Task

Capstone Project – Machine Learning

Visit our website

Introduction

Congratulations on making it this far! This capstone is a milestone in your learning. You will consolidate the knowledge that you have gained and apply it to solve a problem in this project by doing the various preceding tasks, requiring you to analyse and visualise data and build ML models for prediction and classification tasks. Show your creativity as you tackle the set of specified instructions, but the rest is up to you! It's worth spending time and effort to make this a project that you can be proud of and add to your developer portfolio.

DEVELOPER PORTFOLIO

Developers who have the edge are those who find ways to apply their newfound skills from the get-go. As you may know, a **developer portfolio** (a collection of online creations that you have made) allows you to demonstrate your skills rather than just telling people about them. It's a way of bringing your CV to life and introducing yourself to the world. As you learn more skills and put these into practice, each project that you complete will become more efficient and eye-catching.

These capstone projects give you the means to create projects for your very own developer portfolio, allowing you to walk away from this course not only with a certificate but, more importantly, with a head start on your tech career!

THE TASK AT HAND

This task involves an in-depth exploration of the Telco Customer Churn dataset. It encompasses data analysis, preprocessing, and advanced scaling techniques. Through the application of machine learning models such as logistic regression and random forests, you'll predict customer churn, enhance your data science competencies, and foster proficiency in making data-informed decisions.

Instructions

A key focus of this project will be ensuring that your code is correct, well-formatted, readable, and that it adheres to the **PEP 8 style guide**. In this regard, **make sure that you do the following** (and double-check before submitting your work to avoid losing marks unnecessarily!):

1. Identify and remove all syntax, runtime, and logical errors from your code.

2. Make sure that your code is readable. To ensure this, add comments to your code, use descriptive variable names, and make good use of whitespace and indentation.

3. Make sure that your code is modular. Create functions to perform specific units of work.
4. How you choose to write code to create the solution to the specified problem is up to you, but make sure that you write your code as efficiently as possible.
5. Make sure that all outputs that your program provides to a user are easy to read and understand. Label all data that you output.

Take note!

In this task, it's vital to emphasise the role of feature analysis and selection. Not all available features may be relevant for predicting churn effectively. Therefore, your objective extends beyond merely building predictive models; it involves understanding which features have a significant impact on customer churn. Remember, you don't have to use all the features that are available in your models. Thoughtful selection of the most relevant features can lead to more accurate predictions. Moreover, while some features may not directly contribute to predicting customer churn, they can offer valuable insights into customer behaviour. Therefore, consider including these features for exploratory purposes, even if they won't be used in the predictive models. As you proceed through the data inspection, preprocessing, and visualisation phases, keep these guidelines in mind. They will help you make informed decisions about feature selection, enhancing the overall quality of your analysis and predictions.

Practical task

In this task, you will go through the essential stages of data analysis and machine learning. You will work with real-world data from the Telco industry with the goal to effectively predict customer churn. Customer churn occurs when customers decide to end their subscriptions or services with the telecom company. The goal is to create logistic regression and random forest machine learning models that can forecast customer churn. By analysing historical customer data, these models will help us determine the likelihood of a customer either leaving the service (churning) or staying with it. This predictive capability is crucial for the telecom company to implement effective retention strategies.

Follow each step in the phases below:

1. Create a Jupyter notebook named **telco_customer_churn.ipynb**.

2. Begin by importing the necessary libraries for data analysis and visualisation.

3. Load the **Telco Customer Churn** dataset provided in your folder.

Data inspection phase

4. Display the first ten rows of the dataset.

5. Output the names of the dataset's columns.

6. Display the data types of each column.

Data preprocessing phase

7. Convert the “TotalCharges” column to the numeric data type.

8. Remove rows with missing values from the dataset.

9. Remove the “customerID” column from the dataset.

10. Convert the “Churn” column to binary numeric values: replace “Yes” with 1 and “No” with 0.

11. Convert all categorical variables in the dataset into dummy variables. Store the result in a new DataFrame called **telecom_cust_dummies**.

Please note: While converting categorical variables into dummy variables, it's vital to be aware of the **dummy variable trap**. Converting all categorical features can lead to multicollinearity issues (when predictor variables in a model are highly correlated), causing inefficiencies and inaccuracies in your model's predictions. It's important to understand that you may not need to convert every categorical variable into dummies; sometimes, omitting one category or using techniques like one-hot encoding can help prevent similar situations. Be aware that thoughtful feature engineering is a critical aspect of building effective machine learning models.

Data visualisation phase

12. Create a correlation plot to visualise the correlation between features and the target variable “Churn”.
13. Create a histogram of the “tenure” column.
14. Create a scatter plot of “MonthlyCharges” vs. “TotalCharges”.
15. Create a box plot to compare “tenure” for churned and non-churned customers.

Preparations for ML training

16. Scale all the variables to a range of 0 to 1 using min-max scaling.
17. Split the dataset into training and test sets (X_train; X_test; y_train; y_test) with a test size of 25%.

Logistic regression model

18. Import and train a logistic regression model on the training data.
19. Make predictions on the test data.
20. Calculate and print the accuracy score of the logistic regression model.

Random forest model

21. Create a random forest classifier and tune the following hyperparameters:
    a. The number of decision trees in the forest should be 2000.

b. Enable out-of-bag (OOB) error estimation. The OOB estimation metric could be activated by enabling the `oob_score` parameter during model creation. OOB estimation offers a valuable estimate of model accuracy without requiring a separate validation set.

c. The maximum number of features considered for splitting should be set to “sqrt”. The `max_features` parameter controls this and influences model diversity. Setting it to “sqrt” (the default for classification tasks) considers a subset of features for splitting, reducing overfitting while maintaining model power. Experimenting with different values such as “log2”, an integer, or a float can help find the right balance between overfitting and model complexity.

d. The maximum number of leaf nodes in each tree should be 50. The parameter is regulated by `max_leaf_nodes`, and constraints tree complexity. Smaller values foster simpler trees, potentially reducing overfitting, while larger values capture finer details but may risk overfitting.

e. Activate bootstrapping.

22. Fit the model.

23. Make predictions on the new test data.

24. Calculate and print the accuracy score of the random forest classifier.

25. Calculate and print the **OOB error** estimation. Explain how well the model generalises to unseen data. You can determine this estimation by subtracting the OOB score of your model, `(MODEL_NAME.oob_score_)`, from 1.

Enhanced model evaluation

26. Implement code to calculate the confusion matrix for both the logistic regression and random forest models.

27. Compute the precision and recall scores for each model.

28. Discuss what the confusion matrix reveals about the model's performance. Analyse the precision and recall scores to understand the trade-off between correctly and incorrectly identifying churned customers.

29. Compare the logistic regression and random forest models based on these additional metrics. Which model is more suitable for this specific task, and why?

Be sure to place files for submission inside your **task folder** and click “**Request review**” on your dashboard.

Share your thoughts

Please take some time to complete this short feedback **form** to help us ensure we provide you with the best possible learning experience.