Certificate Course in Machine Learning using Python [6 Weeks]

Dashboard › My courses › Certificate Course in Machine Learning using Python [6 Weeks] › Day 27 › Feature Importance & Correlation Matrix

Feature Importance & Correlation Matrix

Feature Importance

Feature importance refers to a class of techniques for assigning scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction.

Feature importance scores can be calculated for problems that involve predicting a numerical value, called regression, and those problems that involve predicting a class label, called classification.

The scores are useful and can be used in a range of situations in a predictive modeling problem, such as:

- Better understanding the data.
- Better understanding a model.
- Reducing the number of input features.

Advantages:

Feature importance scores can provide insight into the dataset. The relative scores can highlight which features may be most relevant to the target, and the converse, which features are the least relevant.

Feature importance scores can provide insight into the model. Most importance scores are calculated by a predictive model that has been fit on the dataset. Inspecting the importance score provides insight into that specific model and which features are the most important and least important to the model when making a prediction.

Feature importance can be used to improve a predictive model. This can be achieved by using the importance scores to select those features to delete (lowest scores) or those features to keep (highest scores).

The correlation matrix

The **correlation matrix** can be used to estimate the linear historical relationship between the returns of multiple assets. You can use the built-in .corr() method on a pandas DataFrame to easily calculate the correlation matrix.

Example: Correlation matrix for admission prediction problem

```
import pandas as pd

df = pd.read_csv("Admission_Predict.csv",sep = ",")

#Printing correlation matrix

df.corr()
```

Displaying correlation matrix as headmap using seaborn

```
import seaborn as sns

plt.figure(figsize=(10, 10))

sns.heatmap(df.corr(), annot=True, linewidths=0.05, fmt= '.2f',cmap="magma")

plt.show()
```

Coefficients as Feature Importance

Linear machine learning algorithms fit a model where the prediction is the weighted sum of the input values. Such as Linear Regression and Logistic Regression.

All of these algorithms find a set of coefficients to use in the weighted sum in order to make a prediction. These coefficients can be used directly as a crude type of feature importance score.

Linear Regression Feature Importance

We can fit a LinearRegression model on the regression dataset and retrieve the *coeff_* property that contains the coefficients found for each input variable. These coefficients can provide the basis for a crude feature importance score. This assumes that the input variables have the same scale or have been scaled prior to fitting a model.

Example: Calculating model_coef_ for admission prediction problem

Calculating feature importance

```
#feature importance

importance = model.coef_

# summarize feature importance

for i,v in enumerate(importance):

    print('Feature:%s, Score: %.2f' % (i,v))
```

A bar chart is created for the feature importance scores..

```
# plot feature importance

pyplot.bar([x for x in range(len(importance))], importance)

pyplot.show()
```

Note: coef_ can also be applied to Logistic Regression model

Decision Tree Feature Importance

Decision tree algorithms like classification and regression trees (CART) offer importance scores based on the reduction in the criterion used to select split points, like Gini or entropy.

The model provides a *feature_importances_* property that can be accessed to retrieve the relative importance scores for each input feature.

Example: Please refer Loan Prediction Problem.

Printing the value of feature_importances for DecisionTreeClassifier algorithm.

```
importance = dtf.feature_importances_
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
```

Printing feature importance for decision tree classifier column wise

```
important_feature= pd.DataFrame({'Feature Value':  dtf.feature_importances_}, index=X.columns). sort_values (by='Feature Value', ascending=False)
important_feature
```

Note: We can obtain the feature_importances_ for Random Forest Regression and Classification, XGBoost Regression and Classification and KNeighborsClassifier algorithms.

Next

PREVIOUS ACTIVITY

◀ Clustering Problem

Jump to...

NEXT ACTIVITY

Course Feedback ▶

Stay in touch

Contact Us

http://nielit.gov.in/gorakhpur/

abhinav@nielit.gov.in or ajay.verma@nielit.gov.in