Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

Machine Learning Project

Gabriel Martins

Emanuele Iaccarino

Vít Šesták

Professor: Aitor Almeida

University of Deusto, 2023

Data Cleaning and Analysis

Prediction of health risks based on various features of the individuals, such as demographic details, health measurements, and lifestyle choices.

Goal: classification model predicting two health-related outcomes: smoking and/or drinking status

Data Cleaning and Analysis

01 Loaded the dataset and performed initial data exploration

02 Utilized pandas, numpy, seaborn, and matplotlib libs for data manipulation and visualization

03 Handling duplicates and Null values

	sex	age	height	weight	waistline	sight_left	sight_right	hear_left	hear_right	SBP
0	Male	35	170	75	90.0	1.0	1.0	1.0	1.0	120.0
1	Male	30	180	80	89.0	0.9	1.2	1.0	1.0	130.0
2	Male	40	165	75	91.0	1.2	1.5	1.0	1.0	120.0
3	Male	50	175	80	91.0	1.5	1.2	1.0	1.0	145.0
4	Male	50	165	60	80.0	1.0	1.2	1.0	1.0	138.0

LDL_chole	triglyceride	hemoglobin	urine_protein	serum_creatinine	SGOT_AST	SGOT_ALT	gamma_GTP	SMK_stat_type_cd	DRK_YN
126.0	92.0	17.1	1.0	1.0	21.0	35.0	40.0	1.0	Y
148.0	121.0	15.8	1.0	0.9	20.0	36.0	27.0	3.0	N
74.0	104.0	15.8	1.0	0.9	47.0	32.0	68.0	1.0	N
104.0	106.0	17.6	1.0	1.1	29.0	34.0	18.0	1.0	N
117.0	104.0	13.8	1.0	0.8	19.0	12.0	25.0	1.0	N

Statistical Analysis and Visualization

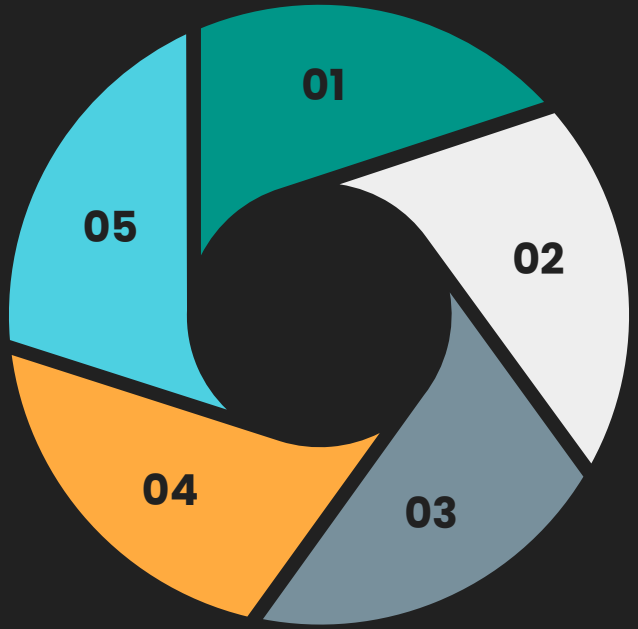
1 Performed statistical analysis and visualization to gain insights

