regression_model

August 7, 2024

0.1 Data processing

0.1.1 There are three techniques to solve the missing values' problem in order to find out the most accurate features, and they are:

0.1.2 * Dropping

0.1.3 * Numerical imputation

0.1.4 * Categorical imputation

[1]:
```python
import pandas as pd
import numpy as np
from scipy import stats
```

[2]:
```python
df = pd.read_csv('/home/student/220962344_ml_lab/Week3/diabetes_csv.csv')
df

## To check for missing values
# print(df.isnull())
df['Has_Missing'] = df.isnull().any(axis=1)

if df['Has_Missing'].any() == True:
    print(yes)
# This code takes the mean of the columns and retains the cols with more than␣
↪60% non-missing value
# Use axis = 1 in mean for rows
threshold = 60
df = df.loc[df.isnull().mean(axis=1) < threshold] # rows
df = df[df.columns[df.isnull().mean() < threshold]] # cols
df

## We use imputation to prevent low training size
# Replace the missing data with a relevant value
df = df.fillna(0)
df = df.fillna(df.median())

## Outlier Identification
# * Z-Score and IQR are good for identifying outliers in univariate data.
# * Box Plots and Scatter Plots offer a visual approach to detecting outliers.
```

```
# * Machine Learning Methods: Isolation Forest and Local Outlier Factor are␣
↪useful for more complex datasets.
df['Z-Score_glucose'] = stats.zscore(df['Glucose'])
df_filtered = df[abs(df['Z-Score']) <= 3]
print(df_filtered.head())

Q1 = df['Glucose'].quantile(0.25)
Q3 = df['Glucose'].quantile(0.75)
IQR = Q3 - Q1

df_filtered = df[(df['Glucose'] >= Q1) & (df['Glucose'] <= Q3)]


## Overfitting
# We can use binning to remove rows ith less importance
df['bin'] = pd.cut(df['Glucose'], bins=[100,250,400,500], labels=["Lowest",␣
↪"Mid", "High"])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File /usr/lib/python3/dist-packages/pandas/core/indexes/base.py:3791, in Index.
↪get_loc(self, key)
   3790 try:
-> 3791     return self._engine.get_loc(casted_key)
   3792 except KeyError as err:

File /usr/lib/python3/dist-packages/pandas/_libs/index.pyx:152, in pandas._libs
↪index.IndexEngine.get_loc()

File /usr/lib/python3/dist-packages/pandas/_libs/index.pyx:181, in pandas._libs
↪index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:7080, in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:7088, in pandas._libs.hashtable.
↪PyObjectHashTable.get_item()

KeyError: 'Z-Score'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Cell In[2], line 27
     22 ## Outlier Identification
     23 # * Z-Score and IQR are good for identifying outliers in univariate data.
```

```
     24 # * Box Plots and Scatter Plots offer a visual approach to detecting␣
  ↪outliers.
     25 # * Machine Learning Methods: Isolation Forest and Local Outlier Factor␣
  ↪are useful for more complex datasets.
     26 df['Z-Score_glucose'] = stats.zscore(df['Glucose'])
---> 27 df_filtered = df[abs(df['Z-Score']) <= 3]
     28 print(df_filtered.head())
     30 Q1 = df['Glucose'].quantile(0.25)

File /usr/lib/python3/dist-packages/pandas/core/frame.py:3893, in DataFrame.
  ↪__getitem__(self, key)
   3891 if self.columns.nlevels > 1:
   3892     return self._getitem_multilevel(key)
-> 3893 indexer = self.columns.get_loc(key)
   3894 if is_integer(indexer):
   3895     indexer = [indexer]

File /usr/lib/python3/dist-packages/pandas/core/indexes/base.py:3798, in Index.
  ↪get_loc(self, key)
   3793     if isinstance(casted_key, slice) or (
   3794         isinstance(casted_key, abc.Iterable)
   3795         and any(isinstance(x, slice) for x in casted_key)
   3796     ):
   3797         raise InvalidIndexError(key)
-> 3798     raise KeyError(key) from err
   3799 except TypeError:
   3800     # If we have a listlike key, _check_indexing_error will raise
   3801     #  InvalidIndexError. Otherwise we fall through and re-raise
   3802     #  the TypeError.
   3803     self._check_indexing_error(key)

KeyError: 'Z-Score'
```

[ ]:
```python
## We do encodeing for non numberci data
from sklearn.preprocessing import LabelEncoder
# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Fit and transform the data
df['Priority_Encoded'] = label_encoder.fit_transform(df['__name__'])
```

