CEE322: Data Analytics for Urban Systems

Lab 8: Predictive Data Analysis

Author: Davyd Tamrazov

Due date: Nov 20, 2020

Stanford University

PART 1

1 Introduction

This part aims to obtain a forecasting model producing day ahead prediction of daily bike rentals to aid the resource planning for the bike sharing program. Multiple models are developed, analyzed and validated using cross-validation technique and mean squared error as the performance metric. Dataset used for the analysis contains variables describing daily weather and the usage of the bike sharing scheme. For the purpose of the analysis, it is assumed that all past data and the current day's date and weather information is available to predict daily bike rental count.

2 Dataset

The dataset contains daily records of calendar, weather, and bike sharing program information for 731 days collected over years of 2011 and 2012. Unique variables, namely, date and ID of the record are removed from the dataset as they carry no predictive power. Further, categorical variables are encoded using Label Encoding to ensure that they are treated as such in the forecasting models. Further, temperature, feeling temperature, humidity, and windspeed features were normalized by the distributor using their maximum values of 41, 50, 100, and 67 correspondingly. The summary of the features used for the analysis is shown in Table 1 and Table 2.

Descriptor	*Mean*	*St.dev.*	*Min*	*25%*	*Median*	*75%*	*Max*
Normalized temperature (C°)	0.495	0.183	0.059	0.337	0.498	0.655	0.862
Normalized feeling temperature (C°)	0.474	0.163	0.079	0.338	0.487	0.609	0.841
Normalized humidity	0.628	0.142	0.000	0.520	0.627	0.730	0.973
Normalized windspeed (mph)	0.190	0.077	0.022	0.135	0.181	0.233	0.507
Number of casual riders	848.2	686.6	2.000	315.5	713.0	1096	3410
Number of registered riders	3656	1560	20.00	2497	3662	4777	6946
Number of total riders	4504	1937	22.00	3152	4548	5956	8714

Table 1: *Summary of the continuous features used for the analysis*

Year		*Month*				*Holiday*		*Weekday*		*Weather Situation*	
2011	365	January	62	July	62	No	709	Sunday	105	Clear	463
2012	365	February	57	August	62	Yes	21	Monday	104	Mist / Cloudy	246
		March	62	September	60			Tuesday	104	Light snow/rain	21
Season		April	60	October	62	*Working day*		Wednesday	104		
Spring	180	May	62	November	60	No	231	Thursday	104		
Summer	184	June	60	December	61	Yes	499	Friday	104		
Fall	188							Saturday	105		
Winter	178										

Table 2: *Summary of the categorical variables used for the analysis*

3 Analysis

The goal of the predictive model in this report is to forecast next day's total count of bike rentals given all of the features described in the previous section. As such, the forecast problem is framed in a way that the prediction day's total count is determined using previous day's bike rental information, as well as a combination of previous and prediction day's features. Therefore, the output of the analysis is set to be the prediction day's total count of bike rentals. Note that in all the tables and figures, prediction day is referred to as (t) and previous day is referred to as $(t-1)$.

It is further assumed that the weather conditions on the prediction day are known with certainty the day before and, thus can be added to the set of input features. Additionally, calendar information for the prediction day is deterministic and is known in advance, meaning that it can be added to the input features list. At the same time, calendar information of the day before the prediction day is assumed to carry no valuable information for the forecasting problem. Finally, the number of registered riders is collinear with casual and total riders and vice versa, since total = casual + registered. As a result, only registered and total numbers are used, thereby removing collinearity but also implicitly defining the number of casual riders. Thus, the final set of input features includes:

- Weather conditions of the prediction day and the previous day;
- Calendar information of the prediction day;
- Count of registered and total riders on the previous day.

