Supervised Learning Approaches for Predicting Income from U.S. Census Data

STAT 432: Final Project[1]

Aria Shetty (aria4), Camille Dolce (cdolc2), Aryaman Kushwaha (ak65)[2]

Abstract

This paper aims to understand how we can best predict whether an individual has an income greater than $50,000, or has an income less than or equal to that value. We use data gathered from the 1994 U.S. Census by Kohavi & Becker (1996) to this binary variable based on a variety of quantitative and categorical variables relating to demographic information (age, race, etc.), employment information, and asset finances. We equip multiple classification models to compare their performance in executing this task. Specifically, we compare the performances of penalized and non-penalized logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), decision trees, and support vector machines (SVMs). Ultimately, we find that all have test errors below 0.25, but that the best performing model, a random forest decision tree, had the lowest classification testing error of 0.1332891.

[1] See appendix for supplementary code repository

[2] Each member made the following contributions:
Camille Dolce: Abstract, Literature Review, Logistic Regression, Support Vector Machine
Aria Shetty: Introduction, Decision Trees and Ensemble Methods, Support Vector Machine, Model Comparison, and Conclusion
Aryaman Kushwaha: Data Preprocessing and Summary Statistics, LDA and QDA, Support Vector Machine, and Conclusion

Introduction

Predicting income levels based on demographic and socio-economic factors is important in understanding income inequality, salary choices, and career prospects. The dataset consists of data from the "Census Income" dataset in the UCI Machine Learning Repository (Kohavi & Becker, 1996), which has 48,842 observations and 14 total variables from the 1994 U.S. Census. This data aims to predict whether annual income exceeds $50,000 a year or not. The data has 13 continuous and categorical predictors consisting of different demographic and socio-economic measures. The continuous predictors are age, capital gain, capital loss, and hours per week. The categorical predictors are race, relationship, sex, occupation, marital status, education, and work class. This report aims to compare various classification methods in the task of predicting the income of a given person taken from a sample of this data. We use various methods such as logistic regression, penalized logistic regression, linear and quadratic discriminant analysis, decision trees, ensemble methods (bagging and random forests), and support vector machines. We aim to identify which method performs best. Our findings revealed that the random forest model achieved the lowest test error which suggests the best predictive performance.

Literature Review

Predicting one's income is an important task for economists in better understanding the determinants of income levels or disparities. For example, past scholarship on the issue has examined the role of demographic factors (marriage, fertility, age) or on the overall income distribution and inequality (Lam, 1997). On the topic of finances, one's capital gains, or profits one gains when selling an asset, have been shown to increase a nation's capital share of income while also increasing income inequality (Robbins, 2018). Given that our data includes a variety of predictors ranging from demographic and employment characteristics to capital gains and losses from one's assets, we will be able to compare these variables against each other through the use of our various classification models. Predicting income is important due to its varying effects on economic development (Stewart, 1999), long-term economic growth in the US (Partridge, 2005), children's development (Cooper & Stewart, 2021), and population health (Lynch & Kaplan, 1997). In our comparison of which classification models are best at predicting and classifying the income level of an individual, our findings will help scholars better predict this characteristic based on a series of both quantitative and categorical predictors.

In executing this task, we use several supervised learning techniques, which classify "a vector into one of several classes by looking at several input-output examples of the function" (p. 2). Past scholarship comparing various supervised learning techniques – such as our models of penalized logistic regression, LDA, QDA, decision trees, and support vector machines (SVMs) – have looked at which specific measurements of accuracy they each perform the best. For example, Osisanwo et al. (2017) found that in their study, SVM was generally more accurate than decision trees, but that decision trees were able to tolerate missing values more than SVMs. Given that selecting the best method is project-dependent, it is important to compare multiple, as we do in our paper. Our first method, logistic regression is used "to predict the probability that a new [observation] belongs to one of the target classes" of the output variable (Bisong, 2019, p. 243). This method has been found by scholars to be beneficial when penalized as a way to deal with multicollinearity and overfitting (Eilers, 2001). Our next models, Linear and Quadratic Discriminant Analysis (LDA and QDA) combine variables in order to maximize "the differences between predefined groups" (Singh et al., 2016, p. 1312) by finding either linear or quadratic boundaries. The difference in whether LDA or QDA performs better appears to be dependent on the dataset and the normality of the variables (Bate-Eya, 2024), so it is necessary to compare the two against each other.

The third model we use, decision trees, are used to classify "a population into branch-like segments that construct an inverted tree with a root node, internal nodes, and leaf nodes" (Ying, 2015, p.1). They are beneficial in their interpretability and their nonparametric nature, meaning that there are no assumptions about the data. In this paper, we fit a tree with the Gini index, and use the ensemble methods of bagging and Random Forest. Bagging and Random Forest are similar in concept given that both "[build] multiple models using different training data" (McClarren, 2021, p. 64), taken by sampling the data with replacement, that are then averaged together. Random Forest, specifically, "the concept of bagging to include random sampling of the independent variable" (McClarren, 2021, p. 65). McClarren argues the benefit in using Random Forest is that the training data and input variables for the splits being chosen randomly means that there is low correlation between the trees, and the average of the tree will have less errors.

Lastly, we use Support Vector Machines (SVMs), which are classifiers that separate data by a hyperplane. SVMs are beneficial in that they "apply a simple linear method to the data but in a high-dimensional feature space non-linearly related to the input space" (Karatzoglou et al., 2006, p. 1). Due to their simplicity and their ability to handle nonlinear data using 'kernel tricks' (a way to use kernel functions to map the data without using a higher dimension), they are quite useful. In reviewing the aforementioned models, it appears that they all have their comparative advantages. Given all of this, it is necessary to compare each model for their performance in a specific task. With predicting a binary categorical predictor, this paper aims to see whether these stated models are successful in identifying when an individual has an income greater than $50,000, or has an income less than or equal to that value.

