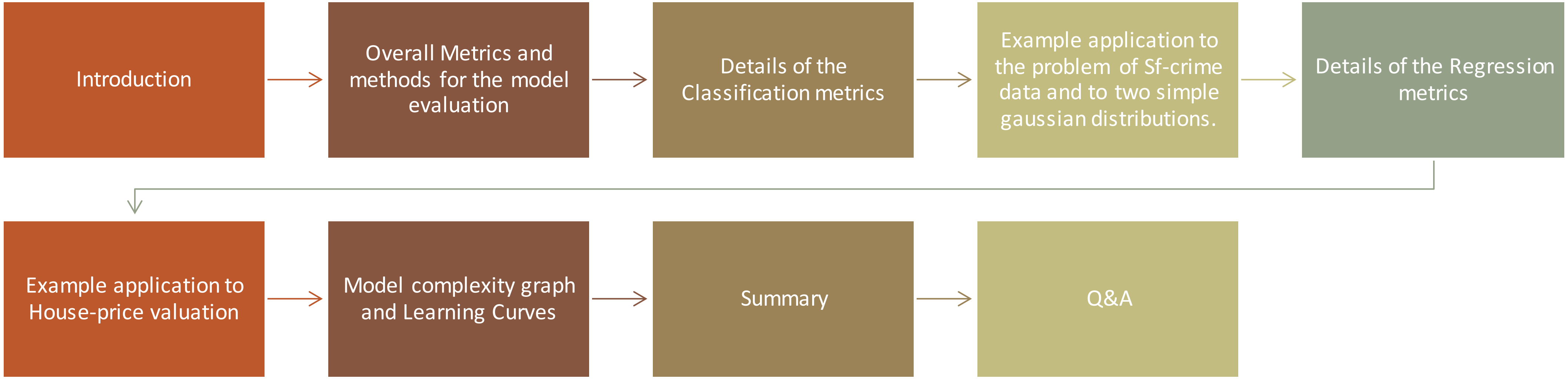
Metrics/Methods for Model Evaluation

Agenda

By Group 15
– Harika, Roy, Kylin, Shane, Jn



Introduction

Overall Metrics and methods for the model evaluation

Details of the Classification metrics

Example application to the problem of Sf-crime data and to two simple gaussian distributions.

Details of the Regression metrics

Example application to House-price valuation

Model complexity graph and Learning Curves

Summary

Q&A

Justify why the chosen Model is Best

Introduction:

The ***machine learning models*** and ***statistical methods*** have been widely used in various fields.

Wide application does not mean that each model can adapt to all business scenarios.

Given the data, ***distributions and patterns are different*** even in a specific business scenario, it is hard to decide which model performs for your business.

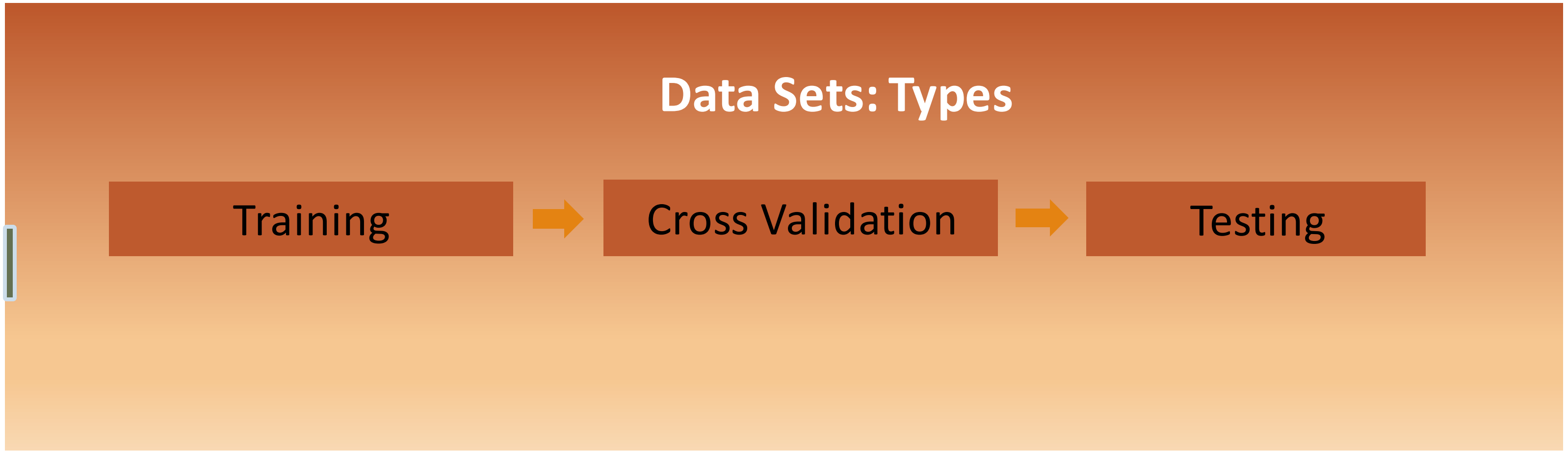
Explain ***why this model has been chosen***? So It is vital to explore correct use of model selection and evaluation techniques.

Explore ***more tools and methodologies***, which helps to explain why the chosen model is best for the given scenario

In ***simple*** and ***understandable*** terms.

Model Evaluation Metrics

- *Classification Metrics (accuracy, precision, recall, F1-score, ROC, AUC..*
- *Regression Metrics (MSE, MAE)*
- *Ranking Metrics (MRR, DCG, NDCG)*
- *Statistical Metrics (Correlation)*
- *NLP Metrics (Perplexity, BLEU score)*
- *Deep Learning Related Metrics (Inception score, Frechet Inception distance)*