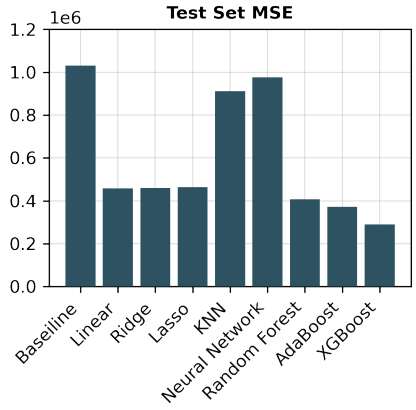
It has to be noted that the last observation in the dataset has to be removed as no next day's observation is available. Such featurization allows splitting the data randomly without compromising the time series nature of the data as current day's and next day's counts are linked for each observation. Thus, the dataset is split into training and test set randomly with the 80/20 ratio resulting in 584 and 146 entries in the respective sets.

Next, the training set is used to fit each of the models and validation testing is performed with 10-fold cross-validation. Since this is a regression problem, performance metric is set to the mean squared error (MSE) and the parameters for each model are tuned to minimize the mean value of the MSE over all validation sets obtained with cross-validation. For models with multiple hyperparameters, the grid search over the parameters was performed to minimize cross-validation error.

3.1 Modeling

Multiple models were fit to the data and fine-tuned using cross-validation on the training set. This section outlines the description of each model as well as the analysis of its performance and hypotheses of why some models perform better or worse than the others. Where possible, model parameters are interpreted.

MSE is used to compare models' performances. The baseline model used for benchmarking is a trivial predictor where prediction day's bike rental count is taken to be exactly the same as that of the previous day. The summary of the cross-validation and test MSE is provided in Figure 1.



Model	*Cross-validation MSE*	*Test MSE*
Baseline	1159064	1029843
Linear Regression	593777	458619
Ridge Regression	574485	459440
Lasso Regression	584854	463212
KNN	1051968	912158
Neural Network	1094650	975066
Random Forest	509883	406949
XGBoost	393115	289829
AdaBoost	476887	371293

Figure 1: *Summary of the models' cross-validation and test set performance*

1. **Linear Regression**

   This type of model fits a line to the data by minimizing the residual sum of squares (RSS). The parameters of the model include intercept and the coefficients for each of the input features. These can be found analytically by solving the normal equation as follows, where X and y are the input matrix and output vector respectively:

   $$\hat{\theta} = (X^\top X)^{-1} X^\top y$$

   In general, linear regression produces a low variance model with the fit of the line not varying significantly between different sets of observations. On the other hand, bias of the model depends on how linear the data is. In this case, linear progression performs significantly better than the baseline model, suggesting the relationship is approximated well with the linear model.

2. **Ridge / Lasso Regression**

   Ridge and Lasso are two variations of the linear regression model where regularization parameter is added to the RSS minimization problem to further reduce the variance and create a more parsimonious model. In case of Ridge regression, $L2$-norm ($||\theta||_2^2$) is used, while in Lasso regression, $L1$-norm ($||\theta||^2$) is added to penalize high coefficient values. In both cases, coefficients are driven closer to zero, although in the former model, coefficient can never reach zero, meaning that none of the variables will be fully excluded from the model. Via experimentation, the optimal regularization strength was found to be 0.3 and 1.0 for ridge and lasso regression models correspondingly.

All regression models are highly interpretable and the parameters obtained for each model are summarized in Table 3. There it may be observed that weather features for previous and prediction days have coefficients with the opposite signs, meaning that it is the difference between weather parameters between days that has an impact on the bike rental count on the prediction day. This observation holds for all three models, apart from the Lasso where windspeed on the previous day is assigned a coefficient of 0.0, suggesting that this feature has limited influence on the bike rental forecast. Next, holiday and working day appear to have opposite effect among all models – on holidays, bike rental drop significantly, what implies that the total bike rental count is primarily driven by commuters throughout the year. Finally, year feature has a large positive coefficient, meaning that there is an increasing trend in bike rentals over time.