Data Preprocessing and Overview

We are looking at a dataset which consists of 48,842 observations and 14 total variables taken from the 1994 U.S. Census (Kohavi & Becker, 1996). When first analyzing the dataset we noticed that it had missing values in some of the predictors. Furthermore, we saw a few columns that had data that were not significant to determining our response. We first dropped the following columns: fnlwgt (Final Weight), education.num (Education as a numeric value), and native.country (Native Country). We dropped fnlwgt because it is neither a continuous or categorical predictor (it is a survey weight used when estimating distributions among subgroups), education.num because the representation of the values within are already covered by another predictor "education", and finally native.country because there was not much variation in the data as most of the observations were represented in the USA only. Next, we handled the missing values of the remaining 11 predictors by first implementing the mice package.

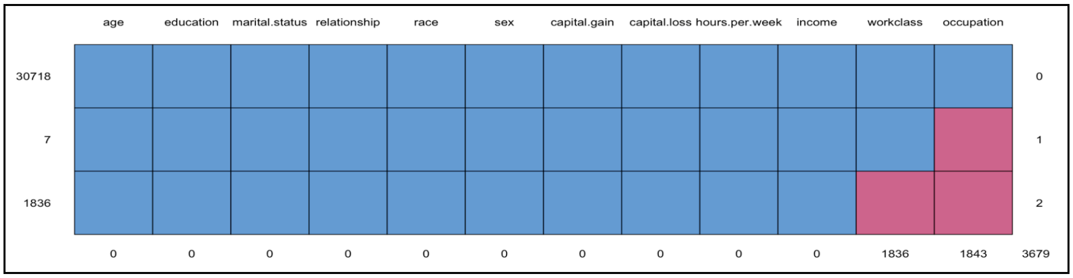
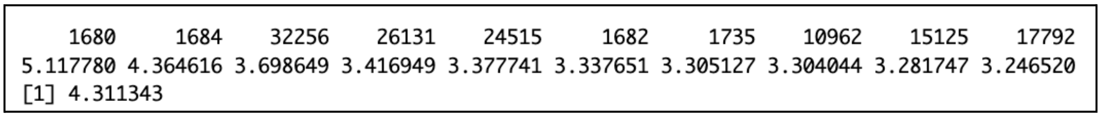


Figure 1: Missing pattern matrix

Based on the following missing pattern matrix, we see that there are a total of 1836 missing entries for “workclass” and 7 missing entries for “occupation” resulting in a combined total of 1843 missing entries for both predictors. Intuitively, if a person is not in the workclass then they also do not have an occupation therefore resulting in both entries per person being empty for the most part. As a result, we dropped this many values out of the entire dataset containing 30K entries. Afterwards, for Exploratory Data Analysis, we had 4 numerical predictors where we plotted histograms and 7 categorical predictors for which we plotted bar plots.

Next, we checked for potential outliers in our dataset using the Bonferroni correction. After computing a critical t-value with Bonferroni correction we have the absolute critical value at 4.311343. After computing the studentized residuals we found that two observations: 1680 and 1684 had studentized residuals of 5.117780 and 4.364616 respectively. Since both residuals are greater than the critical value, we have two outliers in the dataset hence we removed those rows from the entire dataset and refit the model.



```
    1680     1684    32256    26131    24515     1682     1735    10962    15125    17792 
5.117780 4.364616 3.698649 3.416949 3.377741 3.337651 3.305127 3.304044 3.281747 3.246520 
[1] 4.311343
```

Figure 2: List of Studentized Students + Bonferroni Correction Critical Value

Afterwards, we delved into Exploratory Data Analysis where we plotted histograms for the continuous predictors and bar plots for the categorical variables.

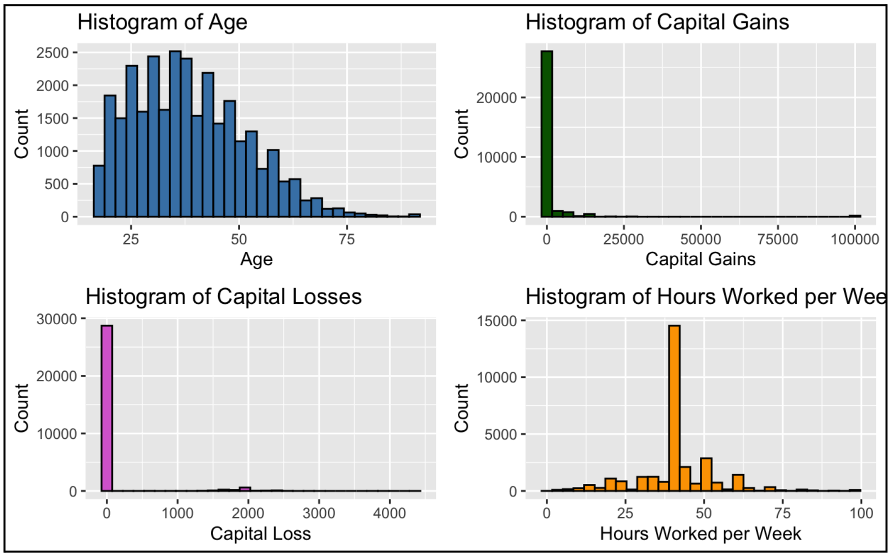


Figure 3: Histograms of the Continuous Predictors

From the above histograms, we see plots of the four continuous variables, Age, Capital Gains, Capital Losses, and Hours Worked per Week. We see that the graph of Age is mostly symmetric with a slight right-tail-skew but the graphs Capital Gains, Capital Loses, and Hours Worked per Week are extremely right-tailed-skewed. This means that the extreme values on the right are pulling the mean of the data to be greater than its median. Intuitively, this makes sense because most of the working class population lies from early 20's to late 50's working their 8 hour jobs throughout the week. Furthermore, very few people sell their assets hence resulting in little capital gains/losses and reflecting on most of the observations being equal to 0.

