# Machine learning for classification of financial services companies in the UK's financial sector – technical report

Date: 28.11.2018

Author(s):Alex Noyvirt

Acknowledgements: Work was carried out in collaboration with ONS' Enchanced Financial Accounts development team, who are working on improving the quality, coverage and granularity of the UK's financial statistics

## 1. Introduction

The goal of this project is to develop a method for automatic algorithmic classification of financial corporations to a detailed financial sub-sector classification.

Analysis by the Bank of England in 2010 estimated that the last financial crisis cost the UK economy around £7.4 trillion in lost output. Since that crisis, there has been international consensus that improved financial statistics can help to identify the build-up of risk in the financial system, better inform monetary and financial policy, and hence mitigate the effects of any future financial crises. Reducing the impact by just 0.01% could lead to an estimated saving for the UK economy of £740 million.

As described in Economic Statistics Transformation Programme: Developing the enhanced financial accounts (UK Flow of Funds) (2016), one of the main areas for improvement is an expansion of the financial corporation subsectors for which financial statistics are published.

The novel approach in this work has been to use firm-level information on financial assets and liabilities, and other business information such as turnover and employment to explore the classification of the financial sub-sectors. Previous work, such as Nesta's dynamic mapping of the creative and information economies, has focused on industry classification by analysis of occupation, and did not covered the financial sector. Financial corporations are, to a large extent, classified by their financial activity, therefore, it might be expected that different patterns of financial and business activity would map to different sub-sectors in the financial sector.

The method is designed to use already available survey and administrative data without any reliance on businesses describing directly their activities. The Standard Industrial Classification 2007: SIC 2007 code is used as standardised output of the method that enables easy comparison

to the existing classification of the companies.

Although, typically all businesses have already SIC 2007 codes, assigned to them at some point of their lifecycle, this is a manual and labour-intensive process. Also, there is a need of revaluating periodically the classification due to dynamic changes of the business activities over time. Failure to do so would most likely result in wrong classification, skewed grouping of the businesses based on the SIC and subsequently in inaccurate aggregation of any statistical results in the financial services sector. This, on the whole, affects the ability to monitor efficiently the flows of funds between the separate company groups and ultimately the risk-build in the sector.

The motivation of the project is to research the feasibility of an algorithm that detects automatically any changes in the activity of a company by using indirect survey and administrative data. This has potential to reduce the manual effort involved in reassessing every single company's SIC, enable shorter reaction time in re-classification and ultimately improve the quality of the aggregated results. In machine learning, such a problem is well-studied and represents a typical supervised learning scenario, where the available manual classification, that is, the available SIC codes, are used as target labels to train a model that later is used to generate new class labels, that is, up-to-date SIC codes, based on the current input data that is fed in the algorithm.

A number of machine learning techniques were tested in order to evaluate their potential for achieving sufficient classification accuracy in predicting the classification of a company using the currently available indirect survey and administrative data. Finally, based on the achieved accuracy, a recommendation is made for an optimal application of automatic classification and anomaly detection in classification of companies of the financial sector.

## 2. Input data

The available data consisted of three separate datasets:

- the Financial Services Survey (FSS) – covering the period of Quarter 3 (July to Sept) 2016
- the Inter-Departmental Business Register (IDBR)
- the Financial Services Register (FSR) of the Financial Conduct Authority (FCA) – representing the regulatory approvals for the operation of the financial services companies

The FSS collects information on the assets, liabilities, income and expenditure of UK businesses classified within the financial industry. The survey is conducted quarterly with a population of approximately 70,000 and sample size is approximately 2,000. All the companies of the population are existing records in IDBR as it is used as the basis of the sampling.

After initial cleaning of the data, in which all obvious outliers and duplicated records were removed, the three datasets were joined together to enable training of the model. Joining FSS and IDBR was trivial due to the fact that the FSS dataset contains unique IDBR record identifiers. However, the joining of FCA and IDBR was far more challenging due to lack of suitable unique identifier keys that are present in both datasets. In fact, the process had to be based on identifying matching pairs of records, as discussed later in this section, for both the name and the registered address of a business in both datasets.

At this stage, the difficulties associated with matching of free text strings from two different datasets started to emerge. In some cases, the mismatch could be attributed to simple typos or different way of abbreviation of common names. In other cases, the mismatch was caused by the presence of a previous name entry or address of a company, for example, after a merger, acquisition or change of address that was not updated in one of the datasets. The problems associated with finding of an exact match for a huge number of records necessitated applying fuzzy matching method for probabilistic matching of the keys for generating likely pairs of records, searching for the pair with minimal distance for each record in IDBR and setting a threshold measure, limiting the likelihood of a wrong match.

After experiments aimed to compare the results of matching a number of applicable methods for computing the distance between strings, it was concluded that the Levenshtein metric should be used due to its higher ability to match records with typical data entry errors. Then, a custom build distributed process algorithm, based on the Levenshtein metric, was developed in Scala language for Apache Spark platform. The algorithm splits the strings into separate words and identifies the best pairs based on the minimum Levenshtein distance. Finally, the sum of the distances of the constituting pairs of words is recorded for pair of strings.

The above splitting of strings into words and the identification of matching pairs of words, in comparison to calculation of the distance for the whole strings, eliminates the problem of computing of an incorrect distance measure in case of swapped pairs of words in one of the records. Furthermore, to reduce the number of false non-matches, the algorithm utilises internally a dictionary of the most common abbreviations by identifying all commonly used abbreviations and excluding them from the total distance for the string pair.

Although the data size did not necessitate using the large storage capability of a cluster, the distributed computation of the data was used to accelerate the computation in the project. In fact, the algorithm was able to compute the distance measures and evaluate the best pair for large number of possible record combinations, representing a cartesian join between the data tables, in reasonable time.

After all possible combinations of the unmatched records were evaluated through the algorithm, the results were sorted on the distance metric and the pair with highest probability of a match was used for record matching. Then, an accumulative threshold level, that is, one

resulting as the sum of the distance metrics for the names and the addresses in the pair, was used to determine the most probable matches. This resulted in approximately 65% matching of the overall datasets. This figure should be considered in the light that a significant number of the entities in each dataset didn't have a corresponding entity of the other. For example, IDBR does not contain partnership firms while FSR records all regulated firms. By varying the threshold measure level for a match, that is, the maximum Levenshtein distance at which a match is accepted, it is possible to vary the sensitivity and specificity of the matching algorithm.

In the project, the optimal threshold distance value was determined empirically by observing the strings pairs around cut-off point. In particular, it was established that the accumulative threshold level of four metric units was providing satisfactory results. Further refinements of the algorithm were left to be investigated as future work. These include:

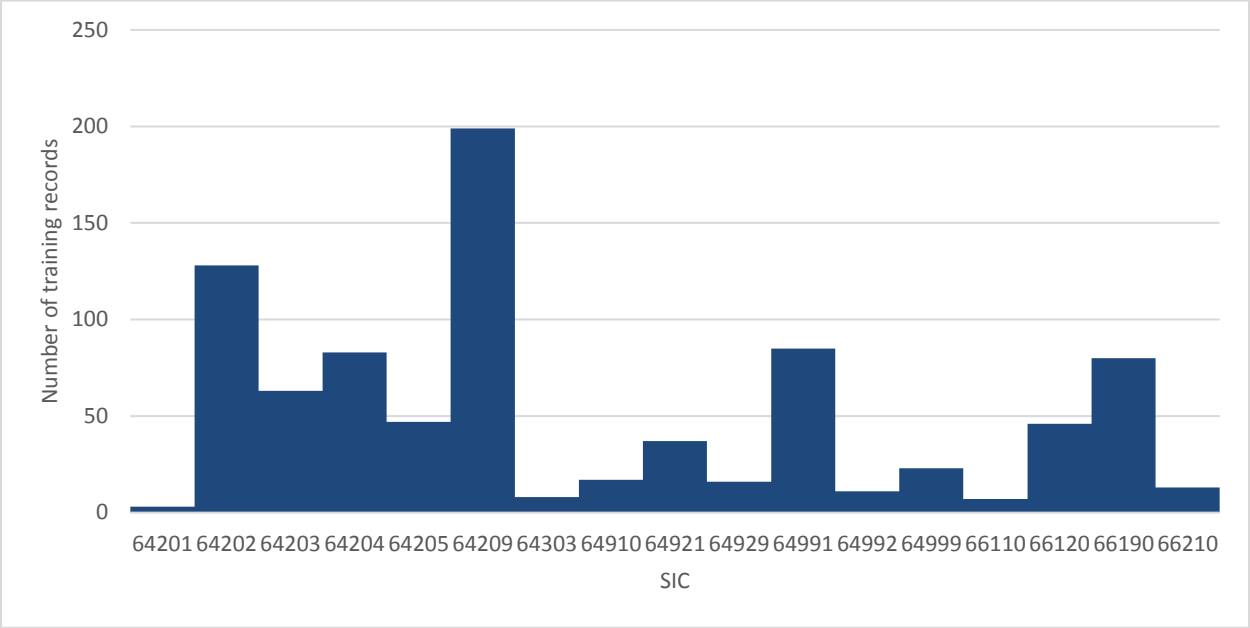
- introducing individual threshold levels for different groups of companies defined by geographic location, SIC code
- using embeddings of the individual words in to strings to capture semantic distances between pairs
- training of an artificial neural network to estimate the posterior probability of a match by considering various factors

Another significant challenge identified at this stage was the high-sparsity level of the training data, resulting from non-responses in the FSS survey dataset. This represented a major detrimental factor, altering the performance of the downstream machine-learning methods by limiting their ability to learn and generalise efficiently.

A further problem identified in the data was the insufficient number of training records, approximately 1,000 resulting from 65% matching of 1,800 FSS records, which given the high number of possible label classes was making over-fitting of the machine model very likely. Finally, the imbalance between the different classes in the training data, observable in Figure 1, necessitated a corrective action in the form of a re-sampling of the dataset. This included both down-sampling of the majority classes and over-sampling of the minority classes. Of course, in practice, the ability to down-sample the majority classes was severely restricted by the limited amount of data. In fact, any further reduction in the training data was making the anticipated over-fitting problem even worse.

Oversampling of the minority class, carried out through duplication of records, was helping to improve the imbalance of the records but it was not benefitting the learning process otherwise as it was not increasing the entropy of data. Therefore, the described re-sampling was not applied aggressively to achieve full balance of the dataset and further data collection efforts are required to achieve this.

**Figure 1: Distribution of training data by Standard Industrial Classification 2007**




In conclusion, the exploratory analysis of the input data was pointing at this stage to a challenging environment for effective application of machine learning.

## 3. Machine-Learning model

Mitigation of the effects of the data quality challenges was a major factor guiding the selection of the machine-learning algorithms. The Random Forest (RF) algorithm, a class from tree-learning ensemble methods, was the first choice as a reasonable balance between the model complexity and its ability to learn from the limited training data, while reducing the decision trees' tendency of overfitting to the training set. Additionally, the XGBoost algorithm, a representative of the gradient-boosted trees (GBT) class of algorithms, was also applied to the data to evaluate its ability to limit the overfitting of the model further.

In general, both RF and GBT are ensemble methods, that is, they build a classifier out of a number of tree-based smaller classifiers and as such they have common characteristics. However, they have some considerable differences.

The main difference is related to the method of training the model. RF trains the decision trees in parallel while the GBT algorithm does this sequentially taking into account the performance of the current performance characteristics of the ensemble. In particular, RF creates a large

number of tree classifiers in parallel, based on bagging, in order to improve the overall prediction accuracy of the ensemble by averaging the results from the individual tree classifiers.

The main principle of bagging is to resample the data multiple times independently and for each sample to train a new classifier to be included in the ensemble. Due to the independency between the separate sampled batches, parallelising the previously described training process is trivial – in fact, an efficient distributed version of RF for Apache Spark exists and has been used in project to accelerate the computation significantly. Combining the trees in RF also reduces the tendency of overfitting the data. Although the individual classifiers have a high tendency of overfitting the data, as each one does it in a different way, using them in an ensemble averages out the overfitting to a certain extent.

In contrast to RF, GBT is a boosting method that builds a series of weak classifiers, again tree-based, but in sequence using a cost function. In theory, the main idea in boosting is that the cost function is optimised over function space by iteratively choosing a function that points in the negative gradient direction. In practice, this is achieved through incrementally increasing the number of classifiers by adding new classifiers trained particularly to improve the currently-trained ensemble based on the pre-defined cost function.

In contrast to the RF, where each training iteration is trained independently from the rest, the parallelisation is not as trivial. However, a parallel implementation of GBT exists in the Apache Spark environment. A comparison between the accuracy of both methods for the dataset was conducted in the project and the results are discussed later.

After selecting the learning methods, further pre-processing measures, aimed at improving the overall performance of the model, were undertaken as described below.

The most important step was the feature selection procedure. Instead of using the full set of features, consisting of 250 individual candidates, the input to the algorithm had to be limited to only a few of the most discriminative ones. As confirmed in experiments, if all available features were fed simultaneously into the model as input, the algorithm was unable to learn to generalise sufficiently well, which resulted in unsatisfactory accuracy on the test dataset – a problem known in machine learning as the “curse of dimensionality”. Therefore, it was considered necessary that both optimisation of the number of input features and discovery of the most suitable feature combinations were needed to improve the accuracy of the model.

After the feature selection, final adjustments of the model were carried out by hyper-parameter tuning in a grid search. This was conducted in a distributed environment of the Apache Spark computing engine to shorten the computation time about from about a minute per feature combination cycle on a single CPU core to less than two seconds on the cluster.

## 4. Feature selection, normalisation and standardisation

As feature selection is one of the most important parts of the machine-learning pipeline, a typical data scientist's toolbox includes several feature selection methods that can be deployed to find the most discriminative features. The possible options include:

- Principal Component Analysis (PCA)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Least Angle Regression (LARS)
- Bayesian model selection or averaging (BSM/A)
- Univariate selection
- recursive feature elimination (RFE)

In order to evaluate their applicability in the particular circumstances of the project, a comparison between a full exhaustive search in three-feature combination space and the available methods for feature selection was carried out. The similarity of the features selected by the above feature selection methods and the results from the exhaustive search highlighted the most suitable feature selection method for the dataset. The exhaustive search was conducted by generating all possible feature combinations for three features and measuring the performance of the RF model that was trained on them.

The length of combinations had to be limited to three features only as the required computational resources for longer feature combinations were prohibitively high even on the cluster environment consisting of 330 central processing unit (CPU) cores. Therefore, a hybrid approach was adopted. It was based on the rationale that after selecting the most appropriate feature-selecting algorithm, that is, that could identify the most similar to the exhaustive search features in the given dataset, then it would be used to generate longer feature combinations that otherwise will be unfeasible to find, in terms of available computational resources, purely through an exhaustive search. After experiments, the most accurate feature selection method for the dataset was found to be the RFE.

Also as part of the pre-processing step, the need for normalisation and standardisation was assessed. Both standardisation, that is, transforming the data to have zero mean and unit variance, and normalisation, that is, scaling all numeric variables in the range [0,1], were found to be unnecessary as the selected machine-learning algorithms are not influenced by un-normalised and non-standardised data.

