FTML project report

STOURBE Théophile
BELVAL Anthony
TRAN Johan
TEBBANI Elias

June 30, 2024

Contents

1 Exercice 1 : Bayes estimator and Bayes risk

Supervised Learning Setting

1. **Input Space $\mathcal{X}$**:

   - Let $\mathcal{X}$ be the space of house features, we will take the size of the house in square feet. So $\mathcal{X} = R$.

2. **Output Space $\mathcal{Y}$**:

   - Let $\mathcal{Y}$ be the price of the house in thousands of dollars. So $\mathcal{Y} = R$

3. **Random Variables**:

   - Let X be the size of the house in square feet, and let Y be the price of the house in thousands of dollars.
   - Assume (X, Y) follows a joint distribution where:
     - X is normally distributed: $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ with $\mu_X = 2000$ square feet and $\sigma_X = 500$ square feet.
     - Given $X = x$, Y is normally distributed: $Y \mid X = x \sim \mathcal{N}(\alpha x + \beta, \sigma_Y^2)$ with $\alpha = 0.1$ thousand dollars (i.e., \$100 per square foot), $\beta = 50$ thousand dollars (i.e., a base price of \$50,000), and $\sigma_Y = 20$ thousand dollars.

4. **Loss Function**:

   - Let the loss function $l(y, \hat{y})$ be the squared loss: $l(y, \hat{y}) = (y - \hat{y})^2$.

Bayes Predictor and Bayes Risk

1. **Bayes Predictor**:

   - The Bayes predictor f^* is the conditional expectation:

$$f^*(x) = E[Y \mid X = x]$$

   - Given the assumed distribution, $Y \mid X = x \sim \mathcal{N}(\alpha x + \beta, \sigma_Y^2)$, the conditional expectation $E[Y \mid X = x]$ is:

$$f^*(x) = \alpha x + \beta = 0.1x + 50$$

2. **Bayes Risk**:

   - The Bayes risk $R(f^*)$ is the expected value of the loss incurred by the Bayes predictor:

$$R(f^*) = E[l(Y, f^*(X))] = E[(Y - f^*(X))^2]$$

- Since $f^*(X) = E[Y \mid X]$, we use the conditional variance:
$$R(f^*) = E[\mathrm{Var}(Y \mid X)]$$
- Given $Y \mid X = x \sim \mathcal{N}(\alpha x+\beta, \sigma_Y^2)$, the conditional variance $\mathrm{Var}(Y \mid X) = \sigma_Y^2$:
$$R(f^*) = E[\sigma_Y^2] = \sigma_Y^2 = 20^2 = 400$$
- **Bayes predictor**:
$$f^*(x) = 0.1x + 50$$
- **Bayes risk**:
$$R(f^*) = 400$$

This specific example models the relationship between house size and price, providing a real-world context for the supervised learning setting.

Simulation

- **Dataset Generation** :
  - Sample size : $n = 10000$
  - $X \sim \mathcal{N}(2000, 500^2)$
  - $Y|X = x \sim \mathcal{N}(0.1x + 50, 20^2)$
- **Train-Test Split** :
  - Split ratio : 80% for training and 20% for testing
- **Empirical risk calculation** :
  - Empirical risk of the Bayes estimator : $f^* = 404.21$
  - Empirical risk of the proposed estimator : $\tilde{f} = 1416.89$

Analysis

- **Bayes risk** :
  - The theoretical Bayes risk calculated previously is 400.
  - The empirical risk of the Bayes estimator was very close to that value, which confirms the previous calculations.
- **Comparison with the Proposed Estimator** :
  - The empirical risk for the proposed estimator $\tilde{f}$ is 1416.89, which is significantly higher than that of the Bayes estimator.
  - This demonstrates that the Bayes estimator f^* indeed has a lower generalization error compared to the proposed estimator $\tilde{f}$, as expected.