Data Sets: Types

Training → Cross Validation → Testing

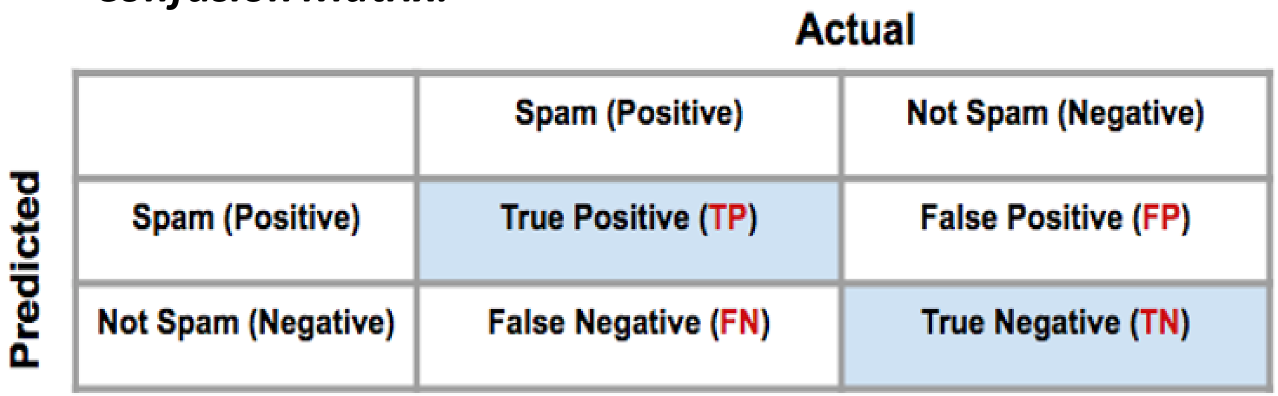
Evaluation Metrics - Classification

There are several evaluation metrics for classification problems, some important ones:

- Accuracy, precision, Recall
- F1-score, F-Beta Score
- ROC Curve
- AUC
- Loss function/Cross Entropy

Evaluation Metrics - Classification

Confusion Matrix:



Actual

Predicted	Spam (Positive)	Not Spam (Negative)
Spam (Positive)	True Positive (TP)	False Positive (FP)
Not Spam (Negative)	False Negative (FN)	True Negative (TN)

- ❖ In case of binary classification with **imbalanced classes**, accuracy evaluation is of little reference value, Therefore, **precision** and **recall** are more useful in the problem context.

$$\textbf{Accuracy} = \frac{\text{True Positives + True Negatives}}{\text{TP + FP + FN + TN}}$$

$$\textbf{Precision} = \frac{\text{True Positives}}{\text{TP + FP}}$$

$$\textbf{Recall} = \frac{\text{True Positives}}{\text{TP + FN}}$$

- ❖ **Precision** focuses on the “**predicted positive**" values in your dataset. You can determine if you are doing a good job of predicting the positive values, as compared to predicting negative values as positive.

Example: **Spam Email classifier** should be **high precision** model.

- ❖ **Recall** focuses on the “**actual positive**" values in your dataset. You can determine if you are doing a good job of predicting the positive values without regard to how you are doing on the actual negative values.

Example: **Diagnosing the heart disease** should be **high recall** model

Ref: https://manoa.hawaii.edu/exploringourfluidearth/physical/world-ocean/map-distortion/practices-science-precision-vs-accuracy

Evaluation Metrics - Classification

❖ F-Beta Score:

In order to look at **a combination of metrics** at the same time, there are some common techniques like the F-Beta Score (where the F1 score is frequently used)

$$F_\beta = (1 + \beta^2) \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

Beta <1, more focus on precision

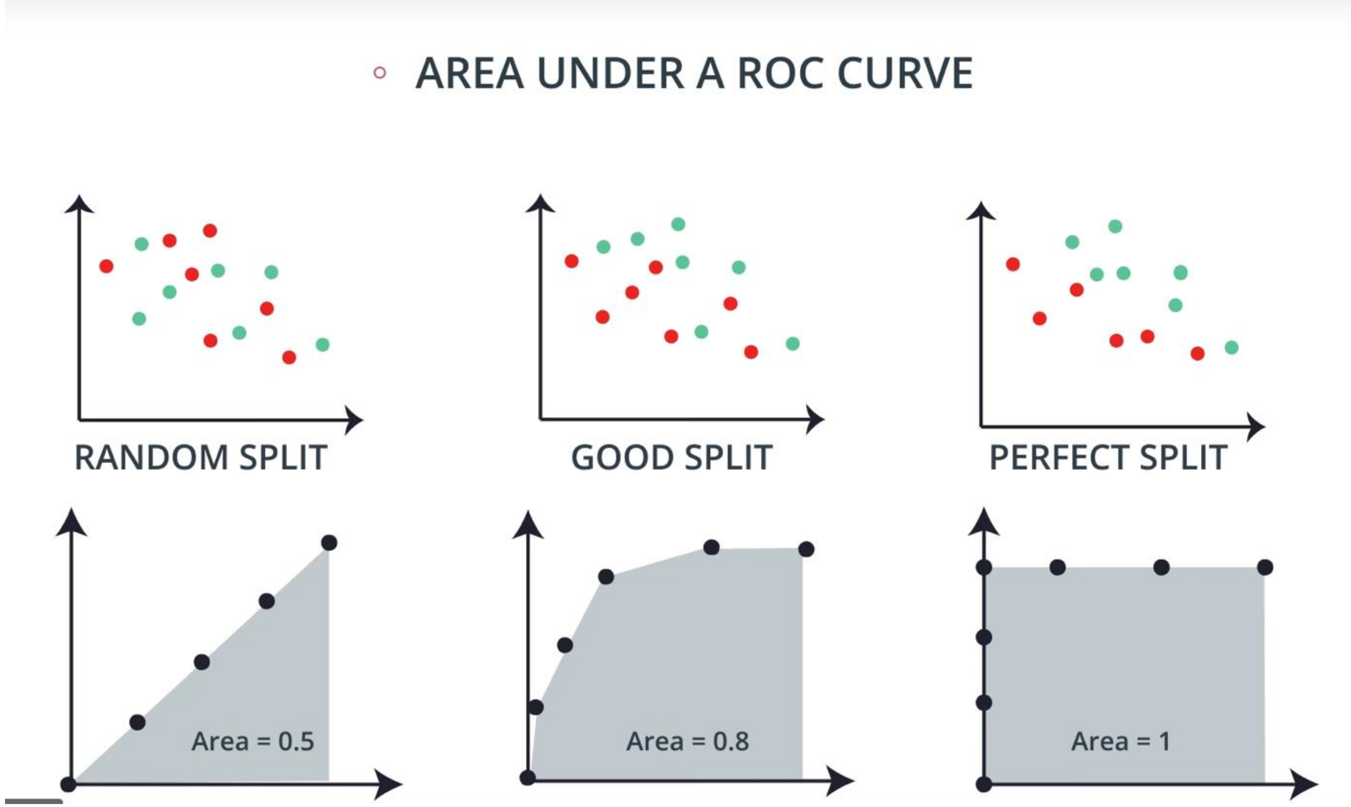
Beta >1, more attention on recall

Beta =1, both are important

Ref: https://en.wikipedia.org/wiki/F1_score

Evaluation Metrics - Classification

ROC Curve & AUC



- ❖ By finding different thresholds for our classification metrics, we can measure the area under the curve (where the curve is known as a ROC curve).

