ANOMALY DETECTION

Lesson 3: Linear Methods—Regression, PCA, SVM

Learning objectives

You will be able to:

- Describe linear methods for anomaly detection
- Apply linear regression models
- Apply principal component analysis (PCA)
- Apply one-class support vector machines (SVM)
- Use Python* to perform anomaly detection data with these methods

Linear methods for anomaly detection

Introduction

- In the previous lesson (lesson 2), we used probabilistic and structural (geometric) models for anomaly detection
- Here we are going to look at other models for anomaly detection
  - Linear regression models
  - Principal component analysis (PCA)
  - Support vector machines (SVM)

Linear methods for anomaly detection

Why are these “linear” methods?

- Example of a linear function: $y = b_0 + b_1 x$
- Such functions are at the foundation of both linear regression models and SVM
  - Provide the score for anomaly detection
- Linear functions are simple: if the original problem is non-linear, it is usually advantageous to transform it into a linear problem

$$y = b_0 + b_1 x^2 \rightarrow y = b_0 + b_1 z$$

Linear methods for anomaly detection

Why are these “linear” methods (continued)?

- There is another use of “linear”: a linear map
- Formal definition:

$$f(\boldsymbol{x}+\boldsymbol{y}) = f(\boldsymbol{x}) + f(\boldsymbol{y})$$

$$f(c\boldsymbol{x}) = cf(\boldsymbol{x})$$

- It is a transformation between two vector spaces (**x** and **y**) that preserves vector addition and multiplication by a scalar (*c*)
- Linear maps are an essential part of PCA
  - Transform original data into principal components

LINEAR REGRESSION MODELS

Linear regression models

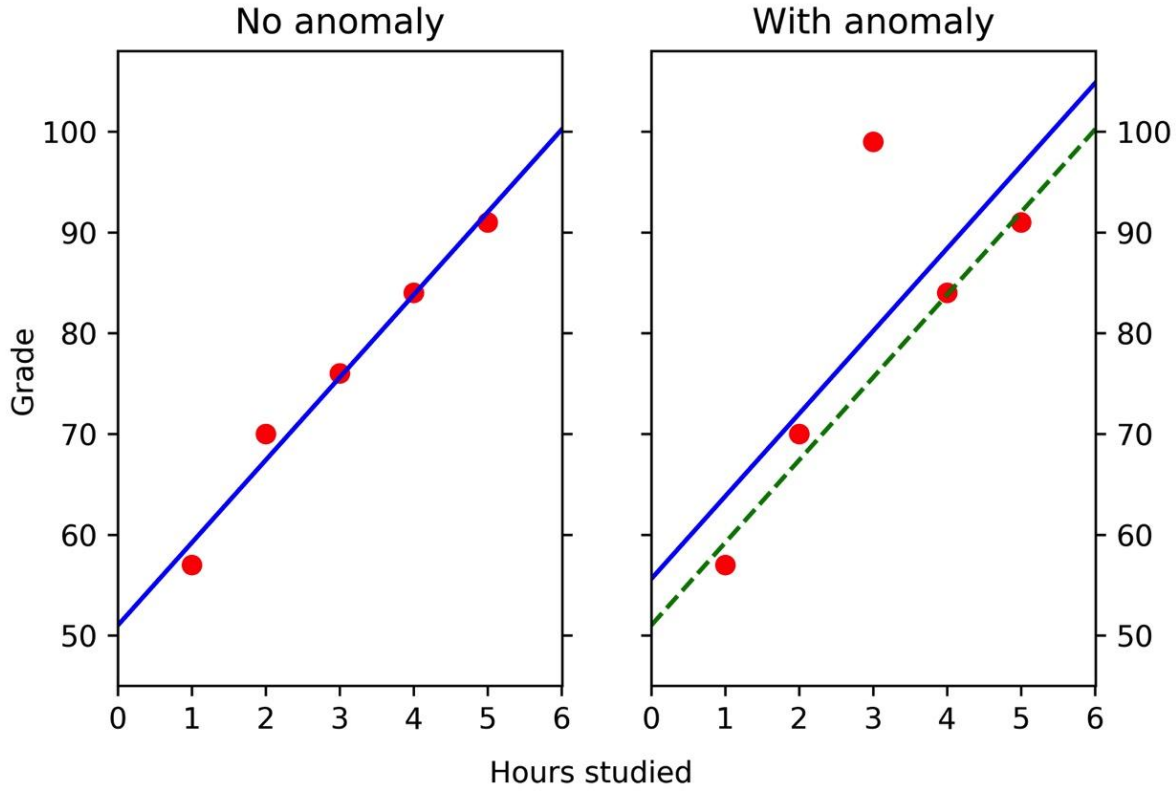
Introduction

- Look for a linear relationship between responses (dependent variables) and explanatory variables (independent variables)
- For simplicity, we will start with just one dependent and one independent variable
  - Simple linear regression
  - 2D data: y vs. x
  - Look for straight line that best fits the data