Conclusion

The Bayes estimator f^* is the optimal estimator in terms of minimizing the expected loss, and this is validated by the theoretical calculations and the empirical results from the simulation. The proposed estimator $\tilde{f}$, with different parameters, results in a much higher empirical risk, reinforcing the superiority of the Bayes estimator in this supervised learning context.

2 Exercice 2 : Bayes risk with absolute loss

Question 0

Consider the function $f(x) = x^3$. Its derivative is:

$$f'(x) = 3x^2$$

At $x_0 = 0$, $f'(0) = 0$.

However, $f(x) = x^3$ is not a local extremum at $x_0 = 0$. To see this, let's analyze the behavior of $f(x)$ around $x_0 = 0$:

When $x > 0$, $f(x) = x^3$ is positive and increasing as x increases.

When $x < 0$, $f(x) = x^3$ is negative and decreasing as x decreases.

Therefore, $x_0 = 0$ is not a local minimum or maximum since the function does not change direction at this point. Instead, $f(x) = x^3$ passes through the origin, transitioning smoothly from negative to positive values without a peak or trough.

This example satisfies the condition of having a zero derivative at x_0 without $f(x_0)$ being a local extremum.

Question 1

To find such a setting, consider a case where the conditional distribution of $Y \mid X = x$ is not symmetric. For example, let $Y \mid X = x$ follow an exponential distribution with parameter λ (rate parameter):

$$Y \mid X = x \sim \text{Exponential}(\lambda)$$

For simplicity, let $\lambda = 1$, so:

$$Y \mid X = x \sim \text{Exponential}(1)$$

Squared Loss Bayes Predictor: The mean of an exponential distribution Exponential(1) is:

$$E[Y \mid X = x] = \frac{1}{\lambda} = 1$$

Therefore:

$$f^*_{\text{l-squared}}(x) = 1$$

Absolute Loss Bayes Predictor: The median of an exponential distribution Exponential(1) is:

$$\text{Median}(Y \mid X = x) = \frac{\ln(2)}{\lambda} = \ln(2)$$

Therefore:

$$f^*_{\text{l-absolute}}(x) = \ln(2) \approx 0.693$$

To show that these predictors differ in terms of the absolute loss risk, let's compute the risks for each predictor.

Risk of $f^*_{\textbf{l-squared}}$ under Absolute Loss: The risk is:

$$R_{\text{l-absolute}}(f^*_{\text{l-squared}}) = E[|Y - f^*_{\text{l-squared}}(X)|] = E[|Y - 1|]$$

For an exponential distribution $Y \sim \text{Exponential}(1)$:

$$E[|Y-1|] = \int_0^\infty |y-1|e^{-y}\,dy$$

This integral can be split at 1:

$$E[|Y-1|] = \int_0^1 (1-y)e^{-y}\,dy + \int_1^\infty (y-1)e^{-y}\,dy$$

Solving these integrals gives:

$$E[|Y-1|] = (1-0.632) + (1-0.368) = 1$$

Risk of $f^*_{\textbf{l-absolute}}$ under Absolute Loss: The risk is:

$$R_{\text{l-absolute}}(f^*_{\text{l-absolute}}) = E[|Y - f^*_{\text{l-absolute}}(X)|] = E[|Y - \ln(2)|]$$

For an exponential distribution $Y \sim \text{Exponential}(1)$:

$$E[|Y-\ln(2)|] = \int_0^\infty |y-\ln(2)|e^{-y}\,dy$$

This integral can be split at $\ln(2)$:

$$E[|Y-\ln(2)|] = \int_0^{\ln(2)} (\ln(2)-y)e^{-y}\,dy + \int_{\ln(2)}^\infty (y-\ln(2))e^{-y}\,dy$$