2 Created box plots for numerical variables like vision measurements, blood pressure, cholesterol, enzyme levels, etc

3 Explored relationships using scatter plots and count plots

4 Utilized count plots for categorical variables such as smoking status and sex

5 Generated histograms for various features such as age, height, waistline, blood sugar, serum creatinine, etc

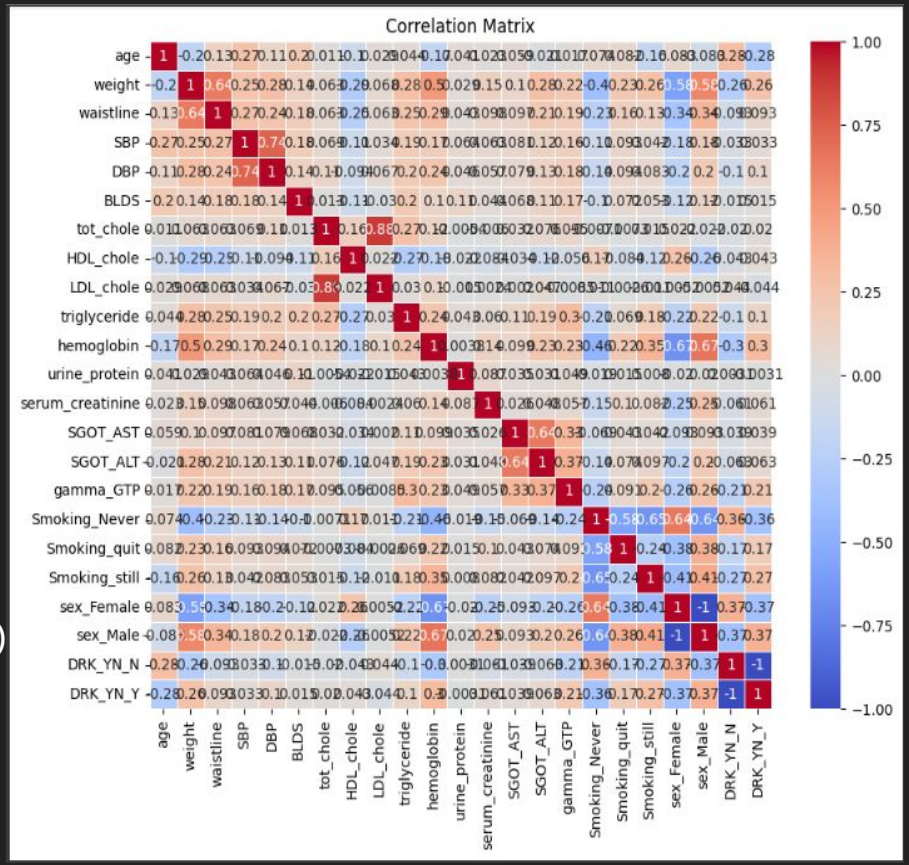


Correlation Matrix

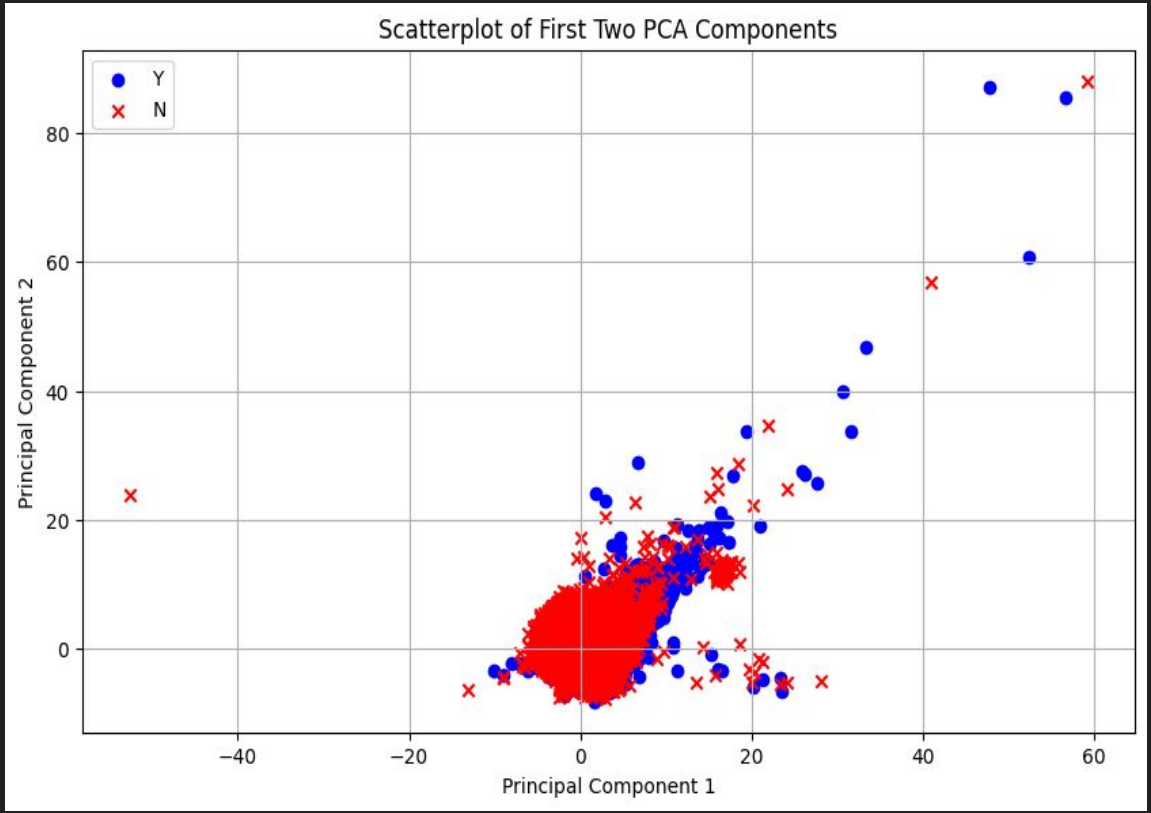
01 Created a correlation matrix to identify feature relationships

