Comparing classification alogirthms using mortgage dataset

***Abstract*—This paper looks looks at the mortgage dataset and uses 2 classification algorithms – Naïve Baye, KNN and Decision Tree and compare algorithms performance after tuning parameters to see if tuning parameters will increase algorithm performance or not. The results show that when predicting if a customer will pay back mortgage on tuning all 3 algorithms parameters will increase accuracy. KNN perofrms the best with accuracy of 0.687% followed by Naïve Baye 68.20% and lastly Decision Tree with accuracy of 67.34.**

Keywords—mortgage dataset, machine learning, KNN, Decision Tree, Naïve Bayw

I. Introduction

Machine learning has become a very useful tool to help predict trends and hidden patterns. Machine learning not only analyses data but also helps to predict future events such as whether someone will be able to pay off their mortgage in the future.

In our study we are using a mortgage dataset from a bank describing 2000 of its previous mortgage customers. The data tells us the status of the Mortgage two years after acceptance. we aim to look at which attributes are important to decide whether a person is paying their mortgage on time and we will also compare different classification model's accuracy to determine which classification gives the best accuracy.

The machine learning algorithms we are using are Decision Tree, K Nearest Neighbour and SVM.

II. Machine Learning Algorithms

We have chosen three machine learning algorithms to compare for our experiment which are K Nearest Neighbors, Naïve Baye, Decision Trees and Random Forest.

A. K Nearest Neighbor

The k -nearest neighbor (k -NN) is a supervised machine learning algorithm , K-NN is widely used and is considered one of the simplest algorithms.

KNN algorithm been applied successfully to a wide range of problems, such as image processing and statistical pattern recognition [1].

KNN is an example of a type of learning known as lazy learning it relies on labeled input data to learn a function which can then produces a suitable output when given new data which isn't labeled. KNN stores all available input data and classifies new input data based on a similarity measure (e.g., distance functions).

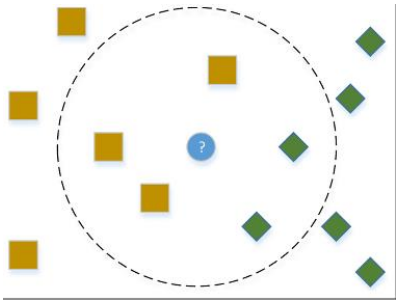


Fig.1. Illustration of k-nearest neighbors.

The above figure(1) helps to illustrate how KNN works, in the circle there are 3 yellow squares and 2 green diamonds the new input which is the blue circle should be classified as yellow square class, this is because yellow square are the most common class in the circle.

Parameters we are going to tune when experimenting whith KNN are its distance functions. The are 3 distance function Euclidean, Manhattan and Minkowski.
Euclidean is the most popular function to use, it will be interesting to see which distance function gives us the greatest accuracy.

We will also be tuning the number of K nearest neibours value to see if the number of neighbours affect the accuracy of the results.

We are going to use KNN because it works well with small dataset KNN Struggle when there are a high number of independent variables.

B. Naïve Bayes classifier

The classifier is based on the Bayes' Theorem with an assumption of independence ie features are assumed to be independent. This assumption, though, is almost certainly not true and is therefore termed a naive assumption. Applying Bayes' Theorem with this naive assumption gives us the Naive Bayes Classifier.

Naive Bayes model is useful for both small and large data sets, our data set is small and is suitable to use with this classifer. Bayes theorem probability the equation is below:

Bayes theorem probability the equation is below:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

C. Decision Tree

Decision Trees are a very popular algorithm, in WEKA .Begum et al points out in their study one reason that the decision tree algorithm is popular is because it is very easy to implement it does not take

long to construct the model compared to other algorithms such as Radom Forest it is also very reliable.

Decision trees fall under the category of supervised learning and can be used as both regression and classification algorithm therefore making the algorithm suitable for our study.

