Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

The US has in recent years experienced a shortage of truck drivers. This has resulted in a high turnover of professional truck drivers for trucking companies (carriers) and an ever increasing need to recruit new drivers. Youcruit is a Swedish based company that aims to use technology to simplify and streamline the recruitment process for both drivers and carriers.

A recruitment process generally has four distinct phases - attracting, shortlisting, selecting and finally the appointing of suitable candidates to positions. Youcruit's business model is primarily concerned with the first two phases - attracting and shortlisting candidates. Carriers appoint Youcruit to attract potential drivers, these candidates are then screened by Youcruit to insure they fulfill that particular carriers requirements. Youcruit then shortlists the most high-caliber of these candidates and it is these shortlisted candidates that Youcruit recommend to the carrier. These candidates then undergo the carriers selection process usually in the form of an interview and/or orientation. The candidate is then rejected by the carrier if they are deemed unsuitable or offered a position if the are deemed qualified and suitable.

Machine learning is a class of algorithm that take input data and use statistical analysis to predict outcomes or classify observations without needing to be explicitly programmed. Supervised classification algorithms are a subcategory of machine learning algorithm used to create models which can successfully predict an output variable as a category or group based on past observations.

1.2 Problem Description

It is important to Youcruit that only quality candidates, and the candidates most likely to receive an offer of employment get shortlisted to go through the carriers selection process. It is this second phase of the recruitment process, the shortlisting of potential candidates, that will be the focus of this thesis.

Data has being gathered by Youcruit through an initial candidate screening process as well as responses from carriers regarding the selection outcome for each candidate. The topic of this thesis is to apply and evaluate machine learning algorithms to this

data in order to predict which candidates are of the highest caliber and thereby most likely to receive an offer of employment from the carrier.

As will be discussed there were four fundamental challenges in attaining a model which could accurately classify candidates. The first relates to the structure of the data obtained through the screening process. The second regards an imbalance in the response data. Thirdly feature engineering and selection in order to find variables that were suitably correlated with the outcome and thereby having good prediction qualities. The forth issue relates to model evaluation and determining which model is "best".

2

Theory

2.1 Machine Learning Classifiers

2.1.1 Classification Trees

Tree based algorithms are commonly used in data mining. Classification tree algorithms are a subcategory of tree based algorithms that aim to create models that can predict the class of a target variable based on several input variables. A classification tree is built through an iterative process of splitting the data into partitions.

Consider a set of training data $\{(x_i, y_i), ..., (x_N, y_N)\}$, where N is the number observations, and p is the number of features $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$. We wish to partition the feature space into M regions $\{R_m\}_1^M$ (commonly termed terminal nodes or leaves) and model the response as a constant $k(m)$ in each region:

$$f(x) = \sum_{m=1}^{M} k(m) 1\{x \in R_m\}, \tag{2.1.11}$$

where $k(m)$ for $k = 0, 1, ...K$ is the class label for the response variable. The estimate of the proportion of class k observations in each region R_m is given by

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} 1\{y_i = k\}, \tag{2.1.12}$$

where N_m is the number of observations in region R_m.

The observations in region R_m can now be defined as the majority class

$$k(m) = \underset{k}{\operatorname{argmax}}(\hat{p}_{mk}). \tag{2.1.13}$$

In order to determine where each branch of the tree should split we need the notation of node impurity i.e how homogeneous the category labels are in each region. A region in which most of the labels are similar would be considered more pure than a region with many dissimilar category labels. The following three measures of node impurity can be used:

- Misclassification error:

$$\frac{1}{N_m} \sum_{x_i \in R_m} 1\{y_i \neq k(m)\} = 1 - \hat{p}_{mk(m)}. \tag{2.1.14}$$

- Gini index:

$$\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}). \tag{2.1.15}$$

- Cross-entropy or deviance

$$-\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}. \tag{2.1.16}$$

Ideally the node impurity would be minimized for each split in order to find the optimal binary partition. This however is not computationally feasible so a 'greedy' algorithm approach is taken. The tree is initially grown greedily until it reaches a certain predetermined (large) size. This is done as a small tree risk missing important elements in the structure of the data.

Splitting the feature space into two half-regions we obtain

$$R_1(j, s) = \{X | X_j \leq s\} \quad \text{and} \quad R_2(j, s) = \{X | X_j > s\}, \tag{2.1.17}$$

where j is the split variable and s the split point that solve the minimization

$$\min_{j,s} \left[\min_{k(1)} IM(1) + \min_{k(2)} IM(2) \right], \tag{2.1.18}$$

and $IM(m)$ defined as the impurity measure at region m.

Thus in the case of a binary classification with node impurity measured by the misclassification error given in equation (2.1.14), (2.1.18) can be expanded to become

$$\min_{j,s}\left[\min_{k(1)}(1-\hat{p}_{1k}(1))+\min_{k(2)}(1-\hat{p}_{2k}(2))\right]$$
$$=\min_{j,s}\left[\min_{k(1)}\left(1-\frac{1}{N_1}\sum_{x_i\in R_1(j,s)}1\{y_i\neq k(1)\right)\right. \tag{2.1.19}$$
$$\left.+\min_{k(2)}\left(1-\frac{1}{N_2}\sum_{x_i\in R_2(j,s)}1\{y_i\neq k(2)\right)\right].$$

The splitting process is repeated until the tree has reached a predetermined size. Inherent in this greedy large tree approach is the danger of over-fitting and as such it may be necessary to "prune" the tree. This pruning can be done through the use of a cost complexity criterion where the algorithm successively collapses branches of the tree in order to minimize a given cost complexity function and produce a sub-tree with the lowest node impurity for each leaf. As bootstrap aggregation was used in this project to prevent over-fitting instead of pruning further discussion on the topic of pruning will be omitted. [Friedman, 2009 (1)]

2.1.2 Bootstrap Aggregation

Bootstrapping is a method that relies on random sampling with replacement to access statistical accuracy. Bootstrap aggregating, commonly referred to as bagging, is a machine learning ensemble meta-algorithm used to improve the prediction accuracy of machine learning algorithms. Bagging is particularly useful in the context of classification trees where it can be applied to reduce variance and avoid over-fitting.

Consider a set of training data $Z = \{z_1, z_2, ...z_N\}$ where $z_i = (x_i, y_i)$. Draw B bootstrap samples of the same size from Z, randomly and with replacement. A model is then fit to each bootstrap sample Z^{*b}, $b = 1, 2, ...B$ to obtain the the prediction $\hat{f}^{*b}(x)$ for a given input x.

