range from 0, indicating a perfect disagreement and that responses are completely random; to 1, which is the perfect agreement between raters, meaning they have assigned the same order to the list of features. We will use the same interpretation for W as in [15]:

$$\text{Interpretation} = \begin{cases} weak,\ if\ W \leq 0.3 \\ moderate,\ if\ 0.3 < W \leq 0.6 \\ strong,\ if\ W > 0.6 \end{cases}$$

In case of ties in the rank lists, their effect is to reduce the value of W; however, this effect is small unless there are a large number of ties [16]. This can optionally be accounted for subtracting m times the following correction factor:

$$T = \sum_{j=1}^{m} \sum_{i=1}^{g} (t_i^3 - t_i) \tag{8}$$

where t_i is the number of tied ranks in each i group of g groups of ties found in rank list from method (rater) j. The sum is computed over all m methods to get T.

Friedman's test statistic can be obtained from W using the formula:

$$Q = m(n-1)W \tag{9}$$

If H_0 is true, this quantity converges in distribution to the χ^2 distribution with (n-1) degrees of freedom when $n \geq$5 or $m >$15 [17]. Thus, this allows us to test W for significance with our null hypothesis saying that there is no agreement among the methods or raters ($H_0 : W = 0$) and our alternative hypothesis being that there is some agreement among the raters ($H_1 : W \neq 0$). For a certain significance level α, the null hypothesis H_0 can be rejected if:

$$Q \geq \chi^2_{\alpha,n-1}$$

where $\chi^2_{\alpha,n-1}$ is the tabled value obtained from the chi-square distribution, with n-1 degree of freedom. The p-value is P($\chi^2_{n-1} >$ q) where q is the observed value of the test statistic Q.

2) Top–k overlap: This metric taken from [18] computes the amount of overlap that exists between features at the top-k ranks in relation to the total number of features at the top-k ranks across n feature rank lists. The presence or absence of a given feature in all of the top-k is considered, regardless of its order. It is defined as follows:

$$Top - k\ overlap = \frac{\cap_{i\geq 2}^{n} Features\ at\ top\ k\ ranks\ of\ list_i}{\cup_{i\geq 2}^{n} Features\ at\ top\ k\ ranks\ of\ list_i} \tag{10}$$

D. Noise and Randomness

As any model that deals with massive amounts of data, there is always the possibility of noise and randomness being present in the data. These two terms generally refer to the presence of random variation being present in the data. Machine learning models can benefit from having noise and randomness, as it can be useful in the prevention of overfitting and also improve the generalization of the model [19]. But they can also be detrimental to models, making it hard for patterns to be recognized, as well as giving different results per run. This could heavily influence our final rankings in detriment of agreement among raters. A subset of models will be done on the main dataset where we set a random seed and do cross validation to analyze how our models differ when these methods are not applied.

1) Random Seed: We can set a random seed in the methods used. The models use a random number generator, and over different iterations of a model it can give different results because of this random number. We set it on the subset of models to be equal to 42 across all iterations.

2) Cross Validation: Cross validation involves evaluating models on different subsets of the data. Cross validation splits a dataset into a training set and a test set, where the model is trained in the training set and the performance is then evaluated in the test set, in a way validating the analysis. This process occurs a number of times, and the performance of each model is averaged across all the iterations. Rather than testing the model once, it is done multiple times, increasing its robustness. The results are then averaged to provide an estimate of the models performance. We will do 5 cross validations across our methods. We will be averaging the results in the Elastic Net and Ridge methods, and in Random Forest and CART we will instead look at the results per cross validation to see if there is a big standard deviation among the feature weights that the model has ranked. Depending on the results, we will be able to see just how much the ranking scores deviate per run. Finally, the methods will be run with different parameters set to find the ideal one.

IV. Results

A. Linear Regression

For each dataset, three different models have been fitted using LinearRegression, Ridge, Lasso classes from **sklearn** package [13]. For each, the coefficients or weights have been transformed as explained in III-B. Because Lasso regression shrinks coefficients to 0 it has been ignored. The coefficients obtained for the base dataset can be seen in Figure 3.

