**Report on Support Vector Models for Classification in Washington State Housing Data**

**Abstract**

This report presents a study that utilizes support vector models for classification to predict whether a dwelling in Washington State is occupied by owners or renters. The data used in this analysis is sourced from the US Census via IPUMS USA and includes variables such as age, income, education level, cost of electricity, year of construction, and population density of the surrounding area. Three types of kernel functions - linear, radial, and polynomial - are employed to build support vector models, and their performance is assessed using a subset of the data. The report outlines the methodology used in building the models, presents the mathematical output of the models, and provides interpretations of the results in the context of the problem using the predictors "Age", "Household Income", "Number\_of\_Bedrooms", "Education\_Level", and "Serial household number", "Martial Status", "Ownership\_Status”, “Electricity Cost”. Additionally, interesting findings from the analysis are highlighted, and relevant questions for further investigation are posed.

**INTRODUCTION**

The dataset utilized in this study is obtained from the US Census via IPUMS USA and comprises a variety of predictors related to residences in Washington State. These predictors encompass population density, age of dwellings, household income, number of rooms and bedrooms, number of vehicles, education level, and ownership status and others. The dataset is expected to yield valuable insights into the factors that influence the occupancy status of dwellings, specifically whether they are occupied by owners or renters.

For data analysis, support vector models with three different kernel functions - linear, radial, and polynomial - are employed. Kernel functions are mathematical functions that transform the data into higher-dimensional spaces, allowing for non-linear classification. The utilization of multiple kernel functions enables a comprehensive analysis of the dataset, as each kernel function may capture distinct patterns and relationships within the data.

The main objective of this report is to develop accurate classification models using support vector machines (SVMs) with various kernel functions and evaluate their performance in predicting the occupancy status of residences. The anticipated results include identifying the most effective kernel function for this classification problem and gaining insights into the significance of the predictors in determining the occupancy status of dwellings.

Furthermore, this report will also highlight interesting findings from the analysis, such as identifying the most influential predictors in determining the occupancy status of dwellings, examining the relationships between different predictors, and exploring any patterns or trends observed in the data. Additionally, potential questions for further investigation may be posed, such as whether the findings align with existing literature or if there are any limitations or biases in the dataset or modeling approach.[(usa.ipums)](https://usa.ipums.org/usa/)

**Background**

**SUPPORT VECTOR MACHINE**

The Support Vector Machine (SVM) is a machine learning algorithm that is mainly used for classification purposes but can also perform regression tasks. It plots input data points in an n-dimensional space, where each dimension represents a feature of the data object. The algorithm then iteratively finds a hyperplane that can separate the data points into different categories. Although SVM is primarily designed for binary classification, it can also perform multi-class classification with multiple target output classes.

The SVM model represents the input data points in a graphical space with a clear gap between different categories. The **hyperplane** is a line or a plane that divides the space and separates different categories. This division helps to identify which section of the space belongs to which category. In a 2D space, the hyperplane is a line, while in a 3D space, it is a plane. The SVM algorithm aims to find a hyperplane that maximizes the margin, which is the distance between the two parallel dotted lines on either side of the hyperplane. The support vectors, which are the data points closest to the hyperplane, have the largest impact on the position of the hyperplane.

Chart, scatter chart

Description automatically generated

Figure1: Support Vector Machines (SVM) [techvidvan.com](https://techvidvan.com/tutorials/svm-in-r/)

As illustrated in the figure 1 above, the hyperplane in an SVM model has two dotted parallel lines on either side, and the perpendicular distance between these lines is referred to as the margin. The margin corresponds to the distance between the data points of the two different categories. The data points closest to the **hyperplane**, which are known as support vectors, have the greatest influence on the hyperplane's position. ([techvidvan](file:///Users/hanishak/Desktop/Hanisha/MSDS/spring/ML2/git/1.%20https:/techvidvan.com/tutorials/svm-in-r))

**HYPERPLANE**

A hyperplane is a math concept that is like a line or plane used to separate things. In machine learning, SVM is a tool that helps find the best hyperplane to divide data points into different groups. The goal is to find a hyperplane with the biggest gap between the groups. The hyperplane is important in machine learning because it helps classify and predict things.

**KERNEL (LINEAR, RADIAL, POLYNOMIAL)**

Kernel functions are used in SVM to handle data that cannot be separated in a straight line. The kernel function transforms the data points to a different space where a straight line can be used to separate them by using *f*(*x*)=*β*0+Σ*iϵSαiK* (*x*, *xi*), which separates hyperplane written with kernel K where *K*(*x*,*xi*) gives relationship between two points.

There are different types of kernel functions in SVM. The **linear** kernel function is used when the data points have a straight-line relationship. The formula for the linear kernel function is simple, it is just the dot product of two data points.