When implementing bootstrap aggregation with classification trees, each tree will be build using a different bootstrap sample an as such may have differing feature splits and number of terminal nodes. Subsequently the predictions from each tree can be expected to vary. The bagged classifier then selects the class that received the most predictions from the B bootstrap trees. [Friedman, 2009 (1)]

2.1.3 Support Vector Classifier

The aim of the support vector algorithm is to construct an optimal linear decision boundary (hyperplane) that splits a given data set into different categorical classes. In the case of two-class classification one class will be given a positive label while the other class a negative label. A data point will consequently be categorize as positive or negative depending on weather or it lies above or below the hyperplane.

Beginning with linear classification, the following model is used to classify new data points

$$y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b, \tag{2.1.31}$$

where b, the bias parameter is a real number, $\mathbf{x}$ is the input vector and $\phi(\mathbf{x})$ denotes a fixed feature space transformation. New data points are accordingly classified depending on the sign of $y(\mathbf{x})$. The target values $t_1, ..., t_N$ corresponding to the N training vectors $x_1, ..., x_N$ and are labelled with output values $t_n = \{-1, +1\}$.

Consider a hyperplane that linearly separates two classes of data. It is desirable that hyperplane separate the data with the largest possible margin. I.e the hyperplane is required to maximize the perpendicular distance between the closest data points of contradictory classes. These points are termed support vectors and the distance between them the margin.

As the distance from a hyperplane defined as $y(\mathbf{x}) = 0$ to the point $\mathbf{x}_n$ is given by $y(\mathbf{x}_n)/||\mathbf{w}||$, it follows that the distance from the decision surface to the point $\mathbf{x}_n$ is given by

$$\frac{y(\mathbf{x}_n)}{||\mathbf{w}||} = \frac{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)}{||\mathbf{w}||}. \tag{2.1.32}$$

The margin is thus maximized by optimizing the parameters $\mathbf{w}$ and b at the point of minimum perpendicular distance to the decision surface:

$$\underset{\mathbf{w},b}{\text{argmax}} \left\{ \frac{1}{||\mathbf{w}||} \min_n \left[t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \right] \right\}. \tag{2.1.33}$$

The factor $||\mathbf{w}||^{-1}$ is taken outside of the optimization as $\mathbf{w}$ is not dependent on n.

Finding a direct solution to this optimization problem is complex. By rescaling $\mathbf{w}$ and b by a constant the distance does not change but it accords the freedom to set

$$t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) = 1, \tag{2.1.34}$$

for the point that is closest to the decision boundary. In this case, all data points will satisfy the constraint

$$t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \geq 1, \quad n = 1, ...N. \tag{2.1.35}$$

As maximizing $||\mathbf{w}||^{-1}$ is equivalent to minimizing $||\mathbf{w}||^2$ the optimization problem can be restated as the simpler convex optimization problem

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}||\mathbf{w}||^2, \tag{2.1.36}$$

subject to the constraints given in inequality (2.1.35). It is worth noting that while the parameter b is not explicit in the optimization it is implicit in the constraints, as changes in $||\mathbf{w}||$ will automatically be counteract by changes in b.

Employing the method of Lagrange multipliers with $\mathbf{a} = (a_1, ..., a_N)^T$ and $a_n \geq 0$, the optimization problem can be solved through the Lagrange function:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{n=1}^{N} a_n[t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - 1]. \tag{2.1.37}$$

Setting the derivatives of (2.1.37) with respect to $\mathbf{w}$ and b equal to 0, the following conditions are obtained

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{2.1.38}$$

$$0 = \sum_{n=1}^{N} a_n t_n. \tag{2.1.39}$$

Substituting equations (2.1.38) (2.1.39) into equation (2.1.37) the so-call *dual representation* is obtained

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m), \tag{2.1.310}$$

constrained by

$$a_n \geq 0, \quad n = 1, ...N \tag{2.1.311}$$

$$\sum_{n=1}^{N} a_n t_n = 0, \tag{2.1.312}$$

where $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ defines the kernel function.

The kernel function maps the data into a higher dimension feature space ensuring that the function $\widetilde{L}(\mathbf{a})$ is positive definite, bounded below and thereby ensuring a well defined optimization problem. In addition through the use of the kernel function, data that is not linearly separable in the data space becomes linearly separable in the nonlinear feature space defined implicitly by the nonlinear kernel function.

In order to classify new data points $y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$ can be expressed in terms of the kernel function and parameters $\{a_n\}$ by substituting $\mathbf{w}$ with the expression in (2.1.38):

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b. \tag{2.1.313}$$

A constrained optimization of this form satisfies the Karush-Kuhn-Tucker conditions meaning that

$$a_n \geq 0 \tag{2.1.314}$$

$$t_n y(\mathbf{x}_n) - 1 \geq 0 \tag{2.1.315}$$

$$a_n \{t_n y(\mathbf{x}_n) - 1\} = 0. \tag{2.1.316}$$

Only the points in the training data that fulfill $t_n y(\mathbf{x}_n) = 1$ are included in the summation in equation (2.1.313) i.e. only the points that lie on the maximum margin hyperplanes are included. It is these points that are termed the support vectors. The data points for which $a_n = 0$, will not appear in the summation and therefor not play any role in classifying new data. This gives a property central to usefulness of SVM, namely only the support vectors are retained after training the model thereby reducing the dimensionality of the problem.

Using the fact that any support vector $\mathbf{x}_n$ satisfies $t_n y(\mathbf{x}_n) = 1$, the value of the parameter b can be obtained using equation (2.1.313) to give

$$t_n\left(\sum_{m\in S} a_m t_m k(\mathbf{x}_n,\mathbf{x}_m)+b\right)=1, \tag{2.1.317}$$

where S denotes the indices of the support vectors. Using the fact that $t_n^2=1$, both sides of (2.1.317) can be multiplied by t_n to solve for b:

$$b=\frac{1}{N_S}\sum_{n\in S}\left(t_n-\sum_{m\in S} a_m t_m k(\mathbf{x}_n,\mathbf{x}_m)+b\right). \tag{2.1.318}$$

where N_S is the total number of support vectors.

When considering two-class non-linear classification where there exists overlap between class distributions it becomes necessary to allow for some misclassification of data. This may be done through the use of *slack variables*, $\xi_n \geq 0$, $n=1,...,N$ where one slack variable is assigned to each point in the training data. The constraints in (2.1.35) can then be replaced with

$$t_n y(\mathbf{x}_n)\geq 1-\xi_n. \tag{2.1.319}$$

This ensures that for the data points for which the slack variable equals zero, $\xi_n=0$, the classification will be correctly. While points for which $0<\xi_n\leq 1$, lie inside the margin and on the correct side of the decision boundary. Point for which $\xi_n>1$, lie on the wrong side of the decision boundary and will be will be misclassified.

Thus it becomes necessary to minimize

$$C\sum_{n=1}^{N}\xi_n+\frac{1}{2}||\mathbf{w}||^2. \tag{2.1.320}$$

As any point where $\xi_n>1$ will be misclassified, the parameter $C>0$ can be thought of governing the trade-off between the slack variable penalty and the margin.