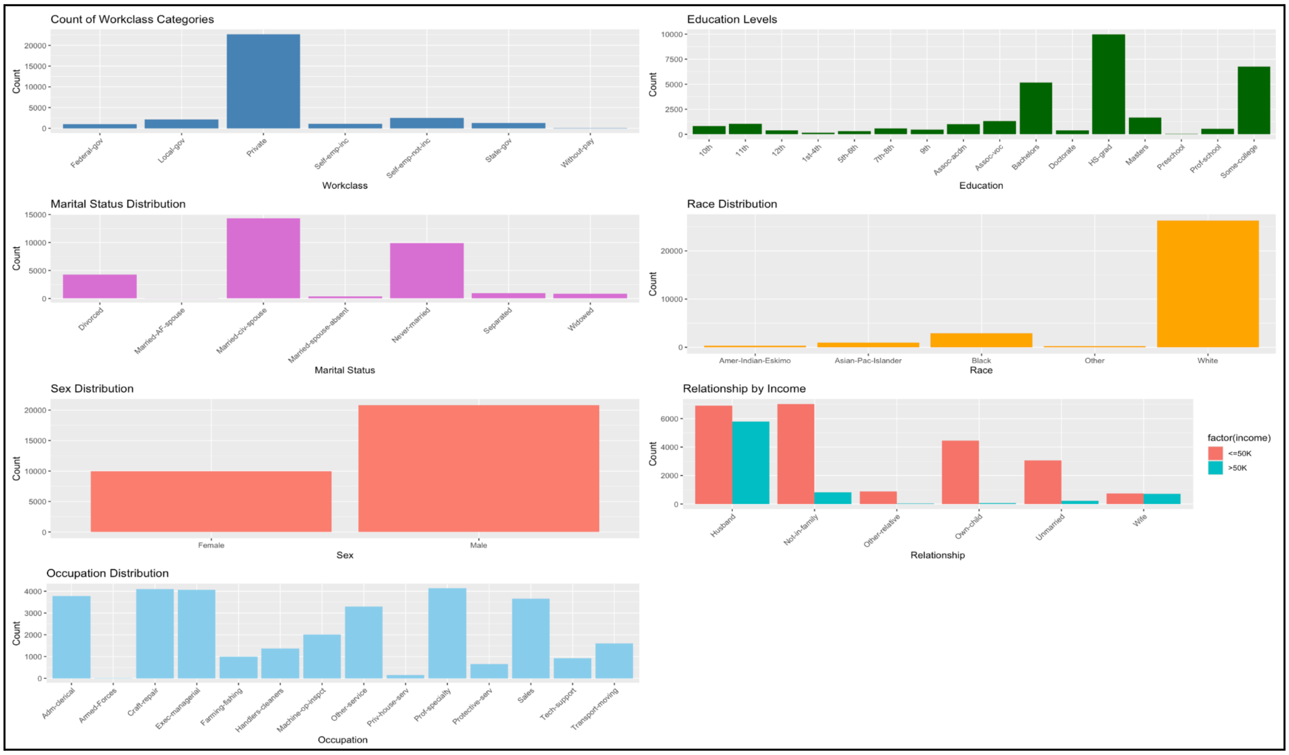


Figure 4: Bar plots of the Categorical Predictors

From the 7 bar plots of the categorical predictors we see that people in the workforce are mostly working in a private sector (~83% of the sample) and that most of the sampled population were at least high school graduates. Under the marital status distribution, most people were either married or unmarried which indicates the general age of the population. For race and sex distributions, most of the people sampled were white (~80% of the sample) and males (~66% of the sample) which indicates that the dataset is not demographically balanced. Furthermore, this disproportion is also represented within the "relationship by income" plots especially amongst those who earn more than 50K salary. Finally, most of the people sampled worked jobs in sales, craft-repair, exec-managerial, and prof-specialty as compared to the other professions.

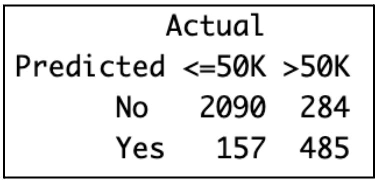
Methods

In the following subsections, we show how we trained each model, and their respective results. For all parts of our projects, the data is split randomly, with 90% of the data in the training set and the remaining 10% in the testing set. The seed for the code for each model and the data splitting function is set to 7, to ensure the replicability of our results and of our training-testing split.

I. Logistic Regression

Since our output variable is categorical (income is either >50k or <=50k), the first method we use is logistic regression, training multiple logistic regression models using the training data, and then predicting responses with the test data. In using multiple models, we aim to compare whether or not the penalized regression models we used (Elastic Net, Lasso, and Ridge) perform better than the general logistic regression model.

We first run a general logistic regression model on the training data with the output variable as the categorical income variable and all other variables mentioned as input variables. We find that for this model, all quantitative variables and at least one category for all input variables was statistically significant[3]. We then use this model to predict responses for the testing data. Using a confusion matrix with a cutoff of 0.5, we compare how successful the logistic regression model is in accurately predicting the income output on the testing data, as depicted in the confusion matrix in Figure 5. It appears that the model was generally successful in doing this task. The test error rate is reported as 0.146220159151194, which is not ideal but still relatively low. This model notably performs well in identifying true negatives, as the specificity score is 0.930129060970182.



```
           Actual
Predicted <=50K >50K
       No  2090  284
      Yes   157  485
```

Figure 5: Confusion matrix for the general logistic regression model

Next, we fit various penalized regression models to the training data in order to compare it to the general model. Adding a penalty to the coefficients of our model through penalized logistic regression aids in maximizing the log likelihood while minimizing the penalty term in order to find the best fit model. In this paper, we compare the general logistic regression model with three penalized models: Elastic Net, Lasso, and Ridge. We use the same training dataset for each model, and the same testing set to predict the model onto.