Linear regression models

Example

- Data on exam grade and hours studied
  - One set of normal data
  - One with an anomaly (that replaces a normal point)
- Plot data as grade (y) vs. hours (x)
- Fit a straight line to the data


No anomaly
With anomaly
Grade
Hours studied
100
90
80
70
60
50
0
1
2
3
4
5
6

Linear regression models and anomaly detection

Discussion

- Vertical distance from straight line fit is used to score points
  - Anomalies are far from line (predicted value differs a lot from actual value)
- Ideally, split the data into train and test datasets
- Use the train data
  - to get the parameters of the straight line fit
  - determine the distance threshold for anomalies
- Apply the results to test data to detect anomalies

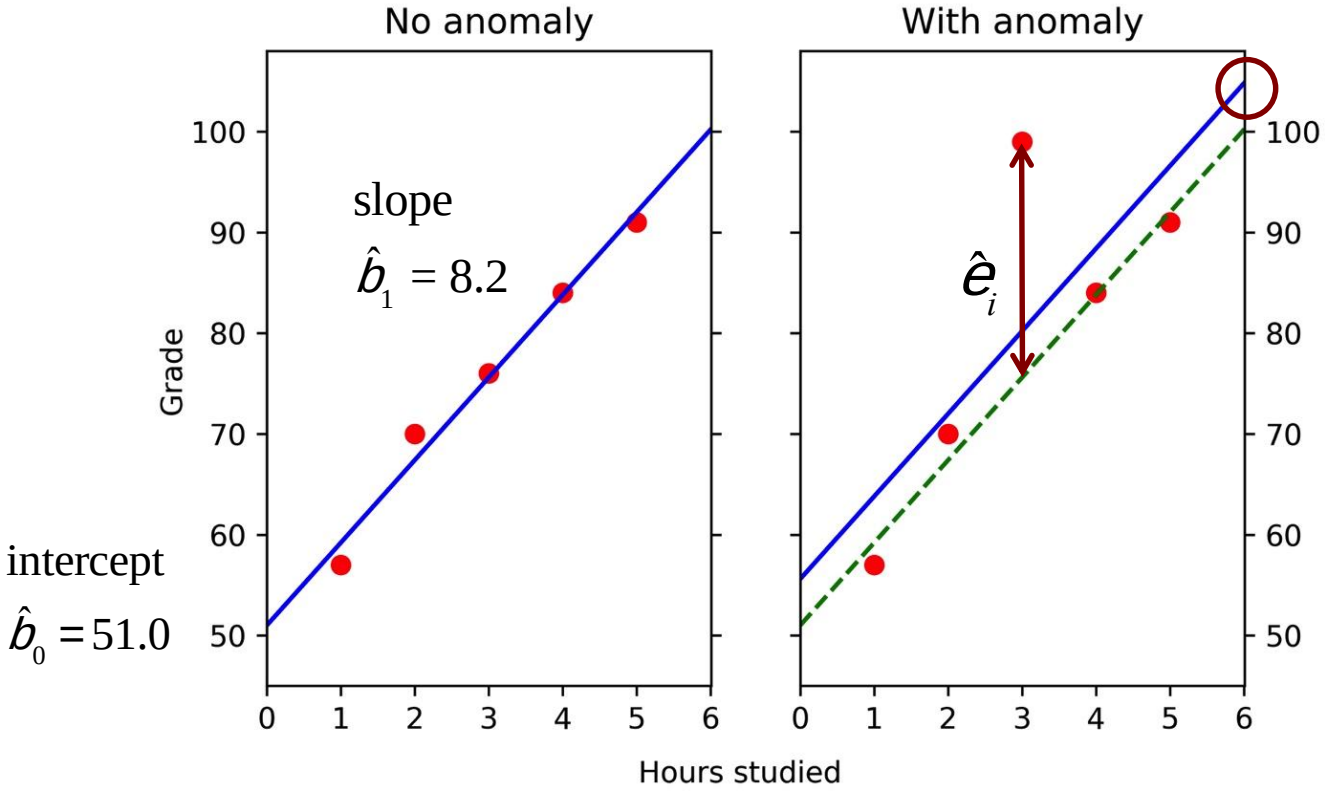
Linear regression models and anomaly detection