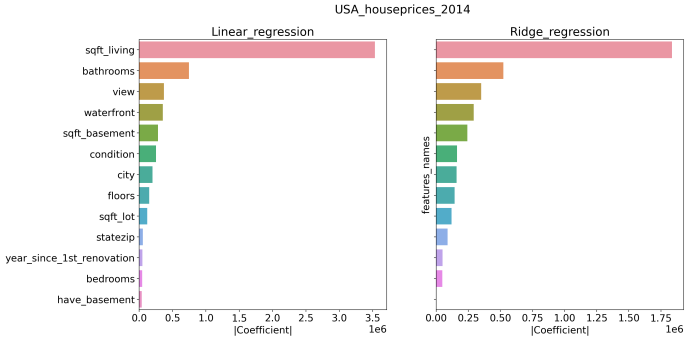

Fig. 3. Coefficients obtained for main dataset using linear models, in absolute value

B. Elastic Net

Each dataset had Elastic Net applied to it, using functions from the **sklearn** package. The coefficients obtained for the main dataset can be seen in Figure 4.

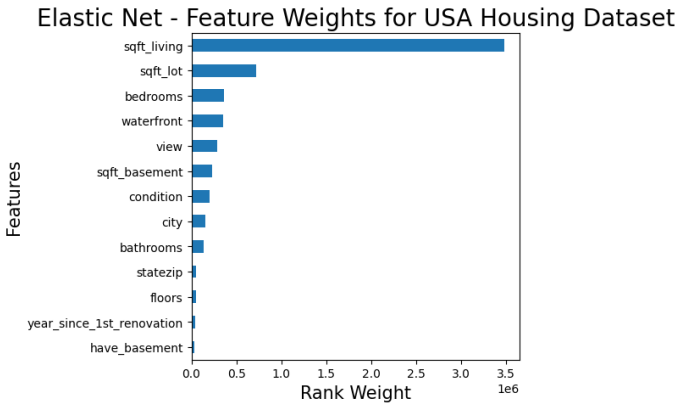

Fig. 4. Weights obtained for main dataset using Elastic Net

C. CART

CART regression decision tree was implemented for the different datasets, using *DecisionTreeRegressor* from the **sklearn** package. The feature importance score for each feature in CART regression is the impurity as was explained in section III-A3. We got these feature scores when we built the decision tree. Moreover, all the feature importance scores have been transformed in the following way III-B. We should interpret the importance score as the larger values are more important (1 is the biggest) than the smaller ones, 0 means there is no relationship between the feature and the target value. The visualized result of the regression decision tree is shown in the following Figure. 5.

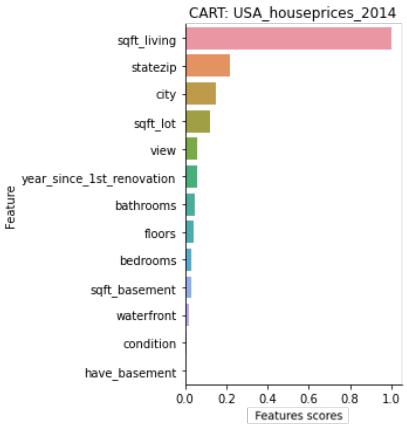

Fig. 5. Features scores obtained for the base dataset using CART regression

D. Random Forest

The features scores for the main dataset can be seen in Figure 6.

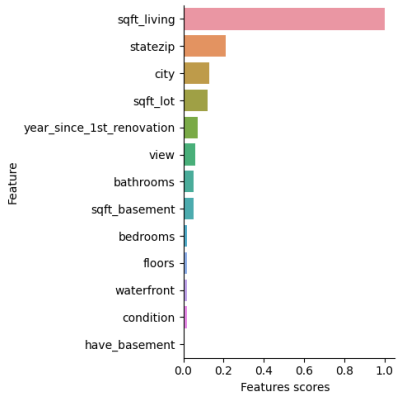

Fig. 6. Features scores obtained for base dataset using Random Forest

E. Feature rankings

The ranking of each feature of the main dataset can be seen, with each of its respective methods in table III. Numbers with .5 mean a tie took place.