Feature	Linear	Ridge	Lasso	*Feature*	Linear	Ridge	Lasso
Season (t)	195.8	199.0	195.3	Windspeed (t-1)	462.4	204.6	0.000
Year (t)	944.5	946.0	944.2	Temperature (t)	3862	2742	3089
Month (t)	-22.87	-23.71	-22.95	Temperature (t-1)	-1609	-1144	-863
Holiday (t)	-313.5	-294.9	-267.2	Feeling Temperature (t)	1747	2183	1744
Weekday (t)	32.79	30.55	27.95	Feeling Temperature (t-1)	-1765	-1455	-1698
Working day (t)	66.99	72.30	66.83	Weather Situation (t)	-507.9	-546.5	-539.2
Humidity (t)	-2110	-1900	-1936	Weather Situation (t-1)	108.6	127.2	125.0
Humidity (t-1)	2114	1907	1896	Number of registered riders (t-1)	0.151	0.158	0.169
Windspeed (t)	-2589	-2247	-2251	Total count (t-1)	0.416	0.407	0.396

Table 3: *Summary of the linear, ridge, and lasso regression parameters*

3. K Nearest Neighbors

K nearest neighbors is a non-parametric model that calculates the Euclidean distance between the observations in the multidimensional feature space and forecasts a prediction based on the average of the k nearest neighbors. Bias and variance of the model largely depends on the number of neighbors selected – variance reduces with the higher number of neighbors used to forecast the points. As such, a grid search was run on the cross-validation set which determined that 44 neighbors produced the optimal result. Nevertheless, model's performance on the test set is only marginally better than that of the baseline model. This is because the high dimensionality of the problem (20 features) is detrimental to this method because the distance between observations become less meaningful, as all vectors are approximately equidistant in the feature space.

4. Neural Network

Neural Network is a non-linear model that involves setting up a number of hidden layers, where, in a fully-connected case, each 'neuron' corresponds to an activation function applied to the linear combination of the input nodes or hidden nodes from the previous layer. The purpose of the activation function is introduce non-linearity to the formulation, what enables approximating any form of the non-linear function. The model is iteratively trained with backpropagation where gradient descent is used to solve for the optimal parameters in each of the hidden nodes. In this case, multi-layer perceptron regressor is trained with the adaptive learning rate and ReLU activation function. However, due to high complexity of the model, Neural Networks tend to easily overfit small tabular data, what results in the worst performance among the considered models in this case.

5. Random Forest

Random Forest is a bagging-based decision tree method that works by averaging a set of randomly generated and uncorrelated decision trees. Each tree is fit on a random subset of features where each split in the tree is selected to minimize MSE. Considering only a subset of features allows to create uncorrelated trees where strong features are not necessary present, thereby resulting in unbiased predictions. The only two parameters defining this method are the tree depth and the number of estimators. It is generally preferred to have smaller tree depth to reduce the predictive strength of each tree and avoid overfitting. These parameters were determined by performing the grid search, resulting in the maximum depth of 10 with 300 estimators. Notably, this models performs considerably better than the regression and baseline models, as decision tree is able to capture non-linear behavior of the data more accurately.

6. AdaBoost

AdaBoost is one of the first developed boosting algorithms which combines an ensemble of weak (low accuracy) decision tree learners to produce a forecasting model that generalizes well to the data. In each iteration, misclassified observations from the previous models are upweighted an the next decision tree is fit by minimizing the training error. The benefit of using such model is that boosting inherently performs dimensionality reduction of the problem as irrelevant features are simply not used in the tree splitting which is defined by the maximization of MSE reduction, similar to the Random Forest. The output of the model is produced by taking the weighted average of the weak learners. The maximum depth of the tree in each iteration is limited to 7 and 1000 estimators to balance the strength of each individual tree and potential for it overfitting the data. The performance of the model is visibly better than all of the previously considered models primarily due to flexibility of decision tree modeling that allows approximating non-linear functions. Further, this model outperforms Random Forest due to the robustness of the AdaBoost algorithm that assigns higher weights in the error calculation to the misclassified examples, thereby iteratively improving the fit of the ensemble.