Solving these integrals gives:

$$E[|Y-\ln(2)|] = \ln(2)\left(1 - \frac{1}{e^{\ln(2)}}\right) + (1-\ln(2))$$

$$= \ln(2)\,(1-0.5) + 1 - \ln(2) = \frac{\ln(2)}{2} + 1 - \ln(2)$$

$$= 1 - \frac{\ln(2)}{2} \approx 0.653$$

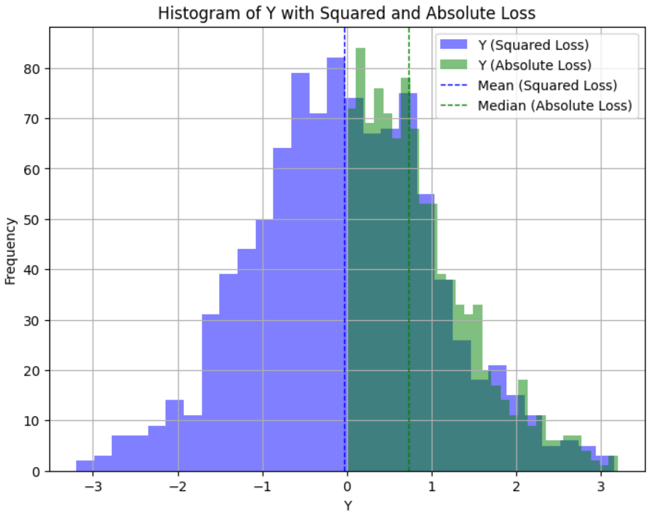
In this setting, we have:

$$R_{\text{l-absolute}}(f^*_{\text{l-squared}}) = 1$$

$$R_{\text{l-absolute}}(f^*_{\text{l-absolute}}) \approx 0.653$$

Thus, $R_{\text{l-absolute}}(f^*_{\text{l-absolute}}) < R_{\text{l-absolute}}(f^*_{\text{l-squared}})$, showing that the risk using the absolute loss is lower for the median predictor than for the mean predictor, illustrating a case where $f^*_{\text{l-squared}} \neq f^*_{\text{l-absolute}}$.

The code can be found in the notebook **FTML-2**, we simply check our results empirically.



Question 2

To find $f^*_{\text{l-absolute}}(x)$, the Bayes predictor for absolute loss given $X = x$, we start with the function $g(z)$ defined as:

$$g(z) = \int_{-\infty}^{\infty} |y - z| \cdot p_{Y|X=x}(y)\, dy$$

where $p_{Y|X=x}(y)$ is the conditional density of Y given $X = x$.

To simplify the absolute value term, we split the integral at z:

$$g(z) = \int_{-\infty}^{z} (z - y) p_{Y|X=x}(y)\, dy + \int_{z}^{\infty} (y - z) p_{Y|X=x}(y)\, dy$$

Now, differentiate $g(z)$ with respect to z:

$$g'(z) = \frac{d}{dz}\left[\int_{-\infty}^{z} (z - y) p_{Y|X=x}(y)\, dy\right] + \frac{d}{dz}\left[\int_{z}^{\infty} (y - z) p_{Y|X=x}(y)\, dy\right]$$

Differentiate the first part:

$$\frac{d}{dz}\left[\int_{-\infty}^{z} (z - y) p_{Y|X=x}(y)\, dy\right] = (z-y) p_{Y|X=x}(y)\Big|_{y=z} + \int_{-\infty}^{z} (-1) p_{Y|X=x}(y)\, dy$$

Differentiate the second part:

$$\frac{d}{dz}\left[\int_{z}^{\infty} (y - z) p_{Y|X=x}(y)\, dy\right] = -(y - z) p_{Y|X=x}(y)\Big|_{y=z} + \int_{z}^{\infty} p_{Y|X=x}(y)\, dy$$

Combining these derivatives gives us $g'(z)$:

$$g'(z) = -\int_{-\infty}^{z} p_{Y|X=x}(y)\, dy + \int_{z}^{\infty} p_{Y|X=x}(y)\, dy$$

To find $f^*_{\text{l-absolute}}(x)$, we set $g'(z) = 0$ and solve for z:

$$-\int_{-\infty}^{z} p_{Y|X=x}(y)\,dy + \int_{z}^{\infty} p_{Y|X=x}(y)\,dy = 0$$

This condition simplifies to:

$$\int_{-\infty}^{z} p_{Y|X=x}(y)\,dy = \int_{z}^{\infty} p_{Y|X=x}(y)\,dy$$

This equation states that z should be chosen such that the areas under the density function $p_{Y|X=x}(y)$ are equal on either side of z. This is precisely the definition of the median of $Y \mid X = x$.