The missing values in the Financial Services Survey (FSS) were filled with zeros as this was considered to be the most likely assumed answer by the survey respondents. Finally, the data were split into training and test datasets at an 80 to 20 ratio. It was decided that a validation or development dataset will not be used under the circumstances due to the need to preserve the maximum amount of data for training.

## 5. The most discriminative features

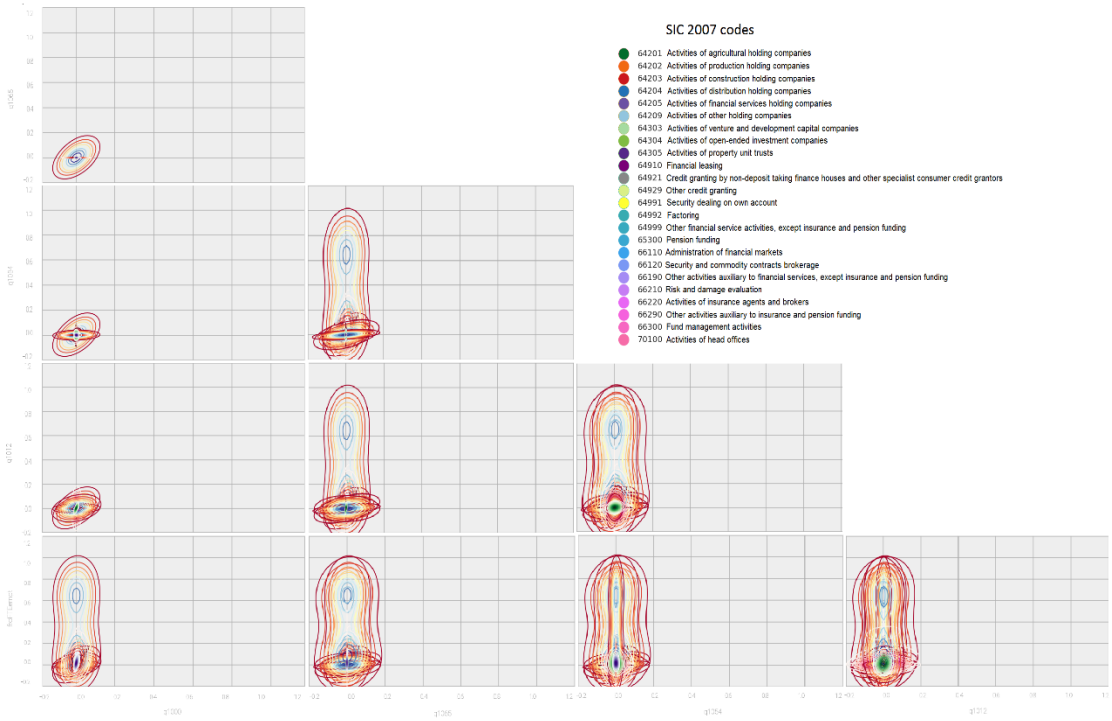
The five most discriminative features identified by the exhaustive search in the feature space were found to be: Q1000, Q1065, Q1054, Q1012 and FTEempt. Their description is provided in Table 1.

### Table 1: Description of the five most discriminative features

| Feature | Description |
|---|---|
| Q1000 | The value of company's holdings of transferable deposits held with banks or building societies located in UK |
| Q1065 | The value of the holdings of listed equity in institutions or businesses located outside of UK |
| Q1054 | The outstanding balance receivable from loans with an original maturity of more than one year from businesses in UK |
| Q1012 | The value of the company's holdings of Treasury Bills issued by Her Majesty's Treasury (HMT) |
| FTEempt | Number of employees, full-time equivalent |

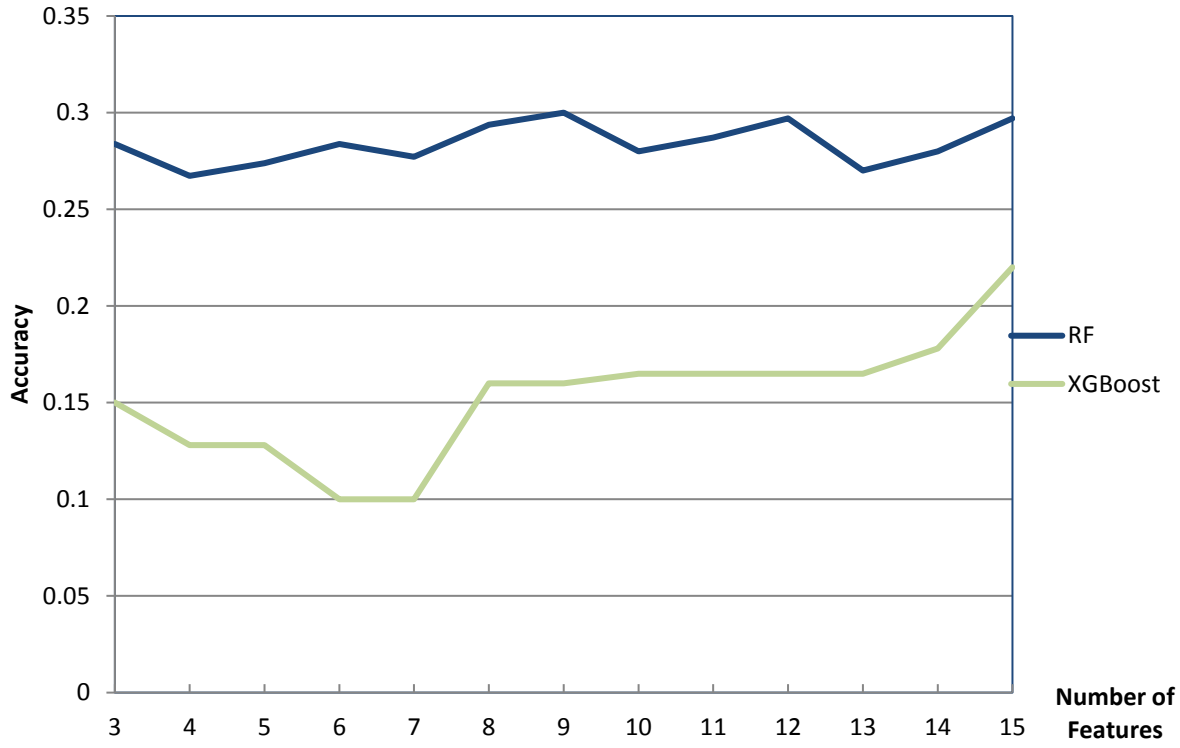
The pair plots of the features in Table 1 are presented graphically in Figure 2 using Kernel Density Estimation (KDE) . In the plots, the label class corresponding to the Standard Industrial Classification 2007: SIC 2007, representing the constant classification code over time of a company, is colour-coded to allow easier visual evaluation of the class members' distribution in each feature pair. When using only two features (in the two-dimensional feature space), the overlapping circles indicate that there is no obvious pair of the features that clearly separate the label classes into well-defined clusters. These features would be depicted by separate circles on individual plots, if present. Therefore, the focus of search had to be directed on higher-dimensional feature spaces.

**Figure 2: Pair plots of the KDE of the most discriminative features**




In addition to discovery of the most discriminative features, several experiments have been conducted to determine the optimal length of the input feature vector. In particular, the list of most discriminative features, as identified by the recursive feature elimination (RFE) feature selection method and the exhaustive feature search, was used to generate the input vectors of higher dimensionality, that is with higher number of features. These input vectors were then evaluated in both Random Forest (RF) and XGBoost algorithms and the results are presented in Figure 3.