- ❖ When the AUC is higher (closer to 1), this suggests that our model performance is better than when our metric is close to 0.

Ref: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Evaluation Metrics - Classification

Loss Function

The loss function is generally used in training, but we can still use it to evaluate the performance of the model we get, which is to evaluate the difference between the calculated results of the model and the real value.

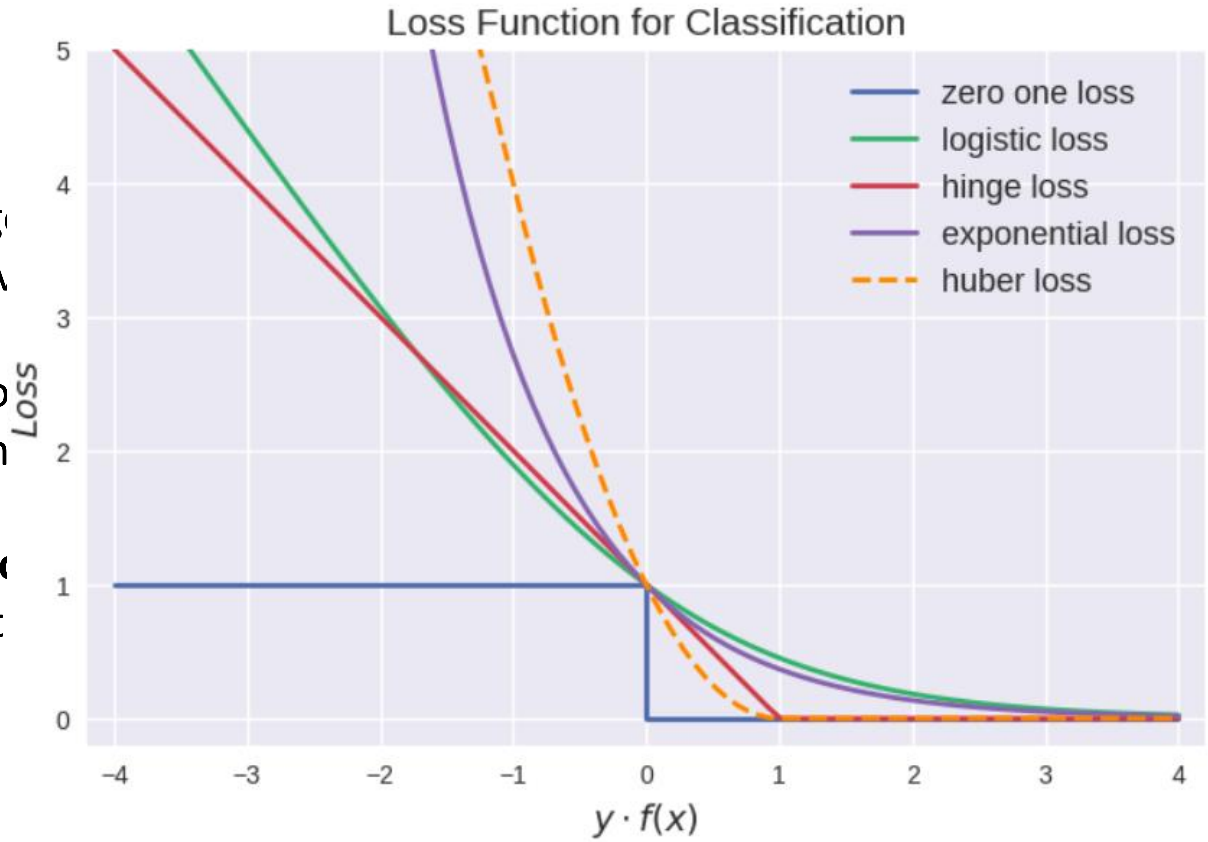
The most common loss function in multi-class classification problems is multi-class logarithmic loss, or the cross entropy loss function.

It comes from **logistic regression** and can be used for multiple classifications. It uses the idea of maximum likelihood. When outputting results, return a set of predicted probabilities (one for every class), and then we can calculate its logloss.

$$L_{\log}(Y, P) = -\log \Pr(Y|P) = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{i,k} \log p_{i,k}$$

Evaluation Metrics - Classification

Loss Function



The loss function is g
performance of the model
we get, which is to ev
ɔdel and the real value.

The most common lo
logarithmic loss, or the cross
entropy loss function

It comes from **logisti**
; the idea of maximum
likelihood. When out
every class), and then we can
calculate its logloss.