In order to choose our hyperparameter lambda (λ) for each model, we do cross-validation on each model with 30 folds to ensure a lower bias. We then easily compute the value of lambda that is the minimum mean-cross validated error (λ_{min}) and the largest value that is within 1 standard error (1 SE) of the errors of that minimum (λ_{1se}). We ultimately choose to use the values for λ_{1se} for each model given that λ_{1se} allows for more regularization. In implementing the models in R, we use the standard alpha (α) parameters for Lasso (1) and Ridge (0), and we set α to 0.75 for Elastic Net. We set the number of folds to 30 for each model. Given Lasso's advantage in being able to perform variable selection, it is worth noting that the model we chose with λ_{1se} removed the variable "workclass". While Elastic Net also performs variable selection, in this case with λ_{1se}, it did not.

Figure 10 in the appendix and Table 1 below shows the confusion matrices, test error, sensitivity, specificity, and precision rates for each model in comparison to the general model. Interestingly, we can see that the logistic model reports the lowest test error rate out of the 4 models. However, the specificity and precision scores of the penalized models are very high, suggesting that they are better at identifying true negatives and are less likely to predict false

[3] See appendix for further details on which dummy variables within which categorical variables were significant.

positives. Conversely, the general logistic regression model has a significantly higher sensitivity score (yet still not ideal), meaning it is more successful in finding true positives.

	Test error	Sensitivity	Specificity	Precision
General Model	0.146220159151194	0.630689206762029	0.930129060970182	0.755451713395639
Elastic Net	0.215848806366048	0.174252275682705	0.992879394748554	0.893333333333333
Lasso	0.213196286472149	0.183355006501951	0.993324432576769	0.903846153846154
Ridge	0.219496021220159	0.166449934980494	0.990654205607477	0.859060402684564

Table 1: Error Rates for each logistic regression model

Another way in which we can compare these 4 models is by their area under the curve (AUC) and receiver operating characteristic (ROC) plot. Figure 6 below shows the AUC values of each model, and Figure 11 in the appendix shows their respective ROC plots. We can see that all models perform well at the threshold of AUC=0.8, but that the general logistic regression performs the best with an AUC of 0.910892315313657 and with the best performing ROC curve.

	General Model	Elastic Net	Lasso	Ridge
AUC Value	0.910892315313657	0.808391249016901	0.808727776321332	0.806173004549345

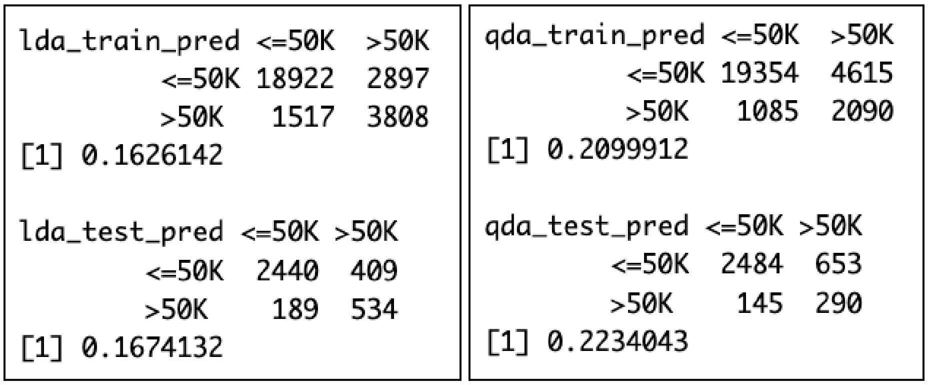
Figure 6: AUC scores for each logistic regression model

Overall, the general logistic regression model performs better than penalized models (Elastic Net, Lasso, and Ridge) in test errors (test error & sensitivity) and AUC analysis. Lasso has the highest specificity and precision scores, however the logistic model is able to sufficiently balance high sensitivity and specificity scores, while the other models have low sensitivity scores.

II. Comparison of LDA vs. QDA

Next on our 90-10 training split we compared the models LDA and QDA by first fitting both models on the training dataset. From our earlier logistic regression analysis we saw that the data carries a linear decision boundary. With LDA and QDA, we wanted to further test whether the same linear boundary has a better overall performance or whether the flexibility of a quadratic boundary improves the performance. We evaluated the models on both the training and testing sets to assess their performances and to check for overfitting using the classification error metric generated from two separate confusion matrices. As a result, LDA's error values on both the training and testing sets were 0.1626142 and 0.1674132. For QDA, the respective error values on both the training and testing sets were 0.2099912 and 0.2234043. Based on these results, we see LDA resulting in lower error terms and higher accuracy values of 83.7% and 83.3% respectively. In contrast, QDA resulted in higher error terms and lower accuracy values of

79.0% and 77.7%, indicating that the linear decision boundary captured by LDA outperforms the quadratic boundary of QDA for this dataset[4].



```
lda_train_pred <=50K  >50K
         <=50K 18922  2897
         >50K   1517  3808
[1] 0.1626142

lda_test_pred <=50K >50K
        <=50K  2440  409
        >50K    189  534
[1] 0.1674132
```