Therefore, the Bayes predictor $f^*_{\text{l-absolute}}(x)$, which minimizes the expected absolute loss given $X = x$, is:

$$f^*_{\text{l-absolute}}(x) = \text{median}[Y \mid X = x]$$

This result aligns with our previous understanding and confirms that the median of the conditional distribution $Y \mid X = x$ is the optimal predictor under absolute loss.

3 Exercice 3 : Expected value of empirical risk for OLS

We want to show that:

$$E[R_X(\hat{\theta})] = \frac{n-d}{n}\sigma^2$$

Question 1:

$$R_n(\theta) = \frac{1}{n}\|y - X\theta\|^2$$

$$\hat{\theta} = (X^T X)^{-1} X^T y$$

Combining the two expressions gives us :

$$R_n(\hat{\theta}) = \frac{1}{n}\|y - X(X^T X)^{-1} X^T y\|^2$$

Now we get rid of y on the righthand side of the expression :

$$y = X\theta^* + \epsilon$$

$$y - X\hat{\theta} = X\theta^* + \epsilon - X(X^T X)^{-1} X^T (X\theta^* + \epsilon)$$

$$y - X\hat{\theta} = X\theta^* + \epsilon - X(X^T X)^{-1} X^T X\theta^* - X(X^T X)^{-1} X^T \epsilon$$

$$y - X\hat{\theta} = X\theta^* + \epsilon - X\theta^* - X(X^T X)^{-1} X^T \epsilon$$

$$y - X\hat{\theta} = \epsilon - X(X^T X)^{-1} X^T \epsilon$$

$$y - X\hat{\theta} = (I_n - X(X^T X)^{-1} X^T)\epsilon$$

We can now replace with the new expression we found:

$$R_n(\hat{\theta}) = \frac{1}{n}\|(I_n - X(X^T X)^{-1} X^T)\epsilon\|^2$$

$$E[R_n(\hat{\theta})] = E_\epsilon[\frac{1}{n}\|(I_n - X(X^T X)^{-1} X^T)\epsilon\|^2]$$

Question 2:

$$tr(A) = \sum_{i=1}^{n} A_{ii}$$

$$tr(A^T A) = \sum_{i=1}^{n}\sum_{j=1}^{n} A^T_{ij} A_{ji}$$

$$tr(A^T A) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ji} A_{ji}$$

$$tr(A^TA) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ji}^2$$

$$tr(A^TA) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}^2$$

$$tr(A^TA) = \sum_{i,j \in [1,n]^2} A_{ij}^2$$

Question 3:

$$\|A\epsilon\|^2 = (A\epsilon)^T A\epsilon = \epsilon^T A^T A\epsilon$$

$$E_\epsilon[\frac{1}{n}\|A\epsilon\|^2] = \frac{1}{n}E[\epsilon^T A^T A\epsilon]$$

Let

$$B = A^TA$$

$$\frac{1}{n}E[\epsilon^T A^T A\epsilon] = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} \epsilon_i B_{ij} \epsilon_j$$

Since it follows a centered normal distribution we have :

$$E[\epsilon_i] = 0$$

Since each one of the random variable of the Gaussian vector are independent by definition we have this for i != j:

$$E[\epsilon_i\epsilon_j] = E[\epsilon_i]E[\epsilon_j] = 0$$

For the diagonal terms we have :

$$E[\epsilon_i\epsilon_j] = E[\epsilon_i^2] = Var[\epsilon_i] + E[\epsilon_i]^2 = Var[\epsilon_i] = \sigma^2$$