**Figure 3: Best multiclass classification accuracy achieved with Random Forest and XGBoost algorithms with different length of the feature vector**

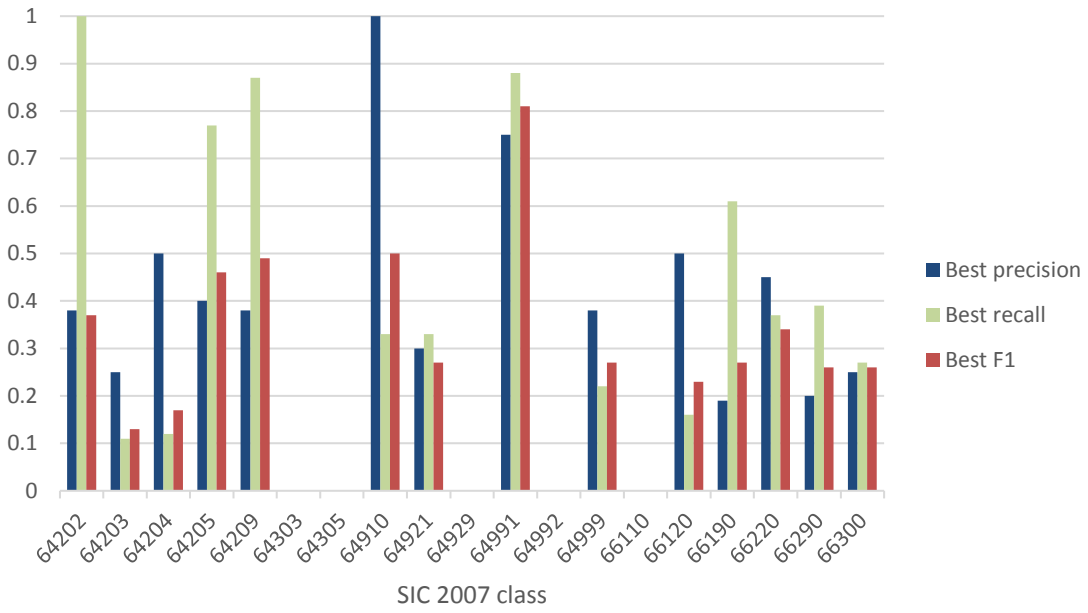



A general tendency can be observed from Figure 3 that under XGBoost the accuracy was increasing continuously with the increase in the length of the input vector. At the same time RF, which has started at a higher accuracy level, was maintaining it approximately constant regardless of the length of the input vector. However, it should be noted that improvement in the accuracy of XGBoost was achieved at the expense of significant hyper-parameter tuning for each input sequence.

This difference of behaviour between the two algorithms could be explained by the presence of internal regularisation mechanisms within XGBoost – simply it was able to optimise its use of the input features better than RF and extract more value of the additional columns. Although RF does not have many parameters to tune in, overall, it resulted in better accuracy in most cases. More detailed results of the experiments are presented in the Appendix A.

It is interesting to see which classes are classified more precisely by the machine learning methods. In Figure 4 the best classification results broken down per SIC 2007 class are shown.

**Figure 4: Best precision, recall and F1 measures achieved per Standard Industrial Classification 2007 class**




It can be observed that by combining different features and hyperparameter settings the performance of the classification algorithm can be optimised individually per SIC 2007 class. Although the SIC 2007 is the prediction target of the classification, there is prior information that can be used to select the best performing set of algorithm, features and hyper parameters. Based on this concept, it is possible for a hybrid classifier to be developed that is able to switch the mode of the underlying algorithm in runtime, in order to optimise the classification performance by using the last known SIC 2007 class for the company.

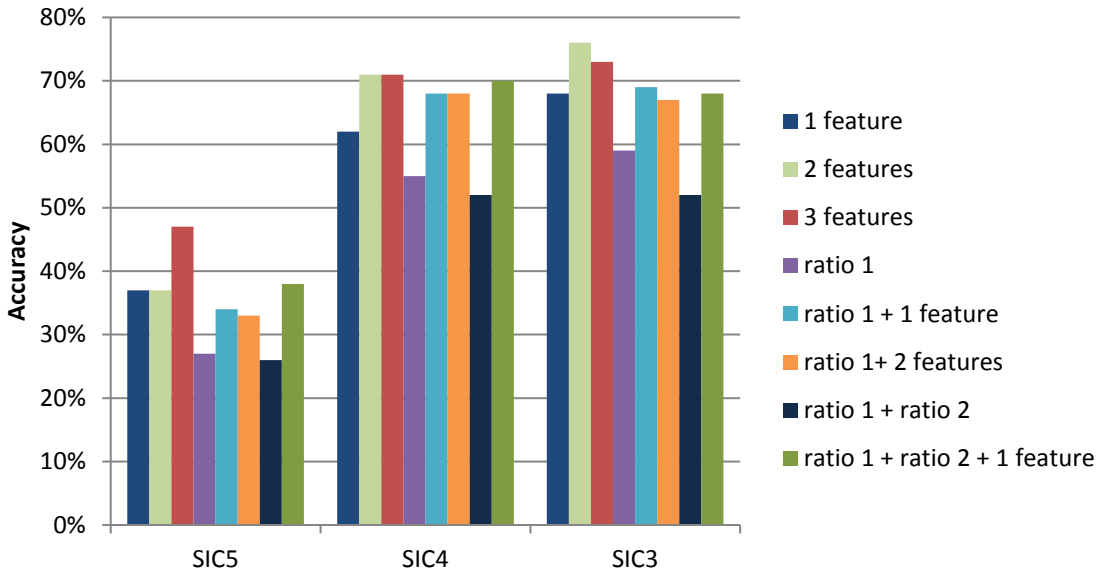
## 6. Results of the exhaustive search

After a large number of experiments based on the exhaustive search, the most discriminative feature configurations and the highest-achieved accuracy for each case was recorded. The results are presented in Figure 5. In particular, the experiments included the following configurations:

- single feature
- a combination of two features
- a combination of three features
- a ratio between two features
- a combination of a ratio of two features and a single feature
- combination of a ratio and two features
- combination of two ratios
- a combination of two ratios and two features

The classification accuracy was evaluated for predicting different levels of aggregation in the Standard Industrial Classification 2007: SIC 2007 structure, given in Appendix B , that is, SIC groups (three digits), SIC classes (four digits) and SIC subclasses (five digits), denoted by SIC3, SIC4 and SIC5 respectively in Figure 5.

**Figure 5: Best multiclass accuracy achieved in the exhaustive feature combination search with the distributed Random Forest classifier, using up to three features or ratio of features and three different levels of SIC 2007 codes granularity**




Part of the experiments followed on from the idea that higher accuracy could be achieved using derivative features that could capture the underlying data patterns in a better way. These experiments were designed to test the accuracy of the derivate features based on ratios. For example, ratios based on relative company indicators, that is, those based on the amount of particular asset or liability reported in survey response per employee, make good economic sense as they allow comparison of businesses with different sizes. However, in tests, using such ratios were not found to lead to increase in the classification accuracy.

The feature combinations that achieved the best accuracy, shown in Figure 5. Full description of the features can be found in the Financial Services Survey: Quarterly Return of Assets and Liabilities documentation.