Decision tree structure is like a flowchart which can be seen in the figure below

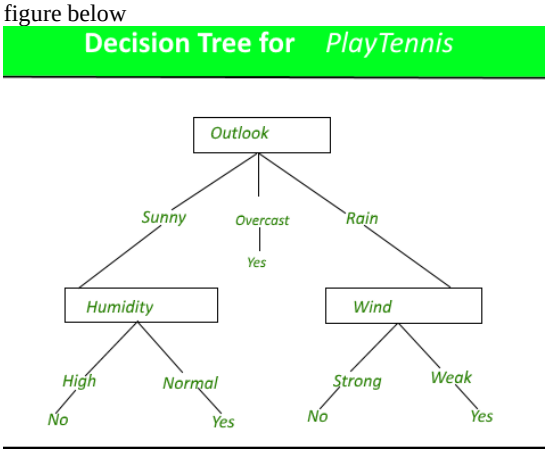


Fig.5. Example of a desion tree

In a decision tree each internal node denotes a test on an attribute an example would be is a toss coin is head or tails, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label for example win or lose.
An Disadvantage of decision tree is that it doesn't work well for continuous data unless it is numerical data therefore when we clean our mortgage dataset all categorical variable will be transformed to numerical/binary variables.

III. LITERATURE REVIEW

Certain efforts using machine learning models have been made in the past by researchers using different datasets for predicting whether a person will pay back mortgage and loans on time.

A. *Design and Comparison of Data Mining Techniques forPredicting Probability of Default on a Loan[2]*

Korhan et al completed a study using the Default of Credit Card Clients(Taiwan) dataset which is avilanle on the UCI machine learning repository.
The dataset from the Taiwan bank in 2005 was used to predict how many cardholders have defaulted with their payment.

The aim of the study was to compare the different algorithms and see which algorithm gave the highest accuracy.
Korhan et al looked at 2 machine algorithms which we are looking at which are KNN and SVM, the study also used the Naïve Baye algorithms and a custom algorithm called LoanAL.

Results of the study are KNN gave the highest accuracy of 77% followed by Naïve Baye (59.62%), SVM (81.18%) and LoanA1(80.8%).

The drawback of this study is, it is only useful to know which algorithm performed better when parameters are not optimized because only default values of the algorithms was used and Korhan et al didn't try to tune the algorithm parameters.
Another problem with this study is no information is given on the feature selection, we do not know if all attributes were used from the dataset or if they used a selection process such as wrapper method.

However Korhan et al did describe in detail how they cleaned the data during data pre-processing stage. Pre-processing includes data integration, data cleaning, normalization and transformation of data.

B. *Data Mining Classification Algorithms Determining the Default Risk[3]*

Begum et al compared different machine learning classification algorithms and see which model performs the best when predicting default risks. Begum et al also identified which attributes contributes to increase risk of loan default and can be used by banks to calculate the risk involved when giving a customer a loan.

The data used for this study was obtained from the Turkish Statistical Institute 2015 survey containing 20275 rows of data and 12 variables including age,gender, martial status, education, work, health, region, housing , revenue, home loan , bills and class.
Weka was used for this study, Begum et al chose 6 classification algorithms which were Decision Tree, Random Forest, Naïve Baye ,BayesNet, Multilayer Perception and Logistic.

Unlike Korhan et al, Begum et al does not go into details about the data preprocessing step, however both studies are similar as they didn't tune algorithm parameters, using default parameters.

Begum et al evaluated classification algorithm performance based on the following criteria classification accuracy, speed, robustness, scalability, Interpretability and rule structure.

In terms of accuracy Logistic performed the best (83.1%) followed by Naïve Baye (82.53%), BayesNet(85.52%), Decision Tree(82.47%) and Random Forest(82.11%).

Whilst analyzing the attributes Begum et al used chi-square and odds ratio per attribute and concluded women were more likely to pay back loans compared to males. Compared to married unmarried were more likely to pay back loans and had a lower default risk. Health also is an important factor someone with good health are less likely to default their loan.

C. *Loan Default Prediction and Indentification of Interesting relations between attributes of peer to peer loan applications [4]*

Zakaria et al completed a study and aimed to predict peer to peer loan default whilst also looking at which attributes contribute to peer to peer loan default. Similar to Begum et al's study Weka was used for this study.

The dataset they used belongs to Landing Club's in California however the year isn't known attributes include loan amount, interest rate, payment term, loan grade, home ownership status, annual income and loan purpose.

Similar to Korhan et al's study Zakaria et al describe in detail their study data preprocessing steps. Data preprocessing steps include searching for missing data/values, removing irrelevant and highly correlated attributes, removal of outlier and balancing data.

Using Weka experiments were conducted where 6 classification algorithms parameters were tuned to find the best performance for each algorithm. The classification algorithms used in this study were Decision Tree, Naïve Baye, K Nearest Neighbour, Random forest, ANN and 1R.

