CIS-445 Machine Learning
Weka – Individual Project 3 Report

Student Name ____Helen Le_____

The tutorial is worth 100 points.

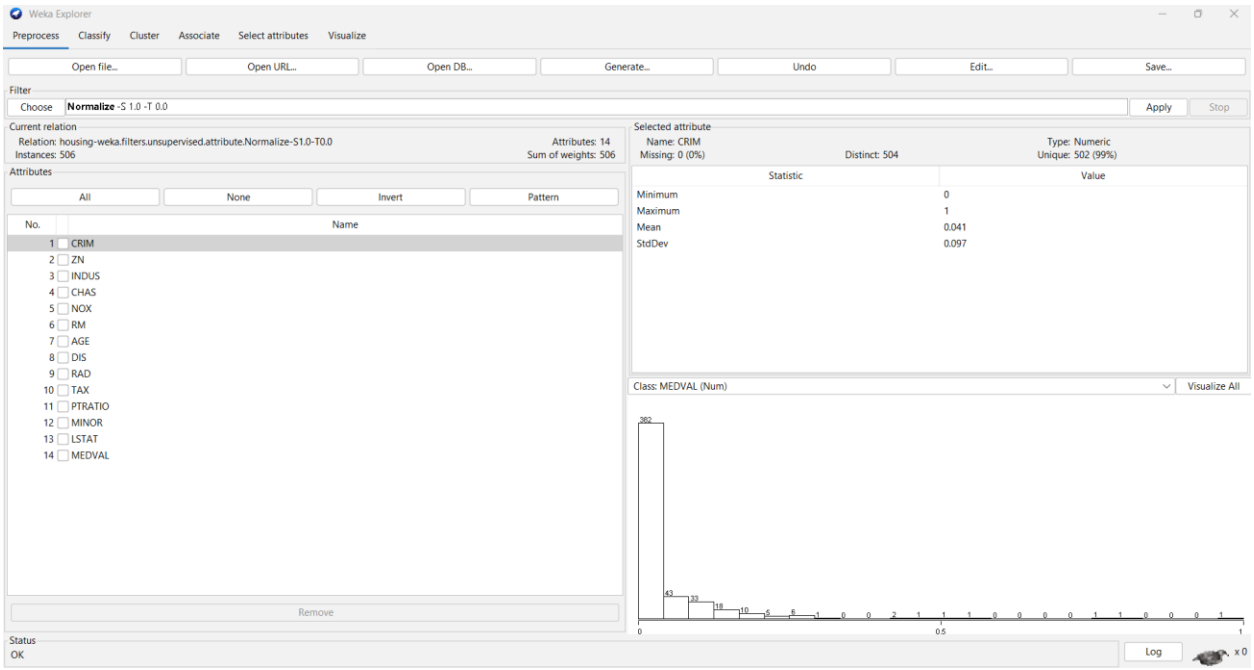
Objectives: Learn about Weka, the Machine Learning Workbench

This project contains two parts: Part A and Part B. In both parts you will use Weka Explorer and Weka Experimenter. In Part A you will work with the `housing.arff` dataset and essentially follow the given instructions. In Part B, however, you will work with `Salary.arff` dataset and explore the performance of several models on your own. Some parts of the project describe Weka and machine learning algorithms. You need to read these parts. There are also other parts which you have to work hands-on, step by step, in Weka. Note that you need to copy some of the results and answer the questions which I ask in the project and paste them into this document so that I can check how well you followed all steps in the project. Because this is the second version of the Weka – Individua Project 3, I would appreciate your comments and suggestions regarding the organization, clarity, spelling, content, and/or anything else.

Please note that this file which now contains 3 pages may grow to a few more pages after you copy and paste here the screen images, tables, answers to questions, etc. from Weka – Individual Project 3 file.

Page 7

Copy and paste Figure 6 below. The Figure should be from your run, not from my run.



Do all numeric attributes are mapped into the [0,1] range?