7. XGBoost

XGBoost is a a highly optimized boosting algorithm which, similarly to AdaBoost, leverages a set of weak learning to produce a strong forecasting model. However, unlike the previous algorithm, XGBoost fits each model by interpreting gradient boosting as an optimization problem expressed by the residual errors of the previous model. Thus, at each iteration the decision tree is fit to the residuals using gradient descent, what results in a highly accurate model. The model parameters were selected using grid search on the cross-validation data, resulting in the maximum tree depth of 4 with 1000 estimators, with each tree found by performing gradient descent at each iteration with the learning rate of 0.015. Meanwhile, an additional parameter called subsampling was set to 0.5, meaning that only half of the training data is randomly sampled for each tree, thereby mitigating the risk of overfitting. Overall, the XGBoost model performs exceptionally well, achieving test MSE almost 100,000 lower than that of the AdaBoost model. The success of this model is explained by the high versatility of the decision tree learners and a very efficient tree boosting process performed with the XGBoost algorithm.

Using decision tree based models allows interpreting the results by considering importance of each feature in the splitting of nodes in the decision trees. This importance is defined by how much error is reduced on average in each tree due to the split in a given feature, as illustrated on Figure 2. There, it is evident that Random Forest and AdaBoost result in a similar set of important features, with the count of total and registered user on the previous day having significantly higher relative importance. Additionally, weather features on the prediction day tend to dominate the list of important features, with temperature and feeling temperature on the prediction day being common between models. On the other hand, XGBoost finds an interesting influence of the humidity and the windspeed from the previous and prediction days, suggesting that these four features contribute the most to forecasting the total bike rental count. Similarly to the Random Forest and AdaBoost models, temperature and the count of total and registered riders plays an important role in the forecasting model, underlining the importance of these features.

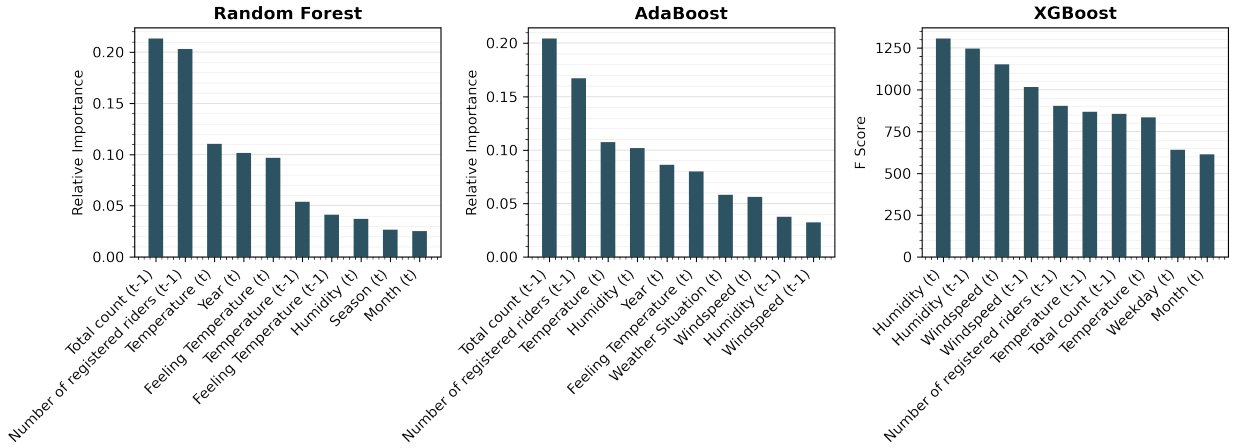


Figure 2: *Comparison of MSE for cross-validation and test sets with developed models*

Finally, as can be seen from Figure 1, decision tree models result in a significantly better performance than the rest of the considered models, with XGBoost outperforming baseline model by three-fold. Figure 3 shows the scatter of the predicted versus true values for each observation in the test set. There, it is clear that all models perform well, with consistently low deviation from perfect prediction. Nevertheless, XGBoost clearly performs considerably better for the values between 4000-5000, where the distribution of points is tighter, what illustrates why this model is the optimal one among those considered. Thus, XGBoost is selected as the final model.