$$\log \qquad N \sum_{i=0} \sum_{k=0}$$

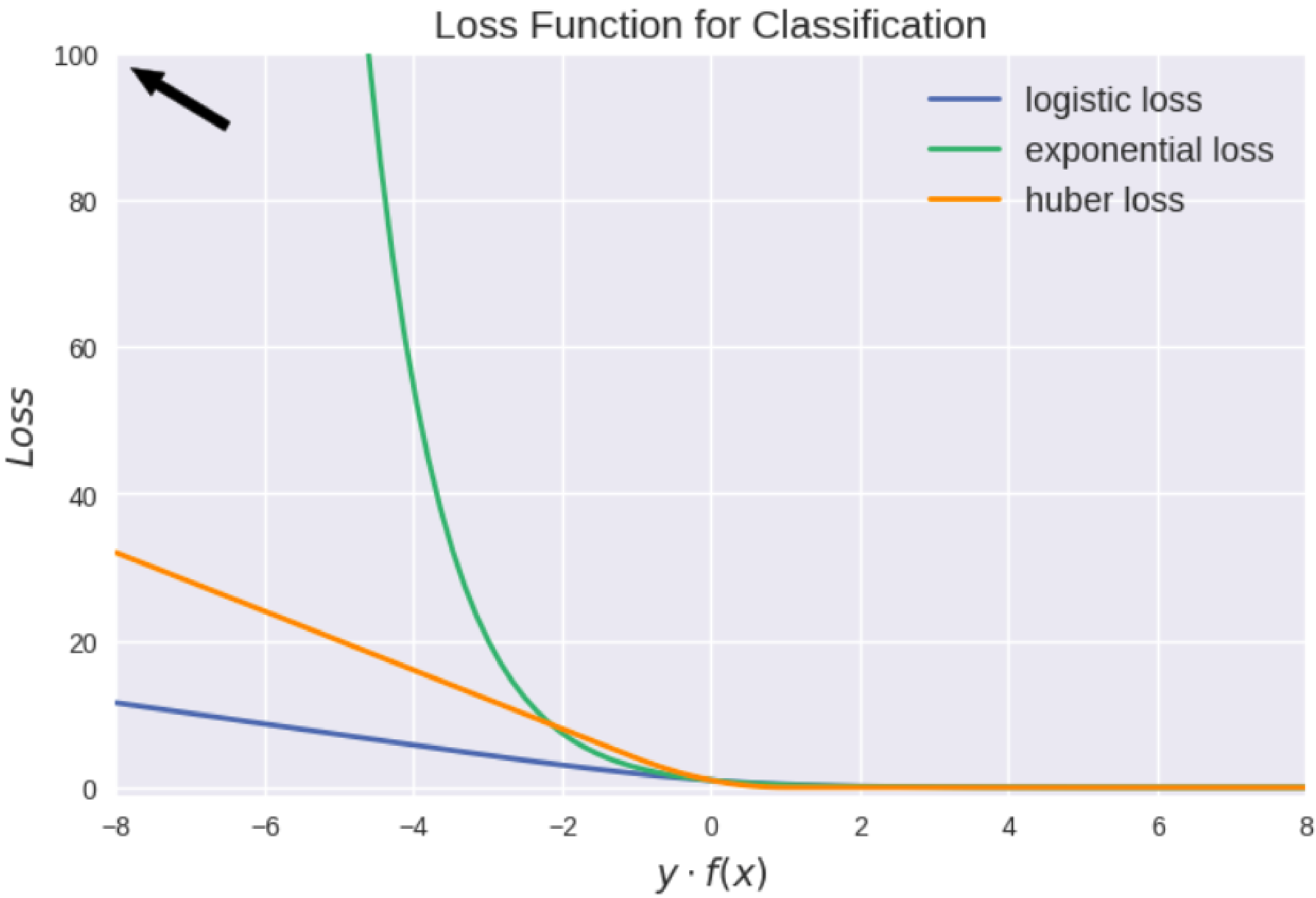
Evaluation Metrics - Classification

Loss Functior

The loss function is g rmance of the model
we get, which is to e ınd the real value.

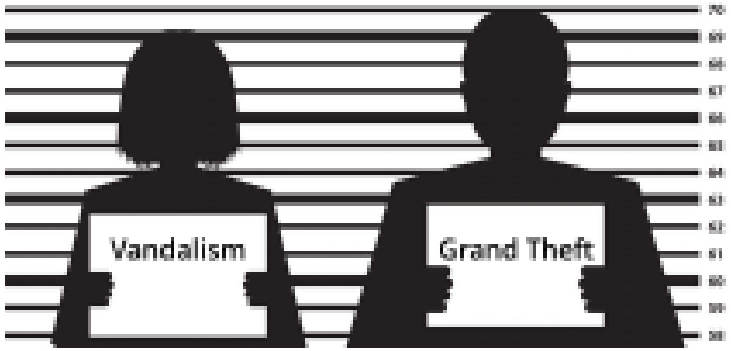
The most common l thmic loss, or the cross
entropy loss functio

It comes from **logisti** dea of maximum
likelihood. When ou class), and then we can
calculate its logloss.



Classification Example Application

Data Set: **San Francisco Crime Classification**



Nearly 12 years of crime reports from across all of San Francisco's neighbourhoods.

- ❖ For this data set, given the time and location, we need to predict the category of the crime that has been occurred.
- ❖ ***Algorithms applied:*** KNN, Naïve-bayes, RandomForest, Adaboost, Xgboost, Lightgbm.
- ❖ ***Metrics used to evaluate the models*** : Accuracy, Logloss, Precision, Recall, F1 score, ROC curve & AUC.

Accuracy, Logloss, Precision, Recall & F1-Score

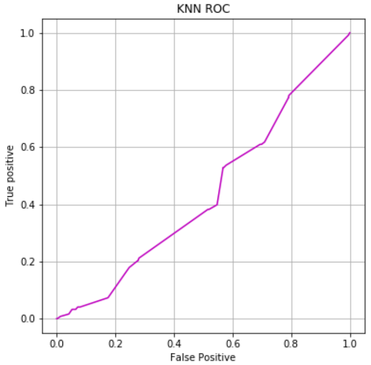
	Accuracy ↑	**Logloss ↓**	**Precision ↑**	**Recall ↑**	**F1 score ↑**	**Compare**
KNN	18.37	21.80	0.08	0.07	0.07	_
Naïve Bayes	22.71	**2.55**	0.03	0.04	0.03	Logloss
Random Forest	26.43	**2.44**	0.09	0.06	0.05	Logloss
Adaboost	20.29	**3.62**	0.02	0.03	0.02	Logloss
Xgboost	26.24	**2.44**	0.10	0.07	0.06	Logloss
Lightgbm	**29.53**	**2.34**	**0.17**	**0.11**	**0.11**	**Logloss**
Compare	Lightgbm	**Lightgbm**	Lightgbm	Lightgbm	Lightgbm	**Lightgbm/Logloss**

So for sf crime dataset, we choose Logloss Metrics to evaluate. Result: Lightgbm is the best model.

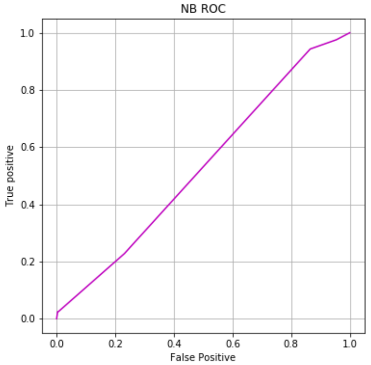
ROC curve & AUC

- ❖ ROC – Receiver Operating Characteristic curve
- ❖ AUC – Area Under the ROC Curve.

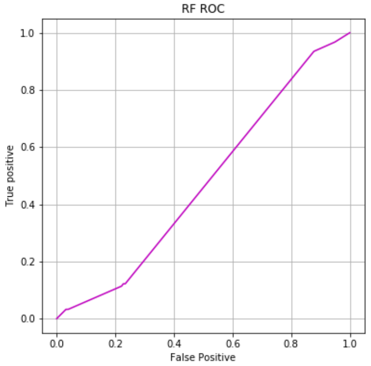
Note: All the AUC results of sf dataset are not very good(Area is not over 0.55 for all the models and ROC curve is not as expected like in the example figure below).