How to fit the data

- For a simple linear regression assume: $y_i = b_0 + b_1 x_i + e_i$
- Goal: find estimated values of fit parameters $\hat{b}_0, \hat{b}_1$ estimates of b_0, b_1
- Use least-squares approach: minimize sum of squares of residuals

$$\sum_{i=1}^{n} \hat{e}_i^2 = (y_i - \hat{b}_0 + \hat{b}_1 x_i)^2 \rightarrow \quad \hat{b}_1 = \frac{\mathrm{cov}(x, y)}{\mathrm{var}(x)}$$

$$\hat{b}_0 = \bar{y} - \hat{b}_1 \bar{x}$$


No anomaly
With anomaly
slope
$\hat{b}_1 = 8.2$
$\hat{e}_i$
intercept
$\hat{b}_0 = 51.0$
Grade
Hours studied
100
90
80
70
60
50
0
1
2
3
4
5
6

Linear regression models and anomaly detection

Scoring the anomaly

- Often the square of the residual is taken as a score $\hat{e}_i^2$
  - Large values (either above or below the fit) are anomalies
- Another approach is to use the z-score
  - Calculate the standard deviation of the residuals: σ
  - The z-score is given by $z = \hat{e}_i / s$

  (by definition the mean of the residuals is zero)

Linear regression models and anomaly detection

A note of caution

- If possible, the fit should be carried out on only normal data
- Anomalies affect the fit itself and the effect can be large
- Example shows how straight line moves towards the anomaly
  - Makes anomaly less anomalous (closer to line than it should be)
  - Makes normal points more anomalous (further from line than they should be)

Linear regression models: generalization

Beyond the simple linear regression

- More than one independent variable (multiple linear regression)

$$y_i = b_0 + b_1 x_{1i} + \ldots b_p x_{pi} + e_i$$

  - Independent variables can be non-linear
  - Model remains linear as long as dependent variable is linear in β's
- Approach to anomaly detection remains the same
  - Use residual for scoring anomaly

PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA)