Feature Name	CART	Linear	RF	Ridge	Elastic Net
sqft_living	1.0	1.0	1.0	1.0	1.0
statezip	2.0	11.5	2.0	10.0	10.0
city	3.0	8.0	3.0	7.5	8.0
sqft_lot	4.0	2.0	4.0	9.0	2.0
view	5.5	5.0	6.0	3.0	5.0
year_since_1st_renovation	5.5	11.5	5.0	11.0	12.0
bathrooms	7.5	9.0	8.0	2.0	9.0
floors	7.5	11.5	11.0	7.5	11.0
sqft_basement	9.0	6.0	7.0	5.0	6.0
bedrooms	10.5	3.0	10.0	12.0	3.0
waterfront	10.5	4.0	12.0	4.0	4.0
condition	12.0	7.0	9.0	6.0	7.0
have_basement	13.0	11.5	13.0	13.0	13.0

TABLE III
RANKINGS FOR ALL DIFFERENT METHODS

F. Evaluation Metrics

The different values for Kendall's W can be seen in table IV for all datasets, as well as their interpretation following equation III-C1. Both forms for the coefficient - W and W', corrected for ties - have been calculated. Three decimals have been included to notice the slight but present differences between the two calculations. Note that m is always 5, which are the total number of methods or raters being compared, but n varies depending on the number features of the dataset.

dataset	n	m	W	strength	W'	strength
USA_houseprices	13	5	0.508	moderate	0.508	moderate
Bike_sharing	12	5	0.708	strong	0.709	strong
Airbnb_price	18	5	0.545	moderate	0.546	moderate
austin_housing	27	5	0.644	strong	0.645	strong
cars	34	5	0.665	strong	0.666	strong

TABLE IV
KENDALL'S W FOR ALL DATASETS

In Figure 7 one can see the calculated Top-k overlap calculated for all datasets. For the sake of interpretation note that they all reach 1 when k is equal to the total number

of features for that dataset (for that reason they reach it at different values of k), which represents the point when the whole rank list is selected for comparison. The specific values for the first 5 overlaps are presented in table V.

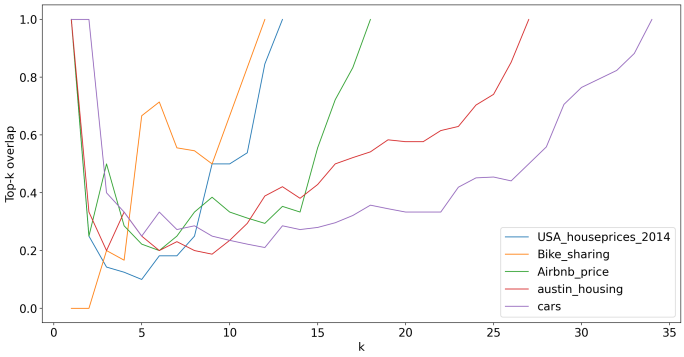

Fig. 7. Top-k overlap for all datasets

dataset_name	Top-1	Top-2	Top-3	Top-4	Top-5
USA_houseprices	1.0	0.25	0.14	0.12	0.10
Bike_sharing	0.0	0.00	0.20	0.17	0.67
Airbnb_price	1.0	0.25	0.50	0.29	0.22
austin_housing	1.0	0.33	0.20	0.33	0.25
cars	1.0	1.00	0.40	0.33	0.25

TABLE V
TOP-K OVERLAP FOR K FROM 1 TO 5 FOR ALL DATASETS

G. Hypothesis Testing

Because we observed that it was quite common for the methods to have ties for different features, we used W' for calculating the test statistic. The different values obtained for the test statistic Q as well as the critical chi-square value $\chi^2_{\alpha,k}$ can be seen in table IV for all datasets. Also, the final decision on the H_0 of the test can be seen. It must be noted that a significance level of $\alpha = 0.05$ was chosen for all tests. Finally, p_{value} obtained is also described.

dataset	df	Q	$\chi^2_{\alpha,df}$	$reject\ H_0$	p_{value}
USA_houseprices	12	30.48	5.23	True	.0024
Bike_sharing	11	38.99	4.57	True	.0001
Airbnb_price	17	46.41	8.67	True	.0001
austin_housing	26	83.85	15.38	True	$< .0001$
cars	33	109.89	20.87	True	$< .0001$

TABLE VI
HYPOTHESIS TESTING

H. Noise reduction

Elastic net was run with different l1 ratios, identifying if a better model was produced when using a heavier l_1 penalization or heavier l_2, 100 alphas will be ran. Ridge Regression was run with multiple different alpha values, seeing which regularization strength was best. CART used the entire depth of the tree and all features in every cross validation. Random Forest was run with 100 trees and with no max depth.

