

Top 30 ML Python Program & Algorithm Questions

1. How do you implement Linear Regression in Python?

Using scikit-learn:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

2. What is the difference between NumPy and Pandas?

NumPy is used for numerical operations, while Pandas is used for data manipulation and analysis.

3. How do you handle missing values in a dataset using Python?

Using Pandas:

```
df.fillna(value)
df.dropna()
```

4. What is the difference between a list and a NumPy array?

A list is a collection of Python objects, while a NumPy array is optimized for numerical operations and provides better performance.

5. How do you implement Logistic Regression in Python?

Using scikit-learn:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
```

6. What is the role of the random_state parameter in scikit-learn?

It ensures reproducibility by controlling the random number generator.

7. How do you normalize data in Python?

Using MinMaxScaler:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
```

8. How do you split a dataset into training and testing sets in Python?

Using train_test_split:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

9. How do you evaluate a classification model in Python?

Using metrics like accuracy, precision, recall, and F1-score from sklearn.metrics.

10. How do you implement a Decision Tree classifier in Python?

Using scikit-learn:

```
from sklearn.tree import DecisionTreeClassifier  
model = DecisionTreeClassifier()  
model.fit(X_train, y_train)
```

11. How do you tune hyperparameters in ML models?

Using GridSearchCV or RandomizedSearchCV from scikit-learn.

12. What is the difference between Bagging and Boosting?

Bagging trains models in parallel, while Boosting trains models sequentially to correct errors.

13. How do you implement K-Means Clustering in Python?

Using scikit-learn:

```
from sklearn.cluster import KMeans  
model = KMeans(n_clusters=3)  
model.fit(X)
```

14. What is PCA, and how do you implement it in Python?

PCA reduces dimensions:

```
from sklearn.decomposition import PCA  
pca = PCA(n_components=2)  
X_pca = pca.fit_transform(X)
```

15. How do you implement a Neural Network in Python?

Using TensorFlow/Keras:

```
from tensorflow import keras  
model = keras.Sequential([...])  
model.compile(...)  
model.fit(X_train, y_train)
```

16. What is the difference between CNN and RNN?

CNN is used for images, while RNN is used for sequential data like text or time series.

17. How do you implement a Support Vector Machine (SVM) in Python?

Using scikit-learn:

```
from sklearn.svm import SVC  
model = SVC()
```

```
model.fit(X_train, y_train)
```

18. What is an Activation Function?

It introduces non-linearity in neural networks. Examples: ReLU, Sigmoid, Tanh.

19. How do you implement a Random Forest in Python?

Using scikit-learn:

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

20. How do you implement Gradient Boosting in Python?

Using XGBoost:

```
import xgboost as xgb
model = xgb.XGBClassifier()
model.fit(X_train, y_train)
```

21. How do you visualize data distributions in Python?

Using Seaborn:

```
import seaborn as sns
sns.histplot(df['column'])
```

22. What is the difference between L1 and L2 regularization?

L1 (Lasso) encourages sparsity, while L2 (Ridge) penalizes large weights.

23. How do you implement Time Series forecasting in Python?

Using ARIMA:

```
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(data, order=(p,d,q))
model.fit()
```

24. How do you handle imbalanced datasets?

Using techniques like SMOTE, class weighting, or oversampling/undersampling.

25. How do you save and load ML models in Python?

Using joblib:

```
import joblib
joblib.dump(model, 'model.pkl')
model = joblib.load('model.pkl')
```

26. What is Transfer Learning?

Using a pre-trained model like VGG16 or ResNet and fine-tuning it for a specific task.

27. How do you interpret a Confusion Matrix?

It shows TP, FP, TN, FN, helping evaluate classification model performance.

28. How do you implement an LSTM model in Python?

Using TensorFlow/Keras:

```
from keras.layers import LSTM
model.add(LSTM(50, return_sequences=True))
```

29. What is an Autoencoder?

A neural network used for unsupervised learning that compresses and reconstructs data.

30. How do you deploy a Machine Learning model?

Using Flask, FastAPI, or Streamlit to serve predictions via a web interface.