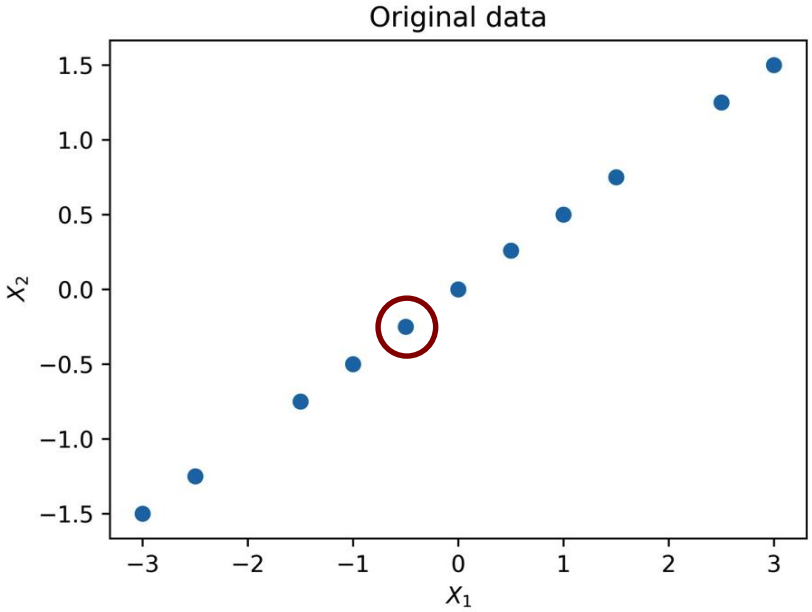
Introduction

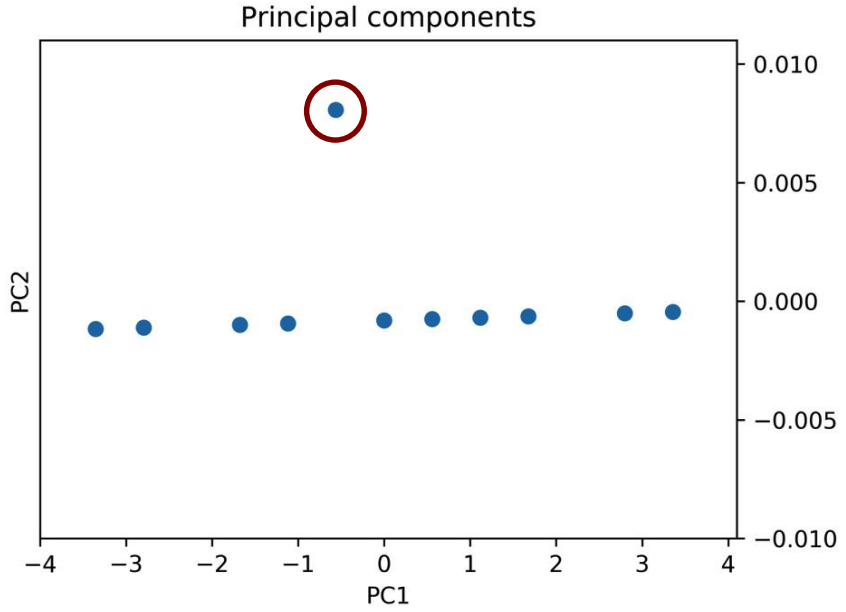
- Linear regression selects one feature of the data (the dependent variable) and compares it with all other features (the independent variables).
- While sometimes it is useful to single out one of features as the dependent variable, there are other times where it makes more sense to treat all variables in the same way
- PCA analyzes data of correlated variables to extract important information
- This information is expressed as a set of uncorrelated variables called principal components

Principal component analysis (PCA)

Example

- 2D data: X_2 and X_1
- As an illustration of PCA, data is essentially linear
  - Points follow $X_2 = X_1 / 2$ with one exception (very small deviation)
- Plot data in two ways
  - Traditional X_2 vs. X_1
  - Principal component 2 (PC2) vs. principal component 1 (PC1)


Original data
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
X_2
−3
−2
−1
0
1
2
3
X_1

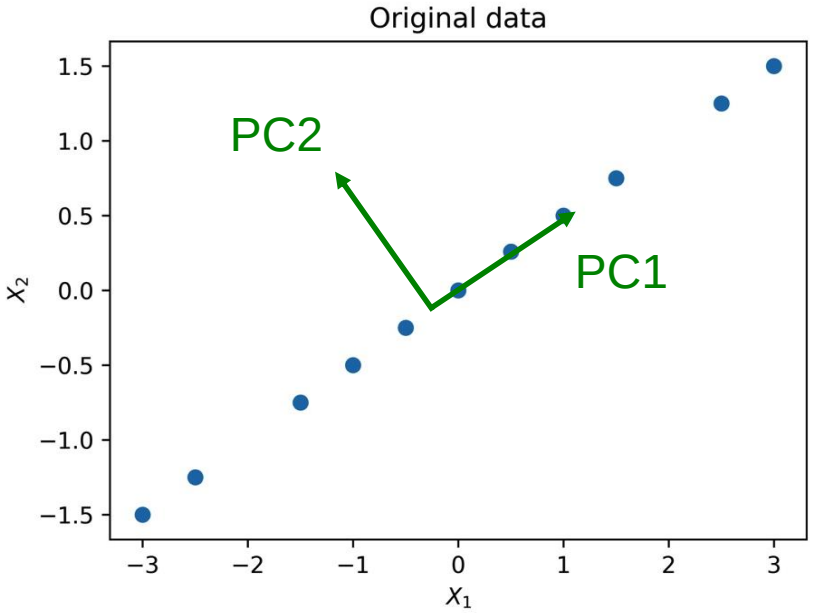

Principal components
0.010
0.005
0.000
−0.005
−0.010
PC2
−4
−3
−2
−1
0
1
2
3
4
PC1