02 Some relations are worth a further analysis:

- Smoking_Never & Sex_Female (positive)
  - Smoking_Still & Age (Negative)
- Drinking_Yes & Smoking_Never (Negative)
  - Smoking_Never & Hemoglobin (Negative)



PCA to identify target distribution and outliers



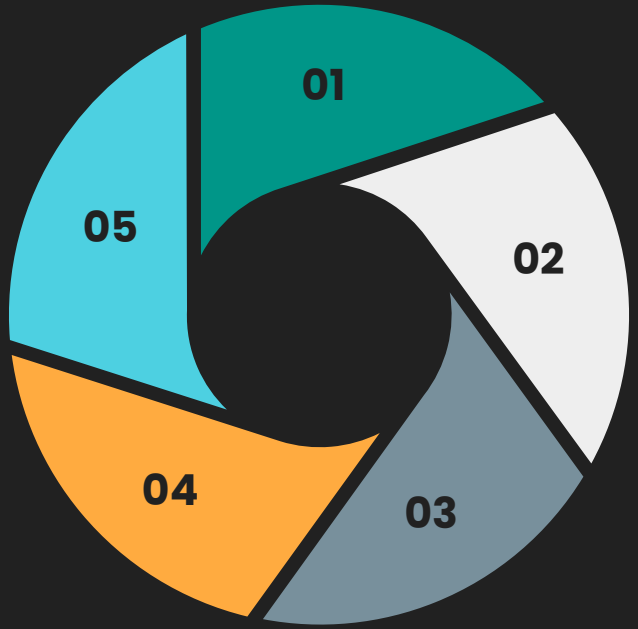
The partition of our target variable is not completely clear so we can't suppose our classification model to perform the task with a high accuracy.

Not enough outliers data to proceed with an outlier detection. Outliers detection techniques may brings a loss of information and decrease the performance of the model.

Model Examined

5 Stochastic Gradient Descent

1 Decision Trees



2 Random Forest Classifier

4 Support Vector Machines:

3 Gaussian Naive Bayes and Binomial Naive Bayes

Introducing Optuna

Framework to automate hyperparameter search, particularly designed for Machine Learning and Reinforcement Learning.

Many benefits (outperforms GridSearchCV):

- Fast
  - Searches for optimal settings simultaneously using different threads or processes
- Efficient
  - Efficiently navigates through hyperparameter configurations, quickly identifying and discarding less promising options
- TPE Sampler
  - we are gonna use Bayesian hyperparameter optimization algorithms that are able to learn about the relationship between the hyperparameters and the objective function

- Better results :
  - With Random Forest
    - Smoking : 0.6993392848136379
    - Drinking : 0.7296716598577697