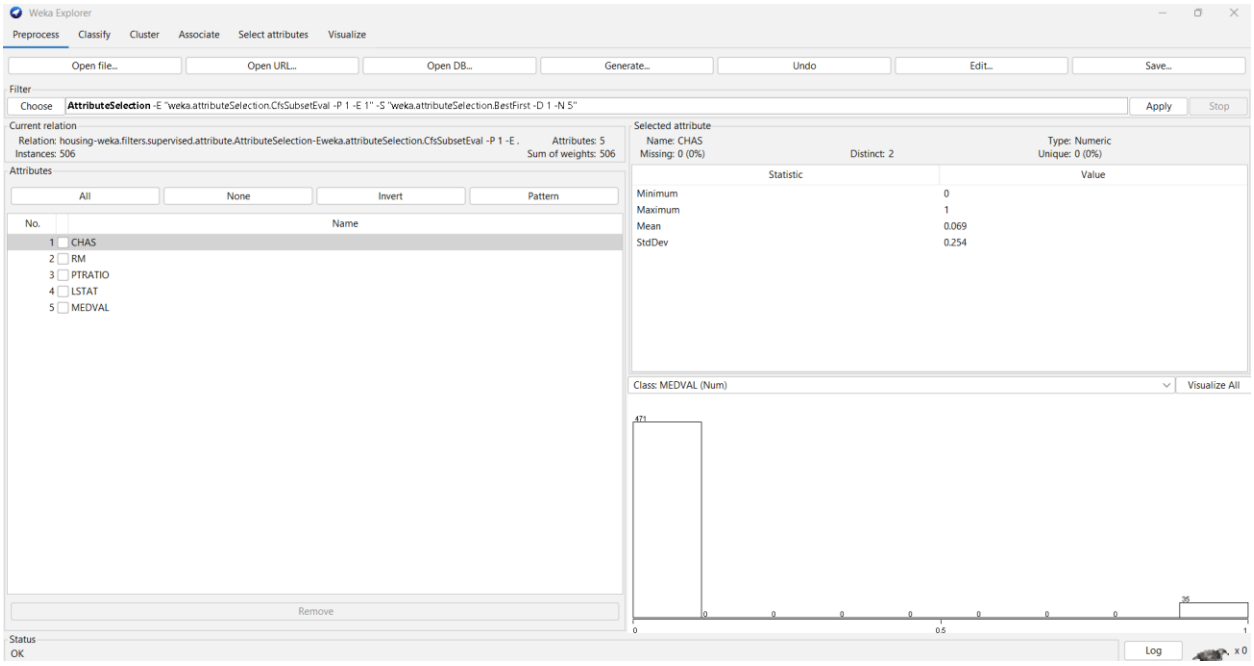
Yes

How about the MEDVAL attribute? Is it normalized too?

No, not unless ignoreClass parameter is set to True in the Normalize filter.

Page 10

Copy and paste Figure 8 here. The Figure should be from your run, not from my run.



How many and which attributes were retained by the attribute selection filter?

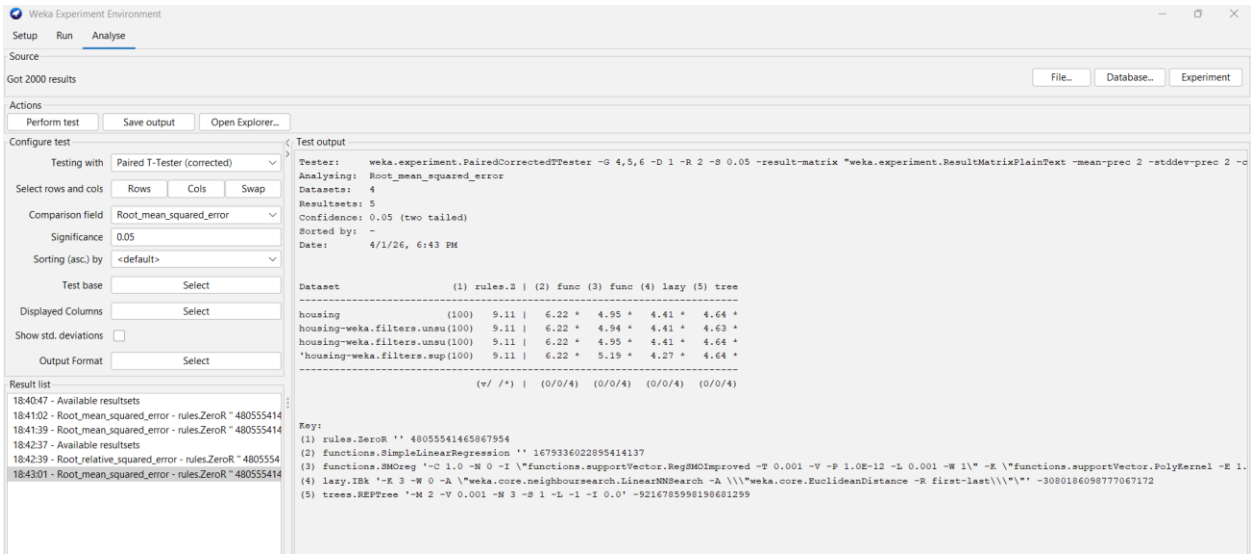
Five attributes were retained.