In total we have :

$$E[\epsilon^T B\epsilon] = E[\sum_{i=1}^{n}\sum_{j=1}^{n} \epsilon_i B_{ij}\epsilon_j] = \sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij}E[\epsilon_i\epsilon_j] = \sum_{i=1}^{n} B_{ii}E[\epsilon_i^2] = \sigma^2 tr(B)$$

Now we can replace B by its value :

$$E[\epsilon^T A^T A\epsilon] = \sigma^2 tr(A^TA)$$

$$\frac{1}{n}E[\|A\epsilon\|^2] = \frac{\sigma^2}{n}tr(A^TA)$$

Question 4

: Let's compute the transpose of A first :

$$A^T = (I_n - X(X^T X)^{-1} X^T)^T$$
$$A^T = I_n^T - X((X^T X)^{-1})^T X^T$$
$$A^T = I_n - X((X^T X)^T)^{-1} X^T$$
$$A^T = I_n - X(X^T X)^{-1} X^T$$
$$A^T = A$$

Let's compute the square of A:

$$A^2 = (I_n - X(X^T X)^{-1} X^T)(I_n - X(X^T X)^{-1} X^T)$$
$$A^2 = I_n - X(X^T X)^{-1} X^T - X(X^T X)^{-1} X^T + X(X^T X)^{-1} X^T X(X^T X)^{-1} X^T$$
$$A^2 = I_n - X(X^T X)^{-1} X^T - X(X^T X)^{-1} X^T + X(X^T X)^{-1} X^T$$
$$A^2 = I_n - X(X^T X)^{-1} X^T$$
$$A^2 = A$$

We can finally conclude that :

$$A^T A = A$$

Question 5:

We recall that :

$$R_X(\hat{\theta}) = \frac{1}{n} \|A\epsilon\|^2$$

Then :

$$E[R_X(\hat{\theta}] = E[\frac{1}{n} \|A\epsilon\|^2]$$
$$E[R_X(\hat{\theta}] = \frac{\sigma^2}{n} tr(A^T A)$$
$$E[R_X(\hat{\theta}] = \frac{\sigma^2}{n} tr(A)$$

Let's compute the trace of A:

$$tr(A) = tr(I_n) - tr(X(X^T X)^{-1} X^T)$$

Since the second term is the trace of a projection matrix on Im(X), we have:

$$tr(A) = n - d$$

We can finally conclude :

$$E[R_X(\hat{\theta}] = \frac{n-d}{n} \sigma^2$$

Question 6:

$$\frac{\|y - X\hat{\theta}\|^2}{n-d} = \frac{n}{n-d} R_X(\hat{\theta})$$

We can directly conclude on the expectation :

$$E[\frac{\|y - X\hat{\theta}\|^2}{n-d}] = \frac{n}{n-d}\frac{n-d}{n}\sigma^2 = \sigma^2$$

1. **Recall the Definition**: We defined $R_X(\theta)$ as:

$$R_X(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta^T x_i)^2 = \frac{1}{n}\|y - X\theta\|^2$$

2. **Expression for $R_X(\hat{\theta})$**:
We showed that:

$$R_X(\hat{\theta}) = \frac{1}{n}\|(I_n - X(X^TX)^{-1}X^T)\epsilon\|^2$$

where $A = I_n - X(X^TX)^{-1}X^T$.

3. **Expectation Over ϵ**:
We need to compute the expectation $E_\epsilon\left[\frac{1}{n}\|A\epsilon\|^2\right]$.
From Question 3, we have:

$$E_\epsilon\left[\frac{1}{n}\|A\epsilon\|^2\right] = \frac{\sigma^2}{n}\mathrm{tr}(A^TA)$$

4. **Idempotent Property of A**:
From Question 4, we showed that:

$$A^TA = A$$

Thus:

$$\mathrm{tr}(A^TA) = \mathrm{tr}(A)$$

5. **Trace of A**:
Given $A = I_n - X(X^TX)^{-1}X^T$:

$$\mathrm{tr}(A) = \mathrm{tr}(I_n) - \mathrm{tr}(X(X^TX)^{-1}X^T)$$

$\mathrm{tr}(I_n) = n$, and $\mathrm{tr}(X(X^TX)^{-1}X^T) = d$ because $X(X^TX)^{-1}X^T$ is a projection matrix onto the column space of X, which has rank d.