Minimizing (2.1.320) subject to $\xi_n\leq 0$ and the constraints (2.1.319) gives the Lagrange function

$$L(\mathbf{w},b,\mathbf{a})=\frac{1}{2}||\mathbf{w}||^2+C\sum_{n=1}^{N}\xi_n-\sum_{n=1}^{N}a_n\{t_n(y(\mathbf{x}_n)-1+\xi_n)\}-\sum_{n=1}^{N}\mu_n\xi_n, \tag{2.1.321}$$

where $\{a_n \leq 0\}$ and $\{\mu_n \leq 0\}$ are the Lagrange multipliers and the Karush-Kuhn-Tucker conditions are

$$a_n \geq 0 \tag{2.1.322}$$

$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0 \tag{2.1.323}$$

$$a_n\{t_n y(\mathbf{x}_n) - 1 + \xi_n\} = 0. \tag{2.1.324}$$

$$\mu_n \geq 0 \tag{2.1.325}$$

$$\xi_n \geq 0 \tag{2.1.326}$$

$$\mu_n \xi_n = 0 \tag{2.1.327}$$

Using a similar method as in the linear case dependence on $\mathbf{w}, b$ and $\{\xi_n\}$ is eliminated to obtain the dual representation

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m), \tag{2.1.328}$$

which is subject to the constraints

$$0 \leq a_n \leq C \tag{2.1.329}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{2.1.330}$$

Using a similar substitution as in the linear case, $y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$ can be expressed in terms of the kernel function and parameters $\{a_n\}$ to obtain

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b. \tag{2.1.331}$$

Correspondingly, data points where $a_n = 0$ do not contribute to the predictive model leaving only the support vectors satisfying

$$y_n f(\mathrm{x}_n) = 1 - \xi_n. \tag{2.1.332}$$

It then follows that for data points where $a_n < C$, the data points will lie on the margin and be correctly classified. If $a_n = C$ the data points will either lie within the margin and $\xi_n \leq 1$ they will be correctly classified and incorrectly if $\xi_n > 1$.

Consequently the expression for b is given by

$$b = \frac{1}{N_M} \sum_{n \in M} \left(t_n - \sum_{m \in M} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right), \tag{2.1.333}$$

where M denotes the set of indices for data points fulfilling $0 < a_n < C$. [Bishop, 2006 (2)]

2.1.4 Naive Bayes

The Naive Bayes Classifier is a conditional probability classifier based on applying Bayes theorem under the assumption that the features are independent.

Given a problem instance to be classified, represented by a vector $\mathbf{x} = (x_1, ..., x_n)$ of n features, the Bayes probability model assigns to this instance conditional probabilities

$$P(C_k | x_1, ..., x_n) \tag{2.1.41}$$

for each of the K possible classes, C_k, conditional on the feature variables x_1 through to x_n. As this model is problematic if there is a large number of features or if a feature can take on a large number of values, the expression (2.1.41) can be reformulated in terms of Bayes' theorem giving

$$P(C_k | x_1, ..., x_n) = \frac{P(C_k) P(x_1, ..., x_n | C_k)}{P(x_1, ..., x_n)} \tag{2.1.42}$$

More plainly (2.1.42) can be written as

$$\text{posterior} = \frac{\text{prior} \cdot \text{likelihood}}{\text{evidence}}. \tag{2.1.43}$$

As the denominator of (2.1.42) is dependent only on the given feature values $\mathbf{x}$ and not on C it is the numerator that is of primary interest. Given that the numerator is equivalent to the joint probability model $P(C_k, x_1, ..., x_n)$ it can be expressed using repeated applications of the definition of conditional probability:

$$\begin{aligned} P(C_k, x_1, ..., x_n) &= P(C_k)P(x_1, ..., x_n|C_k) \\ &= P(C_k)P(x_1|C_k)P(x_2, ..., x_n|C_k, x_1) \\ &= .... \end{aligned} \tag{2.1.44}$$

Using the naive assumptions of conditional independence, each feature x_i is conditionally independent of every other feature x_j for $i \neq j$. This means that

$$P(x_i|C_k, x_j) = P(x_i|C_k). \tag{2.1.45}$$

Hence the joint model can be expressed as

$$\begin{aligned} P(C_k, x_1, ..., x_n) &= p(C_k)P(x_1|C_k)P(x_2|C_k)P(x_3|C_k)\dots P(x_n|C_k) \\ &= P(C_k)\prod_{i=1}^{n} P(x_i|C_k). \end{aligned} \tag{2.1.46}$$

This means that under the independence assumption, the conditional distribution over the class variable C_k can be written as the probability of a class times by the product of one dimensional densities:

$$P(C_k|x_1, ..., x_n) = \frac{1}{Z}P(C_k)\prod_{i=1}^{n} P(x_i|C_k), \tag{2.1.47}$$

where Z is a scaling factor dependent only on $x_1, ..., x_n$, that is, a constant if the values of the feature variables are known. [Murty, 2011 (3)]

The Naive Bayes Classifier combines this Bayes probability model with a decision rule. One common rule known as the maximum a posterior (MAP) decision rule, picks the hypothesis that is most probable. The corresponding classifier is the function that assigns a class label $\hat{y} = C_k$ for some k

$$\hat{y} = \underset{k\in\{1,...,K\}}{\operatorname{argmax}}\ P(C_k)\prod_{i=1}^{n} P(x_i|C_k) \tag{2.1.48}$$

One of the benefits of a model of this form is its manageable as it only uses prior probabilities of classes $P(C_k)$ and independent probability distributions $P(x_i|C_k)$. A class's prior may be estimated by calculating the class probability from the training

set. The form of the class conditional density density depends on the type of each feature some possibilities are given below [Murphy, 2012 (4)]:

- The Gaussian distribution can be used in the case of real-valued features:

$$P(\mathbf{x}|C_k, \theta) = \prod_{i=1}^{n} \mathcal{N}(x_i|\mu_{iC_k}, \sigma^2_{iC_k}) \tag{2.1.49}$$

  where μ_{iC_k} is the mean of feature i in objects of class C_k, and $\sigma^2_{iC_k}$ is its variance.

- The Bernoulli distribution can be used in the case of binary features:

$$P(\mathbf{x}|C_k, \theta) = \prod_{i=1}^{n} \mathcal{B}(x_i|\mu_{iC_k}) \tag{2.1.410}$$

  where μ_{iC_k} is the probability that feature i occurs in class iC_k.