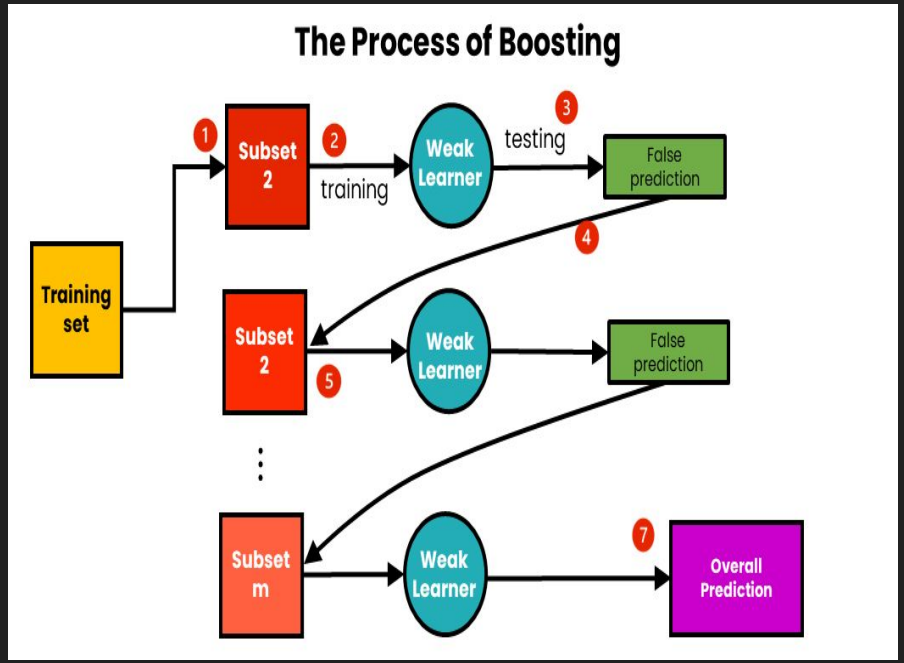
- Next: improve results with Boosting algorithms

Boosting Algorithms

- Boosting algorithms are effective because they combine multiple weak learners (often shallow decision trees) to form a strong and accurate model.

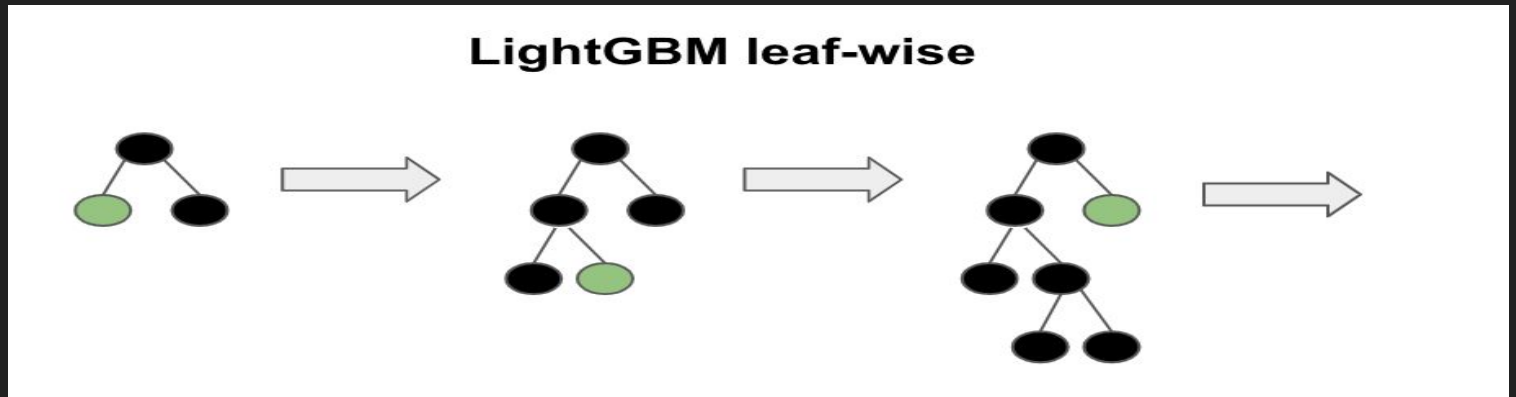
- We used:
  - LightGbm
  - AdaBoost
  - XGBoost

- With those algorithms our results got better.



