Assignment

Q1. You are working on a machine learning project where you have a dataset containing numerical and
categorical features. You have identified that some of the features are highly correlated and there are
missing values in some of the columns. You want to build a pipeline that automates the feature
engineering process and handles the missing valuesD
Design a pipeline that includes the following steps"
Use an automated feature selection method to identify the important features in the datasetC
Create a numerical pipeline that includes the following steps"
Impute the missing values in the numerical columns using the mean of the column valuesC
Scale the numerical columns using standardisationC
Create a categorical pipeline that includes the following steps"
Impute the missing values in the categorical columns using the most frequent value of the columnC
One-hot encode the categorical columnsC
Combine the numerical and categorical pipelines using a ColumnTransformerC
Use a Random Forest Classifier to build the final modelC
Evaluate the accuracy of the model on the test datasetD
Note! Your solution should include code snippets for each step of the pipeline, and a brief explanation of
each step. You should also provide an interpretation of the results and suggest possible improvements for
the pipelineD

Ans: elow is a code template that outlines each step of the pipeline using Python and scikit-learn. Please note that you may need to adapt the column names, target variable, and other details based on your specific dataset.

python

Copy code

```
from                 import
from                  import
from                import
from                         import
from                  import
from                       import
from                           import
from                 import
```

```
                                                                      42



'imputer'                          'mean'
'scaler'





'imputer'                          'most_frequent'
'encoder'                                'ignore'






'num'                        'numerical_column1'  'numerical_column2'
'cat'                          'categorical_column1'  'categorical_column2'






'feature_selection'
'preprocessing'
'classifier'                                       42




       'target_column'      1

  'target_column'
         0.2
            42
```

```
print "Accuracy:"
```

Interpretation of the Results:

- The pipeline incorporates automated feature selection, numerical feature imputation and scaling, categorical feature imputation and one-hot encoding, and combines them using a `ColumnTransformer`.
- The final model is a Random Forest Classifier.
- Accuracy is used as the evaluation metric.

Possible Improvements:

- Fine-tuning hyperparameters of the Random Forest model for better performance.
- Experimenting with different feature selection methods.
- Considering more advanced imputation techniques or exploring other handling strategies for missing values.
- Exploring different models or ensemble methods for potential improvements.

This is a basic template, and you might need to tailor it according to the characteristics of your specific dataset and the goals of your machine learning project.

Q2. Build a pipeline that includes a random forest classifier and a logistic regress#on classifier, and then

use a voting classifier to combine their predictions. Train the pipeline on the iris dataset and evaluate its

accuracy.

Ans:Below is a code template that demonstrates how to build a pipeline with a Random Forest Classifier and a Logistic Regression Classifier, and then combine their predictions using a Voting Classifier. The dataset used in this example is assumed to be an iris dataset.

python

Copy code

```
from                  import
from                  import
from                      import
from                         import
from                 import
from                  import
from                import
from                       import
from                 import
```

`0.2`

`42`

```
                                                        42

                                                   42







    'imputer'                          'mean'

    'scaler'










                'rf'                    'lr'

          'hard'
```

```
'num'                           0  1  2  3
```

```
'preprocessor'

'voting_classifier'
```

```
print "Accuracy:"
```

In this example:

- The pipeline includes a preprocessing step for numerical features, which involves imputation and scaling.
- The individual classifiers are a Random Forest Classifier (`rf_classifier`) and a Logistic Regression Classifier (`lr_classifier`).
- The Voting Classifier (`voting_classifier`) combines the predictions of the individual classifiers using hard voting.

You can adjust the pipeline components based on your specific dataset and requirements. Also, you can experiment with different combinations of classifiers and voting strategies for the Voting Classifier.