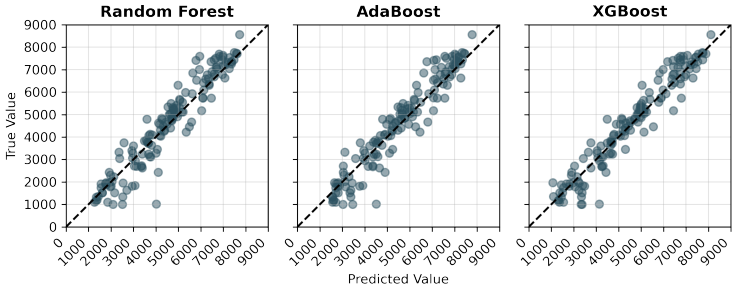


Figure 3: *Comparison of MSE for cross-validation and test sets with developed models*

4 Conclusion

In conclusion, a selection of models was developed to forescast prediction day's bike rental count given weather, count, and calendar information. Using cross-validations, parameters of each model were fine-tuned and the performance was compared on the test set. It was observed that decision tree models offer considerably more accurate predictions that regression and other models because they are able to capture non-linear nature of the distribution well without overfitting the dataset. In particular, XGBoost model was able to achieve exceptionally good performance on the test set.

PART 2

1 Introduction

The second part of the report aims to predict whether the US is the country of origin of the car given a set of car characteristics. A number of classifiers are developed and validated using cross-validation technique and the performance of each model is quantified using area under the receiver operator curve. Models' performances are all compared to the benchmark that is defined as the random classifier that assigns equal probability of the observation belonging to either class and all of the features are used in developing classifiers in further sections.

2 Dataset

The dataset contains 392 cars each described with a set of specific characteristics as well as a flag to identify if the US is the country of origin. There are 245 cars with the US origin and 137 originated elsewhere. The summary of the car characteristics variables is provided in Table 1. No pre-processing was performed on the dataset as it appears to be cleaned by the provider.

Descriptor	*Mean*	*St.dev.*	*Min*	*25%*	*Median*	*75%*	*Max*
MPG	23.45	7.81	9.000	17.00	22.75	29.00	46.60
Cylinders	5.472	1.706	3.000	4.000	4.000	8.000	8.000
Displacement	194.4	104.6	68.00	105.0	151.0	275.8	455.0
Horsepower	104.5	38.49	46.00	75.00	93.50	126.0	230.0
Weight	2978	849.4	1613	2225	2804	3615	5140
Acceleration (0-60 mph)	15.54	2.759	8.000	13.78	15.50	17.03	24.80
Model Year	75.98	3.684	70.00	73.00	76.00	79.00	82.00

Table 1: *Summary of the car characteristics*

3 Analysis

The aim of the classifier is to predict the origin of the car given its characteristics. Since observations are not sequential and are independent of one another, the data is randomly split into training and test sets using 80/20 ratio, resulting in 313 and 79 observations in each set respectively.

3.1 Modeling

The training set is then split into 10-folds of training and validations subsets to be used to fit each classifier with cross-validation. The performance of the models is quantified using the area under the receiver operator curve (AUROC) as it shows the generalized accuracy of the model without having to pick a certain threshold. Table 2 summarizes cross-validation and test set performance, while Figure 1 illustrates models' performance on the test (random classifier is shown in dashed line). Note that accuracies and the confusion matrices are computed using a standard threshold of 0.5. For cross-validation, all metrics are averaged over 10 folds.