The results with cross validation and a set seed yielded interesting results, and proves that setting a seed and doing cross validation is always a viable and recommended. The results can be seen in table VII.The ranks from CART and Random Forest were identical with each other, with only a couple of ranks deviating. Ridge and Elastic on the other hand had exactly the same ranks. The only difference between the cross-validated/set seed models and the original models was that *year_since_1st_renovation* and *view* changed order, as did *floors* and *bathroom*.

Feature Name	CART	Ridge	RF	Elastic
sqft_living	1	1	1	1
statezip	2	10	2	10
city	3	8	3	8
sqft_lot	4	2	4	2
year_since_renovation	5	12	5	12
view	6	5	6	5
floors	7	11	11	11
bathrooms	8	9	7	9
sqft_basement	9	6	8	6
bedrooms	10	3	10	3
waterfront	11	4	12	4
condition	12	7	9	7
have_basement	13	13	13	13

TABLE VII
RANKS WITH SEED AND CROSS VALIDATION

V. DISCUSSION

A. Linear Regression

From figure 3 it can be seen that the two linear methods agree a lot on the feature importance. It must be noted that their high value is due to target variable *price* having such scale, which is reflected in the coefficient even when the independent variables or features have all the same scale [0,1]. Also, there seems to clearly be a dominant feature, *sqtf_living* that is most heavily correlated to response variable and thus most important. For the rest of features there seems to be order of importance, so ranks derived from this should not present any ties. However, values do not vary much meaning there is no evident order of importance after first feature.

B. Elastic Net

The Elastic Net results can be interpreted in much the same way as those from the Linear Regression section. The high value is due to the target variable having a wide scale, and the dominant feature is as well *sqft_living*. The argument could be made to drop the features *have_basement*, *year_since_1st_renovation*, *floors*, and *state_zip* as they have an extremely low weight. If computational power or speed was essential in this project, dropping these 4 variables could be recommended. The remainder of the variables hold some importance, though not as strong as *sqft_living*.

C. CART

In figure 5 it is clearly visible that the *sqtf_living* is highly the most dominant feature, so it is by far the most important. However, after *sqtf_living* the difference between the feature importance is much more consistent and gradual, and only two values are not relevant statistically the *condition* and *have_basement*. In case, we dropped these features with low

importance scores than similar outcomes would be achievable with less computational power. Furthermore, we should notice that there are some features, which importance are really close to each other (e.g.: *bedrooms*, *sqft_basement*). Moreover, we should take into account that CART has a tendency to be biased toward features that have more unique values (e.g. numerical features) and the first 3 features have relatively many unique values compared to others, which might bias the result. The performance of the decision tree can be improved by using pruning to avoid overfitting. However, it would have made more difficult the interpretation so it was out of the scope of the project.

D. Random Forest

Random forest, similar to the rest of methods, computed sqft_living as the most important feature for predicting the price. Statezip is on the second place and is slightly more dominant compare to features lower in ranking. Compared to CART, random forest more radically decreased importance among the rest of the features. These can be a result of combining several decision tree and averaging feature score output, which is helpful in disregarding more irrelevant features.

E. Comparison between methods

Linear, Ridge, and Elastic Net Regression are all closely related to each other, with each version adapting methods from the one before. Linear regression does not penalize for the weights, Ridge regression penalizes the model so the weights have smaller values and are more evenly distributed. Lastly, Elastic Net combines the penalties of Lasso and Ridge, and in the case used in this report, it has an l1_ratio parameter equal to 0.5, which means the penalization is perfectly balanced between l_1 and l_2. Due to this, the results of the rankings are similar. The rankings from Linear and Elastic Net regression are identical, while those from Ridge Regression differ slightly, particularly in the ranking of the feature *sqft_lot*, which ranked 2nd in Linear and Elastic Net but 9th in Ridge.

The similarity between CART and Random Forest is significant because both models depend on similar approaches. CART builds a single decision tree, whereas Random Forest combines several decision trees and gets the average of them.

The CART is simpler, interpretable, and implementable and needs less computational power than Random Forest. However, Random Forest usually has better performance. Since this dataset's outcome is really similar for the CART and Random Forest, the CART's implementation would make more sense because of the following reasons.