**Table 2: The best feature combinations from exhaustive search for predicting the sub-class level of Standard Industrial Classification 2007 (SIC5)**

| Feature vector | The feature vector composition that achieves best accuracy for SIC5 |
|---|---|
| 1 feature | Q1000 – value of company's holdings of transferable deposits held with banks or building societies located in the UK |
| 2 features | Q3018 – acquisition cost, capital expenditure on major improvement and construction work<br>FTEempt – number of full-time employees |
| 3 features | Q1000 – value of company's holdings of transferable deposits held with banks or building societies located in the UK<br>Q1012 – value of holding of UK Treasury Bills<br>FTEempt – number of full-time employees |
| ratio 1 | Q1000/Q1080 - value of company's holdings of transferable deposits held with banks or building societies located in the UK or amount recorded in the balance sheet for goods and services that have been provided to customers, but have not yet received payment for |
| ratio 1<br>+ 1 feature | Q1002/Q2006 – balances with banks or building societies held in current accounts outside the UK in sterling or outstanding balance payable from loans with an original maturity of one year or less<br>FTEempt – number of full-time employees |
| ratio 1<br>+ 2 features | Q1002/Q2006 – balances with banks or building societies held in current accounts outside the UK in sterling or outstanding balance payable from loans with an original maturity of one year or less<br>FTEempt – number of full-time employees<br>Q1000 – value of company's holdings of transferable deposits held with banks or building societies located in the UK |
| ratio 1<br>+ ratio 2 | Q1000/Q1001 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or balances with banks or building societies held in current accounts outside the UK in non-sterling<br>Q1004/Q2014 – value of holdings of any other deposits located in the UK in sterling or outstanding balance payable from loans with an original maturity of more than one year to banks or building societies located in the UK in sterling |
| ratio 1<br>+ ratio 2<br>+ 1 feature | Q1000/Q1001 – value of company's holdings of transferable deposits held with banks or building societies located in UK or balances with banks or building societies held in current accounts outside the UK in non-sterling<br>Q1002/Q1007 – balances with banks or building societies held in current accounts outside the UK in sterling or value of holdings of any other deposits located outside of the UK in non-sterling<br>FTEempt – number of full-time employees |

**Table 3: The best feature combinations from exhaustive search for predicting the class level of Standard Industrial Classification 2007 (SIC4)**

| Feature vector | Feature vector composition that achieves best accuracy for SIC4 |
|---|---|
| 1 feature | FTEemp –- number of full-time employees |
| 2 features | Q1003 – balances with banks or building societies held in current accounts outside the UK in non-sterling |

| | |
|---|---|
| | FTEempt – number of full-time employees |
| 3 features | Q1000 – value of company's holdings of transferable deposits held with banks or building societies located in the UK<br>Q1007 – value of your holdings of any other deposits located outside of the UK in non-sterling<br>FTEempt – number of full-time employees |
| ratio 1 | Q1000/Q1005 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or value of holdings of any other deposits located in the UK in non-sterling |
| ratio 1<br>+ 1 feature | Q1000/Q1007 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or value of your holdings of any other deposits located outside of the UK in non-sterling<br>FTEempt – number of full-time employees |
| ratio 1<br>+ 2 features | Q1000/Q1007 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or value of your holdings of any other deposits located outside of the UK in non-sterling<br>Q1003 – balances with banks or building societies held in current accounts outside the UK in non-sterling<br>FTEempt – number of full-time employees |
| ratio 1<br>+ ratio 2 | Q1000/Q1001 – value of company's holdings of transferable deposits held with banks or building societies located in UK or balances with banks or building societies held in current accounts outside in UK in non-sterling<br>Q1005/Q2036 – value of holdings of any other deposits located in the UK in non-sterling |
| ratio 1<br>+ ratio 2<br>+ 1 feature | Q1000/Q1001 – value of company's holdings of transferable deposits held with banks or building societies located in UK or balances with banks or building societies held in current accounts outside in UK in non-sterling<br>Q1002/Q1007 – balances with banks or building societies held in current accounts outside the UK in sterling or value of your holdings of any other deposits located outside of the UK in non-sterling<br>FTEempt – number of full-time employees |

**Table 4: The best feature combinations from exhaustive search for predicting the group level of Standard Industrial Classification 2007 (SIC3)**

| Feature vector | Feature vector composition that achieves best accuracy for SIC3 |
|---|---|
| 1 feature | FTEempt – number of full-time employees |
| 2 features | Q1000 – value of company's holdings of transferable deposits held with banks or building societies located in the UK<br>FTEempt – number of full-time employees |
| 3 features | Q1000 – value of company's holdings of transferable deposits held with banks or building societies located in the UK<br>Q1005 – value of holdings of any other deposits located in the UK in non-sterling |

| | |
|---|---|
| | FTEempt – number of full-time employees |
| ratio 1 | Q1000/Q1001 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or balances with banks or building societies held in current accounts outside in the UK in non-sterling |
| ratio 1<br>+ 1 feature | Q1000/Q1006 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or value of holdings of any other deposits located outside the UK in sterling<br>FTEempt – number of full-time employees |
| ratio 1<br>+ 2 features | Q1001/Q1005 – balances with banks or building societies held in current accounts outside in the UK in non-sterling or value of holdings of any other deposits located in the UK in non-sterling<br>Q1006 – value of holdings of any other deposits located outside the UK in sterling<br>FTEempt – number of full-time employees |
| ratio 1<br>+ ratio 2 | Q1000/Q1001 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or balances with banks or building societies held in current accounts outside in the UK in non-sterling<br>Q1005/Q2036 – value of holdings of any other deposits located outside the UK in sterling/ amount recorded in the balance sheet for goods and services that you have received, but not yet paid for |
| ratio 1<br>+ ratio 2<br>+ 1 feature | Q1000/Q1001 – value of company's holdings of transferable deposits held with banks or building societies located in the UK or balances with banks or building societies held in current accounts outside in the UK in non-sterling<br>Q1002/Q1007 – balances with banks or building societies held<br>in current accounts outside the UK in sterling or value of your holdings of any other deposits located outside of the UK in non-sterling<br>FTEempt – number of full-time employees |

Further work on more complex derivate features is required to explore the feature space exhaustively.

## 7. Conclusions

### Machine-learning aspect

Overall, in contrast to the expectations, the Random Forest (RF) algorithm outperformed the XGBoost algorithm in all experiments in which both algorithms were compared. Perhaps this result could be attributed to an insufficient amount of training data required to train the relatively more complex XGBoost algorithm.

In general, the results achieved in the exhaustive search were higher than those achieved when using feature combinations generated by a feature-selection algorithm. However, these are not fully comparable as the results were arrived at by using two different implementations of RF and it is possible that they are slightly different. For example, the highest-recorded multiclass classification accuracy for the sub-classes level of the Standard Industrial Classification 2007: SIC code was 46% in the exhaustive search and around 30% when using combinations generated by feature-selection methods. It is also

possible that some variations in the achieved maximum accuracy occur due to the random sampling of the train and test split dataset between different runs of the algorithm.

Although, a higher accuracy of around 60% was measured for shorter SIC 2007 codes, that is, the higher group and class levels of the SIC 2007, these results were not considered fully representative due to losing of the class balance further in the class aggregation process. For an approximate evaluation of the results, a random guess of the SIC would have resulted in 5.5% accuracy for the full sub-classes level of SIC and 16.7% and 25% for the group and class levels of SIC respectively.

The proposed hybrid classification algorithm, which is able to switch its mode of operation based on the prior information about the company, that is, previously recorded SIC 2007, can be applied to improve the classification accuracy.

**Applicability aspects**

As observed from detailed results presented in the Appendix A, very good results can be achieved for certain classes of companies. For example, using RF with more than four features, selected among the list of the most discriminative features, enables achieving very accurate classification for the 64910 class, “Financial Leasing”, that is, precision measure up to 100%, or classification without false positives.