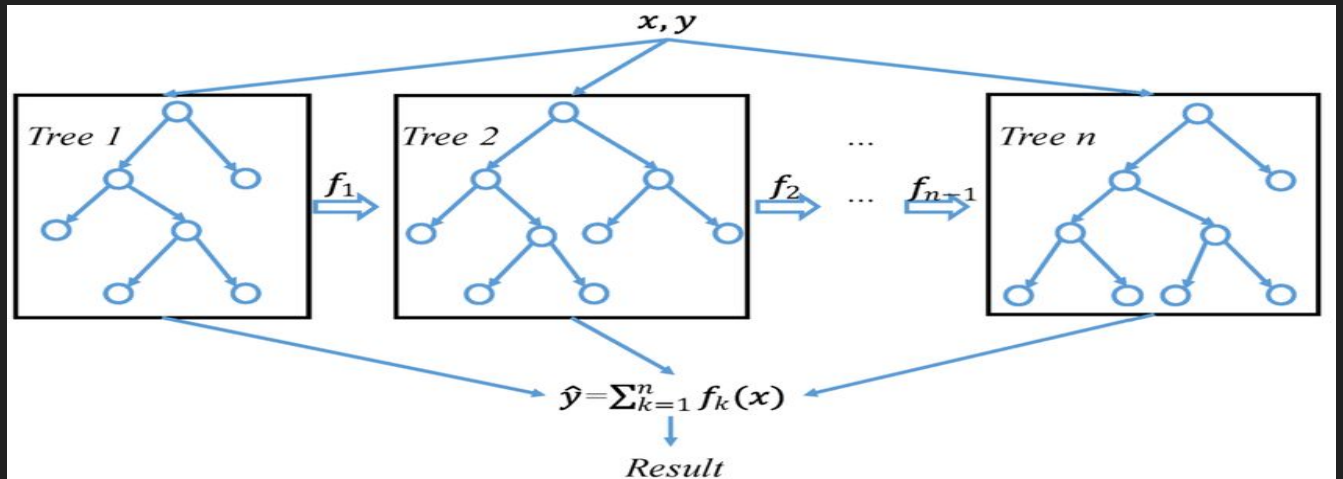
LightGBM: Light Gradient Boosting Machine

- **Gradient Boosting Algorithm**
  - strong predictive model is built by combining the predictions of multiple weak models, often decision trees
- **Lightweight and Fast**
  - lightweight, making it particularly suitable for large datasets and high-dimensional data
- **Leaf-Wise Tree Growth**
  - grows the tree vertically, extending one branch (or leaf) at a time. Selects the leaf that provides the maximum reduction in the loss function (Log loss in our case) and expands it. This allow to capture intricate patterns and relationships in the data, potentially leading to a more expressive and powerful model.



XGBoost: eXtreme Gradient Boosting

- **Gradient Boosting Algorithm**
- **Regularization Techniques**
  - incorporates regularization techniques, such as L1 (Lasso) and L2 (Ridge) regularization, to prevent overfitting and enhance the model's generalization capabilities.
- **Tree Pruning**
  - employs a process of tree pruning during the building phase, removing branches that do not contribute significantly to reducing the loss function. This results in more efficient and less complex trees.