Text

Description automatically generated with low confidence

Fig2: Linear kernel

The **polynomial** kernel function allows for curved boundaries between data points. The formula for the polynomial kernel function is a bit more complicated, it takes the dot product of two data points and raises it to a positive integer power.

Diagram, schematic

Description automatically generated

Fig3: Polynomial Kernel

The **radial** kernel function allows for even circular boundaries. The formula for the radial kernel function is exponential and involves the distance between two data points.

Text

Description automatically generated with low confidence

Fig 4: Radial Kernel

**MAXIMAL MARGIN CLASSIFIER**

The maximal margin classifier is a linear model that separates data points into two classes using a straight-line decision boundary. The boundary is defined as the line that is equidistant from the closest points of each class, known as support vectors. The aim of the maximal margin classifier is to maximize the distance between the decision boundary and the support vectors.

**SUPPORT VECTOR CLASSIFIER**

The support vector classifier is a variant of the maximal margin classifier that allows for some misclassifications of data points. This is used when the data is not completely separable using a linear boundary. In such cases, the decision boundary is allowed to have some misclassifications, with a penalty term assigned for each misclassified point. The goal of the support vector classifier is to minimize the penalty terms and maximize the margin.

**FITTING**

To fit the support vector classifier, the data points are first transformed using a kernel function that maps the points to a higher-dimensional feature space where linear separation is possible. The support vector classifier then tries to find the optimal hyperplane that separates the transformed data points while minimizing the penalty term.

**TUNING PARAMETERS: COST, GAMMA, DEGREE**

The performance of the support vector classifier is dependent on the choice of the kernel function and its hyperparameters. The kernel function and its hyperparameters need to be tuned to achieve the best classification accuracy. Cross-validation is commonly used to tune the hyperparameters of the kernel function. The process involves splitting the data into training and validation sets, fitting the model on the training set, and evaluating the performance on the validation set. The hyperparameter values yielding the highest classification accuracy are selected.

The cost parameter in SVM controls the misclassification rate based on the side of the test observation, with a higher cost resulting in a wider margin but more support vectors. Cross-validation is used to determine the optimal value of C, along with gamma for radial models and degree for polynomial kernel models. This process helps to balance bias and variance in the non-linear model.

**SVMS WITH MORE THAN TWO CLASSES**

SVMs can also classify problems with more than two classes by finding a decision boundary that separates the data points into multiple classes. There are two common approaches: One-vs-One and All-pairs Classification. One-vs-One trains a binary SVM classifier for every pair of classes, while All-pairs trains a binary SVM classifier for each class against all the other classes. Both approaches define the decision boundary for each binary classifier by maximizing the margin between the two classes. The kernel function and its hyperparameters need to be chosen and tuned for the best classification accuracy. Cross-validation can also be used to select the best approach and tune the hyperparameters.[(ISLR2)](https://hastie.su.domains/ISLR2/ISLRv2_website.pdf)

**METHODOLOGY**

**DATA CLEANING AND VARIABLE SELECTION**

The data cleaning and processing pipeline results in a more manageable and interpretable dataset that can be used for further analysis. The dataset is a representative sample of the US population and includes 75,000 records from the IPUMS USA 2019 Census. The code first subsets the data to select relevant columns and groups the data by SERIAL to obtain summary statistics. The summary statistics are then used to create new variables such as encoding categorical variables and checking if spouse is present or not, which are important in the housing context. The resulting dataset after data cleaning and processing contains 28,123 observations and 12 variables. The variables include age, education, household income, cost of electricity, number of bedrooms, ownership status, number of people in the household, and indicators for marital status, such as isSpousePresent, isSpouseAbsent, isDivorced, isSingle, and isWidowed. The summary statistics are calculated for each household, including the mean age of individuals over 30, the highest level of education attained by any individual over 30, and the maximum income and electricity cost for the household.

The second step involves encoding categorical variables, such as ownership and marital status, into factors for modeling purposes. Ownership status is converted to a factor with two levels, "Owned" and "Rented." Marital status is then transformed into separate factors as mentioned above. The "isSpousePresent" variable is created based on the assumption that the highest age person in the household who is married is considered to have a spouse present. Similarly, the "isSpouseAbsent," "isDivorced," "isSingle," and "isWidowed" variables are created based on the highest age person's marital status. We ensured that the dataset had no missing values by eliminating any observations with NA values. These steps helped to prepare a clean dataset that is appropriate for analysis using support vector models.

After cleaning, the dataset was further processed by removing unnecessary columns and scaling the numerical variables. A random sample of 2000 observations were selected with replacement using set. seed () function. Categorical variables were excluded from scaling. The dataset was split into training and testing datasets using a 70-30 split, and 70% of the observations were randomly selected as the training set. The remaining 30% of the observations were assigned to the testing set.