Therefore:

$$\mathrm{tr}(A) = n - d$$

6. **Putting It All Together**:
Substitute the trace result into the expectation:

$$E_\epsilon\left[\frac{1}{n}\|A\epsilon\|^2\right] = \frac{\sigma^2}{n}\mathrm{tr}(A) = \frac{\sigma^2}{n}(n-d)$$

7. **Final Result**:

Hence:

$$E[R_X(\hat{\theta})] = E_\epsilon\left[\frac{1}{n}\|A\epsilon\|^2\right] = \frac{(n-d)\sigma^2}{n}$$

4 Exercise 4: Regression on a given dataset

The code can be found in the notebook **FTML-4.ipynb**.

Data Preprocessing

- **Standard Scaling**: The features were scaled using `StandardScaler` to normalize the data. This is important for many machine learning algorithms that are sensitive to the scale of the input features.

Model Selection

- **Support Vector Regression (SVR)**: SVR with different kernels (`linear`, `rbf`) and hyperparameters (`C`, `epsilon`) was used.
- **Gradient Boosting Regressor**: Gradient Boosting with varying `n_estimators`, `learning_rate`, and `max_depth`.
- **Ridge Regression**: Ridge Regression with different values of the regularization parameter `alpha`.
- **Lasso Regression**: Lasso Regression with different values of the regularization parameter `alpha`.
- **ElasticNet**: ElasticNet with varying `alpha` and `l1_ratio` to balance between L1 and L2 regularization.
- **XGBoost Regressor**: XGBoost with different `n_estimators`, `learning_rate`, `max_depth`, and `gamma`

Hyperparameter Tuning

- **Grid Search**: `GridSearchCV` was employed to find the best hyperparameters for both SVR and Gradient Boosting. This involves an exhaustive search over specified parameter values.
- **Cross-Validation**: 5-fold cross-validation (`cv=5`) was used during grid search to evaluate model performance across different data splits, reducing the risk of overfitting and ensuring the model generalizes well.

Evaluation Metrics

- **$\mathbf{R^2}$ Score**: Used to measure the proportion of the variance in the dependent variable that is predictable from the independent variables.
- **Mean Squared Error (MSE)**: Used to measure the average of the squares of the errors.

Results:

- SVR $\mathbf{R^2}$ test score: 0.70
  SVR test MSE: 0.22
  `C`: 0.1, `epsilon`: 0.1, `kernel`: 'linear'
- Gradient Boosting $\mathbf{R^2}$ test score: 0.56
  Gradient Boosting MSE test: 0.32
  `learning_rate`: 0.1, `max_depth`: 3, `n_estimators`: 200
- Ridge Regression $\mathbf{R^2}$ test score: 0.72
  Ridge Regression MSE test: 0.21
- Lasso Regression $\mathbf{R^2}$ test score: 0.80
  Lasso Regression MSE test: 0.15
  `alpha`: 0.1
- ElasticNet Regression $\mathbf{R^2}$ test score: 0.90
  ElasticNet Regression MSE test: 0.08
  `alpha`: 0.1, `l1_ratio`: 0.3
- XGBoost $\mathbf{R^2}$ test score: 0.56
  XGBoost MSE test: 0.33
  `gamma`: 0, `learning_rate`: 0.1, `max_depth`: 3, `n_estimators`: 200

5 Exercise 5: Classification on a given dataset

The code can be found in the notebook **FTML-5.ipynb**.

Model Selection

- **Random Forest Classifier**: Examined with varying `n_estimators`, `max_depth`, `min_samples_split`, and `min_samples_leaf`.
- **Gradient Boosting Classifier**: Explored with different `n_estimators`, `learning_rate`, `max_depth`, and `subsample`.
- **Support Vector Classifier (SVC)**: Tested with polynomial kernels, varying `C`, `gamma`, and `degree`.