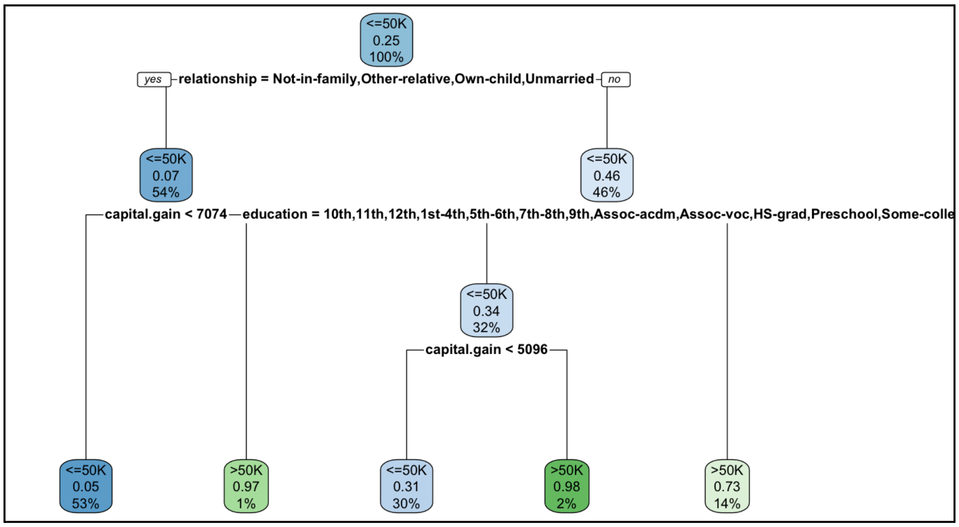
```
qda_train_pred <=50K  >50K
         <=50K 19354  4615
         >50K   1085  2090
[1] 0.2099912

qda_test_pred <=50K >50K
        <=50K  2484  653
        >50K    145  290
[1] 0.2234043
```

Figure 7: Confusion matrices of both LDA and QDA models

III. Decision Trees and Ensemble Methods

We fit a classification decision tree to the training set using the Gini index as the splitting criterion. The decision tree was visualized to reveal key splits based on relationship, education, and capital gain.

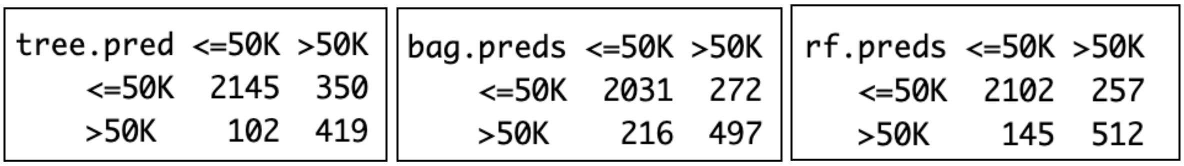


<=50K
0.25
100%
yes
relationship = Not-in-family,Other-relative,Own-child,Unmarried
no
<=50K
0.07
54%
<=50K
0.46
46%
capital.gain < 7074
education = 10th,11th,12th,1st-4th,5th-6th,7th-8th,9th,Assoc-acdm,Assoc-voc,HS-grad,Preschool,Some-colle
<=50K
0.34
32%
capital.gain < 5096
<=50K
0.05
53%
>50K
0.97
1%
<=50K
0.31
30%
>50K
0.98
2%
>50K
0.73
14%

Figure 8: Decision Tree using the Gini index splitting criterion

[4] When fitting both the LDA and QDA models on the training dataset using our 11 predictors, we experienced no errors for LDA, but when fitting the QDA model we received a grouping error that says that there is a rank deficiency in the group of people who earn more than 50K salary. We initially thought that this was an issue when fitting specifically the categorical variables because for QDA, each class has its own covariance matrix whereas for LDA all classes share the same one. We also tried converting all 7 categorical variables into a factor using the factor() function for R to handle the dummy variables produced by those categorical variables. However, after refitting the model we still had the same rank deficiency error. Therefore, we resolved this issue by removing the categorical predictors and instead fitted the model on the training dataset with the 4 remaining continuous predictors (as they have full rank) to predict the training and testing errors for QDA as shown in figure 7. Error from R Output:
```
Error in qda.default(x, grouping, ...) : rank deficiency in group >50K
```

For example, individuals with lower capital gain were more frequently classified as earning less than $50,000. Predictions were made on the test set, and performance was evaluated by calculating the classification error rate. Figure 9 below shows the confusion matrices for each model.



```
tree.pred <=50K >50K
    <=50K  2145  350
    >50K    102  419
```

```
bag.preds <=50K >50K
    <=50K  2031  272
    >50K    216  497
```

```
rf.preds <=50K >50K
   <=50K  2102  257
   >50K    145  512
```

Figure 9: Confusion matrices for the general Decision Tree, Bagging Model, and Random Forest Model

The decision tree model had a test classification error of 0.1498674. However, decision tree models can have low predictive power due to high variance of estimates. So, we compared this model with ensemble methods in order to improve predictions. We fit a bagging model to the training set to reduce the variance of decision trees, and we made predictions on the test set. Surprisingly, the bagging model achieved a classification error rate of 0.1614721, which is higher than the single decision tree model. Next, we fit a random forest model to the training data and made predictions on the test set. The random forest model had the lowest classification error of 0.1329576. Therefore, the random forest model outperformed the single decision tree model and bagging model with the lowest classification error.

IV. Support Vector Machine

We also implemented three support vector machine models using linear, radial, and polynomial (degree 2) kernels. We tuned the cost hyperparameter with 5-fold cross-validation. For the linear kernel, the optimal cost selected was 0.1. Using this value for cost, the training error was 0.1530357, and the testing error was 0.1448939. For the radial kernel, the optimal cost selected was 10. Using this value for cost, the training error was 0.1428308, and the testing error was 0.1405836. For the polynomial kernel, the optimal cost selected was also 10. Using this value for cost, the training error was 0.1473991, and the testing error was 0.1465517. Ultimately, we find that this radial kernel with a cost of 10 was the best performing model. We find that this model has 9,557 support vectors. While this may seem like a lot initially, it should be noted that the training dataset we fit this model on consisted of 27,144 observations. Compared to the models mentioned in the aforementioned subsections, this model was able to produce a relatively similar training and testing error.