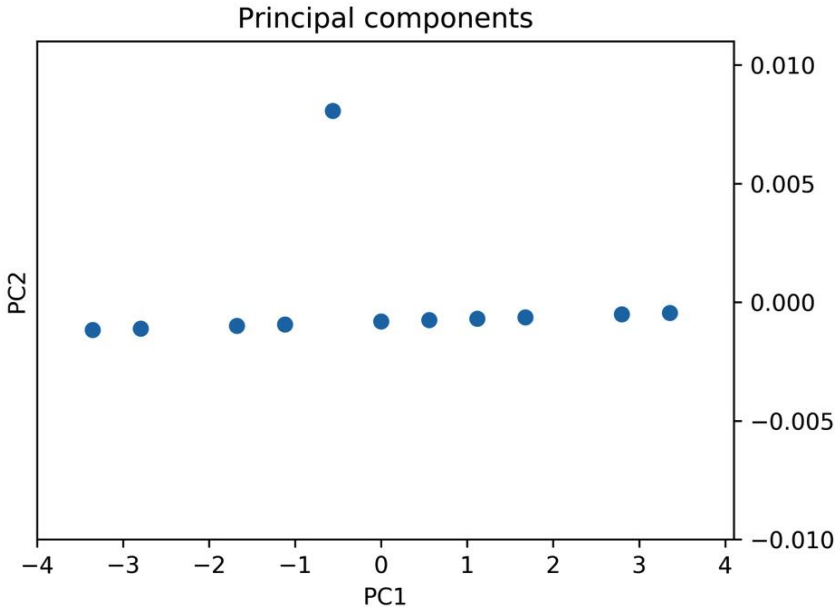
Principal components (PCs)

What are they?

- The principal components are linear combinations of the original features
- These are constructed as follows:
  - PC1: minimize total distance between data and its projections onto PC1*
  - PC2: same construction as PC1 with extra requirement that PC2 is uncorrelated with PC1
  - PC3: same construction as PC1 but uncorrelated with PC1 and PC2
- Maximum number of PCs is minimum of (# data points, # features)

PCs are directions of maximal variance


Original data
PC2
PC1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
X_2
−3
−2
−1
0
1
2
3
X_1


Principal components
PC2
0.010
0.005
0.000
−0.005
−0.010
−4
−3
−2
−1
0
1
2
3
4
PC1

PCA and anomaly detection

The underlying idea

- Often find that only a few PCs matter
  - Most of the data aligns along a lower-dimensional feature space
  - This subspace captures most of the variances of the data
- Anomalies are those points that don't align with this subspace
  - In previous example, all of the data aligned along PC1 except one point
- Distance of the anomaly from the aligned data can be used as anomaly score