Both Random Forest and CART try to select features that contribute most to model performance. On the other hand, both the Linear family of regressions and Elastic Net rely on the coefficients which in the end is a measure of correlation between dependent and independent variables. For this reason, it could be argued to some extent that we have only selected two different methods in terms of approach used. This can make our analysis less robust, so future work on this could be made. However we believe there are enough differences between them to be considered separately.

F. Feature rankings

All methods aggreed on the most important feature, which is *sqft_living*. Most of them pointed feature *have_basement* as the least imporant feature. CART and Random Forest had strong similarities in their rankings, and Linear, Ridge and Elastic Net on the other hand shared similar features as well among them.

Some of the models had weights that fluctuated very little. An argument can be made here that they could be of roughly equal importance. For Random Forest for example, the features *view*, *sqft_basement*, *bathrooms*, have rank weight of 0.06, 0.05 and 0.06 respectively. *bedrooms*, *floors*, *waterfront* and *condition* all had rank weights of 0.02 approximated.

CART shows a similar event happening, with *view*, *year_since_1st_renovation* sharing very close weights.

Elastic Net and other linear regression methods had the same occurrence as well. This can be visualized in Figure 4 and Figure 3.

G. Significant test

Because the p-value obtained was lower than $\alpha = 0.05$ for all dataset, the test is significant in all cases and we have enough evidence to reject the null hypothesis $H_0 : W = 0$. For all cases we have very strong evidence ($< .01$) against H_0 (except for the case of *Airbnb*, where we only have strong evidence) and this tells us that the likelihood that the results supporting the null hypothesis are not due to chance with a confidence level of 95%.

These results were expected, since accepting the null hypothesis would mean that either methods completely disagree, or that they produce random rankings, both cases meaning that these methods do not make good models for predicting the target variable. Still, is is worth double checking our everyday assumptions when building our models.

After having proven that different methods do not rank features randomly and that agree on the importance of them to some extent, it is necessary to quantify this degree of agreement at least based on our observed data to draw further conclusion. We have done this in section V-H

H. Evaluation metrics

It must be noted that W value is only slightly increased after correction for ties. This is due to the small amount of ties present in our final rank lists. Because W is a measure of agreement between the raters, we can actually draw some conclusions already. We can see that there is an overall reasonable agreement between all methods, since all values lay in the threshold between moderate and strong interpretation according to III-C1, so the overall level of agreement is moderate-strong, which actually most of the cases strongly agreeing. From these results, we interpret that all methods are retrieving the inherent characteristics of the dataset and assign to a great extent the different importances of each features to predict the target variable.

Regarding Top-k overlap value as k increases, we observe a common pattern for all datasets except for *Bike_sharing*, which we will analyze separately. They start with a total top-1 overlap of 1, quickly decreasing for the next values of k, although still quite high for $k = 2$ and $k = 3$, observing vlaues such as 1 and 0.4 for *cars* dataset and 0.25 and 0.5 for *Airnb_price*. We think that this is due to the fact that datasets normally have few features that best explain the target variable, while having many others that just refine this prediction. Thus, because we believe different models agree on feature importance, they specially agree on this small subset as can be reflected e in the top-k overlap for small values of k (the exact number depending on each dataset). For larger values of k that approach n the analysis is not very relevant, since it simply shows the natural result of increasingly encompassing more and more features selected in the lists compared thus increasing probability of any overlap being present. In any case, there seems to be a good agreement in the top features.

Dataset *Bike_sharing* seems to be an exception to this behaviour, which we believe can be due to three reasons: a) dataset being noisy b) absence of these "most important features" c) a lot of "most important features" being present. We incline towards option c), since we observe an top-6 overlap of almost 0.8, only including half of the features, which seem to be this group we talked about. Also, if dataset was noisy, W would be lower.

Overall, the values obtained for these metrics seem to suggest that all methods agree to a great extent so as to the order of importance of the feature for each dataset. We believe that this is due to them being able to extract the inherent characteristics of the dataset, that is, a natural order of importance of the features given the data.

I. Noise reduction

Doing Cross Validation and setting a random seed was informative to understand the importance of having robust models. Furthermore, although there are differences outputted throughout the models, there is an agreement between the most important feature, *sqft_living*, as well as some other minor ones. For CART and Random Forest, we outputted the weights of each feature per cross validation to perform statistical analysis on them. However, there was extremely low fluctuation between them, with the highest standard deviation being 0.0134 for Random Forest, and the mean being 0.5544 on the same feature. Since the results had such little deviation, we can assume that there is a very low chance of randomness negatively affecting these models. Had the results per cross validation fluctuated highly instead, the application of a statistical test would have been recommended to see if the randomness or other factor is making the models work incorrectly, and a different conclusion could have been made.