Such a high accuracy enables automatic detection of companies of this type from the survey and administrative data. Similarly, for SIC 64202 code, “Holding companies in production sector”, an excellent result for the recall measure, approximate value of 90% to 100%, can be achieved. This result confirms that the algorithm is very sensitive identifying all holding companies in production that are present in the data without producing false negative classifications for this class.

Suggested immediate next steps for improvements include expanding the training dataset either vertically through stringing together of several sequential Financial Services Survey (FSS) periods or horizontally, through addition of more features by joining other relevant datasets. In the longer-term, if a significantly larger training dataset is collected, training of a more complex model like an artificial neural network becomes a feasible option. Also, it is possible to restructure the FSS to include questions that potentially have more discriminative power in classification of the companies. The refinement of FSS could be done gradually by testing the impact of the newly-added questions on the classification accuracy until the most optimal set of question is arrived at.

In conclusion, the current accuracy of classification does not allow an automatic classification process for any arbitrary class of companies but only for a few selected classes. However, the method can be part of a semi-automated anomaly detection system, where the classification of the automatically highlighted companies is later further checked by including them in a more focused survey with direct questions about the nature of their activity. It is expected that the accuracy of the algorithm can be improved by increasing the size of the training dataset or by enhancing the discriminative power of the features. The latter can be achieved by deriving more discriminative features through linking the existing dataset with additional data sources. This could be addressed as part of future work.

## 8. Appendix A

The results of parameter tuning of the XGboost and Random Forest (RF) algorithms with features generated by the feature selection algorithms are presented in the tables in this appendix. It is interesting to observe both how the accuracy increases with the increase of the number of features and the difference in increase between XGBoost and RF algorithms which can be explained with the differences between both algorithms.

## Table 5: XGBoost algorithm with three features

**Features**: Q1000, Q1128, FTEempt

Accuracy:0.151815

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.21 | 0.82 | 0.33 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.02 | 0.02 | 0.02 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.00 | 0.00 | 0.00 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.16 | 0.71 | 0.26 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.00 | 0.00 | 0.00 | 19 |
| 66190 | 0.00 | 0.00 | 0.00 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.00 | 0.00 | 0.00 | 19 |
| 66290 | 0.00 | 0.00 | 0.00 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.04 | 0.15 | 0.06 | 303 |

## Table 6: XGBoost algorithm with four features

**Features**: Q1000, Q1065, Q1054, FTEempt

Accuracy:0.12871

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.23 | 0.94 | 0.36 | 34 |
| 64203 | 0.14 | 0.11 | 0.12 | 18 |
| 64204 | 0.01 | 0.06 | 0.02 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.00 | 0.00 | 0.00 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.00 | 0.00 | 0.00 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.00 | 0.00 | 0.00 | 19 |
| 66190 | 0.09 | 0.13 | 0.11 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.00 | 0.00 | 0.00 | 19 |
| 66290 | 0.14 | 0.04 | 0.07 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.05 | 0.13 | 0.06 | 303 |

## Table 7: XGBoost algorithm with five features

**Features**: Q1000, Q1065, Q1054, Q1012, FTEempt

Accuracy:0.128713

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.23 | 0.94 | 0.36 | 34 |
| 64203 | 0.14 | 0.11 | 0.12 | 18 |
| 64204 | 0.01 | 0.06 | 0.02 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.00 | 0.00 | 0.00 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.00 | 0.00 | 0.00 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.00 | 0.00 | 0.00 | 19 |
| 66190 | 0.09 | 0.13 | 0.11 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.00 | 0.00 | 0.00 | 19 |
| 66290 | 0.14 | 0.04 | 0.07 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.05 | 0.13 | 0.06 | 303 |

## Table 8: XGBoost algorithm with six features

**Features**: Q1000, Q1128, Q1054, Q1012, Q9101, FTEempt

Accuracy:0.108911

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.20 | 0.06 | 0.09 | 34 |
| 64203 | 0.15 | 0.11 | 0.13 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.12 | 0.17 | 0.14 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.09 | 0.88 | 0.16 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.14 | 0.05 | 0.08 | 19 |
| 66190 | 0.18 | 0.09 | 0.12 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.17 | 0.05 | 0.08 | 19 |
| 66290 | 0.33 | 0.13 | 0.19 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.10 | 0.11 | 0.07 | 303 |

## Table 9: XGBoost algorithm with seven features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, FTEempt

Accuracy:0.108911

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.20 | 0.06 | 0.09 | 34 |
| 64203 | 0.15 | 0.11 | 0.13 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.12 | 0.17 | 0.14 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.09 | 0.88 | 0.16 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.14 | 0.05 | 0.08 | 19 |
| 66190 | 0.18 | 0.09 | 0.12 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.17 | 0.05 | 0.08 | 19 |
| 66290 | 0.33 | 0.13 | 0.19 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.10 | 0.11 | 0.07 | 303 |

## Table 10: XGBoost algorithm with eight features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, FTEempt

Accuracy:0.165017

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.21 | 0.94 | 0.34 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.06 | 0.33 | 0.10 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.20 | 0.25 | 0.22 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.16 | 0.16 | 0.16 | 19 |
| 66190 | 0.11 | 0.09 | 0.10 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.33 | 0.16 | 0.21 | 19 |
| 66290 | 0.09 | 0.09 | 0.09 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.09 | 0.17 | 0.09 | 303 |

## Table 11: XGBoost algorithm with nine features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1052, Q1072, Q1062, FTEempt

Accuracy:0.165017

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.21 | 0.94 | 0.34 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.06 | 0.33 | 0.10 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.20 | 0.25 | 0.22 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.16 | 0.16 | 0.16 | 19 |
| 66190 | 0.11 | 0.09 | 0.10 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.33 | 0.16 | 0.21 | 19 |
| 66290 | 0.09 | 0.09 | 0.09 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.09 | 0.17 | 0.09 | 303 |

## Table 12: XGBoost algorithm with ten features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1052, Q1072, Q1062, Q1040, FTEempt

Accuracy:0.165017

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.21 | 0.94 | 0.34 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.06 | 0.33 | 0.10 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.20 | 0.25 | 0.22 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.16 | 0.16 | 0.16 | 19 |
| 66190 | 0.11 | 0.09 | 0.10 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.33 | 0.16 | 0.21 | 19 |
| 66290 | 0.09 | 0.09 | 0.09 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.09 | 0.17 | 0.09 | 303 |

## Table 13: XGBoost algorithm with eleven features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1052, Q1072, Q1062, Q1040, Q1045, FTEempt

Accuracy:0.165017

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.21 | 0.94 | 0.34 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.06 | 0.33 | 0.10 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.20 | 0.25 | 0.22 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.16 | 0.16 | 0.16 | 19 |
| 66190 | 0.11 | 0.09 | 0.10 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.33 | 0.16 | 0.21 | 19 |
| 66290 | 0.09 | 0.09 | 0.09 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.09 | 0.17 | 0.09 | 303 |

## Table 14: XGBoost algorithm with twelve features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1052, Q1072, Q1062, Q1040, Q1045, Q1048, FTEempt

Accuracy:0.165017

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.21 | 0.94 | 0.34 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.06 | 0.33 | 0.10 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.20 | 0.25 | 0.22 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.16 | 0.16 | 0.16 | 19 |
| 66190 | 0.11 | 0.09 | 0.10 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.33 | 0.16 | 0.21 | 19 |
| 66290 | 0.09 | 0.09 | 0.09 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.09 | 0.17 | 0.09 | 303 |

## Table 15: XGBoost algorithm with thirteen features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1052, Q1072, Q1062, Q1040, Q1045, Q1048, Q1071, FTEempt