All the classification was experimented firstly will all attributes, secondly attributes suggested by wrapper method and finally using filter feature selection.

Decision tree were tuned by lowing confidence factor from 0.25 to 0.15. Zakaria et al also tested the above with and without tree pruning.

Random forest was tuned by changing seeds to 2 and experimenting with 100 and 200 trees.

K Nearest Neighbour was tuned by changing number of K neighbours – 5, 10 and 15. Zakaria et al didn't use wrapper method for the experiment as the algorithm was taking a long time to load. It would have been interesting if Zakaria et al also experiment with all 3 distance functions Euclidean, Manhattan and Minkowski instead of only experimenting with Euclidean to see if changing distance function had any effect on overall algorithm performance.

Begum et al concluded that for decision trees lowering confidence factor reduced model accuracy from 63% to 62%. Less pruning also increases accuracy 65.39%.

Random forest gave better results when all attributes are used 71.75% with 200 trees and seed equal to 2, using feature selection gave results with less than 70%.

K Nearest Neighbour gave greater accuracy when nearest neighbour was small (K=5) 61.18% however there was very little difference in result when K was 10 and 15 61.05% and 61.08.

Overall Random forest gave the greatest accuracy followed by Decision Tree, Naïve Baye and K nearest neighbor at the bottom.

IV. Dataset Description

The dataset we have chosen contains raw unprocessed data from a bank describing 2000 of its previous mortgage customers. The data tells us whether or not each customer repaid the mortgage successfully during the year 2010 indicated by 'G' and if they were bad payers 'B'. The source of the dataset is Attar Software.

The table below (Fig. 2) describes the attributes in the mortgage dataset

TABLE I. Mortgage dataset description

	Mortgage dataset description		
Field	***Type***	***Description***	***Values/Range***
Application_id	Auto Increment numerical value	Mortgage application ID	10032 to 12517
Total_advance	Numeric	Total mortgage required including extras	20 to 1387
Subscription	Numeric	Mortgage subscription	3 to 2082
Term	Numeric	Mortgage term	5 to 40
MortgageType	Discrete	Type of mortgage	3 (End, P/P, REP)
ExistBorrowe	Binary	Does customer borrow any money currently either as an existing customer or a loan	2 (1=Yes, 0=No)
Age1	Numeric	Customer Age	18 to 91
Income 1	Numeric	Income from job 1 in thousands £	13 to 1106
Income 2	Numeric	Income from job 2 in thousands £	0 to 3511
Paymethod	Discrete	How customer will pay mortgage	2 (B/A, Other)
PropertyType	Discrete	Type of property	6 (Flat, Semi,Det, Terr, SemiB,DetB)
YearBuilt	Numeric	Year property was built	0 to 9
PurchPrice	Numeric	Purchase price of the property	49 to 2875
Occupation1	Discrete	Prime occupation of the applicant	0 to 9
Occupation2	Discrete	Second occupation of the applicant	0 to 10
Remortgage	Binary	Is this a remortgage?	2 (1=Yes, 0=No)
appSource	Discrete	How are they applying for the mortgage	4 (Blank, Counter, Agent, Other)
LTV	Numeric	Loan to Value	5 to 214
Sub_to_in	Numeric	Related to buy to let mortgages	0 to 217
Adv_to_INC	Numeric	% interest rate to be included on advance	0.1 to 16.7
IncMult	Numeric	Mortgage valuation fee	-22.6 to 20
Region	Discrete	Region of country property is located	5 (N, E, W, SE, SW)
Mosaic	Numeric	Mortgage Account Fee in £	0 to 99
SecondCharge	Binary	If coming out of existing deal early	2 (1=Yes, 0=No)
Joine_Income	Numeric	If more than 1 applicant joint income in thousands £	17 to 3651
Outcome	Discrete	status of the Mortgage two years after acceptance ('Good' = not in arrears, 'Bad' = 3 or more months in arrears)	2 (G=Good, B=Bad)

Fig. 2. Table describiton attributes in the mortage dataset

Identify applicable funding agency here. If none, delete this text box.

Data preprocessing steps included looking for Data cleaning, removing irrelevant attributes, normalising/scaling data and identifying any outliers. Weka was used for data preprocessing.