List the retained attributes below.

CHAS – Charles River dummy variable
RM – average number of rooms per dwelling
PTRATIO – pupil-teacher ratio by town
LSTAT - % lower status of the population
MEDVAL – median value of owner-occupied homes in $1000's

Page 13

Copy and paste Listing 1 here. The listing should be from your run, not from my run.



In Listing 1 which algorithm is used as the baseline to which the remaining algorithms are compared?

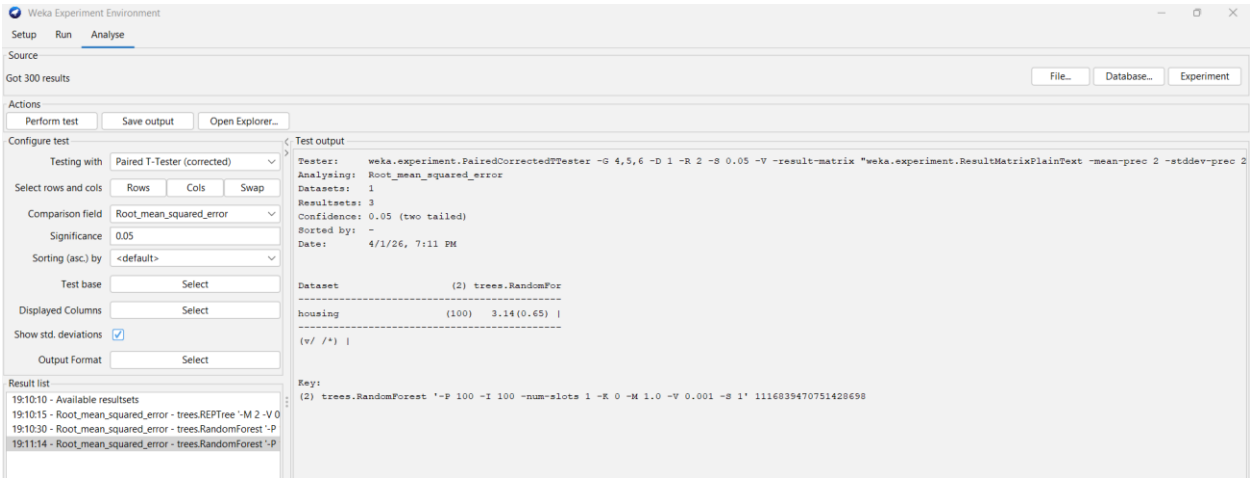
ZeroR, which is actually a great benchmark to compare against more complex models. It ignores all predictor attributes, focusing on the target variable and predicting the majority class.

How did the other 4 algorithms do in terms of performance (RMSE) compared to the baseline algorithm? Did they do statistically better, worse or the same? Remember the objective in regression is to have the RMSE as small as possible. Comment.

As indicated by the asterisk (*) next to all tests, the classifiers performed statistically different and lower than the baseline classifier (ZeroR). We know that a lower RMSE is better because it signifies that a model's predictions are closer to the actual observed values, indicating higher accuracy and better fit. We see that the baseline classifier actually produced the highest RMSE of 9.11 and all other classifiers produced lower ones. Since a lower RMSE is better, these algorithms actually performed better than ZeroR.

Page 25

Copy and paste Listing 9 here. The listing should be from your run, not from my run.



Which model turned out to be the best on terms of RMSE?

Random Forest, which combines multiple decision trees using bagging to randomly select a subset of features at each split and aggregating the predictions of individual trees.

What is the value of RMSE and its standard deviation?

RMSE = 3.14 (thousands of dollars)
Standard Deviation = 0.65

Page 28

Discuss briefly the distribution of attributes.