0.2 Lab Questions

[3]:
```python
'''
Consider the hepatitis/ pima-indians-diabetes csv file, perform the following␣
 ↪date pre-processing.
```

```python
1. Load data in Pandas.
2. Drop columns that aren't useful.
3. Drop rows with missing values.
4. Create dummy variables.
5. Take care of missing data.
6. Convert the data frame to NumPy.
7. Divide the data set into training data and test data.
'''

import pandas as pd
import numpy as np

df = pd.read_csv('hepatitis_csv.csv')

# the class is the target variable, live or die
df.head()

df_cleaned = df.dropna()
df_cleaned.head()

df_dummies = pd.get_dummies(df_cleaned, columns=['sex', 'class'])
df_dummies.head()

numpy = df_dummies.to_numpy()
print(numpy)

# shuffle
df_dummies = df_dummies.sample(frac=1)

# Define the split ratio
train_ratio = 0.8
train_size = int(df_dummies.shape[0] * train_ratio)

# Split into training and testing sets
train_df = df_dummies[:train_size]
test_df = df_dummies[train_size:]

print("Train")
print(train_df)

print("Test:")
print(test_df)
```

```
[[34 True False … False False True]
 [39 False True … False False True]
 [32 True True … False False True]
 …
 [31 False False … False False True]
```

```
 [53 False False … True False True]
 [43 True False … False True False]]
Train
     age steroid  antivirals fatigue malaise anorexia liver_big liver_firm  \
96    30   False       False    True    True    False      True       True
23    42    True       False   False   False    False      True      False
125   34    True       False    True    True     True     False       True
25    27   False       False    True    True    False      True      False
109   33   False       False    True    True    False      True      False
..   …      …          …      …     …     …      …       …
133   72    True        True    True   False    False      True       True
60    37    True       False   False   False    False      True      False
54    30    True       False    True   False    False      True      False
77    34   False        True   False   False    False      True       True
135   25    True       False    True   False    False     False       True

    spleen_palpable spiders  … bilirubin alk_phosphate   sgot  albumin  \
96            False    True  …       0.8         147.0  128.0      3.9
23            False   False  …       0.9          60.0   63.0      4.7
125           False    True  …       0.7          70.0   24.0      4.1
25            False   False  …       0.8          95.0   46.0      3.8
109           False   False  …       0.7          63.0   80.0      3.0
..              …      …  …       …            …    …      …
133           False   False  …       1.0         115.0   52.0      3.4
60            False   False  …       0.7          26.0   58.0      4.5
54            False   False  …       0.7          50.0   78.0      4.2
77            False   False  …       0.6          30.0   24.0      4.0
135            True    True  …       1.3         181.0  181.0      4.5

     protime  histology  sex_female  sex_male  class_die  class_live
96     100.0       True        True     False      False        True
23      47.0      False        True     False      False        True
125    100.0       True       False      True      False        True
25     100.0      False        True     False      False        True
109     31.0       True        True     False       True       False
..       …         …          …        …         …          …
133     50.0       True        True     False      False        True
60     100.0      False        True     False      False        True
54      74.0      False        True     False      False        True
77      76.0      False       False      True      False        True
135     57.0       True        True     False      False        True

[64 rows x 22 columns]
Test:
     age steroid  antivirals fatigue malaise anorexia liver_big liver_firm  \
89    38   False       False    True    True     True     False       True
98    47    True       False   False   False    False      True      False
138   47    True       False    True    True    False      True       True
```

```
19    38   False        True   False   False    False     False       True
75    32   False        True    True    True    False      True      False
38    42   False       False   False   False    False      True      False
53    40    True        True    True   False    False      True       True
94    59   False       False    True    True    False      True       True
39    65    True       False    True    True    False      True       True
15    38   False       False    True    True     True      True      False
85    28   False       False    True    True     True      True       True
5     34    True       False   False   False    False      True      False
124   50    True       False   False   False    False      True       True
33    26   False       False   False   False    False      True       True
43    56   False       False    True   False    False      True      False
17    40   False       False    True   False    False      True       True

    spleen_palpable spiders  … bilirubin alk_phosphate   sgot  albumin  \
89            False   False  …       0.6          76.0   18.0      4.4
98            False    True  …       2.0          84.0   23.0      4.2
138           False    True  …       1.0         166.0   30.0      2.6
19            False   False  …       0.7          70.0   28.0      4.2
75            False   False  …       1.0          55.0   45.0      4.1
38            False   False  …       1.0          85.0   14.0      4.0
53             True   False  …       1.2          85.0   31.0      4.0
94             True    True  …       1.5         107.0  157.0      3.6
39             True    True  …       0.3         180.0   53.0      2.9
15            False   False  …       2.0          72.0   89.0      2.9
85            False   False  …       1.6          44.0  123.0      4.0
5             False   False  …       0.9          95.0   28.0      4.0
124            True    True  …       1.0          85.0   75.0      4.0
33            False   False  …       0.5         135.0   29.0      3.8
43            False   False  …       0.7          71.0   18.0      4.4
17            False   False  …       0.6          62.0  166.0      4.0

     protime  histology  sex_female  sex_male  class_die  class_live
89      84.0       True        True     False      False        True
98      66.0       True        True     False       True       False
138     31.0       True        True     False       True       False
19      62.0      False        True     False      False        True
75      56.0      False        True     False      False        True
38     100.0      False        True     False      False        True
53     100.0      False        True     False      False        True
94      38.0       True        True     False       True       False
39      74.0       True        True     False      False        True
15      46.0      False        True     False      False        True
85      46.0      False        True     False      False        True
5       75.0      False        True     False      False        True
124     72.0       True        True     False      False        True
33      60.0      False       False      True      False        True
43     100.0      False        True     False      False        True
```

```
17      63.0       False         True      False       False         True

[16 rows x 22 columns]
```