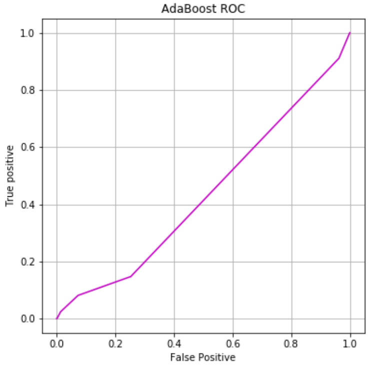
KNN AUC: 0.44



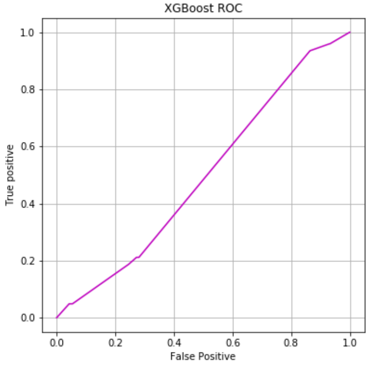
NaiveBayes AUC: 0.53



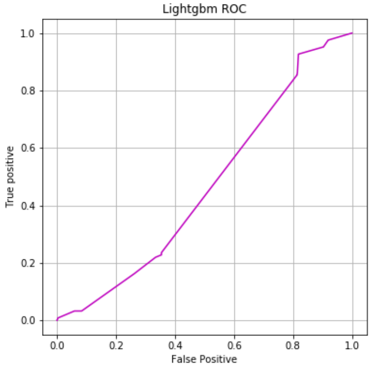
RandomForest AUC: 0.47



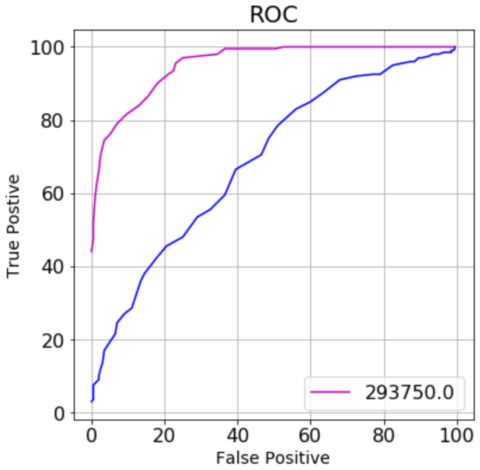
AdaBoost AUC: 0.43



XGBoost AUC: 0.50



Lightgbm AUC: 0.47



Example ROC for simple gaussian distributions

Evaluation Metrics - Classification

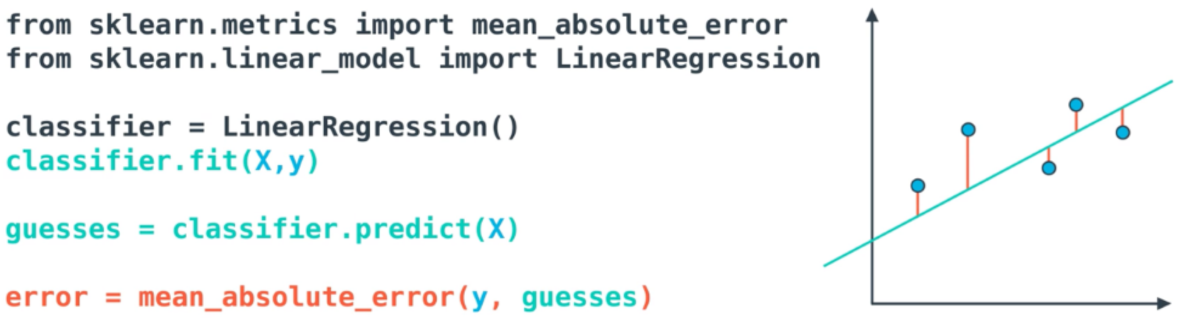
There are three metrics that are commonly used in regression:

- Mean-Squared Error (MSE)
- Mean-Absolute Error (MAE)
- R Squared (R^2)

Evaluation Metrics - Regression

❖ Mean-Absolute Error:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

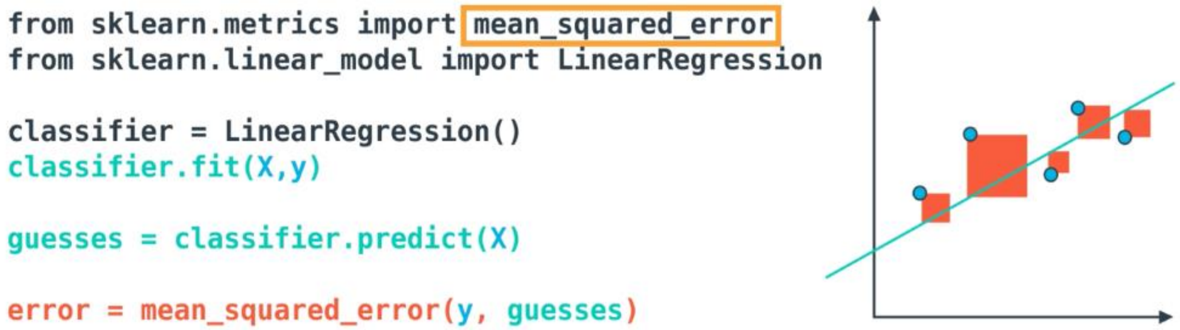


```
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression

classifier = LinearRegression()
classifier.fit(X,y)

guesses = classifier.predict(X)

error = mean_absolute_error(y, guesses)
```

Notes: This is a useful metric to optimize on when the value you are trying to predict follows a skewed distribution However, an absolute value is not differentiable.

❖ Mean-Squared Error:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$



```
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression

classifier = LinearRegression()
classifier.fit(X,y)

guesses = classifier.predict(X)

error = mean_squared_error(y, guesses)
```

Notes: The mean squared error is by far the most used metric for optimization in regression problems. This metric can be greatly impacted by skewed distributions and outliers. In many cases, it is easier to actually optimize on MSE, as a quadratic term is differentiable. This factor makes this metric better for gradient based optimization algorithms.