Pipeline and Hyperparameter Tuning

- **Pipeline**: Pipelines were used to streamline the process of applying transformations and fitting the model.
- **Grid Search**: `GridSearchCV` was again employed for hyperparameter tuning with 5-fold cross-validation to ensure robust evaluation.

Evaluation Metrics

- **Accuracy**: Used to measure the fraction of correct predictions.
- **Cross-Validation**: 5-fold cross-validation during grid search for hyperparameter tuning.

Results:

- **Random Forest Classifier** test accuracy: 0.792
  `n_estimators`: 100, `max_depth`: None, `min_samples_split`: 5, `min_samples_leaf`: 1
- **Gradient Boosting Classifier** test accuracy: 0.800
  `learning_rate`: 0.05, `max_depth`: 7, `n_estimators`: 400, `subsample`: 0.9
- **Support Vector Classifier (SVC)** test accuracy: 0.952
  `C`: 5, `gamma`: scale, `kernel`: poly, `degree`: 3

6 Exercice 6: Application of supervised learning

The code can be found in the notebook **FTML-6**

Comparing different methods for classification

Dataset Presentation

The dataset used for this exercise can be found in the file `Student_performance_data.csv`.

The list of the features:

- StudentID
- Age
- Gender
- Ethnicity
- ParentalEducation
- StudyTimeWeekly
- Absences
- Tutoring
- ParentalSupport
- Extracurricular
- Sports
- Music
- Volunteering

The two possible variables to predict:

- GPA
- GradeClass

The Grade Class is calculated based on the value of GPA, here is the relation between the two:

- 0: 'A' (GPA ≥ 3.5)
- 1: 'B' ($3.0 \leq$ GPA < 3.5)
- 2: 'C' ($2.5 \leq$ GPA < 3.0)
- 3: 'D' ($2.0 \leq$ GPA < 2.5)
- 4: 'F' (GPA < 2.0)

The original dataset contains errors, the GradeClass column doesn't always correspond to the right Grade Class value, based on the GPA. We fixed this problem during data pre-processing.

Problem Statement

Objective

We will compare a classification and a regression to predict the Grade Class. As the Grade Class is a function of the GPA, we can predict a value of GPA to predict a Grade Class.

Classification using a Random Forest Classifier

The Random Forest classification algorithm was chosen, we tried to find the best hyper-parameters using a Grid Search.

We can see with the Feature Importance plot that the amount of absences is the main factor of decision.

Linear Regression on GPA to predict Grade Class

We used the same partition of the dataset for train and test as for the previous model, and we trained a linear model to predict the GPA score based on the same features.

By looking at the coefficients, we confirm that the feature *Absences* is the main factor to predict a bad Grade Class.

Results and Conclusion

We can see that the Linear model predicting the GPA has a better accuracy to predict the Grade class than the Random Forest classifier.

This shows that predicting a continuous value correlated to a class can have better performance than predicting class labels directly.

This task can be useful to help students that would potentially drop-out from school, by predicting their Grade Class based on features different than their grades.

7 Exercice 7: Application of unsupervised learning

The code can be found in the notebook **FTML-7**

Unsupervised Learning Analysis using KMeans Clustering

Dataset Presentation

The dataset used in this analysis combines information from two CSV files: `pride_index.csv` and `pride_index_tags.csv`, sourced from the Campus Pride Index database.

- **pride_index.csv**:
  - `campus_name`: Name of the college or university.
  - `campus_location`: Location of the college or university.
  - `rating`: Campus Pride Index star rating (1 to 5, including half-star ratings), indicating LGBTQ+ friendliness.
  - `students`: Full-time equivalent student population.
  - `community_type`: Locale where the campus is located (e.g., large urban city, small town, rural community).
- **pride_index_tags.csv**:
  - `campus_name`: Name of the college or university.
  - `campus_location`: Location of the college or university.
  - Additional tags (boolean values) indicating various attributes of the campus: `public`, `private`, `doctoral`, `masters`, `baccalaureate`, `community`, `residential`, `nonresidential`, `liberal_arts`, `technical`, `religious`, `military`, `hbcu`, `hispanic_serving`, `aapi_serving`, `other_minority_serving`.