		Baseline	Logistic	Ridge	SVC	QDA	NN	RF	AdaBoost	XGBoost
CV	AUROC	0.5000	0.9481	0.9312	0.9416	0.9382	0.9336	0.9691	0.9638	0.9726
	Accuracy	0.5000	0.8849	0.8595	0.8754	0.8563	0.8658	0.9073	0.9199	0.9135
Test	AUROC	0.5000	0.9510	0.9490	0.9490	0.9497	0.9310	0.9614	0.9559	0.9614
	Accuracy	0.5000	0.8734	0.8987	0.9114	0.8734	0.8734	0.8987	0.9114	0.9114

Table 2: *Summary of the car characteristics (CV = Cross-validation)*

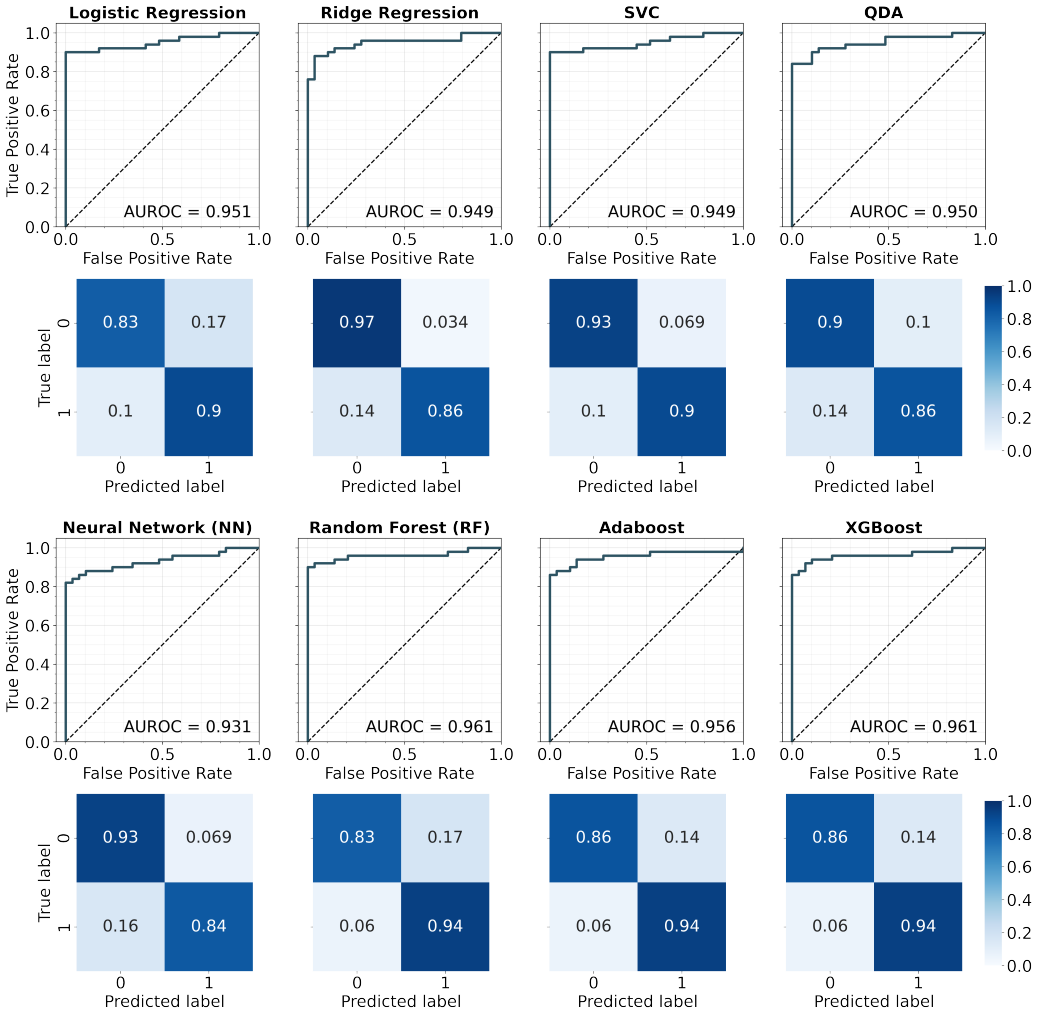


Figure 1: *Reciever operator curve (ROC) and normalized confusion matrix (0.5 threshold) on the test set*

