Short Report on Predicting the Prices of House in the California Area Using Machine Learning

The Shiny app, "Predicting the Prices of the House in the California Area using Machine Learning," gives insights into housing prices in California. We chose to use this data because house prediction models are fundamental as we need certain assumptions on the house prices. This is just an example of the data we used. Later, we can use this idea with different data from a different place. So, it will be very useful for anyone who wants to predict housing prices where they want to buy one. This shiny app has different data visualizations, exploratory data analysis (EDA), and predictive modeling which has different elements inside it. We imported varieties of libraries like dplyr, tidyverse, and janitor for data manipulation and analysis; ggplot2 and plotly for visualization; tidy models, caret, and randomForest for machine learning and `shiny`, shinydashboard, and shinythemes for the shiny app framework.

Data preprocessing was very important in our project. At first, we imported the California house prediction dataset from Kaggle. At first, we loaded the dataset and then changed "median_house_value" into two categorical variables, high and low, as it is the variable we are interested in predicting. After that we used the Tigris package to find the counties for each house using spatial joins with the California county file. We also converted the `ocean_proximity` variable into numeric from categorial. So, in total, we have these variables: longitude, latitude, housingMedianAge, totalRooms, totalBedrooms, population, households, medianIncome, medianHouseValue, oceanProximity, where all of the variables are numeric except for the medianHouseValue.

Our Shiny app is made using a fluid page, which comprises three main tabs: the Description tab, Exploratory Data Analysis, and the Predictive Modeling tab. The Description tab provides an overview of the dataset as well as how to use the app. The Exploratory Data Analysis tab allows users to choose and display either a correlation matrix or cluster analysis. For the correlation matrix, we can see the correlation between all these variables. For the cluster analysis, we make the clusters for the counties based on the mean values of multiple variables that we have, and median house values. It calculates the mean values for each variable for each county, performs K-means clustering to assign each county to one of three clusters, and then visualizes the results with a scatter plot. So, the cluster plot shows the chosen variable on the x-axis and the mean median house value on the y-axis, with points colored by their cluster. The Predictive Modeling tab is the main tab of our app as it is the one that does the machine learning analysis. Moreover, most of the things are interactive with different reactive buttons that allow users to explore wide features of the data. Here, we can select out of two models, a Decision Tree or Variable Importance.

Variable importance is significant as it is useful for feature engineering, to find the relative influence of each variable that we have in our housing data. The app reactively renders images or plots based on user inputs for each relevant county in California. We can also see the Variable importance for all the counties in California. In predictive modeling, we can choose a model type, using `checkboxGroupInput` for Decision Trees and `selectInput` for Important Variables. On the process of making the Random forest Decision Tree, we performed machine learning analysis, by data splitting, recipe creation for the preprocessing, model specification and tuning using cross-validation, and visualization using “rpart.plot”. Moreover, we trimmed the random forest tree to make it more clear and understandable.

We wanted the random forest model to be hidden while the Variable importance is running and vice versa. So, to achieve this, we used Shiny conditional panels in order to show or hide outputs based on user selections. We had to import shinyjs for this feature in the app. Moreover, to make it look good and align it to the right with a decorative border, we wrote a small piece of custom CSS code. Moreover, to make the descriptive text align properly on the page, we used HTML codes. In this way, for our project, we integrated data preprocessing, EDA, and predictive modeling. Combination of all these into an interactive website, provides valuable insights into factors affecting housing prices in California, allowing users to explore different analytical perspectives and predictive models.