PCA and anomaly detection

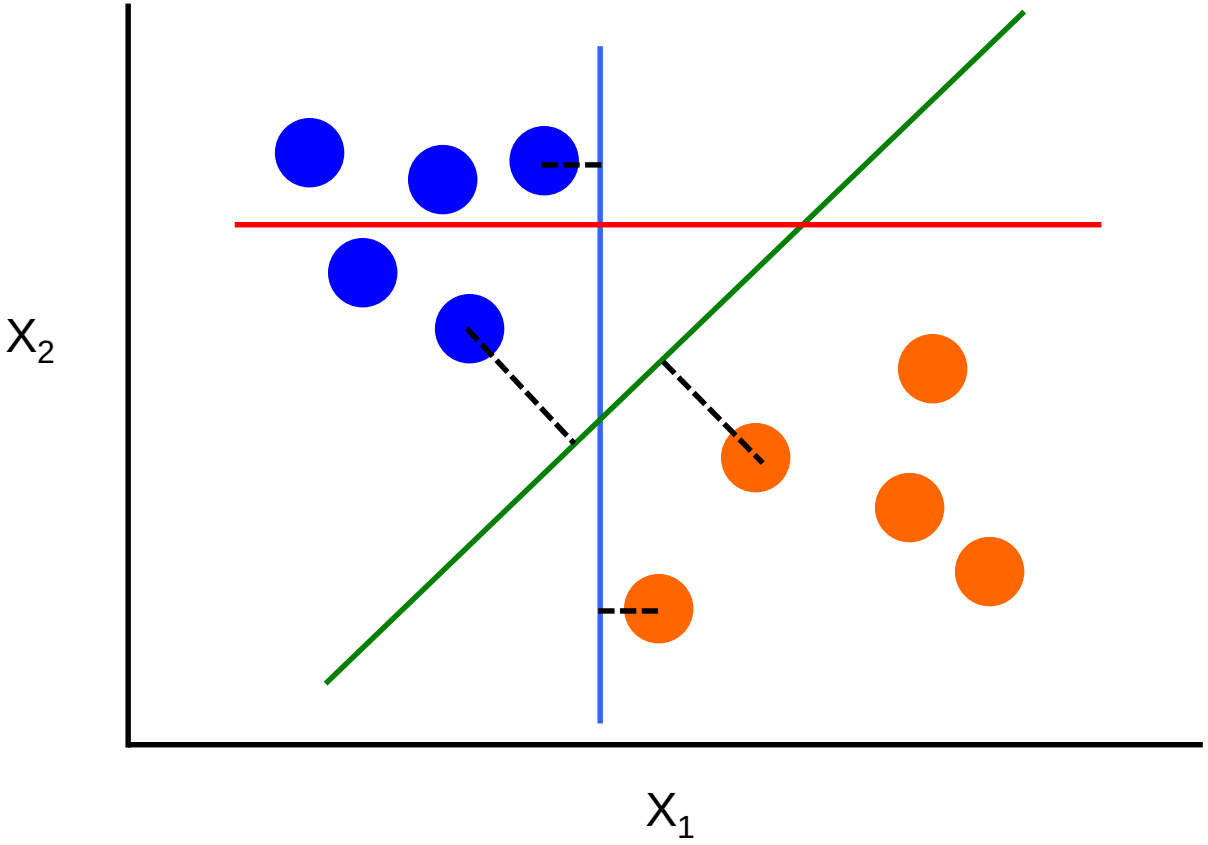
Caveats

- As with linear regression, anomalies can affect the modeling
  - Ideally, find PCs with only normal data
  - If this is not possible, after detecting anomalies, remove them from data and repeat analysis to check PCs haven't changed significantly
- Preprocessing of data may be needed
  - standardizing (mean = 0 and standard deviation = 1)
- Also, not all data patterns are suitable for PCA
  - For example, non-linear patterns such as spherical shells of points

Support vector machines (SVMs)

An overview

- SVMs are supervised learning models used for classification
- Typically used to classify data in two classes
- Training data requires labeled examples of both classes
- Find hyperplane with largest separation (“margin”) between two classes (the decision boundary)
- New data is classified according to which side of the hyperplane it falls on


X_2
X_1

Support vector machines (SVMs)

Next steps

- What if you can't neatly separate the classes?
  - Not linearly separable (No suitable hyperplane)
- Transform data to a high-dimensional space where data is linearly separable
- This transformation is known as the "kernel trick"
  - Use kernel function to efficiently calculate decision boundary
- Choice of kernel will depend on the type of data

Support vector machines and anomaly detection

Two approaches

- Have labeled normal data and anomalies
  - Use SVMs as a supervised learning model (discussed in a future lesson)
- Have unlabeled data
  - Use one-class SVMs (discussed here)
  - Since data is unlabeled require assumptions to proceed