1. **Logistic Regression**

Logistic regression models a probability of the target class (US origin in this case). It uses the sigmoid function to convert linear combination of the input features to the probability of the target class as follows:

$$p(y = 1 \mid X) = \frac{1}{1 + \exp(-\theta^\top X)}$$

The parameters of the logistic regression (θ) are the intercept and the coefficient corresponding to each feature. These are found using gradient ascent to maximize log likelihood of the set of observations. The performance of the logistic regression is remarkably good given its simplicity, getting an AUROC score of 0.9510 on the test set – significantly better than the random classifier. Additionally, from the ROC curve it may be noticed that the classifier performs well over most of the thresholds. The reason why logistic regression works so well on this problem is explained by the data used, which is likely to be well-separated.

2. **Ridge Regression**

Ridge regression, on the other hand, does not operate in terms of probabilities. Rather, it encodes the output variable as $\{-1, 1\}$ and treats the problem as the regression task with the regularization term added to the formulation corresponding to the $L2$-norm of the coefficients vector. The output of the ridge regression model is the value that is above or below 0 and that is what defines to which class the observation will be assigned to. The probability of each class may be inferred from the deviation of the output from 0. From the results, it is clear that ridge regression does not perform as well as the logistic regression on the cross-validation set, but produces similar performance on the test set. This is because these two regression formulations are similar in nature and given that the data is well separated, performance is expected to be similar.

3. **Support Vector Classifier**

Support Vector Classifier (SVC) is another way to fit a linear boundary between classes, defined as a linear combination of the input variables. Similarly to the ridge regression, output variable is encoded as $\{-1, 1\}$ and the model fits a hyperplane to maximize the margin between two classes – draw the largest possible margin between the closest observations to the boundary in either class. Because, in its nature, SVC doesn't use probabilities and performs a similar fit to the ridge regression, these models' performances are similar with the AUROC score being identical on the test set. Nevertheless, at 0.5 threshold, SVC performs significantly better than both logistic and ridge regression.

All three of the above methods assign a coefficient to each of the input features, making the models very interpretable. These coefficients are summarize in Table 3. There it can be confirmed that indeed boundaries produced by these methods are similar, as most of the coefficients have similar magnitude and sign. Looking closer at the individual coefficient values reveals that later model years tend to be more likely to come from the US, while cars that have higher MPG and more horsepower are likely to be produced elsewhere, as identified by all models. Further, ridge regression has a significantly smaller coefficient for the weight when compared to that in

Features	Logistic Regression	Ridge Regression	SVC
MPG	-0.0901	-0.0475	-0.0869
Cylinders	-1.3402	-0.1368	-0.8535
Displacement	0.1247	0.0126	0.1129
Horsepower	-0.0542	-0.0140	-0.0475
Weight	-0.0043	-0.0003	-0.0040
Acceleration (0-60mph)	-0.0020	-0.0089	0.0043
Model Year	0.1001	0.0581	0.0991

Table 3: *Summary of the linear regression, ridge regression and support vector classifier parameters*

the logistic regression model, suggesting that weight is a less important factor in identifying origin of the car. Similarly, the coefficient corresponding to the number of cylinders in the car is considerably smaller in the ridge regression model, underlining that this feature may be a not important one. Finally, SVC model has the oppositely signed coefficient for the acceleration when compared to the other two models, suggesting that this parameter is likely to be not that important as all models display similar performance.

4. Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a probablistic classifier that fits a multivariate normal distribution to each class and defines the decision boundary at the intersection of class distributions. This method heavily relies on the assumption that the classes are normally distributed what, it appears in this case, is reasonable but not perfect. As a result, the classifier performs well with the AUROC metric on the test set, but at low thresholds, the accuracy is lower than that of the other classifiers.

5. Neural Network

Multi-layer perceptron classifier is fit to the model in a similar fashion as the in Part 1. However, in this case, output of the model consists of the probability of each class being assigned to the observation. Due to the high complexity of the model and a very small dataset, this model does not perform as well because the bias of the fit decreases significantly with closer fit to the training data causing a corresponding increase in the bias on the test set. As a result, NN model overfits the training data and doesn't generalize well to the test set.