1. Data cleaning

There were some data for PropertyType which was spelled incorrect. We changed all data (total 88) spelled as "DetB" to "Det". We changed all data (total 44) spelled as "SemiB" to "Semi".

There was no missing data for Occupation2 attribute there were some rows of data which was "BLANK" however that isn't because data is missing it is because some customers didn't have two jobs this was confirmed by looking at the Income 2 and Joine_Income attribute. If Occupation2 was "BLANK" then Income2 was also 0 and Joine_Income was same as Income 1 value.

However data was missing for AppSource some rows of data was"BLANK" instead of Agent, Counter or Other. We did not delete the data because 2402 of rows was "Blank" therefore instead we created a value called "Unknown" for all the Blank data.

One hot encoding was used to convert categorical variables.

2. Removing irrelevant attributes

Application ID attribute was removed before the machine learning task this is because it is only for indexing/reference purposes and not a useful information for the machine learning task.

3. Normalisation

All data was normalised this is when the values of numeric columns in the dataset such as age and income is changed to a common scale, without distorting differences in the ranges of values. All values are now between 0 and 1.

4. Outliers

Using Weka filter InterquartileRange we found out that we had 266 outliers out of 4875. The outliers were removed from the data using Weka filter RemoveWithValues.

Data was now ready for the machine learning task.

V. Experimental Methodology

The mortgage dataset was chosen because it is not a large dataset therefore the experiments will take less time compared to large dataset therefore being more practical then waiting a long time for series of experiments to take place in Weka.

We chose Naive Bayes,J48, random forest and KNN because these algorithms can be categorical and numerical data they have also been used as discussed in literature review with similar datasets. Our dataset is small, and the 4 chosen algorithms can generate models in a good speed and are very common algorithms therefore having a good usage frequency.

Whilst evaluate the 4 classification algorithms we will use Begum et al's method of evaluating and algorithms [3].
Whilst evaluating algorithms we will be looking for

1. Classification accuracy: the ability of the model to correctly predict the label of class
2. Speed: How long the model takes to run
3. Robustness: accuracy whilst predicting even if there are any missing data or noisy observations
4. Scalability: the ability of a model to be predict accurately and being productive when handling a large amount of data
5. Interpretability: the level of understanding provided by the model
6. Rule Structure: the understandability of the algorithms rule structure

After conducting literature review looking at related studies are hypothesis is

1. KNN will perform better when K value is small <5
2. Reducing decision tree confidence factor will also reduce decision tree accuracy
3. Naïve Baye will perform better when using default parameters

Cross validation with 10 folds will be used for all experiments.

First we will experiment with decision tree and tune MNO value and confiendence Factor

Next we will experiment Naïve Baye and tune parameter called UseKernelEstimator. Kernel estimator is used for numerical attributes instead of normal distribution.

Finally for KNN, K value will be tuned. Values are 1,3,5,7,9,11,13,15,17,19,21,23 and 25. The values are small to large to see if having a large K value is better than having a small K value. KNN weight will be tuned, there are 3 weights Euclidean, Manhattan and Minkowski. Euclidean is the most popular function to use, it will be interesting to see which distance function gives us the greatest accuracy.

We will experiment with feature selection firstly using all attributes and later using a attribute evaluator on Weka called CfsSubSetEval using its default parameters. CfsSubSetEval evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

After running CfsSubSetEval the attributes that were suggested were Subscription, LTV, Sub_to_in and SecondCharge.

VI. Experimental Results and Discussion

A. *Experiment 1 – Compare all algorithms using default parameters and all attributes – cross valiadation 10 folds.*

For the first experiment all 4 classification algorithms (Random Forest, KNN, Decision Tree and Naïve Baye) were ran using default parameters and all attributes using cross validation 10 folds.

Results from the experiment can be seen below (Fig . 3).

TABLE II. Experiment 1 results - Comparison of classification algorithms

	Experiment 1 results						
Agorithm	Accuracy	RMSE	ROC Area	Precision	Recall	Time (s)	F – Measure
Naïve Bayes	66.4	0.5	0.74	0.67	0.66	0.14	0.66
Decision Tree	66.5	0.54	0.66	0.67	0.67	1.03	0.67
KNN	58.9	0.64	0.59	0.59	0.59	0	0.59

Fig. 3. Table showing results for experiment 1.