Accuracy:0.165017

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.21 | 0.94 | 0.34 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.06 | 0.33 | 0.10 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.20 | 0.25 | 0.22 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.16 | 0.16 | 0.16 | 19 |
| 66190 | 0.11 | 0.09 | 0.10 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.33 | 0.16 | 0.21 | 19 |
| 66290 | 0.09 | 0.09 | 0.09 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.09 | 0.17 | 0.09 | 303 |

## Table 16: XGBoost algorithm with fourteen features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1052, Q1072, Q1062, Q1040, Q1045, Q1048, Q1071, Q1041, FTEempt

Accuracy:0.178218

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.20 | 1.00 | 0.33 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.00 | 0.00 | 0.00 | 6 |
| 64921 | 0.20 | 0.33 | 0.25 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.00 | 0.00 | 0.00 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.00 | 0.00 | 0.00 | 19 |
| 66190 | 0.00 | 0.00 | 0.00 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.15 | 0.16 | 0.15 | 19 |
| 66290 | 0.12 | 0.39 | 0.18 | 23 |
| 66300 | 0.25 | 0.27 | 0.26 | 15 |
| avg / total | 0.06 | 0.18 | 0.08 | 303 |

## Table 16: XGBoost algorithm with fifteen features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1052, Q1072, Q1062, Q1040, Q1045, Q1048, Q1071, Q1041, Q1055, FTEempt

Accuracy:0.221122

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.23 | 0.91 | 0.37 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.00 | 0.00 | 0.00 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.00 | 0.00 | 0.00 | 6 |
| 64921 | 0.00 | 0.00 | 0.00 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.29 | 0.92 | 0.44 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.00 | 0.00 | 0.00 | 9 |
| 66120 | 0.00 | 0.00 | 0.00 | 19 |
| 66190 | 0.15 | 0.61 | 0.25 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.00 | 0.00 | 0.00 | 19 |
| 66290 | 0.00 | 0.00 | 0.00 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.06 | 0.22 | 0.10 | 303 |

## Table 17: RF algorithm with three features

**Features**: Q1000, Q1128, FTEemp

Accuracy:0.283828

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.38 | 0.09 | 0.14 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.25 | 0.12 | 0.17 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.34 | 0.87 | 0.49 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.00 | 0.00 | 0.00 | 6 |
| 64921 | 0.17 | 0.17 | 0.17 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.50 | 0.54 | 0.52 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.33 | 0.22 | 0.27 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.00 | 0.00 | 0.00 | 19 |
| 66190 | 0.14 | 0.35 | 0.20 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.32 | 0.37 | 0.34 | 19 |
| 66290 | 0.18 | 0.09 | 0.12 | 23 |
| 66300 | 0.20 | 0.07 | 0.10 | 15 |
| avg / total | 0.23 | 0.28 | 0.22 | 303 |

## Table 18: RF algorithm with four features

**Features**: Q1000, Q10658, Q1054, FTEemp

Accuracy:0.267327

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.29 | 0.12 | 0.17 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.32 | 0.77 | 0.45 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.67 | 0.33 | 0.44 | 6 |
| 64921 | 0.14 | 0.17 | 0.15 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.45 | 0.54 | 0.49 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.17 | 0.11 | 0.13 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.00 | 0.00 | 0.00 | 19 |
| 66190 | 0.17 | 0.35 | 0.23 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.36 | 0.21 | 0.27 | 19 |
| 66290 | 0.17 | 0.17 | 0.17 | 23 |
| 66300 | 0.25 | 0.13 | 0.17 | 15 |
| avg / total: | 0.21 | 0.27 | 0.21 | 303 |

## Table 19: RF algorithm with five features

**Features**: Q1000, Q1065, Q1054, Q1012, FTEemp

Accuracy:0.273927

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.29 | 0.12 | 0.17 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.33 | 0.77 | 0.47 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.17 | 0.29 | 6 |
| 64921 | 0.08 | 0.08 | 0.08 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.48 | 0.62 | 0.55 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66120 | 0.43 | 0.16 | 0.23 | 19 |
| 66190 | 0.15 | 0.26 | 0.19 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.33 | 0.26 | 0.29 | 19 |
| 66290 | 0.20 | 0.22 | 0.21 | 23 |
| 66300 | 0.11 | 0.07 | 0.08 | 15 |
| avg / total | 0.24 | 0.27 | 0.22 | 303 |

## Table 20: RF algorithm with six features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, FTEemp

Accuracy:0.283828

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.29 | 0.12 | 0.17 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.03 | 0.06 | 0.04 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.35 | 0.74 | 0.48 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.33 | 0.50 | 6 |
| 64921 | 0.19 | 0.25 | 0.21 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.69 | 0.75 | 0.72 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.50 | 0.05 | 0.10 | 19 |
| 66190 | 0.19 | 0.43 | 0.26 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.00 | 0.00 | 0.00 | 19 |
| 66290 | 0.20 | 0.30 | 0.24 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.25 | 0.28 | 0.23 | 303 |

## Table 21: RF algorithm with seven features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, FTEemp

Accuracy:0.277228

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.25 | 0.12 | 0.16 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.05 | 0.06 | 0.05 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.36 | 0.74 | 0.48 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.67 | 0.33 | 0.44 | 6 |
| 64921 | 0.12 | 0.08 | 0.10 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.69 | 0.75 | 0.72 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.11 | 0.05 | 0.07 | 19 |
| 66190 | 0.15 | 0.39 | 0.22 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.20 | 0.05 | 0.08 | 19 |
| 66290 | 0.23 | 0.30 | 0.26 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.22 | 0.28 | 0.23 | 303 |

## Table 22: RF algorithm with eight features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, FTEemp

Accuracy:0.293729

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.25 | 0.12 | 0.16 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.35 | 0.81 | 0.49 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.33 | 0.50 | 6 |
| 64921 | 0.18 | 0.17 | 0.17 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.70 | 0.67 | 0.68 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.11 | 0.05 | 0.07 | 19 |
| 66190 | 0.15 | 0.39 | 0.21 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.45 | 0.26 | 0.33 | 19 |
| 66290 | 0.22 | 0.22 | 0.22 | 23 |
| 66300 | 0.17 | 0.07 | 0.10 | 15 |
| avg / total: | 0.25 | 0.29 | 0.24 | 303 |

## Table 23: RF algorithm with nine features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, Q1062, FTEemp

Accuracy:0.300330

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.31 | 0.15 | 0.20 | 34 |
| 64203 | 0.20 | 0.06 | 0.09 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.50 | 0.05 | 0.10 | 19 |
| 64209 | 0.35 | 0.83 | 0.50 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.33 | 0.50 | 6 |
| 64921 | 0.22 | 0.17 | 0.19 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.65 | 0.71 | 0.68 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.20 | 0.11 | 0.14 | 9 |
| 66120 | 0.50 | 0.05 | 0.10 | 19 |
| 66190 | 0.15 | 0.43 | 0.22 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.30 | 0.16 | 0.21 | 19 |
| 66290 | 0.21 | 0.17 | 0.19 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.30 | 0.30 | 0.25 | 303 |

## Table 23: RF algorithm with ten features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, Q1062, Q1040, FTEemp

Accuracy:0.283828

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.26 | 0.15 | 0.19 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.34 | 0.81 | 0.48 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.67 | 0.33 | 0.44 | 6 |
| 64921 | 0.12 | 0.08 | 0.10 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.62 | 0.62 | 0.62 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66120 | 0.25 | 0.05 | 0.09 | 19 |
| 66190 | 0.16 | 0.43 | 0.23 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.25 | 0.05 | 0.09 | 19 |
| 66290 | 0.22 | 0.30 | 0.25 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.22 | 0.28 | 0.22 | 303 |