To visualize the SVM model's classification abilities, Figure 12 in the appendix shows examples of its classifications of income by numerical values such as capital gains, age, hours worked per week, and capital losses. Here, we can see that for capital gains, a variable in which the majority of observations are listed at 0 and a small minority have extreme values, it was classified most of those nonzero values as having an income greater than $50,000. From this, it appears that having a significant amount of capital gains (over around $10,000) is a predictor for having an income greater than $50,000, but age or hours per week are not when plotted against capital gains. Conversely, capital loss by age is less of a strict predictor of income, with several observations in the center classified are from both income groups and are similar in capital loss and age.

Conclusion

I. Model Comparison

To compare model performance across all the models, we examined evaluation metrics such as the classification error, AUC, sensitivity, specificity, and precision. Among all models, random forest had the lowest classification testing error of 0.1329576. The best SVM model with radial kernel had a cost of 10 and had the next lowest classification testing error of 0.1405836, followed by the general logistic regression model with a classification error of 0.14622. Penalized logistic regression models (Elastic Net, Lasso, Ridge) had higher test errors, all above 0.21, but the penalized logistic regression models all maintained high specificity and precision values. However, the general logistic regression model performed better in sensitivity which suggests stronger performance in identifying individuals with income over $50,000. The ROC curves and AUC scores further confirmed these patterns. The general logistic regression model achieved the highest AUC value at 0.910892315313657.

To summarize, we evaluated a range of models to predict income levels, using the UCI Adult dataset. Among all models, the random forest model demonstrated the best predictive performance, achieving the lowest error rate. However, the SVM model with radial kernel and logistic regression had comparable test errors, which suggests these models might be just as effective. Meanwhile, the logistic regression model had a higher classification error, but it provided a good balance between sensitivity and specificity. However, the penalized logistic regression models did not improve overall predictive accuracy. The SVM with radial kernel was the strongest support vector machine model, which suggests that it captured non-linear patterns that the other models might have missed. Based on the results, if predictive accuracy is the main goal, the random forest model seems to be the preferred model. However, the SVM model with a radial kernel is also a strong choice for capturing non-linear patterns.

II. Limitations and Future Research

A limitation of this study is the imbalance in demographic representation within the dataset. From the summary statistics, we know that there is a disproportion in what demographic groups are being represented (white races made up roughly 80% and males made up roughly 66% of the sample). Future work could evaluate certain confusion matrix metrics such as specificity, sensitivity, and accuracy along with finding the AUC for each subgroup of the race and sex predictors to evaluate their performances individually. That way, we are predicting our outcome variable, income, on those individual characteristics instead. Future research can also expand on our work by using additional ensemble methods such as AdaBoost in order to update the weights of observations for each decision tree fitting. Incorporating methods such as AdaBoost could improve performance.

Appendix

I. References

Bate-Eya, A. E. (2024). Empirically comparing the performance of LDA and QDA when classifying customer sales data with different properties of normality and equality of covariance matrices.

Bisong, E. (2019). Logistic regression. *Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners*, 243-250.

Cooper, K., & Stewart, K. (2021). Does household income affect children's outcomes? A systematic review of the evidence. *Child Indicators Research*, *14*(3), 981-1005.

Eilers, P. H., Boer, J. M., van Ommen, G. J., & van Houwelingen, H. C. (2001, June). Classification of microarray data with penalized logistic regression. In *Microarrays: optical technologies and informatics* (Vol. 4266, pp. 187-198). SPIE.

Karatzoglou, A., Meyer, D., & Hornik, K. (2006). Support vector machines in R. *Journal of statistical software*, *15*, 1-28.

Kohavi, R., & Becker, B. (1996). *Adult Income dataset (UCI Machine Learning Repository)*.

Lam, D. (1997). Demographic variables and income inequality. *Handbook of population and family economics*, *1*, 1015-1059.

Lynch, J. W., & Kaplan, G. A. (1997). Understanding how inequality in the distribution of income affects health. *Journal of health psychology*, *2*(3), 297-314.

McClarren, R. G. (2021). Decision trees and random forests for regression and classification. *Machine Learning for Engineers: Using data to solve problems for physical systems*, 55-82.

Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*(3), 128-138.

Partridge, M. D. (2005). Does income distribution affect US state economic growth?. *Journal of Regional Science*, *45*(2), 363-394.

Robbins, J. (2018). Capital gains and the distribution of income in the United States. *cit. on*, 54.

Singh, A., Thakur, N., & Sharma, A. (2016, March). A review of supervised machine learning algorithms. In *2016 3rd international conference on computing for sustainable global development (INDIACom)* (pp. 1310-1315). Ieee.

Stewart, F. (1999). Income distribution and development. United Nations Conference on Trade and Development.

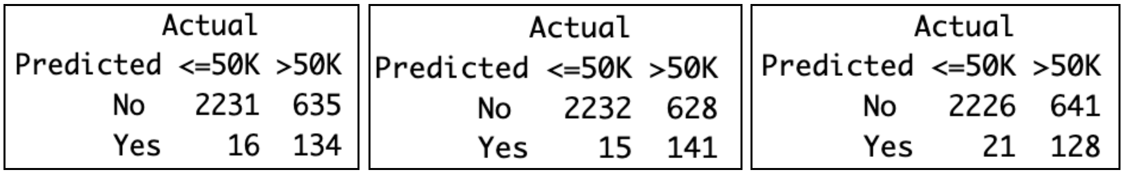
Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, *27*(2), 130.