For the first experiment in terms of accuracy Decision Tree performed the best at 66.5%, Naïve Baye 66.4% and KNN having the worst accuracy of 58.9%.

It is preferred that the root mean squared error is small for better model accuracy. In terms of RMSE the difference between Naïve

Baye and Decision Tree RMSE isn't big 0.5 and 0.54. KNN has the greatest RMSE of 0.64.

ROC curve helps to evaluate algorithm, the closer the value it is to 1 it means the area under the curve is approaching 1 this indicates that the classification was carried out correctly therefore improving accuracy and interpretability of the model. Naïve Baye ROC Area was the closet to 1 at 0.74 the worst was KNN at 0.59.

Comparing the values in F-measure according to the recall criterion, Random forest shows the greatest/best value of 0.73. There was not much difference between the value Naïve Baye and decision tree 0.66 and 0.67.KNN performed the worst with a value of 0.59.

The algorithms that gave the best results for the precision criterion is Naïve Baye and Decision tree performed they similar with value rounded up to 0.67. KNN performed the worst with a value of 0.59.

Whilst evaluating simplicity of the model we will compare the speed it took to run the model the less time it to run the model the greater the model's simplicity is.
KNN took the less time to run the model in less than 0 second Naïve Baye followed next with 0.14 second.

B. *Experiment 2 – Compare all algorithms using default parameters and feature selection*

We attributes suggested from CfsSubSetEval with default parameters to select best attributes for each algorithm

Results from the experiment can be seen below (Fig . 4).

TABLE III. EXPERIMENT 2 RESULTS - COMPARISON OF CLASSIFICATION ALGORITHMS USING FEATURE SELECTION

	Experiment 2 results						
Agorithm	Accuracy	RMSE	ROC Area	Precision	Recall	Time (s)	F – Measure
Naïve Bayes	68	0.461	0.743	0.686	0.680	0.01	0.675
Decision Tree	67	0.464	0.709	0.67	0.673	0.05	0.666
KNN	61.1	0.623	0.611	0.611	0.611	0	0.611

Fig. 4. Table showing results for experiment 2.

For the second experiment in terms of accuracy Naïve Baye performed the best at 68% followed by Decision Tree (67%), and KNN having the worst accuracy of 61.1%. What is interesting is that All algorithms accuracy improved when using feature selection.

It is preferred that the root mean squared error is small for better model accuracy. In terms of RMSE Naïve Baye has the smallest values of 0.461 the difference between Naïve Baye and Decision Tree RMSE isn't big only 0.003.

ROC curve helps to evaluate algorithm, the closer the value it is to 1 it means the area under the curve is approaching 1 this indicates that the classification was carried out correctly therefore improving accuracy and interpretability of the model. Naïve Baye ROC Area was the closet to 1 at 0.743 the worst was KNN at 0.611.

Comparing the values in F-measure according to the recall criterion, Naïve Baye shows the greatest/best value of 0.675. KNN performed the worst with a value of 0.611.

The algorithms that gave the best results for the precision criterion is Naïve Baye at 0.686.

Whilst evaluating simplicity of the model we will compare the speed it took to run the model the less time it to run the model the greater the model's simplicity is.
KNN took the less time to run the model in less than 0 second Naïve Baye followed next with 0.01 second.
It was expected less time would be taken compared to experiment 1 as experiment 1was using all attributes. Overall Naïve Baye performed the best when no algorithms are tuned using CfsSubSetEval feature selection.

C. *Experiment 3 –Decision tree – best algorithm to use*

TABLE IV. EXPERIMENT 3 RESULTS – BEST TREE ALGORITHM TO UE

Dataset	(1) rules.Ze	(2) rules	(3) trees
'mortgage-weka.filters.un(100)	52.15	59.42 v	67.00 v
	(v/ /*)	(1/0/0)	(1/0/0)

Fig. 5. Table showing results for experiment 3.

Result from experiment 3 shows that decision tree J48 is the best algorithm to use we will be using this algorithm for our decision tree.

D. *Experiment 4 –J48 CfsSubSetEval feature selection attributesand tuning MNO (MinNumObject).*

For this experiments CfsSubSetEval feature selection attributes were are tuning MNO with the following values one at a time – 2,5,10,15 and 20 and assess the performance of the different trees

TABLE V. EXPERIMENT 4 RESULTS – TUNING MNO

Dataset	(1) trees.J4	(2) trees	(3) trees	(4) trees	(5) trees
'mortgage-weka.filters.un(100)	67.00	67.09	67.23	67.31	67.26
	(v/ /*)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

Fig. 6. Table showing results for experiment 4.