J. Future work proposal

In our paper, we examined the different feature ranking method's outcomes compared to each other. However, there are further directions where additional studies would be relevant and interesting to do. In our point of view, the following approaches would be an appropriate addition to our project.

Firstly, more state-of-art or "black box" feature ranking methods (e.g.: Boruta) can be examined in the same way. It would be useful since these techniques are widely used currently and these methods usually have better performance than the explained ones. Moreover, adding feature selection methods from other natures (e.g.: Wrapper Methods) would achieve different outcomes. With this upgrade, we would get a more complete result for feature selection for regression.

On the other hand, we can analyze feature selection methods for classification problems as well because our project was only tested for regression. With this modification, we can examine a completely different problem with only a small modification.

In the case of using more complex datasets with a hundred or more feature of features, this paper would achieve even cleared results. Since the number of features can significantly influence our hypothesis testing and evaluation metrics. Furthermore, using more datasets, not 5 can show other outcomes as well.

Finally, another research topic could be done into actually testing these methods on data. The removal of certain low-ranked features could be done on a dataset, and then a regression task could be trained and tested on the new dataset. We could then see how well the performance on the models is, and perhaps compare it to one where features were not removed, therefore seeing if the removal of the features actually increased the accuracy and f1-score of the models.

VI. Conclusion

We have experimented with multiple feature selection methods (Linear and Ridge Regression, Elastic Net, CART and Random Forest) in this paper and concluded that there was a general agreement between them with regard to the importance order assigned to features. First step to come up with this conclusion has been to reject the null hypothesis that there is no agreement at all between methods. Then, we have shown that the level of coincidence between them is actually moderate-strong, according to the interpretation of Kendall's W. Finally, we have observed that this agreement is specially strong when determining the first few most important features, when looking at top-5 overlap. Overall, these results seem to confirm that despite the differences in approach and nature of the methods when selecting features, they will not disagree much in the natural order feature importance, which is useful information for machine learning practitioners when trying to select a representative subset. Although we have obtained satisfactory results that provide good evidence to support our belief, we are well-aware of the limitations of this study. Therefore, we propose as a future line of work to repeat the analysis with more datasets, more metrics, more features and specially a more diverse and abundant set of feature selection methods to make results more robust.

References

[1] Shree, "House price prediction."

[2] N. Sharma, H. V. Bhandari, N. S. Yadav, and H. Shroff, "Optimization of ids using filter-based feature selection and machine learning algorithms," *Int. J. Innov. Technol. Explor. Eng*, vol. 10, no. 2, pp. 96–102, 2020.

[3] S. S. SHAPIRO and M. B. WILK, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, dec 1965. [Online]. Available: https://doi.org/10.1093/biomet/52.3-4.591

[4] J. Yang, S. Rahardja, and P. Fränti, "Outlier detection: how to threshold outlier scores?" in *Proceedings of the international conference on artificial intelligence, information processing and cloud computing*, 2019, pp. 1–6.

[5] E. Pierce, "Austin, tx house listings."

[6] S. Zheng, "Airbnb price prediction."

[7] Kaggle, "Bike sharing demand."

[8] Prassanth, "New cars price 2019."

[9] "Python 3.10, python software foundation. [online]. available," 2023.

[10] M. H. DeGroot and M. J. Schervish, "Linear statistical models," in *Probability and statistics*. Pearson Education, 2012.

[11] T. Hastie, R. Tibshirani, and J. Friedman, "Linear methods for regression," in *The elements of statistical learning*. Springer, 2009, pp. 43–99.

[12] L. Breiman, J. Freidman, R. Olshen, and C. Stone, "Classification and regression trees. wadsworth, monterey, ca." *Classification and regression trees. Wadsworth, Monterey, CA.*, 1984.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] F. Questier, R. Put, D. Coomans, B. Walczak, and Y. Vander Heyden, "The use of cart and multivariate regression trees for supervised and unsupervised feature selection," *Chemometrics and Intelligent Laboratory Systems*, vol. 76, no. 1, pp. 45–54, 2005.