Overall, there are 6,684 instances and 9 attributes which are described below:
Age, a numeric attribute, is a positively skewed distribution, meaning more of the data clusters on the younger end of the spectrum. The minimum age in this dataset is 21 years old, and the maximum age is 62 years old. The average age is 33-34 years old with a deviation of 7.596 years.
Gender is a binary, nominal attribute as there are only two options presented: male and female. This dataset has 3,671 males and 3,013 females, making it decently balanced but ultimately have more males.
Education Level converts what would be nominal to numeric by using a ranking scale: a 0 represents highest level of education being a high school diploma, 1 represents a Bachelor's Degree, 2 represents a Master's Degree, and 3 represents a Ph.D.
Job Title is a nominal attribute with 129 distinct jobs. The most common job is Software Engineer with 809, followed by Data Scientist with 515, and Data Analyst with 329. The less common jobs include Operations Director, Network Engineer, VP of Finance, UX Researcher, and many more with only 1 recorded instance.
Years of Experience, a numeric attribute, is self-explanatory but describes the time each instance has had in this career. The minimum years of experience is 0, and the maximum is 34 years. The average years of experience in this dataset is 8 with a deviation of approximately 6.
Country is a nominal attribute with 5 options that are all incredibly balanced: UK, USA, Canada, China, and Australia.

Race is a nominal attribute with options such as White, Hispanic, Asian, African American, Mixed, and Black. The most populated race is White with 1,957 instances and Asian with 1,599 instances. The least populated races are Hispanic with 322 instances and Welsh with 333 instances.
Senior is a binary attribute where 0 represents not being a senior position and 1 represents being a senior position. The distribution contains many more non-senior positions than senior ones, which makes logical sense.
Lastly, **Salary** is the target attribute and is numeric, meaning we are dealing with a regression problem. It appears to be a bimodal distribution as it has two peaks. The minimum salary is $350, and the maximum salary is $250,000. The mean salary is $115,307.175 with a standard deviation of $52,806.811 which is high, meaning there is lots of variation in the salaries.

Did you normalize numeric attributes or standardize?

I chose to **normalize** numeric attributes, meaning all input attributes now have a minimum of 0 and maximum of 1. Gender, job title, country, and race were not normalized because they are nominal attributes. Salary was not normalized because it is the target attribute and ignoreClass was not activated.

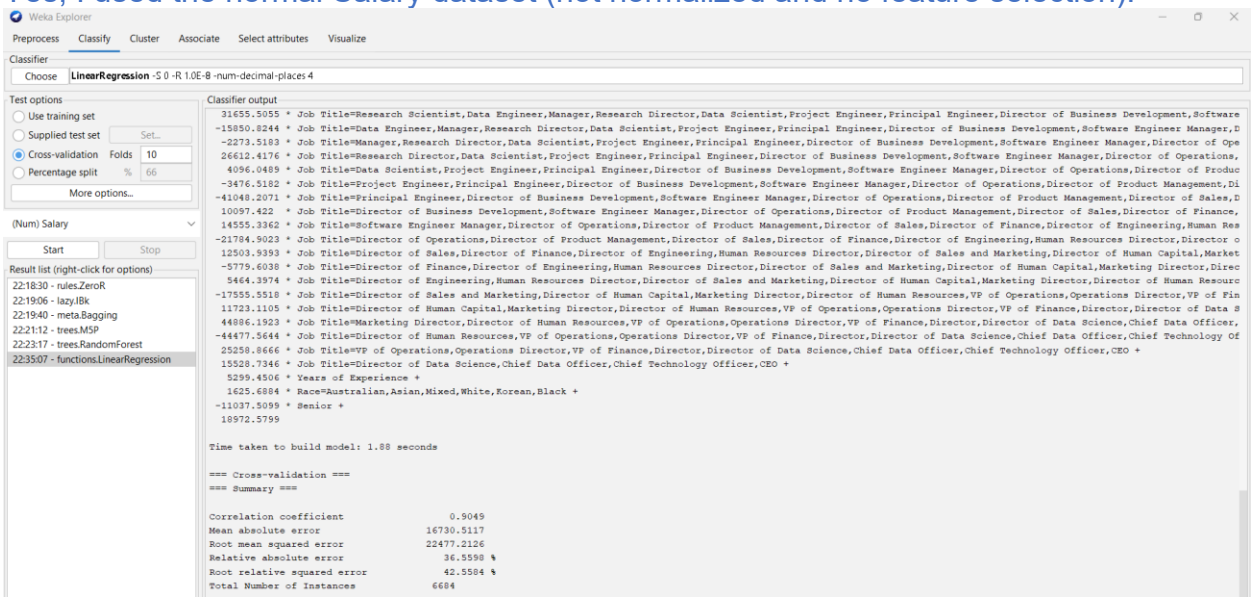
Which attributes were retained by the attribute selection filter? List them below.