Evaluation Metrics - Regression

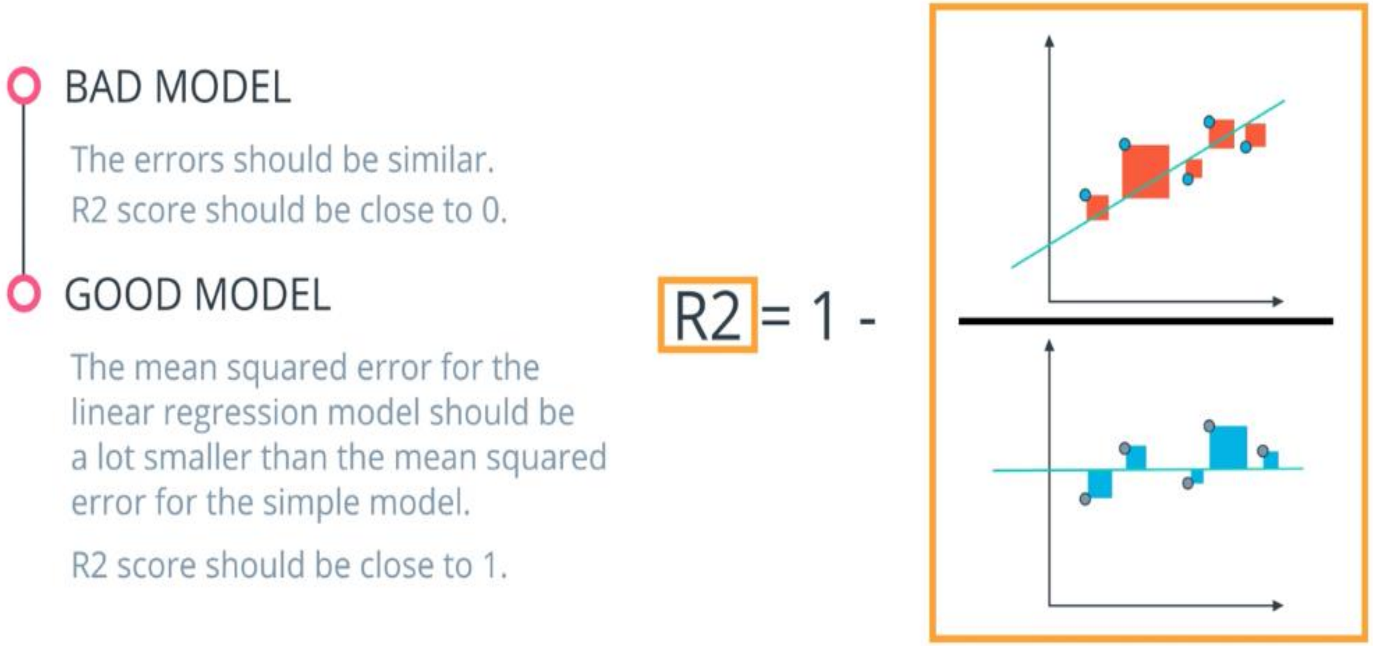
❖ R Squared (R²) :

- ❖ R² is the ratio between how good our model is vs how good is the naive mean model
- ❖ When R² is negative it means that the model is worse than predicting the mean. [1]

❖ The baseline of R²:

$$\mathrm{MSE(baseline)} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

$$R^2 = 1 - \frac{\mathrm{MSE(model)}}{\mathrm{MSE(baseline)}}$$



BAD MODEL

The errors should be similar.
R2 score should be close to 0.

GOOD MODEL

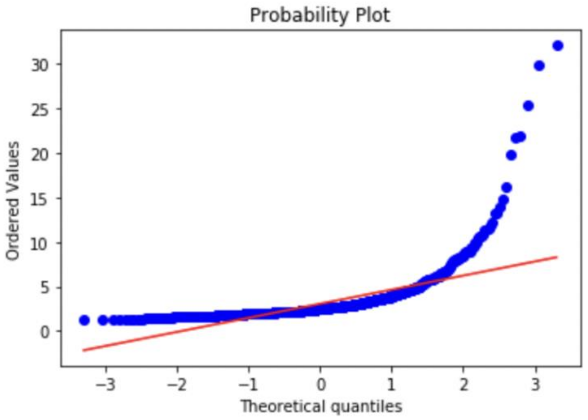
The mean squared error for the linear regression model should be a lot smaller than the mean squared error for the simple model.

R2 score should be close to 1.

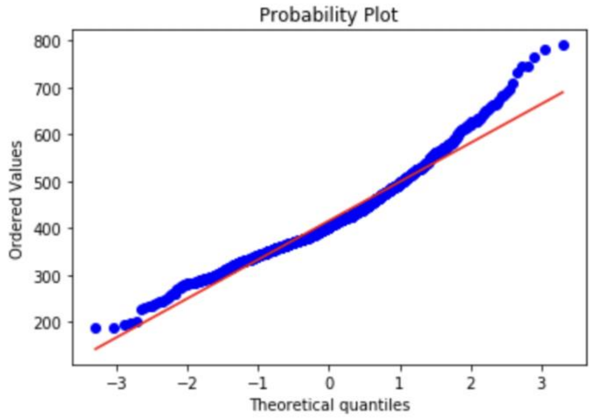
[1]G. Drakos, "How to select the Right Evaluation Metric for Machine Learning Models: Part 1 Regression Metrics", Medium, 2018. [Online]. Available: https://medium.com/@george.drakos62/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-1-regrression-metrics-3606e25beae0. [Accessed: 01- Jan- 2020].

Regression Example Application: **House-price dataset preprocessing**

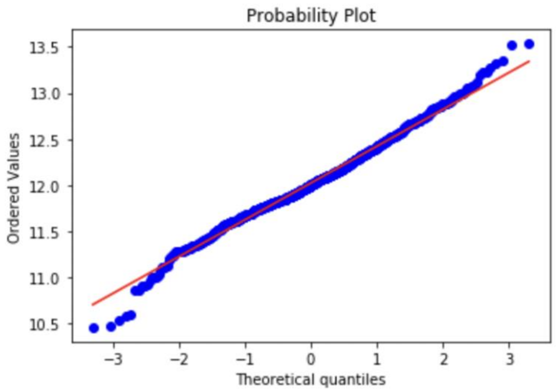
Exponential transformation :



Power function :



log(1+x) function :



- The Q-Q (quantile-quantile) chart is a probability chart used to compare the difference in probability distribution between the observed and predicted values. The comparison object here generally uses a normal distribution. The Q-Q chart can be used to test the similarity of the data distribution.

- The red line is a normal distribution, and the blue line is our data. It can be seen in our data set that it has been seriously deviated from the normal distribution. We try to transform the data as part of the Data pre-processing. Common transformations include exponential transformation, logarithmic transformation, and power function.

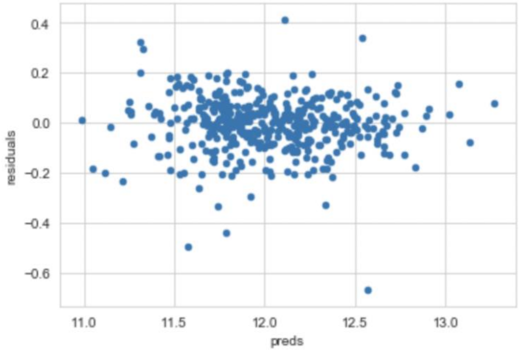
- The comparison of the three functions fits, and the logarithmic conversion is the most consistent, but we know that the logarithm means that it is negative when it is less than 1, which is obviously not consistent with cognition. Log (1 + x), which is log1p, should be used to ensure Validity of x data.