No result is significantly better than the other however the results show that best tree is between 10 (67.23) and 15(67.31) objects, in the next experiment we will check this.

E. *Experiment 5 –Finding best tree MNO (MinNumObject) value.*

For this experiments CfsSubSetEval feature selection attributes were used and are tuning MNO. We already know that the best tree is between 10 (67.23) and 15(67.31) objects. We are going to make sure we have not missed a better one by tuning MNO with the from 10 to 20.

TABLE VI. EXPERIMENT 5 RESULTS – TUNING MNO

Dataset	(1) trees.J4	(2) trees	(3) trees	(4) trees	(5) trees
'mortgage-weka.filters.un(100)	67.23	67.28	67.26	67.29	67.26
	(v/ /*)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

(6) trees	(7) trees	(8) trees	(9) trees	(10) tree	(11) tree
67.31	67.32	67.25	67.20	67.22	67.26
(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

Fig. 7. Table showing results for experiment 5.

No result is significantly better than the other however the results show that best tree is when MNO is 16 with the accuracy being 67.32 next best tree is when MNO is 15 the accuracy is 67.31

We now want to see that happens when we continue to tune MNO with higher values and see if this will increase the accuracy in the next experiment.

F. Experiment 6 –test more severe MNO values.

TABLE VII. EXPERIMENT 6 RESULTS – TUNING MNO FOR SEVERE VALUES

Dataset	(1) trees.J4	(2) trees	(3) trees	(4) trees	(5)
'mortgage-weka.filters.un(100)	67.00	67.26	67.15	66.96	67
(v/ /*)		(0/1/0)	(0/1/0)	(0/1/0)	(0,

(6) trees	(7) trees	(8) trees	(9) trees
66.73	67.06	66.91	66.37
(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

Fig. 8. Table showing results for experiment 6.

MNO values we tested were 2,10,20,30,40,50,100,150,200 and 250.
As we can see from the above table no result is significantly better than the other however performance starts to drop after MNO = 50.

We will next want to tune the confidence factor when MNO is 16 to see if we can further increase the trees accuracy from 67.32

G. Experiment 7 –Tuning Confidence Factor when MNO= 16

TABLE VIII. EXPERIMENT 7

Dataset	(1) trees.J4	(2) trees	(3) trees	(4) trees	(5) trees
'mortgage-weka.filters.un(100)	67.32	67.32	67.34	67.28	67.15
(v/ /*)		(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

Fig. 9.. Table showing results for experiment 7.

Confidence factor values were testing at 0.25, 0.2 0.15, 0.1 and 0.05 when MNO is 16 to see if we are able to further increase trees accuracy.

From the results we can see unfortunatley there isn't much difference in the result however 0.1 performs the best 0.15 and 0.25 ar equal at 67.32.

We will try once more my Confidence Factor values from 0.1 to 0.15 to see if any other Confidence Factor value gives better accuracy.

H. Experiment 8 –Continue tuning Confidence Factor when MNO= 16

TABLE IX. EXPERIMENT 8 – CONTIUING TUNING CONFIDENCE FACTOR WHEN MNO = 16

Dataset	(1) trees.J4	(2) trees	(3) trees	(4) trees	(5) trees	(6) trees
'mortgage-weka.filters.un(100)	67.28	67.31	67.31	67.34	67.35	67.34
(v/ /*)		(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

Fig. 10.. Table showing results for experiment 8.

Confidence factor values were testng at 0.1 to 0.15 when MNO is 16 to see if we are able to further increase trees accuracy.

From the results we can see when Confidence Factor is 0.14 and MNO is 16 this increases accuracy to 67.35.

To conclude the best decision tree has a accuracy of 67.35 when MNO is 16 and Confidence Factor is 0.14.

I. Experiment 9 –Naïve Baye tuning

We can tune Naïve Baye by changing UseKernelEstimator to "true" instead of false (by default) and see if doing so will increase the accuracy of Naïve Bayes. Kernel estimator is used for numerical attributes instead of normal distribution.