Gender
Education Level
Job Title
Years of Experience
Race
Salary

Age, Country, and Senior were all removed.

Did you do the test runs of Linear Regression, IB*k*, Bagging, M5P Tree, and Random Forest in Weka Explorer?

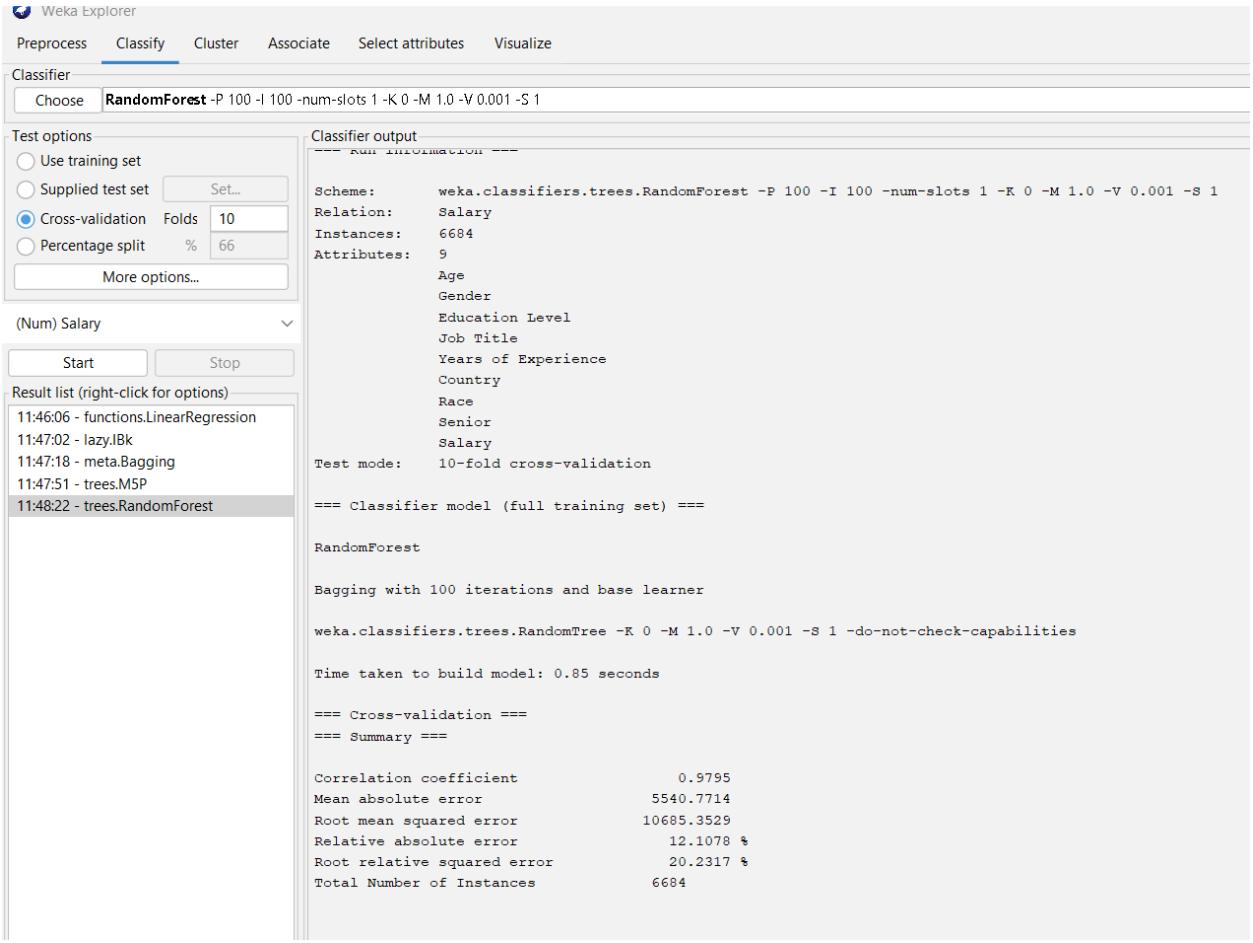
Yes, I used the normal Salary dataset (not normalized and no feature selection).