[15] H. Akoglu, "User's guide to correlation coefficients," *Turkish journal of emergency medicine*, vol. 18, no. 3, pp. 91–93, 2018.

[16] Wikipedia contributors, "Kendall's w — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Kendall%27s_W&oldid=1118580390, 2022, [Online; accessed 3-January-2023].

[17] P. Legendre, "Species associations: the kendall coefficient of concordance revisited," *Journal of agricultural, biological, and environmental statistics*, vol. 10, no. 2, pp. 226–245, 2005.

[18] G. K. Rajbahadur, S. Wang, G. A. Oliva, Y. Kamei, and A. E. Hassan, "The impact of feature importance methods on the interpretation of defect classifiers," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2245–2261, 2021.

[19] S. C. Thundyill, Chhabria, "Impact of noise in dataset on machine learning algorithms," 2019.

Appendix A
Source Code

Te different scripts used for this report can be found in https://github.com/jorgedelpozolerida/KDS_Statistics_GroupProject.git

Appendix B
Data Source

The different datasets re available online and were downloaded from the following sources:

- USA_houseprices_2014: https://www.kaggle.com/datasets/shree1992/housedata
- Bike_sharing: https://www.kaggle.com/competitions/bike-sharing-demand/data
- Airbnb_price: https://www.kaggle.com/datasets/stevezhenghp/airbnb-price-prediction
- austin_housing: https://www.kaggle.com/datasets/ericpierce/austinhousingprices
- cars: https://www.kaggle.com/datasets/prassanth/new-cars-price-2019

APPENDIX C
ALL FEATURE RANKINGS

Feature Name	CART	Linear	RF	Ridge	Elastic Net
livingAreaSqFt	1.0	1.0	1.0	1.0	1.0
zipcode	2.0	11.0	2.0	11.0	9.0
numOfBathrooms	3.0	3.0	3.0	3.0	3.0
latitude	4.0	16.0	4.0	16.0	13.0
avgSchoolRating	5.0	5.5	7.0	5.5	5.0
longitude	7.0	9.5	5.0	9.5	8.0
yearBuilt	7.0	13.0	6.0	13.0	11.0
numOfAppliances	7.0	26.5	12.0	26.5	25.0
avgSchoolSize	9.0	16.0	8.0	16.0	14.0
avgSchoolDistance	11.0	22.5	9.0	22.0	19.0
numPriceChanges	11.0	13.0	10.0	13.0	10.0
hasAssociation	11.0	3.0	11.0	3.0	2.0
numOfPatioAndPorchFeatures	16.0	18.0	17.0	18.0	16.0
numOfParkingFeatures	16.0	16.0	16.0	16.0	15.0
numOfBedrooms	16.0	5.5	13.0	5.5	6.0
MedianStudentsPerTeacher	16.0	19.5	20.0	19.0	17.0
hasSpa	16.0	19.5	24.0	22.0	18.0
homeType	16.0	22.5	14.0	22.0	22.0
lotSizeSqFt	16.0	13.0	19.0	13.0	12.0
hasView	23.5	22.5	25.0	22.0	21.0
hasGarage	23.5	22.5	23.0	22.0	20.0
numOfSecurityFeatures	23.5	26.5	18.0	26.5	24.0
numOfStories	23.5	3.0	27.0	3.0	4.0
numOfWindowFeatures	23.5	25.0	26.0	25.0	23.0
garageSpaces	23.5	7.0	22.0	7.5	26.0
parkingSpaces	23.5	8.0	15.0	7.5	27.0
propertyTaxRate	23.5	9.5	21.0	9.5	7.0

TABLE VIII
RANKINGS FOR ALL DIFFERENT METHODS - AUSTIN DATASET

Feature Name	CART	Linear	RF	Ridge	Elastic Net
room_type	1.0	1.0	1.0	1.0	1.0
longitude	2.0	13.0	2.0	13.0	9.0
bathrooms	3.5	2.0	3.0	2.0	3.0
latitude	3.5	9.5	4.0	9.5	6.0
amenities	5.0	16.5	5.0	16.5	13.0
number_of_reviews	6.0	16.5	6.0	16.5	12.0
accommodates	7.0	5.0	7.0	5.0	4.0
bedrooms	8.0	3.0	8.0	3.0	2.0
review_scores_rating	9.0	16.5	9.0	16.5	10.0
property_type	10.0	16.5	10.0	16.5	11.0
beds	11.5	8.0	16.0	8.0	7.0
cancellation_policy	11.5	11.0	11.0	11.0	15.0
instant_bookable	13.5	6.5	15.0	6.5	8.0
cleaning_fee	13.5	9.5	12.0	9.5	16.0
city	16.0	6.5	13.0	6.5	5.0
host_identity_verified	16.0	13.0	14.0	13.0	18.0
bed_type	16.0	13.0	17.0	13.0	14.0
host_has_profile_pic	18.0	4.0	18.0	4.0	17.0