Metrics Calculated for Evaluating the models

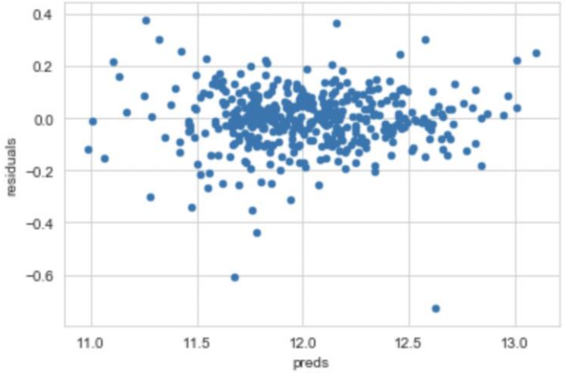
- We trained the data set with the six algorithm models shown on the left, and calculated the evaluation indices through the metrics RMSE and R2-Score.

- The Lasso regression model and the ENet model are best compared to other models.
  - RMSE : 0.1161 and 0.1162.
  - R2-Score: 0.9118 and 0.9117.

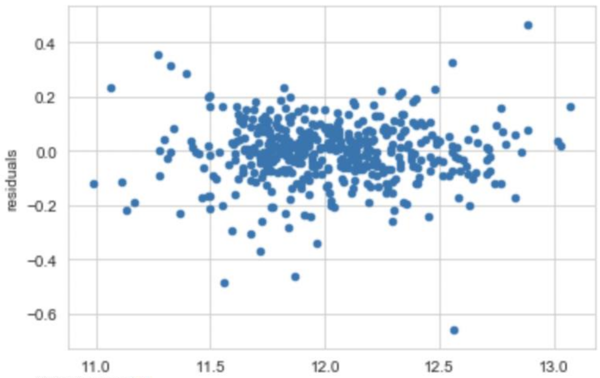
ENet partial residual plot:



GBDT partial residual plot:



LGBM Regressor:

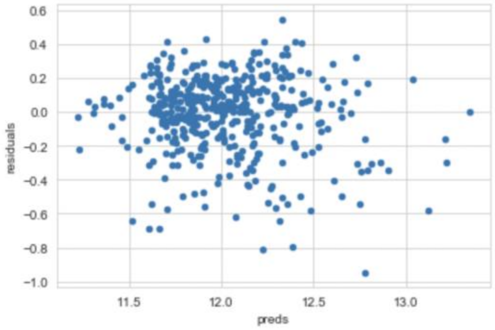


Note: Through the residual plots, we can see that these two models perform the best.

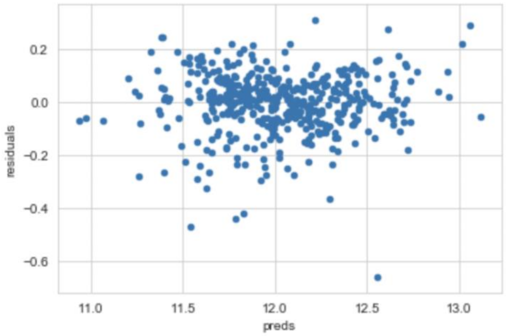
$$RMSE=\sqrt{\frac{\sum_{i=1}^{N}(Predicted_i-Actual_i)^2}{N}}$$

$$R^2 = 1 - \frac{\sum_i \left(\hat{y}^{(i)} - y^{(i)}\right)^2}{\sum_i \left(\bar{y} - y^{(i)}\right)^2}$$

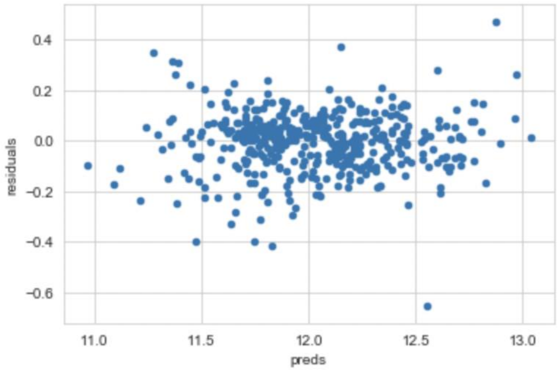
Lasso partial residual plot:



KRR: partial residual plot:



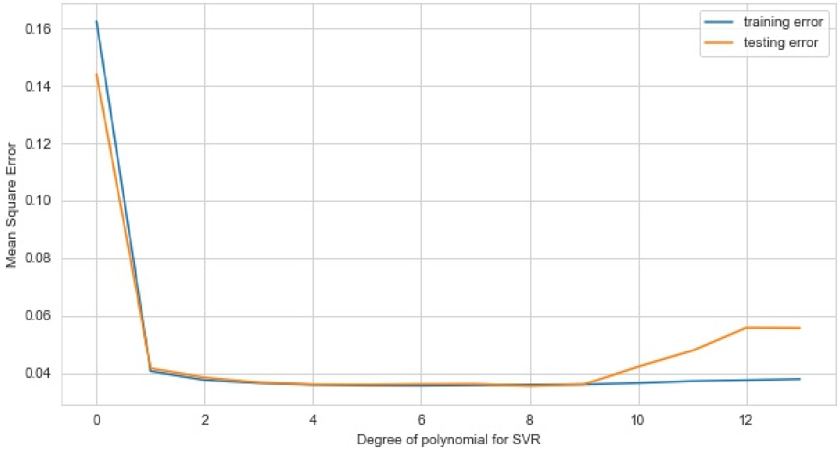
XGB Regressor:

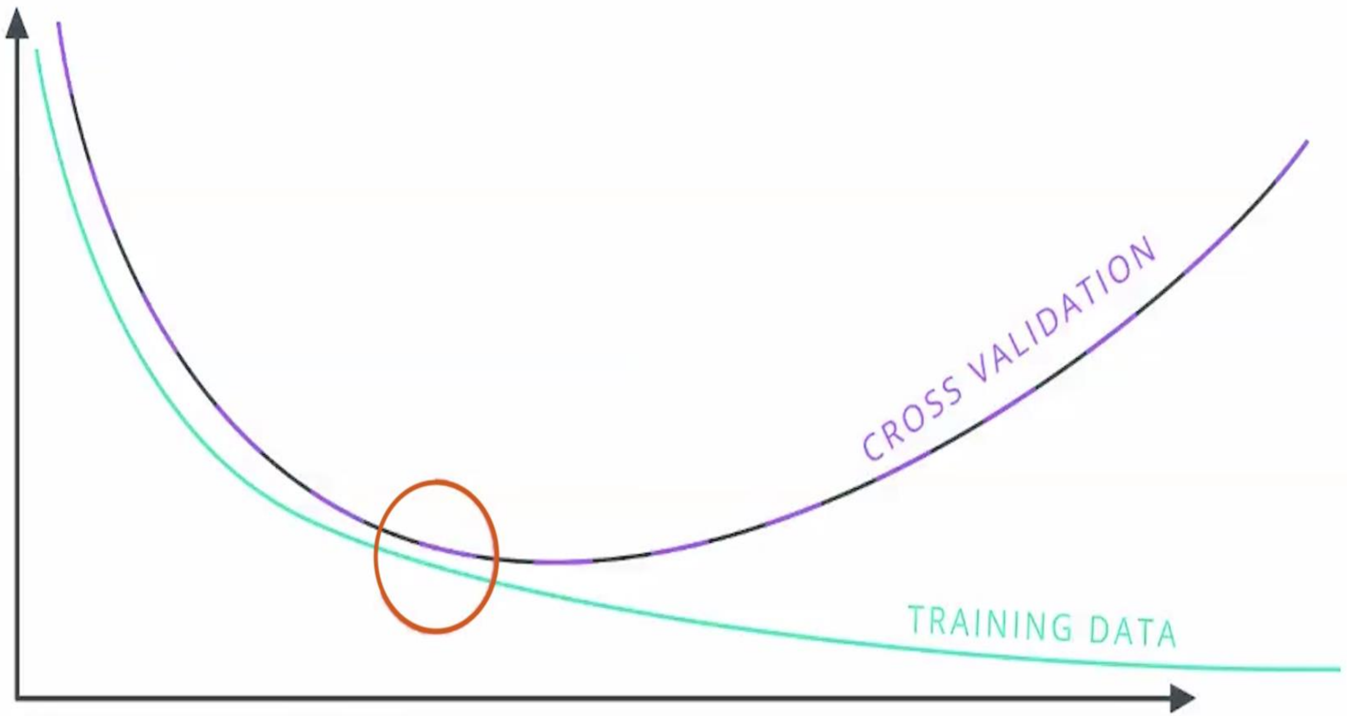


Model Complexity Graph

Training and Cross Validation Errors Vs Degree of polynomial: Under fitting to Over fitting problem

- **Degree of polynomial (hyper parameter)**
- **Compare Training and Validation errors.**
- *Example*: **House-price valuation dataset**





Learning Curves

Plot that show changes in Learning performance over the time i.e with the number of data points.

- **Under Fit Problem**

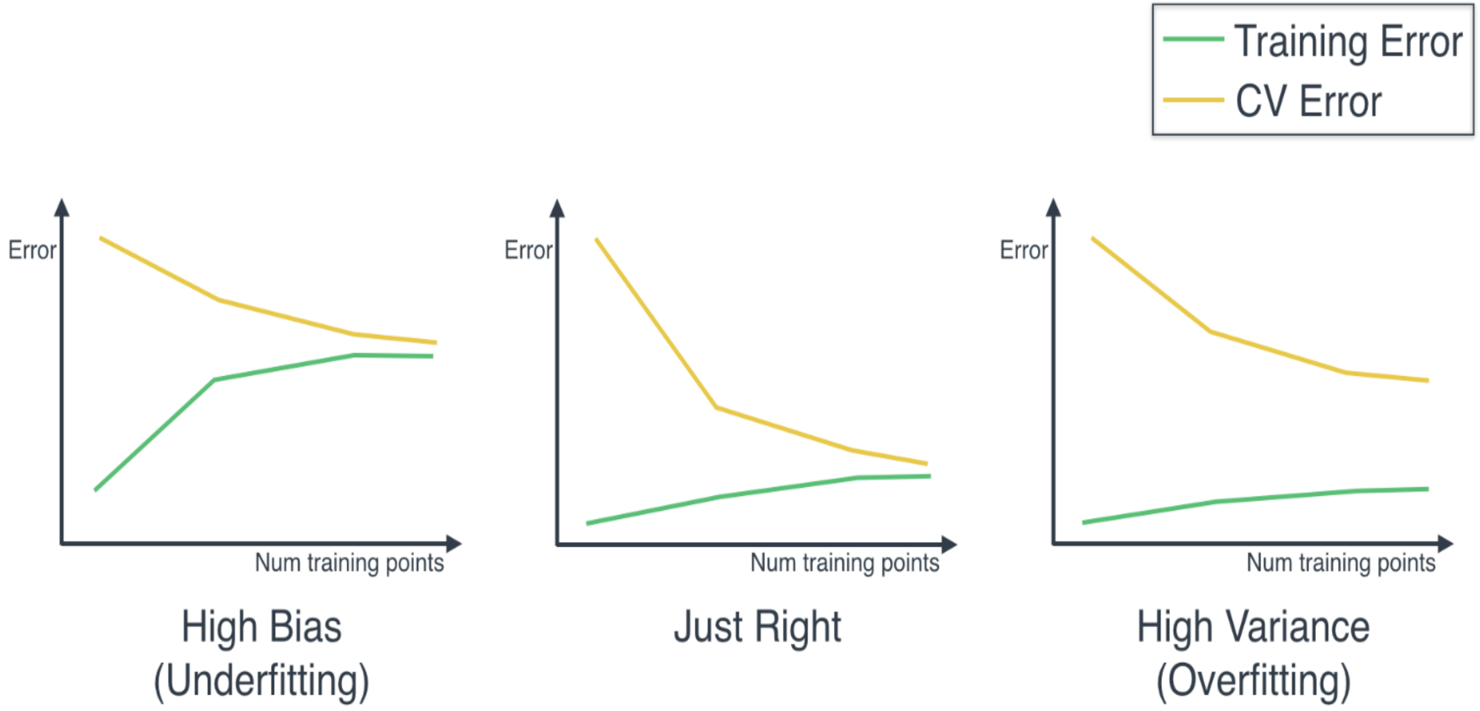
  High Bias: High errors on training and validation data sets.

- **Over Fit Problem**

  High Variance: High difference between the training and validation errors.

- **Solution:**

  Low Bias and Low Variance



High Bias (Underfitting)

Just Right

High Variance (Overfitting)

Ref: https://www.dataquest.io/blog/learning-curves-machine-learning/
https://www.ritchieng.com/machinelearning-learning-curve/

Summary

We have explored the metrics and methods to evaluate the models for Classification and Regression problems in this presentation with example applications applying some statistical methods as well. We have to further investigate about more tools for advanced Machine Learning algorithms like Deep Learning, NLP etc.

Q&A