What are the preliminary results in terms of the performance with respect to the correlation coefficient (R), root mean squared error (RMSE) and mean absolute error (MAE)? Discuss.

	R	MAE	RMSE
Linear Regression	0.9049	16730.5117	22477.2126
IBk	0.8939	12615.6874	24240.4368
Bagging	0.973	6167.647	12186.3229
M5P Tree	0.979	6072.766	10759.5897
Random Forest	0.9795	5540.7714	10685.3529

Linear Regression performed reasonably well with R=0.9049, suggesting there is a strong linear relationship between the data. With lower RMSEs and lower MAEs being desired, this model performed worse than other models.
IBk has a slightly lower R, indicating a still strong but slightly weaker linear relationship. Its MAE is the highest out of all models, and the RMSE is the second highest.
Bagging has a high R value of 0.973, indicating a strong linear relationship. RMSE and MAE are cut in half from IBk, making it a stronger model for this dataset. As it averages out predictions from multiple models, it smooths out the errors seen in IBk.
M5P performs even better than Bagging in all three metrics with a higher R, lower RMSE, and lower MAE.
Lastly, **Random Forest** performs even better than M5P, producing the best performance with the highest R, lowest RMSE, and lowest MAE.
Overall, all five algorithms performed well, but the tree-based models performed the strongest.



Compare the performance of IB*k*, Bagging, M5P Tree, and Random Forest to *Linear Regression* (benchmark) with respect to the Correlation Coefficient (CC), RMSE, and MAE. Are the differences in the models' performance statistically significant at p=0.05?

Correlation Coefficient

	LinearRegression	IBk	Bagging	M5P	RandomForest
Salary	0.91	0.89 *	0.97 v	0.98 v	0.98 v
Salary-normalized	0.91	0.89 *	0.97 v	0.98 v	0.98 v
Salary-feature-selection	0.90	0.92 v	0.96 v	0.96 v	0.96 v

We see that Salary and Salary-normalized produce identical results, with Salary-feature-selection falling slightly behind in all models except for IBk where it performed statistically higher and better as indicated by the v. For Salary and Salary-normalized, IBk had performed statistically worse than the base algorithm, as indicated with the *. Bagging, M5P, and RandomForest then produce stronger correlation coefficients (closer to 1) than the LinearRegression model. Random Forest and M5P produced correlation coefficients closest to 1, making those the strongest models. IBk then produced the lowest correlation coefficients, making it the weakest model.

RMSE

	LinearRegression	IBk	Bagging	M5P	RandomForest
Salary	22465.44	24172.47 v	12241.83 *	10776.99 *	10709.68 *
Salary-normalized	22465.64	24193.68 v	12235.74 *	10775.92 *	10689.71 *
Salary-feature-selection	22573.93	21379.31 *	15124.78 *	14528.68 *	14582.30 *

Again, Salary and Salary-normalized produce nearly identical results, though there is more deviation here than with the correlation coefficient. Salary-feature-selection then deviates more from the Salary and Salary-normalized datasets. Since a lower RMSE is better, the * will indicate a better (lower) RMSE and v will indicate a worser (higher) RMSE. For Salary and Salary-normalized, IBk produces worse results. However, for Salary-feature-selection's IBk and all other algorithms for all three datasets have a *, indicating a statistically significant difference that is lower which again, is better in this case. Thus, the base LinearRegression for all three datasets had worse results than all other algorithms, except for IBk for Salary and Salary-normalized. RandomForest is the strongest model as it produced the lowest RMSEs, which means there is a lower magnitude of prediction errors. IBk and LinearRegression produced the highest RMSEs, making those the weaker models.

MAE

	LinearRegression	IBk	Bagging	M5P	RandomForest
Salary	16717.30	12675.25 *	6216.91 *	6184.95 *	5573.49 *

Salary-normalized	16717.70	12686.78 *	6211.03 *	6184.70 *	5564.27 *
Salary-feature-selection	16765.96	11031.72 *	8338.33 *	9176.97 *	7948.26 *

The same trend repeats – Salary and Salary-normalized produced similar results across all models, while Salary-feature-selection deviates more. A lower MAE is better, and we see that all models produced a lower MAE than the base LinearRegression. RandomForest produced the lowest MAE, meaning there were smaller average differences between predicted and actual values, making it the strongest model. LinearRegression produced the highest MAEs, making it the weakest model.

Do normalization or standardization, or attribute selection which you applied, improve the performance of the models?