TABLE X. EXPERIMENT 9 RESULTS – TUNING NAÏVE BAYE

Dataset	(1) bayes.Na	(2) bayes
'mortgage-weka.filters.un(100)	68.16	68.20
(v/ /*)		(0/1/0)

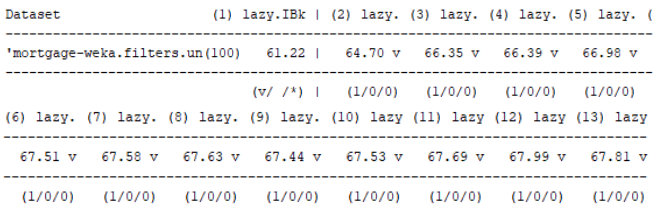
Fig. 11. Table showing results for experiment 9.

From the results we can see that Naïve Baye increases when UseKernelEstimator is set to true however not by a lot first it was 68.18 now it is 68.20.
We can conclude tuning Naïve Baye has increased accuracy from 68.18 to 68.20.

J. Experiment 10 –KNN change K value

We can tune tune KNN by changing K value. For this experiment we have tuned K value to the following odd values 1,3,5,7,9,11,13,15,17,19,21,23 and 25

TABLE XI. EXPERIMENT 10 RESULTS – CHANGE K VALUES



Dataset	(1) lazy.IBk	(2) lazy.	(3) lazy.	(4) lazy.	(5) lazy.	(
'mortgage-weka.filters.un(100)	61.22	64.70 v	66.35 v	66.39 v	66.98 v	
(v/ /*)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)		

(6) lazy.	(7) lazy.	(8) lazy.	(9) lazy.	(10) lazy	(11) lazy	(12) lazy	(13) lazy
67.51 v	67.58 v	67.63 v	67.44 v	67.53 v	67.69 v	67.99 v	67.81 v
(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)

Fig. 12. Table showing results for experiment 10.

From the table above we can see that when K value is 23 the algorihtm performs the best the acuracy is 67.99 the next best value is K=21 after 23 the accuracy drops.

The next parameter we will tune is distance funcrion we will change it from Euclidean to Manhattan and see if it increases the accuracy or not.

K. Experiment 11 –KNN change Distance Function

TABLE XII. EXPERIMENT 11 RESULTS – DISTANCE = MANHANATTAN

Dataset	(1) lazy.IBk	(2) lazy.	(3) lazy.	(4) lazy.	(5) lazy.
'mortgage-weka.filters.un(100)	61.25	65.31 v	66.25 v	66.54 v	67.02 v
(v/ /*)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	

(6) lazy.	(7) lazy.	(8) lazy.	(9) lazy.	(10) lazy	(11) lazy	(12) lazy	(13) lazy
67.99 v	68.28 v	68.03 v	68.10 v	68.15 v	68.23 v	68.14 v	68.15 v
(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)

Fig. 13. Table showing results for experiment 11.

From the table above we can see that using the same K value from last experiment but only changing distance function to Manhattan has resulted in increase of accuracy. The best K value to now K=13 with an accuracy of 68.28, the next best K value is K=21 with an accuracy of 68.23

The next parameter we will tune is distance funcrion we will change it from Manhattan to Minkowski and see if it increases the accuracy or not.

L. Experiment 12 –KNN change Distance Function

TABLE XIII. EXPERIMENT 12 RESULTS – DISTANCE = MINKOWSKI

Dataset	(1) lazy.IBk	(2) lazy.	(3) lazy.	(4) lazy.	(5) lazy.	(6) lazy.
'mortgage-weka.filters.un(100)	61.22	64.70 v	66.35 v	66.39 v	66.98 v	67.51 v
(v/ /*)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	

(7) lazy.	(8) lazy.	(9) lazy.	(10) lazy	(11) lazy	(12) lazy	(13) lazy
67.58 v	67.63 v	67.44 v	67.53 v	67.69 v	67.99 v	67.81 v
(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)	(1/0/0)

Fig. 14. Table showing results for experiment 11.

From the table above we can see that using the same K value from last experiment but only changing distance function to Minkowski has not resulted in increase of accuracy. The best K value in this experiment is K=23 with an accuracy of 67.99, the next best K value is K=25 with an accuracy of 67.81.