- The Multinomial distribution can be used in the case of categorical features:

$$P(\mathbf{x}|C_k, \theta) = \prod_{i=1}^{n} \mathcal{M}(x_i|\mu_{iC_k}) \tag{2.1.411}$$

  where μ_{iC_k} is a histogram over the K possible values for x_i in class C_k

2.2 Imbalanced Data

A data set is imbalanced if there is an unequal distribution of classes within a data set. Machine learning algorithms are generally constructed to improve accuracy by reducing the error, and do not usually take into account the class distribution. As such most algorithms tend to produce inaccurate classifiers when dealing with imbalanced data sets. There are several methods available to address the issue of class imbalance. Two such methods are Over-sampling and SMOTE.

2.2.1 Over-Sampling

Over-sampling is the most straightforward of the two methods and simply consisted of repeatedly sampling from the minority class at random and with replacement until the ratio between the different classes meets some predetermined threshold.

2.2.2 Synthetic Minority Over-sampling Technique

Synthetic Minority Over-sampling TEchnique (SMOTE) is a more sophisticated method in which synthetic samples of the minority class are created to address the imbalance.

		Actual Outcome P	Actual Outcome N
Predicted Outcome	P′	True Positive (TP)	False Positive (FP)
	N′	False Negative (FN)	True Negative (TN)

Figure 2.1: Confusion Matrix

In order to create a synthetic data point the SMOTE technique works by first taking a minority sample from the data set, and determining its k nearest neighbors in the feature space. The Euclidean distance between the current data point and each of its k nearest neighbors is calculated. The distances are then multiplied by a number generated at random from the interval $[0, 1]$ and added to the current data point to create new, synthetic data points. In the case of nominal valued features the Value Distance Metric can be used to compute the nearest neighbors. [Chawla, 2005 (5)]

2.3 Model Evaluation

Accuracy is in not by its self a reliable metric to access the performance of a classifier as it will yield misleading results if the data set is unbalanced. Consider a data set balanced 95 observations of class A and 5 observations of class B. A classifier that classifies all observations to class A would be 95% accurate. This accuracy measure does not however indicate the ability of the model to recognize the minority class B as such more insightful metrics are required.

2.3.1 Confusion Matrix

A classifier is typically evaluated through the use of a confusion matrix as seen in figure 2.1. A confusion matrix is a table with two rows and two columns that reports the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). This allows for a more detailed analysis of the classifier than mere accuracy. [Chawla, 2002 (6)]

From the confusion matrix the following metrics can be derived-

Accuracy: The proportion of correctly classified outcomes in relation to the total number of outcomes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3.11}$$

Precision: The proportion of correctly classified positives in relation to total number of classified positives.

$$Precision = \frac{TP}{TP + FP} \tag{2.3.12}$$

Sensitivity/Recall: The proportion of actual positives that are correctly identified.

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.3.13}$$

Specificity: The proportion of actual negatives that are correctly identified.

$$Specificity = \frac{TN}{TN + FP} \tag{2.3.14}$$

2.3.2 Receiver Operating Characteristic

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the relationship between false positive and true positive rates and is used to access a classifiers performance. An ROC curve is based on the notion of two classes forming a pair of overlapping distributions. Complete separation of the two underlying distributions implies a perfectly discriminating classifier while complete overlap implies no discrimination.

The ROC curve is created by plotting the cumulative distribution function of the true positive rate, also known as sensitivity, against the cumulative distribution function of the false positive rate, calculated as 1 specificity. Frequencies of positive and negative results of the classifier will vary as one changes the "criterion" or "threshold" for positivity on the decision axis.

An ROC curve located progressively closer to the upper lefthand corner of the plot corresponds to a high discriminant classifier. An ROC curve lying on the diagonal line reflects the performance of a classifier that is no better than chance. [Hajian-Tilaki, 2013 (7)]

2.3.3 Area Under the Curve

The Area Under the Curve (AUC) is given by

$$\int_0^1 ROC(u)du \tag{2.3.31}$$

and is a real valued measure of sensitivity and specificity that describes the inherent validity of a classifier. [Hajian-Tilaki, 2013 (7)]

3

Methods

3.1 Data Analysis & Model Building Process

It is quite easy for the data analysis and model building process to become disorganized and confused. In an effort to avoid this the CRoss Industry Standard Process for Data Mining (CRISP-DM) was followed. [Marban, 2009 (8)]

CRISP-DM is a six stage methodology for providing a structured approach to implementing a data mining project. The basic phases of the CRISP-DM Framework are:

- **Business understanding:** Understanding the project objectives and requirements.
- **Data understanding:** Collecting, describing and exploring the data. Accessing data quality.
- **Data preparation:** Prepare and clean the provided data. Feature engineering and data transformation.
- **Modeling:** Implement selection of ML algorithms depending on the business requirements, available data and desired outcome.
- **Evaluation:** Evaluate model (using appropriate metrics), select model, access if it achieves the business objectives.
- **Deployment:** Organized and Present model/results.

It is important to note that the sequence of the phases is not strict. A certain amount of moving back and forth between different phases is required as the insights learned during any particular stage of the process can trigger new questions and developments that may subsequently be implemented into the process.

4

Data

4.1 Data Description

The input data consisted of approx 16 040 observations and the response data 8 703 observations. Upon cleaning the data, removing observations with missing values and matching with the response data the final data set consisted of 1 101 observations. The following two sections detail how the data was gathered and the features included.

4.1.1 Input Data

Each Youcruit candidate undergoes a screening process aimed at gathering information on their driving background, employment history and any criminal offences they may have been convicted of. The initial questions are of a binary nature for example "Have you had any moving violations?" - "Yes" or "No". In the case of a negative answer no further questions are asked regarding that particular aspect of the candidates background. If however a candidate answers positively then further questions are asked regarding that particular aspect. So continuing with the example given above the further questions would be - "What was the charge?", "What was the violation date?", "What was the vehicle type?", "Did you receive a Suspension?", "Was your license revoked?" and "In which state were you charged?". Depending on how a candidate answered these questions a further line of questions may or may not be pursued. In the case that a candidate may have had a moving violation charge of the type "Speeding" then further questions would be asked regarding what was the posted speed and how much over the speed limit they were driving.

As as result of this type of questioning the structure of the screening data was not tabular but instead hierarchical, with different question branches having different branch lengths for each different candidate. In addition there were some issues regarding missing values as not all candidates answered all questions. This led to challenges in compiling a data set suitable in nature to be fed into a Machine Learning algorithm. Using only the initial first level of screenings questions may result in crude feature variables with poor predictive qualities. Using on the other hand too many of layers of the screening questions may result in variables with better predictive qualities but a much smaller data set that is not likely to be representative of

the entire population of candidates.