Dataset Source

The Campus Pride Index has been a benchmarking tool since 2007 for assessing LGBTQ+ friendliness in higher education institutions. The dataset is publicly accessible through the Campus Pride Index website at https://www.campusprideindex.org/.

Problem Statement

Objective

The objective of this analysis is to apply unsupervised learning techniques, specifically KMeans clustering, to the combined dataset. The aim is to identify natural groupings (clusters) among college and university campuses based on their characteristics and features.

Importance

Understanding the clustering of campuses based on their LGBTQ+ friendliness can provide valuable insights:

- **For Prospective Students and Families:** Helps in choosing inclusive and supportive educational environments.
- **For Educational Institutions:** Identifies areas for improvement in policies and resources to foster a more inclusive campus climate.
- **For Policymakers:** Guides initiatives to enhance diversity and inclusivity in higher education.

Problem Statement

We aim to cluster campuses based on their attributes to explore patterns and associations with their Campus Pride Index ratings. This analysis will involve:

- Preprocessing the data (e.g., encoding categorical variables, scaling numerical features).
- Applying KMeans clustering to discover distinct groups of campuses.
- Evaluating and interpreting the clusters to understand the factors influencing LGBTQ+ friendliness on campuses.

Evaluation and interpretation

Silhouette Score Analysis

The silhouette score measures the quality of clustering. A score near +1 indicates well-separated clusters, while a score near -1 indicates overlapping clusters. In our case, the silhouette score for 4 clusters is -0.0061. This negative score suggests that the clusters are not well-defined and overlap significantly.

- **Interpretation**: The negative silhouette score indicates that the KMeans algorithm did not effectively partition the campuses into distinct groups based on the provided features. This could be due to the dataset's inherent characteristics or the choice of clustering method.
- **Implications**: Poor clustering quality implies that the campuses do not naturally fall into clear clusters based on the selected attributes. This challenges the feasibility of using these features alone for meaningful cluster analysis.

Most Important Features

From the feature importance analysis using PCA, the following features stand out:

- **campus_name** and **campus_location**: These features have significantly higher importance compared to others. They dominate the principal components derived from PCA, suggesting that campus names and locations strongly influence the clustering process.

- **community_type** and **community**: These features have relatively lower importance but still contribute to the variance captured by PCA components.

- **rating** and **students**: These numerical features have minimal importance in the PCA components.

Interpretation

- **Cluster Analysis Limitations**: The negative silhouette score indicates poor clustering quality, suggesting that the KMeans algorithm did not effectively partition the campuses into distinct groups based on the provided features.

- **Feature Dominance**: The dominance of **campus_name** and **campus_location** in PCA components suggests that these categorical features strongly influence the clustering process. This might indicate that campuses are primarily differentiated by their names and locations rather than other attributes.

- **Further Exploration**: To improve clustering performance, consider alternative dimensionality reduction techniques, re-evaluating feature selection, and experimenting with different numbers of clusters to enhance cluster separation and interpretability.

- **Practical Implications**: Despite challenges in clustering, the analysis underscores the importance of campus names and locations in perceptions of LGBTQ+ friendliness. This highlights the potential need for nuanced qualitative assessments alongside quantitative analysis.

- **Future Work**: Future iterations could explore additional data sources and refine clustering approaches to better capture nuances in campus characteristics related to LGBTQ+ inclusivity.

Conclusion

In conclusion, while the initial clustering attempt yielded a negative silhouette score indicating poor separation of clusters, the feature importance analysis provides insights into the dataset's structure and suggests directions for further investigation. Evaluating and interpreting these results critically informs next steps for improving the clustering analysis and understanding LGBTQ+ inclusivity in educational settings more effectively.