From completing our experiments, we can conclude that tuning parameters for all 3 algorithms has resulted in increase of accuracy for K Nearest Neighbor the model performs best when K value is 13 and distance function is Manhattan the accuracy is 68.28 . Decision tree performs best when MNO is 16 and Confidence Factor is 0.14 with a accuracy of 67.34.
Naïve Baye increases when UseKernelEstimator is set to true however not by a lot first using default parameters it was 68.18 now it is 68.20.
The table below shows comparison of the 3 algorithms when tuned to the best performance.

TABLE XIV. BEST ALGORITHM WHEN PARAMTERES TUNED

Agorithm	Parameters Tuned	Accuracy	RMSE	ROC Area	Precision	Recall	Time (s)	F – Measure
Naïve Bayes	UseKernelEstimator = True	68.20	0.454	0.744	0.681	0.680	0	0.678
Decision Tree	MNO = 16 CI= 0.14	67.34	0.463	0.703	0.679	0.676	0.02	0.72
KNN	K =13 Distance Function = Manhattan	68.28	0.459	0.726	0.687	0.685	0	0.682

Fig. 15. Table showing results

In terms of accuracy KNN performed the best at 68.28% followed by Naïve Baye 68.20%, and Decision tree having the worst accuracy of 67.34%. There is not significant difference between accuracy variables between the 3 algorithms.

It is preferred that the root mean squared error is small for better model accuracy. In terms of RMSE Naïve Baye has the smallest values of 0.454 the difference between Naïve Baye and KNN RMSE isn't big only 0.005.

ROC curve helps to evaluate algorithm, the closer the value it is to 1 it means the area under the curve is approaching 1 this indicates that the classification was carried out correctly therefore improving accuracy and interpretability of the model. Naïve Baye ROC Area was the closet to 1 at 0.744.

Comparing the values in F-measure according to the recall criterion, Decision tree shows the greatest/best value of 0.72. Naïve Baye performed the worst with a value of 0.678.

The algorithms that gave the best results for the precision criterion is KNN at 0.687.

Whilst evaluating simplicity of the model we will compare the speed it took to run the model the less time it to run the model the greater the model's simplicity is.
KNN and Naïve Baye both took equal less time of 0 seconds.
Random Forest took the longest time with 1.28 second.

Overall KNN performs the best when parameters are tuned.

VII. Conclusion

All 3 algorithms performed better when parameters were tuned.

The accuracy for KNN, Decision tree and Naïve Baye with default parameters were Decision Tree (66.5%), Naïve Baye (66.4%) and KNN having the worst accuracy of 58.9%.

Feature selection was used using Weka CfsSubSetEval top 4 attributes were Subscription, LTV, Sub_to_in and SecondCharge. This resulted increase of accuracy Decision Tree (67%), Random (66.75%) and KNN having the worst accuracy of 61.1%..

Tuning parameters also increased accuracy all for K Nearest Neighbor the model performs best when K value is 13 and distance function is Manhattan the accuracy is . Decision tree performs best when MNO is 16 and CI is 0.14 with a accuracy of 67.34.
Naïve Baye increases when UseKernelEstimator is set to true however not by a lot first using default parameters it was 68.18 now it is 68.20.
Overall KNN performs the best (0.687%) when predicting whether customer will pay mortgage on time followed by Naïve Baye and Decision Tree.

At the start of the experiment we had 3 hypotheses, we reject the first hypothesis because KNN performed between when K value greater than 5. We accept the second hypothesis because decision tree performed better when confidence factor was reduced from 0.25 to 0.14. We also reject the final hypothesis because Naïve Baye performed better than parameter UseKernelEstimator was tuned to true.

References

[1] Barrientos, R.J., Millaguir, F., Sánchez, J.L. et al(2017)'GPU-based exhaustive algorithms processing kNN queries'. *Journal of Super computing* (73) pp. 4611–4634

[2] Akcura, Korhan & Chhibber, Appan. (2018). Design and Comparison of Data Mining Techniques for Predicting Probability of Default on a Loan.

[3] Beg ̈um Çı ̆gs ̧ar and Deniz ̇Unal(2019) 'Comparison of Data Mining Classification AlgorithmsDetermining the Default Risk' *Hindawi Scientific Programming* pp. 1-9.

[4] Zakaria Alomari, Dmitriy Fingerman(2017) 'Loan Default Prediction and Identification of Interesting Relations between Attributes of Peer-to-Peer Loan Applications ' *ZJCHI*, 2(2)