[4]:
```python
'''
2. a. Construct a CSV file with the following attributes:
Study time in hours of ML lab course (x)
Score out of 10 (y)
The dataset should contain 10 rows.
b. Create a regression model and display the following:
Coefficients: B0 (intercept) and B1 (slope)
RMSE (Root Mean Square Error)
Predicted responses
c. Create a scatter plot of the data points in red color and plot the graph of␣
 ↪x vs. predicted y in blue color.
d. Implement the model using two methods:
Pedhazur formula (intuitive)
Calculus method (partial derivatives, refer to class notes)
e. Compare the coefficients obtained using both methods and compare them with␣
 ↪the analytical solution.
f. Test your model to predict the score obtained when the study time of a␣
 ↪student is 10 hours.
Note: Do not use scikit-learn.
'''

import numpy as np
import pandas as pd

study_time = np.random.randint(1,6,100)

score = np.random.randint(0,11,100)

dic = {
    "study_time" : study_time,
    "Score" : score
}
df = pd.DataFrame(dic)
df.head()
```

[4]:

	study_time	Score
0	3	7
1	4	3
2	2	0
3	4	1
4	1	5

[6]:

```python
import numpy as np

class LinearRegression:
    def __init__(self):
        self.weight = None
        self.bias = None
        self.epochs = []
        self.loss = []
    def fit(self, X, y, epochs=1000, alpha=0.01):
        '''
        Train the linear regression model using gradient descent.

        Parameters:
        X (numpy array): Input feature of shape (num_samples,)
        y (numpy array): Target values of shape (num_samples,)
        epochs (int): Number of iterations for gradient descent
        alpha (float): Learning rate
        '''
        num_samples = len(X)
        # Initialize weight and bias
        self.weight = 0.0
        self.bias = 0.0

        # Gradient descent
        for epoch in range(epochs):
            # Compute predictions
            y_pred = self.predict(X)
            # Compute gradients
            weight_gradient = -2 * np.sum((y - y_pred) * X) / num_samples
            bias_gradient = -2 * np.sum(y - y_pred) / num_samples
            # Update parameters
            self.weight -= alpha * weight_gradient
            self.bias -= alpha * bias_gradient

            # Optionally print loss for every 100 epochs
            if epoch % 100 == 0:
                loss = np.mean((y_pred - y) ** 2)
                print(f"Epoch {epoch}, Loss: {loss}")
                self.epochs.append(epoch)
                self.loss.append(loss)


    def predict(self, X):
        '''
        Predict the target values for the input feature.

        Parameters:
```