**Support Vector Machines (SVMs) Predictive Model:**

The aim of the project was to predict whether a house is owned or not, based on a set of predictors, using support vector machines (SVM). After data cleaning, we can split the data into training and testing sets to evaluate the model's performance. The SVM algorithm involves selecting a kernel function, which is used to map the input data into a higher-dimensional space. This is necessary as it allows the SVM to perform classification in a nonlinear feature space. There are several types of kernel functions available, including linear, radial, and polynomial kernels. The choice of kernel function depends on the nature of the data and the problem we are trying to solve.

The SVM models' performance was optimized through tuning the cost parameter, which gives a balance between the classification accuracy on the training data and the decision boundary's complexity. The optimal value was selected using cross-validation by identifying the lowest error rate on the testing data. Once the best model was identified, its performance was evaluated using both the test and train error rates. The test error rate was given greater importance since it measured the accuracy of the model on new data. The summary of the models was used to determine the number of support vectors, and the coefficient check technique was utilized to identify the potential strong predictors of ownership status. The performance of the models was evaluated using plots of paired variable predictors to visualize the decision boundary and identify areas of misclassification. After training, we can evaluate the performance of the model on the testing data by calculating various metrics such as accuracy. We can also generate visualizations such as confusion matrices to better understand the model's performance.

To generate the computational results, the dataset was first cleaned by removing unnecessary columns. The dataset was then randomly sampled with replacement, and the categorical variables were excluded from scaling. The dataset was split into training and testing datasets, with 70% used for training and 30% for testing. The R code used for cleaning, building, and generating the computational results of the models is attached in the appendix.

**RESULTS**

**Linear kernel SVM Model for Predicting Household Ownership Status**

Support Vector Machine (SVM) model is used to predict the ownership status of households based on a set of features. A linear kernel is used, and the parameter 'cost' is tuned using 10-fold cross-validation with the following values: 0.001, 0.01, 0.1, 1, 5, 10, and 100. The best performing model is selected based on the cost parameter, and it is found to be 0.01 with a performance score of 0.1742857.

Chart

Description automatically generated

Fig 5: Age vs Household income using ownership status.

The model is trained on a training dataset, and its performance is evaluated on both the training and testing datasets. The testing accuracy is found to be high at 0.8316667, which suggests that the linear kernel works well for this dataset. However, the number of support vectors is relatively high, indicating that the model may be overfitting the data to some extent. The model correctly predicted the ownership status of 446 Owned households and 53 Rented households in the testing dataset, with a training accuracy of 0.8285714. Overall, the performance of the model is satisfactory, and it can be used for predicting the ownership status of households based on the given set of features.

Chart

Description automatically generated

Fig6: SVM classification plot for number of bedrooms and household income

The SVM plot of the number of bedrooms versus household income is displayed in Figure 6. The plot indicates the ownership status of the houses, but the hyperplane is not visible from this view. This suggests that these two predictors could linearly separate the two classes of ownership status. In other words, a line could be drawn to separate the data points based on these two predictors. The testing accuracy was found to be 0.8316667, which indicates that the linear kernel worked well for the given dataset. However, the number of support vectors was relatively high, which suggests that the model may be overfitting to some extent. Nonetheless, the overall performance of the model was satisfactory, and it can be used for predicting the ownership status of households based on the given set of features. A plot of the age of the neighborhood versus the number of bedrooms of the household members indicated that a combination of these predictors could linearly separate the two classes of ownership status.

Chart

Description automatically generated

Fig7: Hyperplane Plot for Number of Bedrooms and Age of the Member

In order to visualize the decision boundary of the SVM model, we plotted two variables from the scatter plot. Figure 7 shows that the hyperplane of the linear SVM model is easily identifiable, with a series of support vectors in both classes. The presence of a clear hyperplane while plotting the number of bedrooms and the age suggests that these variables could be good predictors of ownership status.

However, it is worth noting that many points fall outside the margins that are classified as support vectors. This indicates that the model is incorrect in these cases, and this is likely because the data does not have a linear boundary for the ownership status classes. Therefore, the non-linear and radial SVM models outperformed the linear model in terms of test error rate, as they are better suited to handle non-linear decision boundaries.

**Radial Kernel SVM Model**  
  
The non-linear SVM model with the radial kernel was used to predict household ownership status. The model was optimized using the tune function in R with various gamma and cost values. The optimal model had a gamma value of 0.2 and a cost of 1, with 649 support vectors. However, the training error rate was 0.13, while the test error rate was 0.15, indicating some overfitting.