Normalization had virtually no impact on performance. Across all three metrics, the Salary and Salary-normalized datasets produced nearly identical results for every model. If there were differences, they were not statistically significant. Tree-based models (M5P, RandomForest, and Bagging) and instance-based models (IBk) are typically insensitive to feature scaling.
Attribute selection then hurt performances. For Correlation Coefficient, all models except IBk show lower values with feature selection compared to the given and normalized datasets. For RMSE, all models produced higher (worse) RMSE with feature selection than the other two datasets. Lastly, for MAE, MAE increased for Bagging, M5P, and RandomForest, while IBk actually improved compared to the other two datasets. Since IBk is sensitive to irrelevant features, removing irrelevant features strengthens its performance. Tree-based models on the other hand can ignore irrelevant features through splitting.

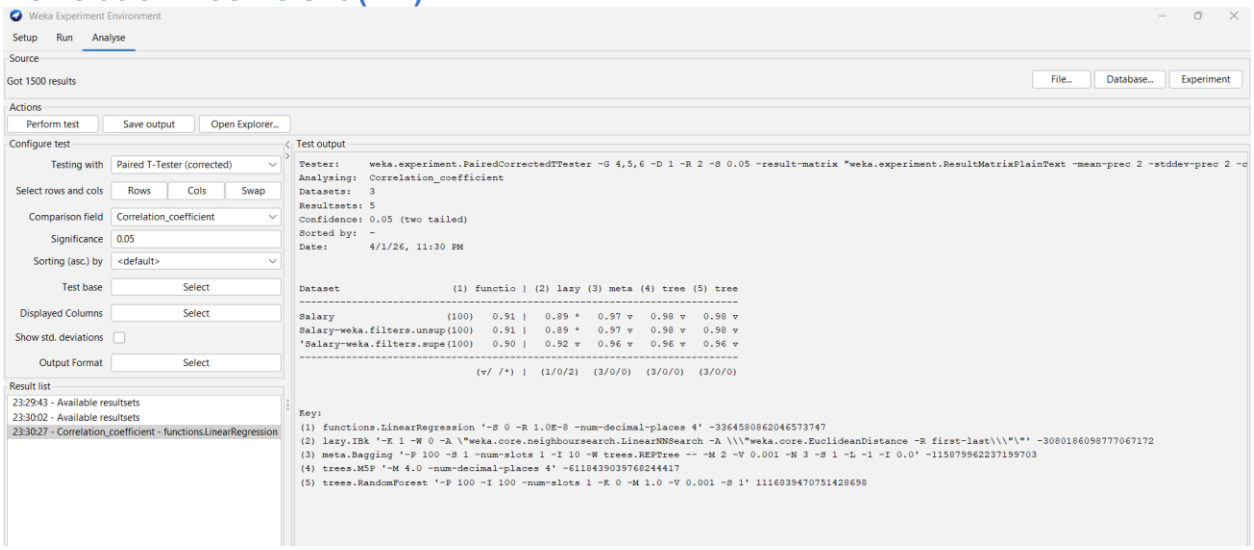
Discuss Part B of the project and the results.

In Part B of this project, I used the Salary dataset from Kaggle. The goal is to be able to predict salary using machine learning algorithms based on nine attributes, including age, gender, education level, race, and more. For this experiment, three sets of data were used: the given one, a normalized version (min = 0, max = 1), and a version using the AttributeSelection that only selected the most relevant attributes in the dataset. To get through all ten repetitions with ten folds each, the experiment took approximately 40 minutes to run.