In deciding which features to include domain knowledge was key in. As an example the question "Do you have a valid CDL license?" was not included as all candidates had a valid license. However questions regarding the specificity of a candidates license were included such as "Do you have any endorsements?", or "Do you have any restrictions?". These question were include in the data set as they had a high level of candidate response in addition to yielding useful information. A candidate that has endorsements has undergone additional training in order to obtain the extra qualification and as such indicates a level of seriousness and professionalism on the candidates part towards their career as a truck driver. Similarly if a candidate has restrictions such as only being licensed to drive automatic vehicles, or not being licensed to drive a night, this may be seen as a limiting factor by carriers looking to recruit drivers. Table A.1 displays the variables included in data set. Note that not all these variables were used in creating the classification models. In addition other variables were generated in the feature engineering process covered in chapter 4.2.

4.1.2 Response Data

The second data source came from carrier submissions detailing if a candidate had been hired or not. After a candidate has gone through a carriers selection process the carrier evaluates each candidate and assigns them to one of the three possible categories: Hired, OK to hire or Rejected. See table 4.1 for more detail. In the event that a candidate was rejected, the carrier provides supplementary information in the form of assigning nine possible reason for rejection given in table 4.2

Table 4.1: Carrier submissions

Submission Status	**Description**
Rejected	Carrier failed to make and offer OR Candidate refused an offer.
Hired	Carrier made and offer AND Candidate accepted.
OK to hire	Carrier in the process of making an offer.

It should be noted that rejecting a candidate does not necessary indicate that a candidate was of low quality or otherwise unsuitable. A candidate may be rejected because they took an offer from a different carrier or that they were given an offer but declined. Additionally a carrier may have wished to offer a candidate a position but were unable to contact the candidate. Analysis of 'Other' category (table 4.2) suggest that while there may have been negative reason for rejecting some candidates, some candidates may have been deemed worthy of an offer by the carrier but the carrier simply did not have any available positions at that time. It would also seem that some carriers chose the category 'Other' for simplicity sake. Rather that give each individual a specific reason for rejection they would assign all candidates

Table 4.2: Rejection reasons

Rejection Reason	Description
Safety rejection	Candidate rejected due to safety concerns.
Failed drug screen	Candidate rejected due to failing carrier drug test.
Criminal background	Candidate rejected due to criminal background.
Candidate no longer interested	Candidate 'rejected' due to no longer being interested in the position.
Candidate took other job	Candidate 'rejected' due to candidate taking other position.
Not able to contact	Candidate rejected due to carrier not being able to contact candidate.
No show to orientation	Candidate rejected due to failing to show up for orientation and/or interview.
Not qualified	Candidate rejected as due to carrier deeming them not qualified for position.
Other	Candidate rejected for unspecified reasons.

to the 'Other' category. A further issue was that carriers were not obligated to give a rejection reason in the earlier iterations of the response process. This lead to further issues of missing values as it was only for the more recently recruited candidates that the response data was more complete.

4.2 Feature Engineering

Feature engineering is the process of using domain knowledge of the data to create features that machine learning algorithms can more readily exploit in order to make more accurate predictions and/or classifications.

4.2.1 Input Feature Engineering

Table B.1 overviews all the variables that were engineered with the aim of isolating characteristics that may provide useful in the model building process. Further motivation for the engineering is given below

While the screening data provides some measure of driver experience with the feature variables OTR months, Tractor-trailer months, Tanker months and Flatbed months, it is possible and even likely that there is overlap between these variables. As an example if a driver states that they have 2 years experience driving OTR

and 1 years experience driving a Tanker there is no way of knowing if the period the driver spent driving a Tanker truck coincided with the period the driver spent driving OTR or if it was in addition to the period spent driving OTR. As such there is no definitive variable that measures total driver experience. In an attempt to address this a new variable TTF months was created as the sum of Tractor-trailer months, Tanker months and Flatbed months. OTR is the most common form of experience as well as the most sort after by carriers, as such it was decided that this warranted it not being included in the TTF summation but left as a standalone feature.

The number of accidents or misdemeanors a candidate has is not of itself a thorough indicator of driver competency. Rather a driver who has had 2 accidents over a 20 year career is arguably a lower risk candidate than a drive who has only had 1 accident during a shorter 5 year carrier. The feature Accident-per-OTR, Accident-per-TTF, Misdemeanors-per-OTR and Misdemeanors-per-TTF in addition to their binary counterparts were all attempts to provide a more comprehensive picture of driver competency.

With regards to Endorsements and Restrictions, as the number of candidates that held either of these was low it was felt that measuring the number of these a driver carried may prove to be more useful than the specific type of endorsement or restriction.

4.2.2 Response Feature Engineering

Youcruit work with many different transport companies, servicing different parts of the transport sector, working in different states and subject to different laws, regulation and insurance requirements. Fundamentally, each carrier will therefore have slightly differing notations of what the ideal candidate looks like. The aim of the thesis is to apply machine learning methods to find the candidates most likely to be given an offer of employment based on the hiring outcomes of past candidates. Under the assumption that "the best candidates get the job", it was decided that candidates should be divided into two categories under a general notion of candidate quality: High Quality Candidates and Low Quality Candidates.

High Quality Candidates (HQC) were defined as candidates that:

- Had been hired **OR** ...
- were deemed suitable to hire but had yet to be given an offer **OR** ...
- were given an offer but were no longer interested in the position **OR** ...
- were given an offer but took another position with another employer.

Low Quality Candidates (LQC) were defined as all other candidates.

It is worth stressing that given the carriers varying needs and requirements what one carrier may deem a worthy hire, the other may deem unworthy, so it is entirely

possible for an individual candidate to be both given an offer of employment from one company and at the same time rejected from another. Given this, it is not certain that the above definition is relevant or even that the concept of an all-round high quality candidate satisfying a wide range of requirements is valid.

4.3 Data Analysis

All negative instances of the binary variables were coded as 0, positive instances as 1. As an example, if a candidate had recorded misdemeanors then the Misdemeanor variable would be coded as 0. This enabled the examination of proportions between the number of negative and positive instances for each feature and each class. E.g. For candidates labeled as low quality the proportion of candidates with misdemeanors to candidates without misdemeanors was $\frac{112}{657} = 0.17$, while for high quality candidates it was $\frac{4}{52} = 0.08$.

Through examination of the above described proportion metric two key pieces of information can be gleamed.

- Does the data support reasonable expectations based on domain knowledge. For example one would expect the proportions of candidates with recorded accidents to be lower among the HQC's then the LQC's. By comparing proportions it can be determined if these expectations are reasonable.
- Examining the difference in the proportions between HQC's and LQC's for each variable gives an indication of the importance or usefulness of that variable when building a model.

Table 4.3 shows the proportion metric for 9 different binary variables. It can be noted that the majority of features have proportions that align with expectations, however in many cases this difference is small.

