Task-by-Task Guide

If you'd like a little more support while completing this project, explore this step-by-step resource to get additional hints and resources to help you along each task of this project.

Task 1 - Import data and data preprocessing

For this project, we will be working with stroke prediction data. The data contains 5110 observations with 12 attributes. You may spend some time reading a well-detailed description of the data here.

In this first task, you will load data, perform data cleaning, data transformation, feature engineering, and missing data imputation. The whole idea for this task is to prepare the data set ready for building prediction models in the next task.

Note: You will spend most of the time working on this task. Therefore, please spend some quality time doing this task well because the quality of your data determines the quality of your model and predictions.

I have provided general hints on how to complete this task.

Load data and install packages

Hint

As you may know that R lives in a world of packages. Although I have installed some packages for you, you may need to install some more packages because this project is open-ended. After installing packages, you need to import the packages using the **library()** function.

You can load the **healthcare-dataset-stroke-data.csv** using the **read.csv() or read_csv()** functions.

Here are some useful resources to help you with this section of the task:
R Packages: A Beginner's Tutorial
Importing data into R tutorial

Describe and explore the data

Ideally, in this section, you want to describe and explore the data using functions and visualizations to uncover insights. Once you have your data, it's a good idea to get acquainted with it. You should show some summary statistics and visually examine your data. Don't forget to write out some insights that you have gained along with your analysis.

Hint

You may decide to use base R functions for data exploration, such as **summary(),** or advanced functions, such as the **skim()** function from the skimr package. You can create data visualizations to explore the data using base R functions or the ggplot2 package.

Also, in this step, it will be useful to clean or preprocess the data, including recoding variables, changing N/A or Unknown to missing values, and converting variables to correct data types (such as numeric, factor, etc.). Some functions in the tidyverse family of packages, including the dplyr and tidyr functions such as **mutate(), mutate_at(),** will be useful.

In this section, you might ask yourself questions such as, "What variables are correlated?" "Which variable provides valuable information for model building" before analyzing data and creating visuals to showcase your findings.

Here are some useful resources to help you with this section of the task:
Four R packages for Automated Exploratory Data Analysis you might have missed
Exploratory Data Analysis
Data visualization with ggplot2

You can read my articles on tidy data here:

- A Gentle Introduction to Tidy Data in R
- Tidy Data Case Studies Using R

Feature engineering and missing values imputation

After you have explored the data set, you may see the need to create new variables, this is referred to as *feature engineering*. Also, you will need to impute missing values in variables.

Hint

Creating new features requires sufficient evidence that comes usually from your data exploration. There are different techniques to impute missing values, such as multiple imputations. The tidyr and mice packages provide very interesting ways to deal with missing values.

Here are some useful resources to help you with this section of the task:
Feature Engineering for Machine Learning in R
Read my article on Handling Missing Values in R using Tidyr
Getting Started with Multiple Imputation in R
Multiple imputation with the mice package
Tutorial on 5 Powerful R Packages used for imputing missing values

When you are satisfied that your data is tidy and ready for model building, please proceed to the next task.

Task 2 - Build prediction models

Once you have your tidy data, your task is to build the stroke prediction model using different classification models.

Hint
There are different classification algorithms you can try, including logistic regression, naive Bayes, K-nearest neighbors, (linear and/or kernel) support vector machine, decision tree, random forest, and XGBoost. In this task, you are allowed to use any approach since this is open-ended, However, there are some steps you should consider including

- dealing with class imbalance
- standardize or normalize numeric variables
- one-hot encode categorical variables
- splitting the data into train and test sets
- training the data on the train sets using cross-validation
- tuning hyperparameters
- going through that iterative process until you are satisfied with your model.

Note: As I said, you can use any approach you desire; there are no right or wrong answers. However, I have found using tidymodels useful in tying up many of these steps.

Here are some useful resources to help you with this task:
A Gentle Introduction to tidymodels
Tidymodels: tidy machine learning in R
Your First Machine Learning Project in R Step-By-Step
A complete guide to fit Machine Learning models in R

Task 3 - Evaluate and select prediction models

After fitting the model using the best hyperparameters on the train set and validating on the cross-validation set, the next step is to evaluate and select prediction models based on evaluation metrics, including accuracy, sensitivity, recall, F1 score, and AUC-ROC curve. Again, there are no right or wrong answers since the decision to select a model is based on your analysis.

Hint

A few common questions to ask are:

- Which model has the highest score across evaluation metrics?
- What is important in this case study? Do you want a model that is highly sensitive or highly specific for stroke prediction?
- Are there evaluation metrics such as an F-score or balanced accuracy that are better for this case study?
- How spread out are the confidence intervals around each evaluation metric?

Here are some useful resources to help you with this task:

Tidymodels: tidy machine learning in R
Metrics to evaluate classification models
Compare Models And Select The Best Using The Caret R Package
Your First Machine Learning Project in R Step-By-Step
A complete guide to fit Machine Learning models in R

Task 4 - Deploy the prediction model

Usually, people build models and stop there. The value in building a model is to deploy the model for use. In this context, you will deploy the best prediction model for stroke prediction in a clinical setting. Once you have decided on the best model, you can continue to deploy the model.

Hint

There are several ways to deploy your model in R, either via an API using Vetiver or via a simple web page using R Shiny. The choice of which to use depends on you.

Note: Using Vetiver is relatively new compared with R Shiny. Both have their pros and cons. For example, you can deploy your pinned `vetiver_model()` via a Plumber API, which can be hosted in a variety of ways. R Shiny only provides an option to deploy your model as a web page.

Here are some useful resources to help you with this task:

Read about Vetiver on Posit website
Read these three articles on create a deployable vetiver model, publish and version your model, and deploy your model as a REST API
Mastering Shiny: Your first Shiny app
R Shiny for Data Science Tutorial – Build Interactive Data-Driven Web Apps

Task 5 - Findings and Conclusions

Finally, we can wrap up the project. You can write a conclusion about your process and any key findings.

Hint

The main components that you will want to include:

- What did you learn throughout the process?
- Are the results what you expected?
- What are the key findings and takeaways?