TABLE IX
RANKINGS FOR ALL DIFFERENT METHODS - AIRBNB DATASET

Feature Name	CART	Linear	RF	Ridge	Elastic Net
Basic Miles/km	1.0	1.0	1.0	1.0	1.0
SAE Net Horsepower @ RPM	2.0	2.0	2.0	2.0	2.0
Model year	3.0	9.5	3.0	11.0	13.0
Wheelbase (in)	4.0	6.0	4.0	6.0	5.0
Drivetrain Years	5.0	4.0	8.0	3.0	3.0
Manufacturer	6.0	22.0	5.0	22.5	16.0
Displacement	7.0	18.0	6.0	22.5	29.0
Front tire rim size	8.0	28.0	9.0	28.0	34.0
Fuel System	9.0	22.0	19.0	22.5	18.0
Turning Diameter-Curb to Curb	10.5	13.5	10.0	11.0	12.0
EPA Fuel Economy Est-City	10.5	22.0	7.0	18.5	27.0
Front tire aspect ratio	12.0	13.5	14.0	9.0	8.0
SAE Net Torque @ RPM	13.0	5.0	11.0	5.0	6.0
Front tire width	14.0	13.5	13.0	15.0	17.0
Drivetrain	17.0	28.0	17.0	28.0	24.0
Corrosion Years	17.0	22.0	20.0	22.5	22.0
Roadside Assistance Miles/km	17.0	33.0	24.0	32.5	33.0
Basic Years	17.0	3.0	12.0	4.0	4.0
Drivetrain Miles/km	17.0	28.0	15.0	32.5	32.0
Trans Description Cont.	21.0	28.0	21.0	28.0	21.0
Traction Control	21.0	13.5	23.0	15.0	11.0
Passenger Capacity	21.0	18.0	18.0	15.0	28.0
Trans Type	26.5	7.0	22.0	7.0	7.0
Rollover Protection Bars	26.5	22.0	28.0	22.5	15.0
Air Bag-Side Head-Front	26.5	28.0	25.0	28.0	30.0
Category	26.5	28.0	30.0	22.5	19.0
Child Safety Rear Door Locks	26.5	18.0	26.0	18.5	14.0
Front Wheel Material	26.5	28.0	29.0	28.0	23.0
Engine	26.5	8.0	16.0	8.0	9.0
Passenger Doors	26.5	13.5	31.0	15.0	20.0
Air Bag-Side Head-Rear	32.5	33.0	32.0	32.5	25.0
Corrosion Miles/km	32.5	9.5	34.0	11.0	10.0
Daytime Running Lights	32.5	33.0	27.0	32.5	26.0
Night Vision	32.5	13.5	33.0	15.0	31.0

TABLE X
RANKINGS FOR ALL DIFFERENT METHODS - CARS DATASET

Feature Name	CART	Linear	RF	Ridge	Elastic Net
hr	1.0	3.0	1.0	3.0	1.0
temp	2.0	5.0	2.0	4.0	3.0
yr	3.0	4.0	3.0	5.0	5.0
workingday	4.0	11.0	4.0	11.0	11.0
hum	5.0	2.0	5.0	1.0	2.0
season	6.0	6.0	6.0	6.0	6.0
atemp	8.5	1.0	8.0	2.0	4.0
mnth	8.5	12.0	9.0	12.0	12.0
weathersit	8.5	10.0	7.0	9.5	8.0
weekday	8.5	9.0	10.0	9.5	10.0
windspeed	11.0	7.0	11.0	7.0	7.0
holiday	12.0	8.0	12.0	8.0	9.0

TABLE XI
RANKINGS FOR ALL DIFFERENT METHODS - BIKE DATASET