Similarly by looking a the mean of the numerical variables for both high quality and low quality candidates ones expectations can be confirmed or disputed. Observing the means given in table 4.4 the first indications of inconsistencies between the data and expectations are seen. In nearly all branches and career fields the more experience a candidate has the more valuable they are. As such an expectation of the high quality candidates having more experience is not unreasonable. This however is not reflected in the data, where for all of the features that measure experience the low quality candidates have a much higher average than the high quality candidates. It is also worthwhile investigating if the the definition of a high quality candidate as given in section 4.2.2 meets reasonable assumptions. While it may be naive, it is not unreasonable to expect a candidate with a "spotless record" to be classified as high quality. By filtering away all candidates with recorded accidents, misdemeanors, moving violations, failed drug test etc. a subset of 73 "spotless" candidates was obtained. The majority of these candidates also had more experience than was the average for each of the experience types. Unfortunately only one of these 73 "spotless" candidates fulfilled the HQC definition, with all other candidates labeled as

Table 4.3: Feature proportions

Variable name	LQC	HQC
Moving Violations	0.74	0.7
Felonies	0.13	0.1
Misdemeanors	0.17	0.08
Accidents	0.47	0.37
Contracts	0.05	0.06
Failed or Refused Drug Test	0.02	0
Suspended of Revoked License	0.19	0.2
Terminated	0.21	0.12
Moving Violations in commercial vehicle	0.23	0.19

Table 4.4: Feature averages

Variable name	LQC	HQC
OTR months	49	32
TTF months	99	50
Tanker months	7	2
Tractor trailer months	79	45
Flatbed months	12	3
Number Accidents	0.4	0.3
Number Moving Violations	0.6	0.6
Number Endorsements	0.8	0.6

LQC despite their "spotless record".

It can as such be concluded that the definition of a high quality candidate as given in section 4.2.2. is conflicted and problematic. In addition the lack of a clear difference between feature characteristics for high and low quality candidates as well as inconsistencies between the data and expectations are likely to severely hinder the accuracy of machine learning algorithms to correctly classify candidates.

4.4 Feature Selection

The variables used to train a machine learning algorithm can have a substantial effect on model performance. Certain features can be expected to be colinearly dependent. In addition, irrelevant features in the data set can decrease model accuracy and increase training time. It then becomes desirable to select only the variables that contribute most to predicting the response variable.

Two methods were used in an attempt to select only the most relevant variables. This resulted in two different feature list. Feature list A consisted of 13 features while feature list B consisted of 6 features. Table 4.5 shows which variables were

included in each list.

Table 4.5: Selected feature

Features A	**Features B**
Misdemeanors	Misdemeanors
Number Endorsements	Number Endorsements
Terminated	Terminated
Accidents-per-TTF	Accidents-per-TTF
Accidents-per-OTR	Accidents-per-OTR
Suspended or Revoked License	Accidents
Contracts	
Number Moving Violations	
Moving Violation in commercial vehicle	
Moving Violations-per-TTF	
Failed or Refused Drug test	
TTF Months	
OTR Months	

To determine which variables to use in feature list A, a collection of logistic regression and random forest models were built using all available features. The results of these models were assessed to build up an understanding of variable importance. The variables deemed most important were selected to make up the preliminary version of data set A. Thereafter several variables were removed based on them displaying high levels of correlation with with other more "important" variables.

To determine which variables to use in feature list B the finding of the data analysis preformed in section 4.3 was exploited. Features were selected if the the difference in the proportion (or mean) between the high and low quality candidates was deemed to be large.

5

Results

Given the data analysis findings outlined in section 4.3 it is unlikely that many machine learning algorithm would be able to deal with the ambiguous nature of the data. As such this becomes less a process of finding a suitable and highly accurate model but rather an exercise in examining the different ways in which the models are inaccurate.

5.1 Rebalancing

As the HQC were the minority class in the data representing approx only 5% of the data, the first step was to re-balance the data as outlined in chapter 2.2. After splitting the data into training and testing data with a 75/25 split, over-sampling and SMOTE were applied so as to increase the proportion of the minority class to approx 25%. Together the features selected in chapter 4.4 the following four data set were obtained:

- **Over A:** The over-sampled date consisting of the A features.
- **Smote A:** The SMOTE-sampled date consisting of the A features.
- **Over B:** The over-sampled date consisting of the B features.
- **Smote B:** The SMOTE-sampled date consisting of the B features.

In the following section the performance of the the different classifiers are presented: Bagging (BAG), Support Vector (SV) and Naive Bayes (NB). All three machine learning algorithms were tested on the four data sets given above. However as the models trained on the over-sampled data resulted in similar or slightly poorer preforming models as those trained on the SMOTE data sets the results have been omitted.

5.2 Bagged Model Results

The confusion matrices for the two bagged models using the A and B features can be seen in tables 5.1 and 5.2. Notably neither models were able to correctly classify a single high quality candidate. The misclassification of low quality candidates was

lower in the model trained on the Smote B data set, i.e. the data set with fewer features.

The evaluation metrics for the bagged models are presented in table 5.3. Both models have precision and sensitivity scores of zero, reflecting the inability of either model to correctly classify a single HQC. Nonetheless specificity and accuracy scores are quite high due to only a small numbed of low quality candidates being misclassified.

Table 5.1: Confusion matrix for Smote A Bagged model.

		Actual	
		HQC	LQC
Predicted	HQC	0	19
	LQC	13	244

Table 5.2: Confusion matrix for Smote B Bagged model.

		Actual	
		HQC	LQC
Predicted	HQC	0	9
	LQC	13	254

Table 5.3: Evaluation Metrics - Bagged Models.

	Accuracy	**Precision**	**Sensitivity**	**Specificity**
Smote A	0.88	0.00	0.00	0.93
Smote B	0.92	0.00	0.00	0.97

In figure C.1 the ROC curve for both bagged models is shown, confirming the low performance of both models.

5.3 Support Vector Model Results

The confusion matrices for the support vector models are shown in table 5.4 Both models were highly accurate with regards to correctly classifying LQC, however both failed to correctly classify a single HQC. This is reflected in the evaluation metrics with a high specificity scores and low sensitivity scores. (The precision score for the Smote A data can not be calculated due to division by zero.)

The ROC plot and corresponding AUC values seen in figure C.2 reflect the inability of both models to correctly classify high quality candidates while correctly classifying low quality candidates.

5.4 Naive Bayes Model Results

The results to the two Naive Bayes models differ notable than those of the previous four models. While both models correctly classified most HQC, they also incorrectly classified a large number of LQC as HQC. See table 5.7. Both models incorrectly

Table 5.4: Confusion matrix for Smote A Support vector model.

Predicted \ Actual	HQC	LQC
HQC	0	0
LQC	13	263

Predicted \ Actual	HQC	LQC
HQC	0	1
LQC	13	261

Table 5.5: Confusion matrix for Smote B Support vector model.

Table 5.6: Evaluation Metrics - Support Vector Models.