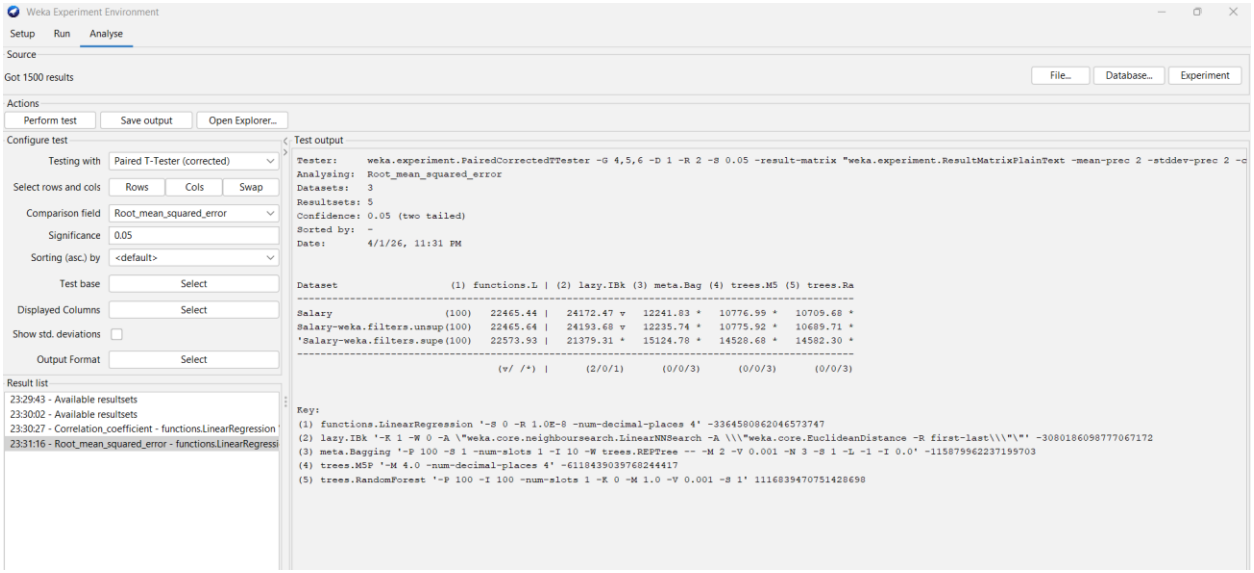
The experiment used Linear Regression as the base algorithm and compared it to IBk, Bagging, M5P Trees, and Random Forest. From there, the metrics obtained were Correlation Coefficient, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). For Correlation Coefficient, the goal is to have a value close to -1 or 1 as it indicates a strong, linear relationship. For RMSE, the goal is to have a low RMSE as it reduces the magnitude of prediction errors, indicating a better-fitting model. A lower MAE also indicates a better-performing model as it signifies smaller average differences between predicted and actual values. We see that RandomForest consistently showed the strongest performance and IBk and LinearRegression showed the weakest performances.

Copy and paste all relevant screen shots from Weka Experimenter below.

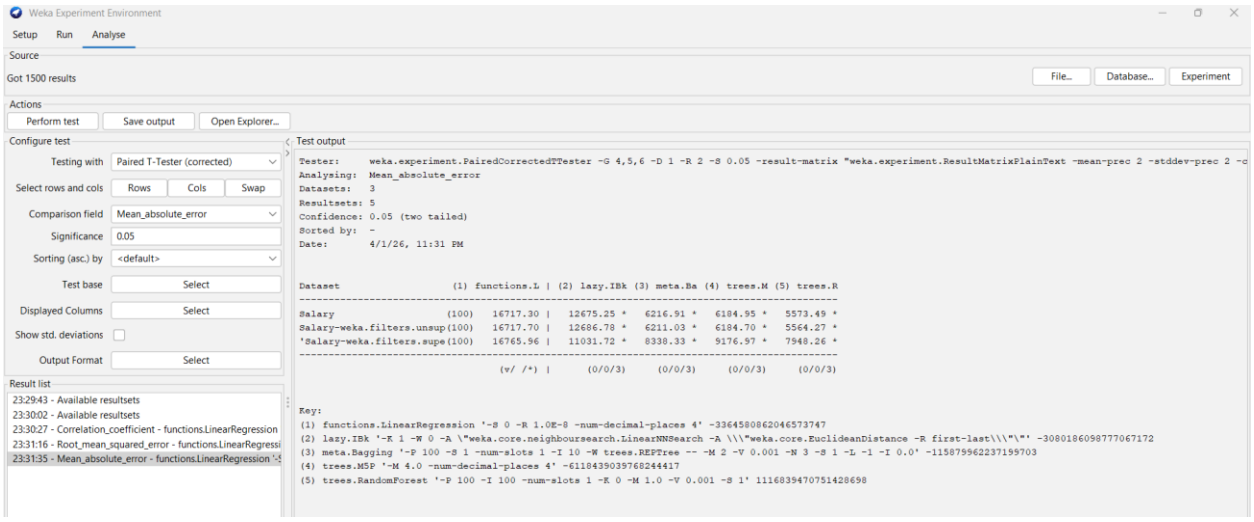
Correlation Coefficient (CC)



RMSE



MAE



Write your comments and suggestions below. (optional)