To evaluate the model's performance, confusion matrices were generated for the training and test sets. The training set confusion matrix indicated that the model correctly classified 1009 "owned" homes and 205 "rented" homes but misclassified 150 "owned" homes and 36 "rented" homes. The test set confusion matrix showed that the model correctly classified 453 "owned" homes and 56 "rented" homes but misclassified 69 "owned" homes and 22 "rented" homes.

Chart, histogram

Description automatically generated

Fig8: Radial kernel for Education Level vs Age. Fig9: Radial kernel for number of bedrooms vs Age

To visualize the decision boundaries of the radial kernel model, scatter plots were created using various combinations of variables. Figure 9 shows the scatter plot of the number of bedrooms versus age, while Figure 8 shows the scatter plot of education level versus age. The plot with education level and age produced a clearer non-linear boundary compared to the plot with age and number of bedrooms in the household, although both plots had misclassifications. These visualizations highlight the importance of selecting appropriate variables for the radial kernel model to predict household ownership status.

**POLYNOMIAL KERNEL SVM MODEL**

In addition to the radial kernel, the polynomial kernel was used to model the household ownership status in the dataset. The tune function in R was utilized to optimize the model with different cost and degree values. The optimal model had a cost of 1 and degree of 3, with 561 support vectors, while the training error rate was 0.15, and the test error rate was 0.15. This indicates that the model was not overfitting or underfitting the data.

To evaluate the performance of the polynomial kernel model, confusion matrices were generated for the training and test sets. The training set confusion matrix indicated that the model correctly classified 1008 "owned" homes and 184 "rented" homes but misclassified 171 "owned" homes and 37 "rented" homes. The test set confusion matrix showed that the model correctly classified 455 "owned" homes and 56 "rented" homes but misclassified 69 "owned" homes and 20 "rented" homes.

Chart

Description automatically generated

Fig10: Polynomial kernel model for Number of Bedrooms vs Electricity cost

To visualize the decision boundary of the polynomial kernel model, a scatter plot was created showing the relationship between the number of bedrooms in a household and the electricity cost. Figure 10 demonstrates the decision boundary, which reveals a clear separation between the owned and rented households. The visualization highlights the importance of selecting appropriate variables for the polynomial kernel model to predict household ownership status.

**DISCUSSION**

The SVM model with a linear kernel is used to predict ownership status based on a set of features, with the best performing model having a cost parameter of 0.01 and a performance score of 0.1742857. The testing accuracy is high at 0.8316667, but the number of support vectors is relatively high, indicating possible overfitting. The scatter plot of the number of bedrooms versus household income suggests that these predictors could linearly separate the two classes of ownership status. However, for hyperplane view many points fall outside the margins, indicating incorrect predictions, likely due to non-linear boundaries.

A non-linear SVM model with the radial kernel was used to predict household ownership status. The optimal model had a gamma of 0.2 and a cost of 1, with 649 support vectors. The model had a training error rate of 0.13 and a test error rate of 0.15, indicating some overfitting. Scatter plots were used to visualize the decision boundaries, and it was found that the variables used can affect the accuracy of the model. The confusion matrices showed some misclassifications, indicating room for improvement.

A polynomial kernel SVM model was used to predict household ownership status, achieving a training and test error rate of 0.15 with 561 support vectors, a cost of 1, and a degree of 3. The training set correctly classified 1008 owned homes and 184 rented homes but misclassified 171 owned homes and 37 rented homes. The test set correctly classified 455 owned homes and 56 rented homes but misclassified 69 owned homes and 20 rented homes. A scatter plot was used to visualize the decision boundary, showing a clear separation between owned and rented households based on the number of bedrooms and electricity cost.

**CONCLUSION**

This study compared the performance of different SVM models in predicting housing ownership status based on demographic and socio-economic variables. The results showed that the polynomial model was the most accurate, while the radial model was prone to overfitting. Variable combinations produced different separation boundaries, emphasizing the importance of exploring multiple combinations.

The study found that household income, number of bedrooms, age, and cost of electricity were strong predictors of home ownership. Those with higher incomes and age tended to own larger homes.

In summary, this study gives significant aspects that need to be considered in addressing societal issues related to home ownership status.

**CITATION**

1. <https://techvidvan.com/tutorials/svm-in-r/>
2. <https://www.listendata.com/2017/01/support-vector-machine-in-r-tutorial.html>
3. Steven Ruggles, Sarah Flood, Matthew Sobek, Danika Brockman, Grace Cooper,  Stephanie Richards, and Megan Schouweiler. IPUMS USA: Version 13.0 [dataset]. Minneapolis, MN: IPUMS, 2023. <https://doi.org/10.18128/D010.V13.0>
4. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data
5. Mining, Inference, and Prediction (2nd Edition). Retrieved from:<https://hastie.su.domains/ISLR2/ISLRv2_website.pdf>