	Accuracy	**Precision**	**Sensitivity**	**Specificity**
Smote A	0.95	-	0	1
Smote B	0.95	0	0	0.99

classified more candidates than they correctly classified.

This is reflected in the evaluation metrics show in table 5.9. Both models have low precision and specificity scores but the sensitivity scores (measure of correctly classified HQC) is relatively high. Also confirmed in by the ROC plot in figure C.3

Table 5.7: Confusion matrix for Smote A Naive Bayes model.

Predicted \ Actual	HQC	LQC
HQC	11	184
LQC	2	79

Predicted \ Actual	HQC	LQC
HQC	9	172
LQC	4	91

Table 5.8: Confusion matrix for Smote B Naive Bayes model.

Table 5.9: Evaluation Metrics - Naive Bayes Models.

	Accuracy	**Precision**	**Sensitivity**	**Specificity**
Smote A	0.32	0.06	0.85	0.3
Smote B	0.36	0.05	0.7	0.34

5.5 Comparison

As anticipated from the data analysis carried out in section 4.3 none of the machine learning algorithms preformed particularly well given the ambiguous nature of the data.

One can ascertain that the Bagging and Support Vector models are not fit for purpose as they all failed to correctly classify a single high quality candidate. With

that said they all had very high rates of specificity. In contrast to both the Naive Bayes models which while managing to correctly classify some of the high quality candidates, also misclassified a very large number of low quality candidates. Compering the ROC curves and their corresponding AUC scores also calls into question the validity of the models with all models preforming similarly to what could be expected from a random classifier.

Regarding which feature list A or B proved most suitable, each pair of models was very similar for each algorithm type. Given this similarity no definitive conclusions can be made on the superiority of one data set over the other. However as feature list B contains fewer variables then feature list A it could be argued that list B is to be preferred in this regard.

6

Conclusion and Improvements

The failure of the machine learning algorithms evaluated in this thesis to successfully classify candidates is not so much a failure of the algorithms to find a pattern in the data rather a reflection of the fact that there is a fundamental absence of a pattern in *this* particular data set. This does not mean that there are no patterns that could eventually be exploited to create accurate machine learning models, rather that improvements in the candidate screening and data gathering process are needed first.

The original screening data consisted of over 16 000 observations, However after removing observations with missing values and matching with the available response data the final data set consisted of only 1101 observations. Section 4.3 highlighted that while high quality candidates had a lower proportion of negative instances than low quality candidates as one would expect they also had lower levels of experience. One conjecture is that candidates with less experience or opportunities are more motivated to answer all screening questions thereby providing a more complete profile. Candidates with more experience, contacts and opportunities are less motivated to answer screening questions and thereby leave incomplete profiles, and consequently not making it into the final data set. This would result in a data set not representative of the whole and would go some way in explaining the conflicted and ambiguous nature of the data set. In this regard the one obvious improvement would be to make the screening process obligatory.

The decision was made in the data tidying phase to try and keep the number of observations as large as reasonable possible. This meant that some of the more granular data on each candidate was sacrificed and it is entirely possible that important features were overlooked. More work could be done here to ascertain if this is indeed the case.

Experience is generally seen as a desirable trait for any potential employee. While the screening process does measure certain types of experience there is a lack of a single over-all measure of experience. Including the question "How many years/-months of experience do you have?" into the screening process would be beneficial and lead to improvements to the accidents-per-OTR(TTF) and misdemeanors-per-OTR(TTF) variables created in section 4.2.1.

Finally, section 4.3 also highlighted the problematic way in which the choice to define

high and low quality candidates based on weather or not the had received an offer of employment led to candidates with what would seem to have spotless records being classified as LQC. Changes could be made in the way the carrier responses are gathered and subsequently labeled. One suggestion would be to simply ask the carriers to rate candidates on a scale and then to use this in conjunction with the recruitment outcome to improve the definition of high and low quality candidates.

Bibliography

[1] Jerome Friedman, Trevor Hastie and Robert Tibshirani. *The Elements of Statistical Learning*. Springer New York, 2009.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Cambridge, CB3 0FB, Microsoft Research Ltd, 2006.

[3] M. N. Murty, V. Susheela Devi. *Pattern recognition : an algorithmic approach*. Springer, London, Dordrecht, Heidelberg, New York, 2011.

[4] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts, London, England, 2012.

[5] Chawla N.V.*Data Mining for Imbalanced Datasets: An Overview.* In: Maimon O., Rokach L. (eds) *Data Mining and Knowledge Discovery Handbook*. Springer, Boston, MA. 2005.

[6] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer. *SMOTE: Synthetic Minority Over-sampling Technique*. In: *Journal of Artificial Intelligence Research 16 (2002) 321–357*.

[7] Karimollah Hajian-Tilaki, *Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation*. In: *Caspian Journal of Internal Medicine. 2013 Spring; 4(2): 627–635*.

[8] Oscar Marban, Gonzalo Mariscal and Javier Segovia. *A Data Mining Knowledge Discovery Process Model*. In: Julio Ponce and Adem Karahoca. *Data Mining and Knowledge Discovery in Real Life Applications*. IntechOpen, 2009.

Appendix A

Table A.1: Input data

Variable name	**Type**	**Description**
Moving Violations	Binary	0 if candidate has registered moving violations, 1 otherwise.
Felonies	Binary	0 if candidate has felony convictions, 1 otherwise.
Misdemeanors	Binary	0 if candidate has registered misdemeanors, 1 otherwise.
Accidents	Binary	0 if candidate has registered accidents, 1 otherwise.
Contracts	Binary	0 if candidate has contractual obligations, 1 otherwise.
Failed or Refused Drug Test	Binary	0 if candidate has previously failed or refused a drug test, 1 otherwise.
Suspended of Revoked License	Binary	0 if candidate has had a suspended or revoked license, 1 otherwise.
Terminated	Binary	0 if candidate has been terminated from previous employment, 1 otherwise.
Moving Violations in commercial vehicle	Binary	0 if candidate received a moving violation while driving a commercial vehicle, 1 otherwise.
Endorsement Type	Categorical	Six endorsement types - 'T', 'P', 'N', 'H', 'X', 'S'. (Also combination of endorsement e.g 'TNX').
Restrictions Type	Categorical	Seven restriction types - 'L', 'Z', 'E', 'O', 'M', 'N', 'V'. (Also combination of restriction e.g 'LM')
OTR Months	Numerical	Number of months experience driving 'Over The Road'.
Tractor-Trailer Months	Numerical	Number of months experience driving 'Tractor-Trailer'.
Tanker Months	Numerical	Number of months experience driving 'Tanker'.
Flatbed Months	Numerical	Number of months experience driving 'Flatbed'.
Number of Accidents	Numerical	Number of accidents registered to candidate.
Number of Moving Violations	Numerical	Number of moving violations registered to candidate.