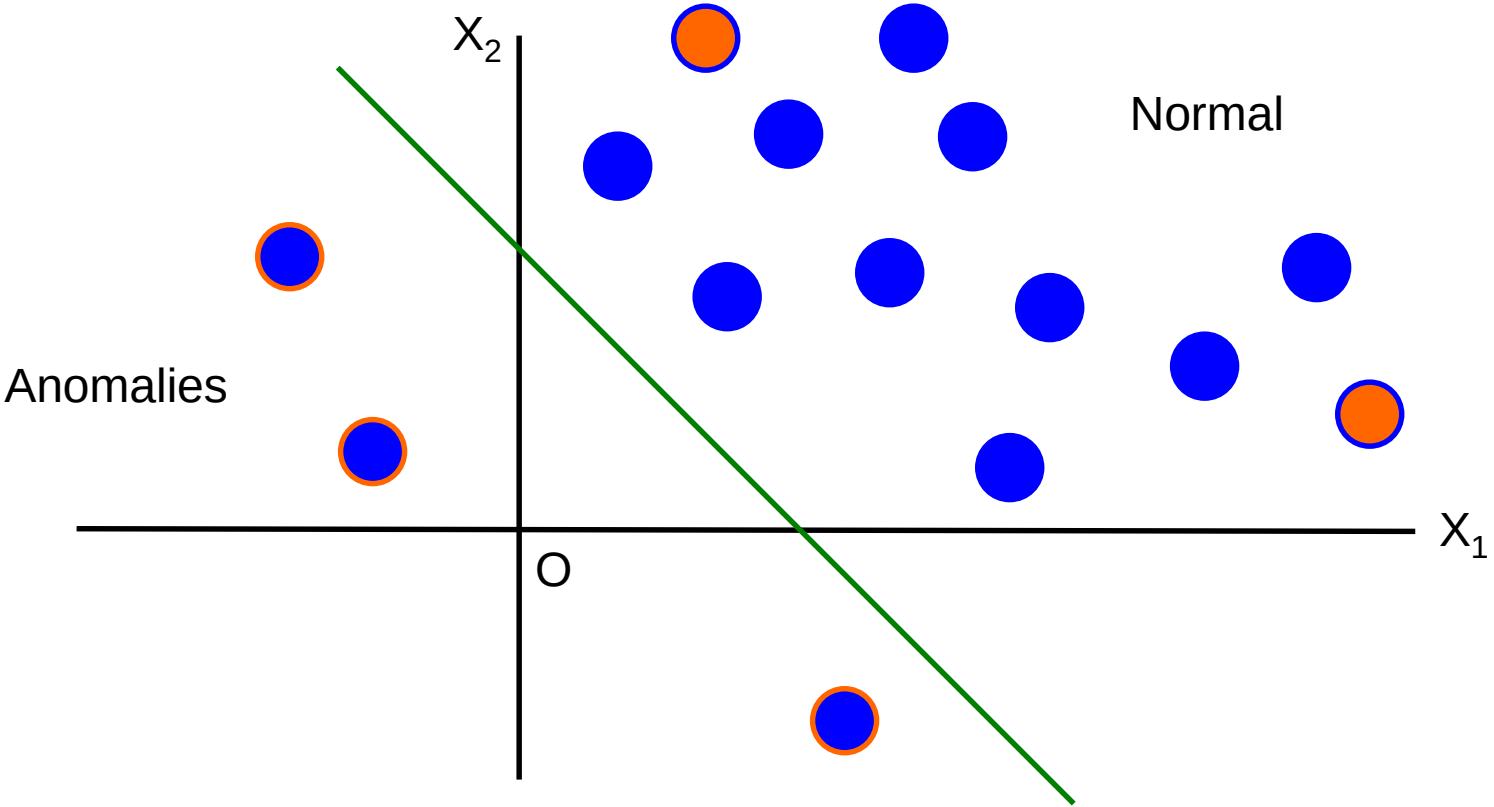
One-class SVM

Two key assumptions

- Data provided all belong to normal class
  - Since data may contain anomalies this results in a noisy model
- The origin belongs to the anomaly class
  - Rarely use data as is. Origin is that of kernel-based transformed data

Consequence

- All data on one side of decision boundary is classified as normal
- Data on other side (with origin) is an anomaly


X2
Normal
Anomalies
O
X1

One-class SVM and anomaly detection

A note of caution

- The shape of the decision boundary is sensitive to the choice of kernel and other tuning parameters of SVMs
- Without deep knowledge of both the data and SVMs, it is easy to get poor results
- To address this issue, sampling of subsets of the data and averaging of scores is recommended (as discussed for linear regression)

CONCLUSION

Use Python* for anomaly detection

Next up is a look at applying these concepts in Python*

- See notebook entitled *Linear_Anomaly_Detection_student.ipynb*

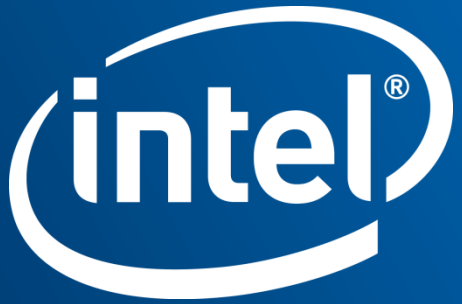
Learning objectives recap

In this session you learned how to:

- Describe linear methods for anomaly detection
- Apply linear regression models
- Apply principal component analysis (PCA)
- Apply one-class support vector machines (SVM)
- Use Python* to perform anomaly detection data with these methods

References

- *An Introduction to Linear Regression Analysis* by D.C. Montogomery, E.A. Peck, C.G. Vining (Wiley 2013)
- *Principal Component Analysis* by H. Abdi, L. J. Williams (2010)
- *A User's Guide to Support Vector Machines* by A. Ben-Hur, J. Weston (2007)


intel®