II. Supplementary Code

Please see the following GitHub repository for the code used for data preprocessing, data visualizations, and for each 4 models:

https://github.com/camilledolce/STAT-432-final-project/tree/main

III. Figures



```
         Actual
Predicted <=50K >50K
      No   2231  635
      Yes    16  134
```

```
         Actual
Predicted <=50K >50K
      No   2232  628
      Yes    15  141
```

```
         Actual
Predicted <=50K >50K
      No   2226  641
      Yes    21  128
```

Figure 10: Confusion matrices for the Elastic Net, Lasso, and Ridge models

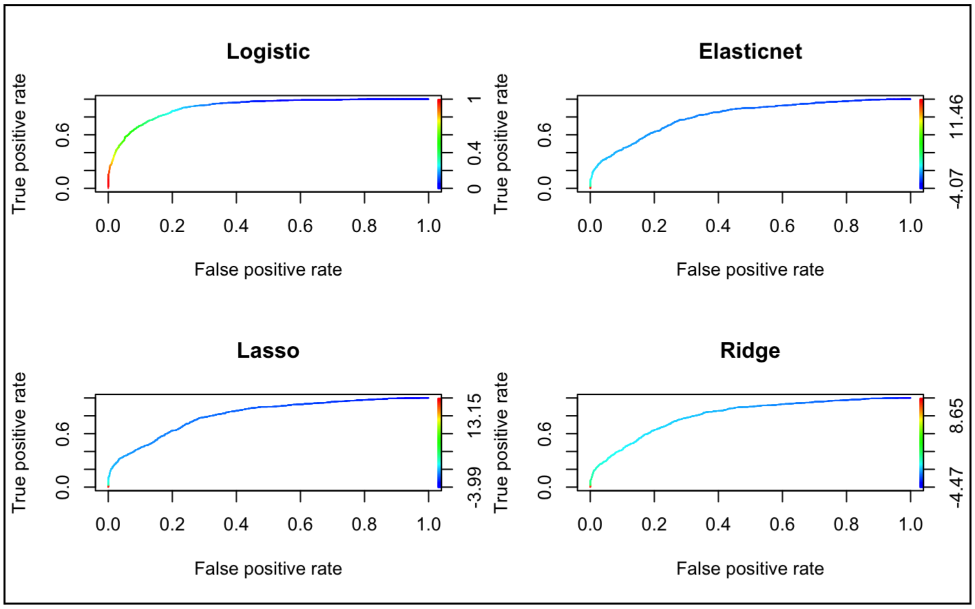
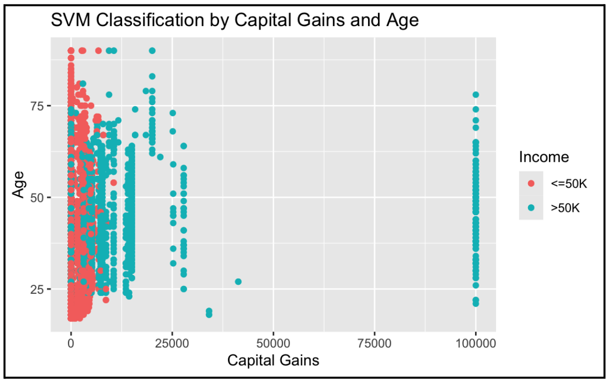
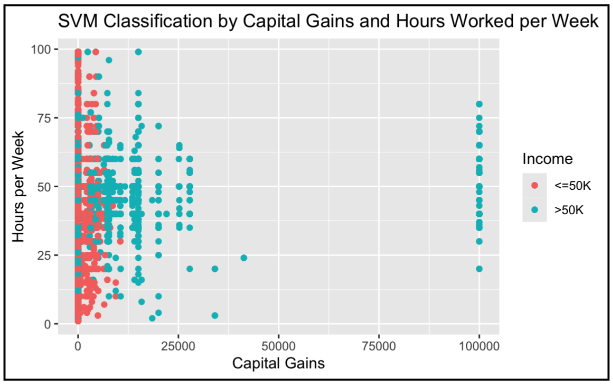


Figure 11: AUC scores for each logistic regression model





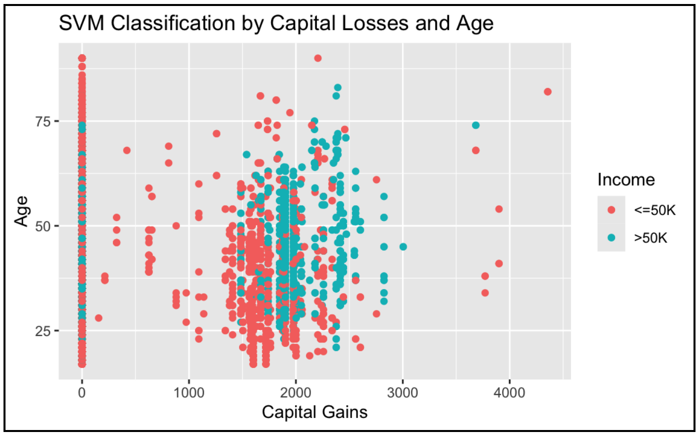
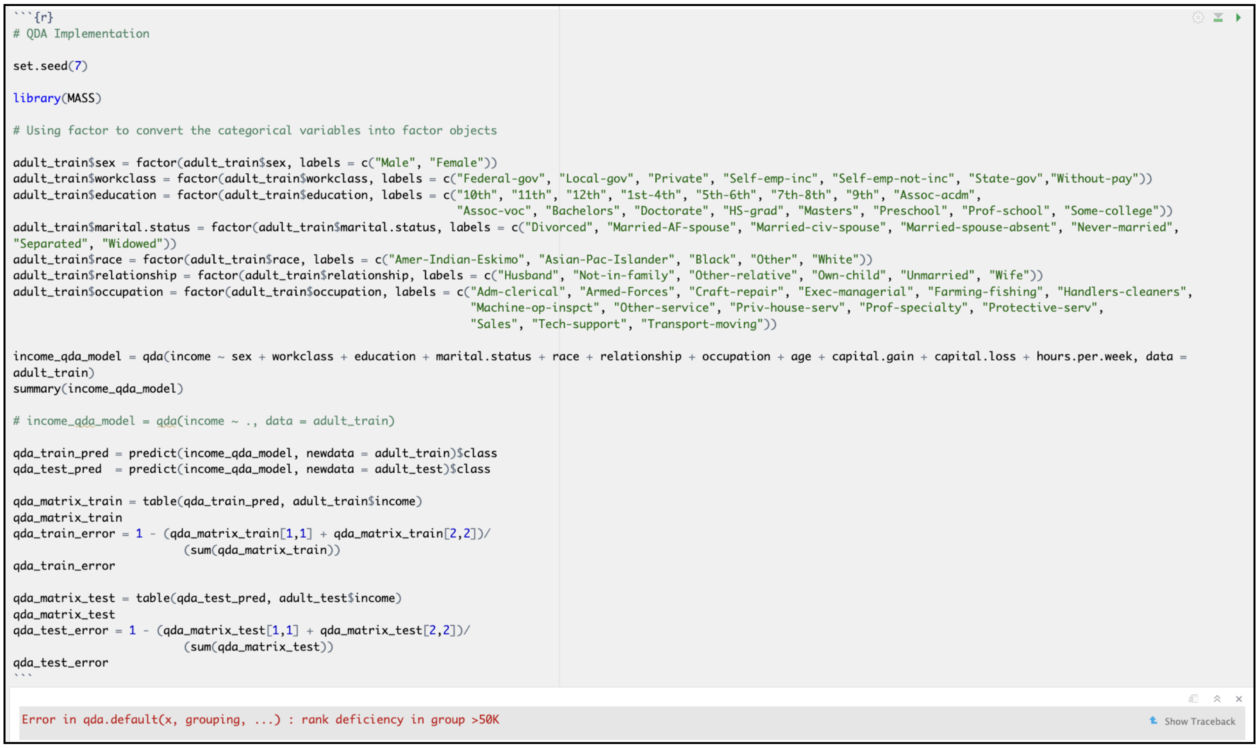


