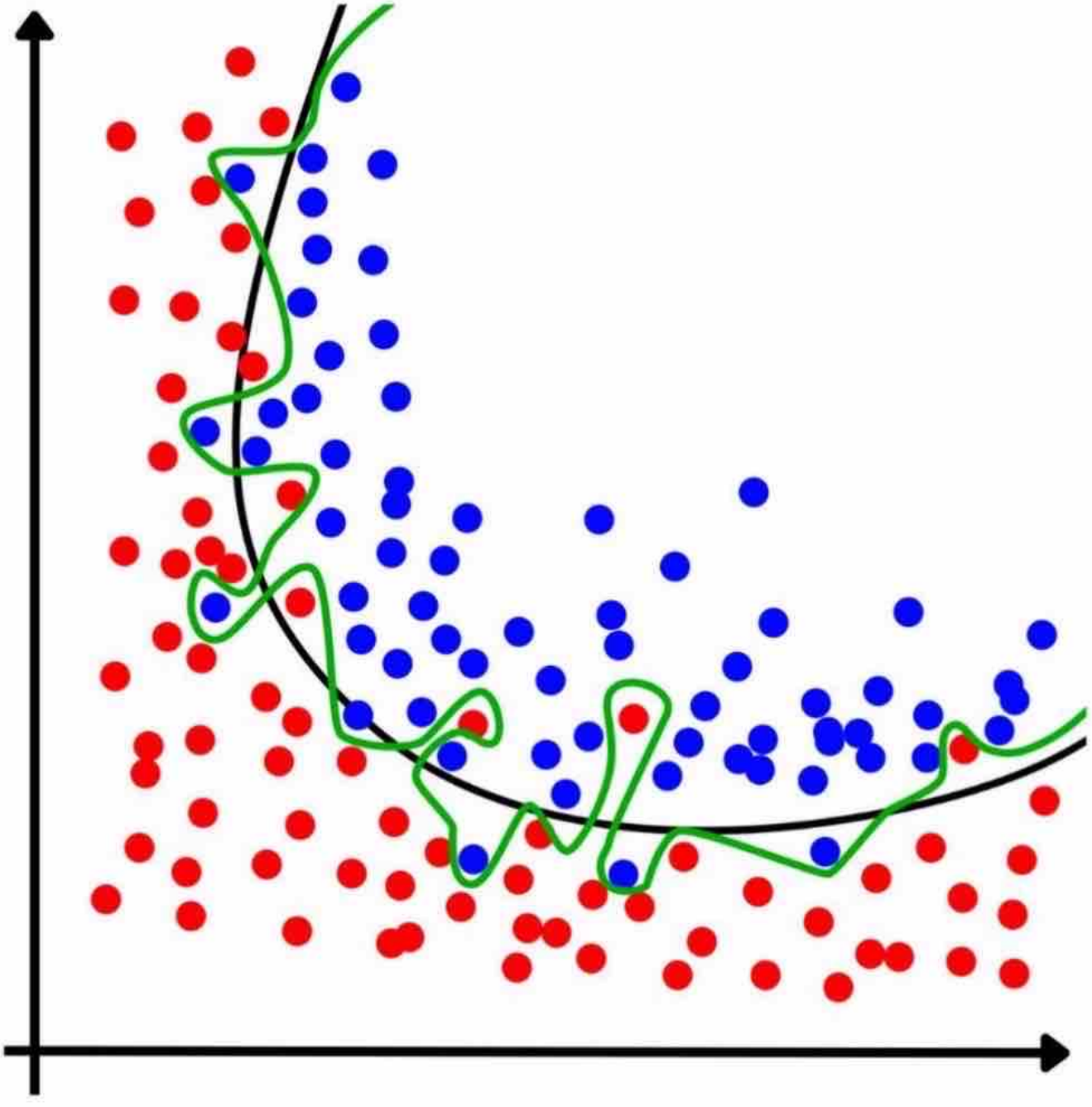
Analytics Vidhya

How to avoid Overfitting in Machine learning



Certainly! Avoiding overfitting is critical for building **robust machine learning models**. Overfitting occurs when a model learns the training data too well, including its noise and outliers, making it perform poorly on unseen data.

Here are some techniques to prevent overfitting along with Python code examples

Analytics Vidhya

Train-Test Split

First and foremost, split the data into training and test sets to validate the model's performance on unseen data.

```
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)
```

Cross-Validation

Use cross-validation to get a more reliable estimate of the model's performance.

```
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier()
scores = cross_val_score(clf, X, y, cv=5)  # 5-fold c
```

Regularization

Analytics Vidhya

Apply regularization methods like L1 or L2 regularization to add some form of penalty to the model.

For Ridge Regression (L2)

```
from sklearn.linear_model import Ridge

ridge = Ridge(alpha=1.0)
ridge.fit(X_train, y_train)
```

For Lasso Regression (L1)

```
from sklearn.linear_model import Lasso

lasso = Lasso(alpha=1.0)
lasso.fit(X_train, y_train)
```

Use Dropout for Neural Networks

Analytics Vidhya

Introduce dropout layers in neural networks to randomly set a fraction of input units to 0 during training.

```
from keras.models import Sequential
from keras.layers import Dense, Dropout

model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

https://leimao.github.io/blog/Dropout-Explained/

https://medium.com/@chirag.sharma0378/dropout-68913941f569

Analytics Vidhya

Early Stopping

Stop training when the model's performance starts to degrade on a held-out validation dataset.

```
from keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss', patience=2)
model.fit(X_train, y_train, epochs=50, validation_split=0.2, callbacks=
[early_stopping])
```

Ensemble Methods

Use ensemble methods like Random Forest or Gradient Boosting to average out biases and reduce variance.

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=50)
rf.fit(X_train, y_train)
```

Feature Selection

Analytics Vidhya

Reduce the dimensionality of the data by selecting only the most important features.

```
from sklearn.feature_selection import SelectKBest, f_classif

selector = SelectKBest(score_func=f_classif, k=5)
X_new = selector.fit_transform(X, y)
```

By using one or a combination of these techniques, you can reduce the risk of your model overfitting the training data. This will result in a model that generalizes better to new, unseen data.