Appendix B

Table B.1: Engineered Features

Variable name	**Type**	**Description**
Binary-Restrictions	Binary	0 if candidate has a restricted license. 1 otherwise.
Binary-Endorsements	Binary	0 if candidate has no endorsements. 1 otherwise.
Number Restrictions	Numerical	Number of license restrictions a candidate has.
Number Endorsements	Numerical	Number of license endorsements a candidate holds.
TTF Months	Numerical	Sum of number Tractor-Trailer Months, Tanker Months and Flatbed Months combined.
Binary-TTF	Binary	0 if candidate has ¡ 84 months experience. 1 otherwise.
Binary-OTH	Binary	0 if candidate has ¡ 24 months experience. 1 otherwise.
Accidents-per-TTF	Numerical	Number of accident divided by number of months experience driving TTF
Moving Violations-per-TTF	Numerical	Number of moving violations divided by number of months experience driving TTF
Accidents-per-OTR	Numerical	Number of accident divided by number of months experience driving OTR
Moving Violations-per-OTR	Numerical	Number of moving violations divided by number of months experience driving OTR

Appendix C

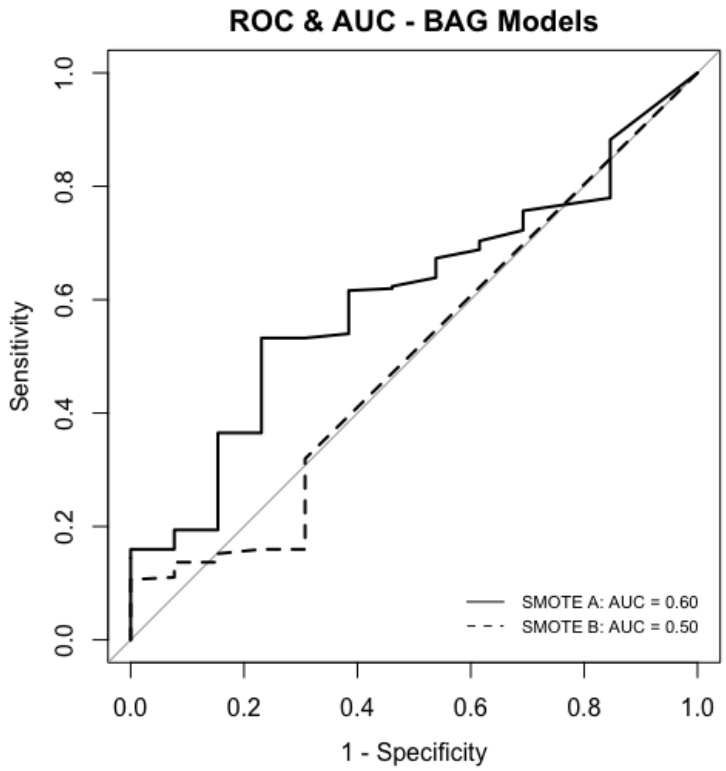

Figure C.1: ROC plot and AUC values for bagged models

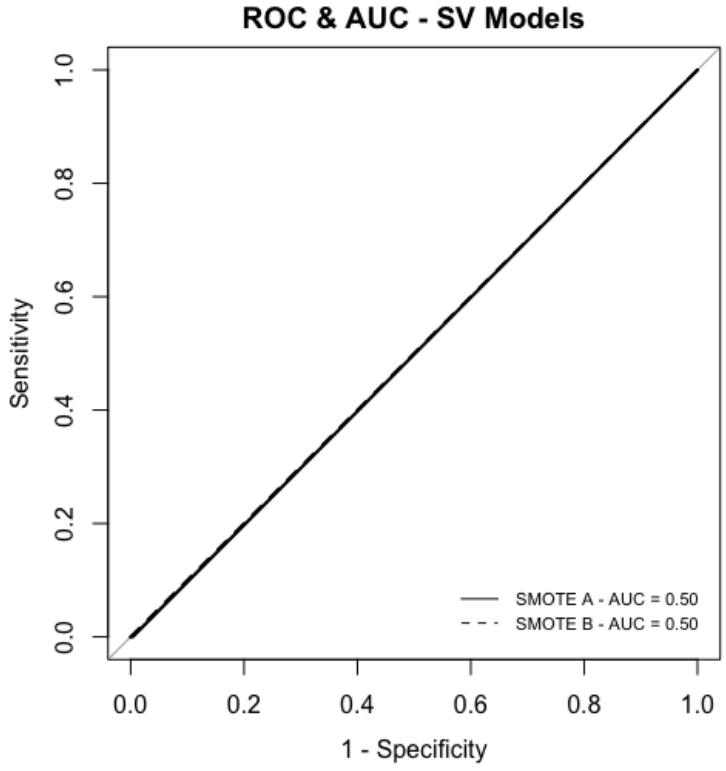

Figure C.2: ROC plot and AUC values for support vector models

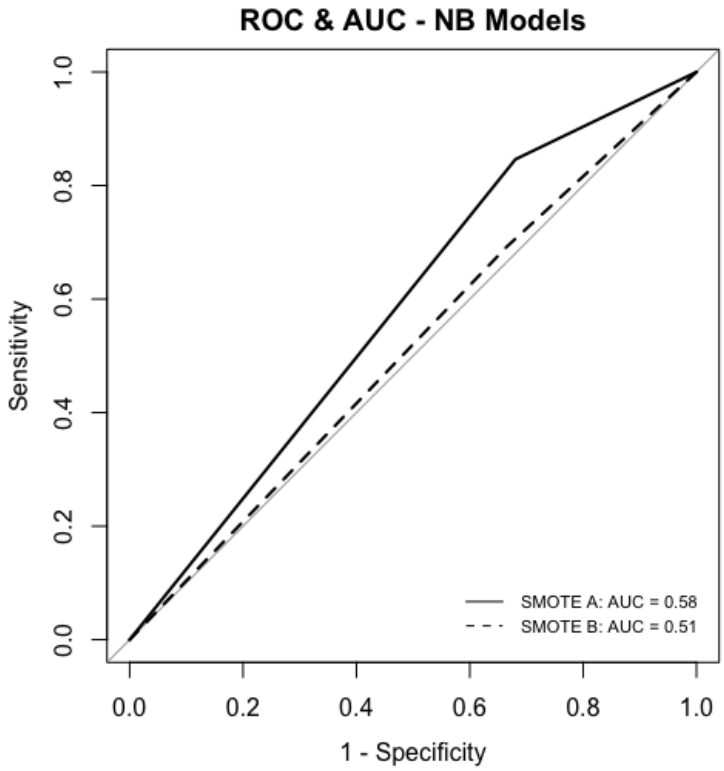

Figure C.3: ROC plot and AUC values for Naive Bayes models