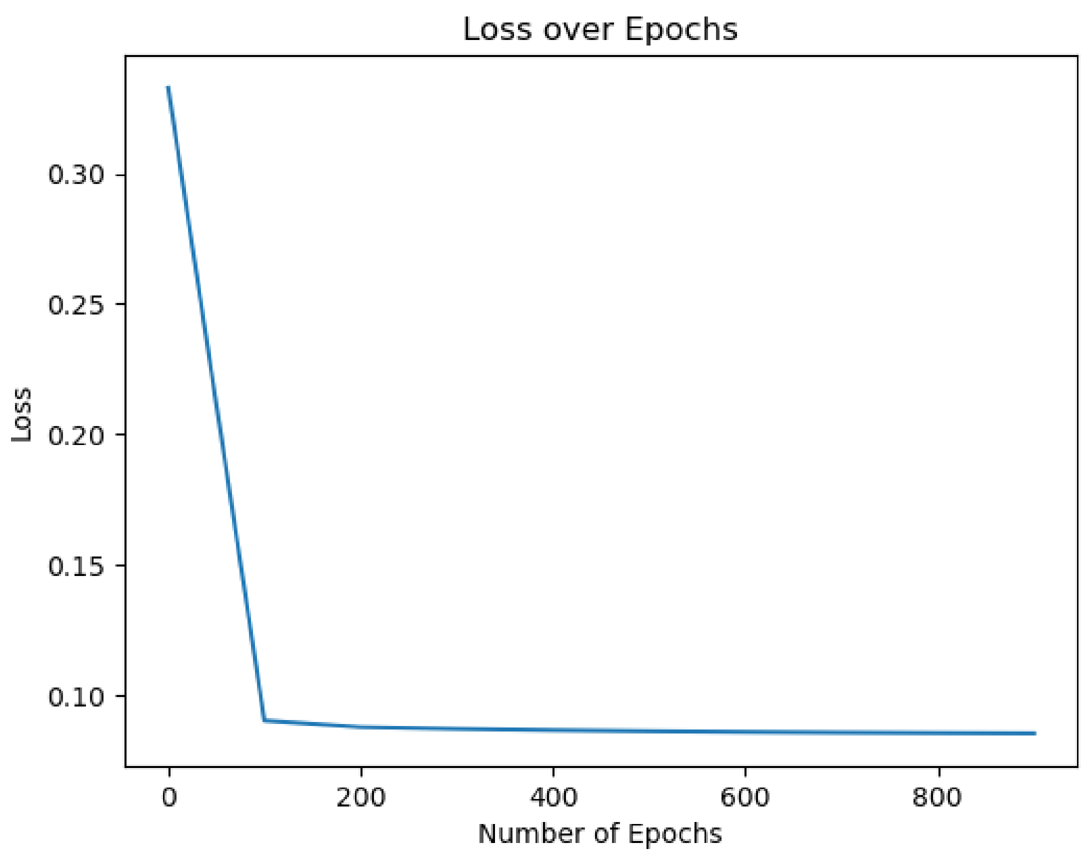
```
        X (numpy array): Input feature of shape (num_samples,)

        Returns:
        numpy array: Predicted target values
        '''
        if self.weight is None or self.bias is None:
            raise ValueError("Model has not been trained yet.")
        return self.weight * X + self.bias

    def mean_squared_error(self, y_true, y_pred):
        '''
        Calculate the mean squared error between true and predicted values.

        Parameters:
        y_true (numpy array): True target values
        y_pred (numpy array): Predicted target values

        Returns:
        float: Mean squared error
        '''
        return np.mean((y_true - y_pred) ** 2)
```

[7]:
```
import numpy as np

# Generate synthetic data
np.random.seed(42)
X = np.random.rand(100)  # 100 samples of input feature
true_weight = 3.5
true_bias = -2.0
y = np.random.rand(100)

# Create and train the model
model = LinearRegression()
model.fit(X, y, epochs=1000, alpha=0.01)

# Predict using the trained model
y_pred = model.predict(X)

# Calculate and print the mean squared error
mse = model.mean_squared_error(y, y_pred)
print(f"Mean Squared Error: {mse}")

# Print learned parameters
print(f"Learned weight: {model.weight}")
print(f"Learned bias: {model.bias}")
```

```
Epoch 0, Loss: 0.33289148886695125
Epoch 100, Loss: 0.09022597115000433
```

```
Epoch 200, Loss: 0.08771998112325295
Epoch 300, Loss: 0.08703019843125777
Epoch 400, Loss: 0.08651855289171552
Epoch 500, Loss: 0.08613318908361896
Epoch 600, Loss: 0.08584289931278948
Epoch 700, Loss: 0.08562422737112567
Epoch 800, Loss: 0.08545950431374577
Epoch 900, Loss: 0.08533542033009127
Mean Squared Error: 0.0852419492930398
Learned weight: 0.023128772113414613
Learned bias: 0.4849278695821864
```

[8]:
```
import matplotlib.pyplot as plt  # Correct import statement

# Assuming `model` has `epochs` and `loss` attributes
plt.plot(model.epochs, model.loss)
plt.xlabel('Number of Epochs')
plt.ylabel('Loss')
plt.title('Loss over Epochs')
plt.show()
```



```
[ ]:
```

```
[ ]:
```