6. Random Forest / AdaBoost / XGBoost

Decision tree framework allows to directly predict the class the observation belongs to without looking into probabilities. Random Forest, AdaBoost, and XGBoost models are the same as those used for the regression as described in Part 1 with the difference being the output which is set to be a choice between class 0 or class 1. Additionally, the error used to define a split at each step in tree definition is called Gini index, which relies on the likelihood of misclassifying each observation. Since all of these methods fit an ensemble of trees, the final prediction is determined by the vote that each tree casts and the tree's weight. As such, in random forest, all trees are equally weighted, meaning that the probability of the observation belonging to a certain class is simply a proportion of votes cast by all trees for that class. On the other hand, AdaBoost and XGBoost have weights associated with each tree in the ensemble and so the probability is now

defined as the weighted average of votes. Parameters for all three models were selected using grid search on the cross-validation set. The number of estimators for Random Forest, AdaBoost, and XGBoost were selected at 300, 300, and 200 while the maximum depth is limited to 10, 4, and 2. Unlike in the previous regression problem, no subsampling is specified for the XGBoost model, since the dataset is too limited.

Once again, decision tree based models outperform the rest in the AUROC score and the ROC line is significantly above the baseline classifier, while AdaBoost and XGBoost do better than all other models in accuracy metric with 0.5 threshold. The reason why decision tree models work so well is that they are able to not only capture some non linearities in the decision boundary between classes but also to produce discontinuous boundary that none of the other model can do, meaning that certain discontinuous hidden trends in the data can be identified.

Model results on the test set can be interpreted by looking at the feature importance graphs shown in Figure 2. There, the importance of features varies between models, but engine displacement consistently appears in the top 3, underlining its importance for classification of the car origin. Conversely, number of cylinders has the lowest on average contribution among all three models, suggesting that this feature is not critical confirming findings from the ridge regression model. The rest of the features appear to have varying contributions among models, most likely meaning that all of them have a statistically significant level of importance in the problem.

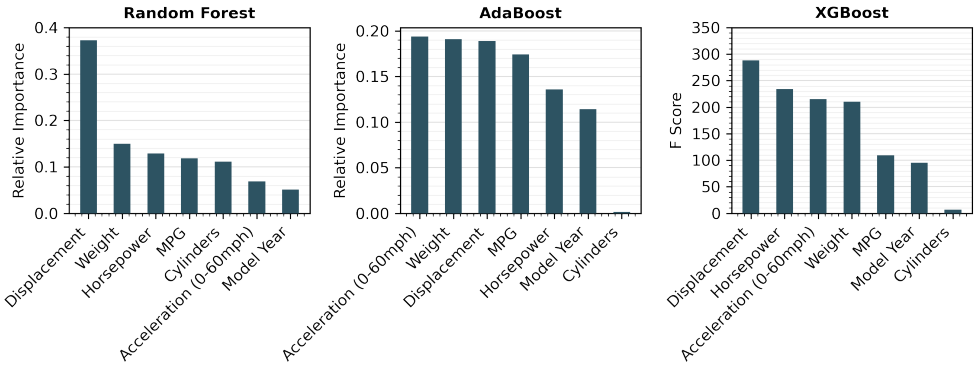


Figure 2: *Comparison of MSE for cross-validation and test sets with developed models*

Finally, considering both AUROC and accuracy at 0.5 threshold metrics on the test set, XGBoost outperforms the rest of the models and, thus, is selected as the best model from the analysis

3.2 Conclusion

To summarize, several classifiers were fit to the data by fine-tuning the parameters of each using the cross-validation set. All fitted models appeared to significantly outperform the baseline random classifier, showing that there are easily identifiable trends in the used dataset and classes are likely to be well separated. Decision tree models have shown to have the best performance primarily due to their ability to create discontinuous boundaries between classes. XGBoost model was determined as the best one based on the AUROC and accuracy at 0.5 threshold metrics on the test set.