Figure 12: SVM classification graphs of income by various quantitative variables



~~~
```{r}
# QDA Implementation

set.seed(7)

library(MASS)

# Using factor to convert the categorical variables into factor objects

adult_train$sex = factor(adult_train$sex, labels = c("Male", "Female"))
adult_train$workclass = factor(adult_train$workclass, labels = c("Federal-gov", "Local-gov", "Private", "Self-emp-inc", "Self-emp-not-inc", "State-gov","Without-pay"))
adult_train$education = factor(adult_train$education, labels = c("10th", "11th", "12th", "1st-4th", "5th-6th", "7th-8th", "9th", "Assoc-acdm",
                                                                 "Assoc-voc", "Bachelors", "Doctorate", "HS-grad", "Masters", "Preschool", "Prof-school", "Some-college"))
adult_train$marital.status = factor(adult_train$marital.status, labels = c("Divorced", "Married-AF-spouse", "Married-civ-spouse", "Married-spouse-absent", "Never-married",
"Separated", "Widowed"))
adult_train$race = factor(adult_train$race, labels = c("Amer-Indian-Eskimo", "Asian-Pac-Islander", "Black", "Other", "White"))
adult_train$relationship = factor(adult_train$relationship, labels = c("Husband", "Not-in-family", "Other-relative", "Own-child", "Unmarried", "Wife"))
adult_train$occupation = factor(adult_train$occupation, labels = c("Adm-clerical", "Armed-Forces", "Craft-repair", "Exec-managerial", "Farming-fishing", "Handlers-cleaners",
                                                                   "Machine-op-inspct", "Other-service", "Priv-house-serv", "Prof-specialty", "Protective-serv",
                                                                   "Sales", "Tech-support", "Transport-moving"))

income_qda_model = qda(income ~ sex + workclass + education + marital.status + race + relationship + occupation + age + capital.gain + capital.loss + hours.per.week, data =
adult_train)
summary(income_qda_model)

# income_qda_model = qda(income ~ ., data = adult_train)

qda_train_pred = predict(income_qda_model, newdata = adult_train)$class
qda_test_pred  = predict(income_qda_model, newdata = adult_test)$class

qda_matrix_train = table(qda_train_pred, adult_train$income)
qda_matrix_train
qda_train_error = 1 - (qda_matrix_train[1,1] + qda_matrix_train[2,2])/
                         (sum(qda_matrix_train))
qda_train_error

qda_matrix_test = table(qda_test_pred, adult_test$income)
qda_matrix_test
qda_test_error = 1 - (qda_matrix_test[1,1] + qda_matrix_test[2,2])/
                         (sum(qda_matrix_test))
qda_test_error
```
~~~

```
Error in qda.default(x, grouping, ...) : rank deficiency in group >50K
```

Figure 13: Error in QDA Implementation

IV. Footnote Comments

As mentioned in footnote 2, there are several variables that factored out into dummy variables, with some being statistically significant and others not. For the following variables, the bolded dummy variables and bolded quantitative variables are those that are statistically significant at the p>0.05 level for the general logistic regression model:

- **Age**

- Workclass: **Local-gov**, **Private**, Sel-emp-inc, **Self-emp-not-inc**, **State-gov**, Without-pay
- Education: 11th, 12th, 1st-4th, 5th-6th, 7th-8th, 9th, **Assoc-acdm**, **Assoc-voc**, **Bachelors**, **Doctorate**, **HS-grad**, **Masters**, Preschool, **Prof-school**, **Some-college**
- Marital Status: **Married-AF-spouse**, **Married-civ-spouse**, Married-spouse-absent, **Never-married**, Separated, Widowed
- Occupation: Armed-Forces, Craft-repair, **Exec-managerial**, **Farming-fishing**, **Handlers-cleaners**, Machine-op-inspect, **Other-service**, Priv-house-serv, **Prof-specialty**, **Protective-serv**, Sales, **Tech-support**, Transport-moving
- Relationship: Not-in-family, Other-relative, Own-child, Unmarried, **Wife**
- Race: Asian-Pac-Islander, Black, Other, White
- Sex: **Male**
- **Capital Gain**
- **Capital Loss**
- **Hours (worked) per Week**