## Table 24: RF algorithm with eleven features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, Q1062, Q1040, Q1045, FTEemp

Accuracy:0.287129

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.29 | 0.12 | 0.17 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.00 | 0.00 | 0.00 | 19 |
| 64209 | 0.33 | 0.77 | 0.46 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.33 | 0.50 | 6 |
| 64921 | 0.13 | 0.17 | 0.15 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.78 | 0.75 | 0.77 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66120 | 0.50 | 0.05 | 0.10 | 19 |
| 66190 | 0.18 | 0.48 | 0.27 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.00 | 0.00 | 0.00 | 19 |
| 66290 | 0.19 | 0.30 | 0.23 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.24 | 0.29 | 0.22 | 303 |

## Table 25: RF algorithm with twelve features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, Q1062, Q1040, Q1045, Q1048, FTEemp

Accuracy:0.297030

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.23 | 0.15 | 0.18 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.03 | 0.06 | 0.04 | 16 |
| 64205 | 0.33 | 0.05 | 0.09 | 19 |
| 64209 | 0.38 | 0.72 | 0.49 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.33 | 0.50 | 6 |
| 64921 | 0.30 | 0.25 | 0.27 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.69 | 0.75 | 0.72 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.17 | 0.11 | 0.13 | 9 |
| 66120 | 0.50 | 0.05 | 0.10 | 19 |
| 66190 | 0.16 | 0.43 | 0.24 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.44 | 0.21 | 0.29 | 19 |
| 66290 | 0.22 | 0.26 | 0.24 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.29 | 0.30 | 0.26 | 303 |

## Table 26: RF algorithm with thirteen features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, Q1062, Q1040, Q1045, Q1048, Q1071, FTEemp

Accuracy:0.273927

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.22 | 0.12 | 0.15 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.04 | 0.06 | 0.05 | 16 |
| 64205 | 0.25 | 0.05 | 0.09 | 19 |
| 64209 | 0.34 | 0.66 | 0.45 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 0.67 | 0.33 | 0.44 | 6 |
| 64921 | 0.08 | 0.08 | 0.08 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.75 | 0.88 | 0.81 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.50 | 0.05 | 0.10 | 19 |
| 66190 | 0.15 | 0.43 | 0.22 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.25 | 0.05 | 0.09 | 19 |
| 66290 | 0.19 | 0.22 | 0.20 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.26 | 0.27 | 0.23 | 303 |

## Table 27: RF algorithm with fourteen features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, Q1062, Q1040, Q1045, Q1048, Q1071, Q1041, FTEemp

Accuracy:0.287129

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.36 | 0.15 | 0.21 | 34 |
| 64203 | 0.00 | 0.00 | 0.00 | 18 |
| 64204 | 0.04 | 0.06 | 0.05 | 16 |
| 64205 | 0.40 | 0.11 | 0.17 | 19 |
| 64209 | 0.36 | 0.70 | 0.47 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.33 | 0.50 | 6 |
| 64921 | 0.17 | 0.08 | 0.11 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.74 | 0.83 | 0.78 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.11 | 0.15 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.11 | 0.05 | 0.07 | 19 |
| 66190 | 0.16 | 0.39 | 0.22 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.17 | 0.05 | 0.08 | 19 |
| 66290 | 0.21 | 0.30 | 0.25 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total | 0.27 | 0.29 | 0.25 | 303 |

## Table 28: RF algorithm with fifteen features

**Features**: Q1000, Q1065, Q1054, Q1012, Q9101, Q1072, Q1052, Q1062, Q1040, Q1045, Q1048, Q1071, Q1041, Q1055, FTEemp

Accuracy:0.297030

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 64202 | 0.25 | 0.12 | 0.16 | 34 |
| 64203 | 0.25 | 0.06 | 0.09 | 18 |
| 64204 | 0.00 | 0.00 | 0.00 | 16 |
| 64205 | 0.33 | 0.05 | 0.09 | 19 |
| 64209 | 0.33 | 0.77 | 0.46 | 53 |
| 64303 | 0.00 | 0.00 | 0.00 | 2 |
| 64305 | 0.00 | 0.00 | 0.00 | 1 |
| 64910 | 1.00 | 0.33 | 0.50 | 6 |
| 64921 | 0.29 | 0.17 | 0.21 | 12 |
| 64929 | 0.00 | 0.00 | 0.00 | 4 |
| 64991 | 0.64 | 0.75 | 0.69 | 24 |
| 64992 | 0.00 | 0.00 | 0.00 | 2 |
| 64999 | 0.25 | 0.22 | 0.24 | 9 |
| 66110 | 0.00 | 0.00 | 0.00 | 0 |
| 66120 | 0.50 | 0.05 | 0.10 | 19 |
| 66190 | 0.18 | 0.35 | 0.24 | 23 |
| 66210 | 0.00 | 0.00 | 0.00 | 4 |
| 66220 | 0.14 | 0.05 | 0.08 | 19 |
| 66290 | 0.20 | 0.39 | 0.26 | 23 |
| 66300 | 0.00 | 0.00 | 0.00 | 15 |
| avg / total: | 0.28 | 0.30 | 0.24 | 303 |

## 9. Appendix B: Standard Industrial Classification 2007 structure for financial and insurance activities

### Financial service activities, except insurance and pension funding

**64.1 Monetary intermediation**

- 64.11 Central banking
- 64.19 Other monetary intermediation
  - 64.19/1 Banks
  - 64.19/2 Building societies

**64.2 Activities of holding companies**

- 64.20 Activities of holding companies
  - 64.20/1 Activities of agricultural holding companies
  - 64.20/2 Activities of production holding companies
  - 64.20/3 Activities of construction holding companies
  - 64.20/4 Activities of distribution holding companies
  - 64.20/5 Activities of financial services holding companies
  - 64.20/9 Activities of other holding companies (not including agricultural, production, construction, distribution and financial services holding companies)

**64.3 Trusts, funds and similar financial entities**

- 64.30 Trusts, funds and similar financial entities
  - 64.30/1 Activities of investment trusts
  - 64.30/2 Activities of unit trusts
  - 64.30/3 Activities of venture and development capital companies
  - 64.30/4 Activities of open-ended investment companies
  - 64.30/5 Activities of property unit trusts
  - 64.30/6 Activities of real estate investment trusts

**64.9 Other financial service activities, except insurance and pension funding**

- 64.91 Financial leasing
- 64.92 Other credit granting
  - 64.92/1 Credit granting by non-deposit taking finance houses and other specialist consumer credit grantors
  - 64.92/2 Activities of mortgage finance companies
  - 64.92/9 Other credit granting (not including credit granting by non-deposit taking finance houses and other specialist consumer credit grantors and activities of mortgage finance companies).
- 64.99 Other financial service activities, except insurance and pension funding.
  - 64.99/1 Security dealing on own account
  - 64.99/2 Factoring
  - 64.99/9 Other financial service activities, except insurance and pension funding, (not including security dealing on own account and factoring).

| | | | | |
|---|---|---|---|---|
| **65** | **Insurance, reinsurance and pension funding, except compulsory social security** | | | |
| | **65.1** | **Insurance** | | |
| | | 65.11 | Life insurance | |
| | | 65.12 | Non-life insurance | |
| | **65.2** | **Reinsurance** | | |
| | | 65.20 | Reinsurance | |
| | | | 65.20/1 | Life reinsurance |
| | | | 65.20/2 | Non-life reinsurance |
| | **65.3** | **Pension funding** | | |
| | | 65.30 | Pension funding | |