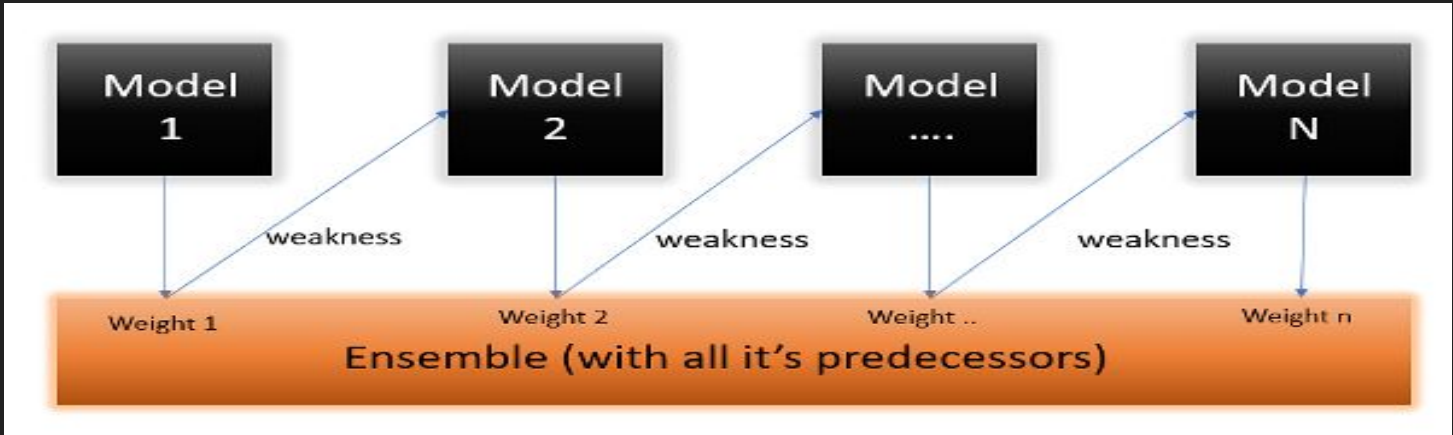
Adaboost: Adaptive Boosting

- **Boosting**
  - combines the predictions of multiple weak learners sequentially to create a robust ensemble model. The focus is on correcting the errors made by previous weak learners in the ensemble.

- **Weighted Instances**
  - During training, each instance in the dataset is assigned a weight. Misclassified instances from the previous weak learners are given higher weights, so subsequent weak learners focus more on correcting these mistakes.
- **Sequential Learning**
  - Weak learners are trained sequentially, with each new weak learner emphasizing the mistakes of the ensemble up to that point.The final prediction is a weighted sum of the predictions from all weak learners.
- **Combining Weak Classifiers**
  - Commonly, decision trees with a depth of 1 (stumps) are used as weak learners in AdaBoost, although other algorithms can also be employed. Stumps focus on a single decision based on one feature, making them weak but effective in combination.

Workflow

- **Initialize Weights:** Assign equal weights to all instances in the training dataset.
- **Train Weak Learner:** Train a weak learner on the weighted dataset.
- **Compute Error:** Calculate the error of the weak learner on the weighted dataset.
- **Compute Learner Weight:** Calculate the weight of the weak learner in the final ensemble based on its error.
- **Update Weights:** Increase the weights of misclassified instances, making them more important for subsequent training.
- **Repeat:** Repeat the process with a new weak learner, updating weights at each iteration.
- **Final Prediction:** Combine the predictions of all weak learners, with each learner's contribution weighted by its performance.



Conclusion

- Even though the normal algorithms were good, with boosting and optuna we could get better results.

  The best results achieved:

  - **LGBM :**
    - Smoking : 70,263 %
    - Drinking : 73,633 %
  - **XGBoost:**
    - Smoking : 70,240 %
    - Drinking : 73,595 %
  - **AdaBoost:**
    - Smoking : 70,066 %
    - Drinking : 73,519 %
  - **RandomForest:**
    - Smoking : 69,934 %
    - Drinking : 72,967 %

```
#Smoking Y/N
random_state = 42

#[I 2023-11-12 07:18:32,836] Trial 99 finished with value: -0.7026288181414679

lgb_params = {
    "n_estimators": 266,
    "max_depth": 12,
    "learning_rate":0.04885993133945042,
    "reg_alpha":0.9053259867272333,
    "reg_lambda": 0.5646689446221783,
    "subsample": 0.8957582593900227,
    "colsample_bytree": 0.8299845443203071,
    "min_child_samples": 37

}
```

```
#Drink Y/N
random_state = 42

#[I 2023-11-12 01:03:17,431] Trial 99 finished with value: -0.7363263123915588


lgb_params = {
    "n_estimators": 174,
    "max_depth": 9,
    "learning_rate":0.12863006286750658,
    "num_leaves": 50,
    "reg_alpha":0.3088088189585255,
    "reg_lambda": 0.2053013407939033,
    "subsample": 0.8989862633330897,
    "colsample_bytree": 0.6756762046103896,
    "min_child_samples": 42

}
```

Table of results

Classifier	Smoking Accuracy (%)	Drinking Accuracy (%)
RandomForestClassifier	69,171	72,748
GaussianNB	66,311	68,545
SVM		
SGDClassifier (w GridSearch)	66,883	70,378
RandomForest (w Optuna)	69,934	72,967
LGBMClassifier (w Optuna)	**70,263**	**73,633**
XGBClassifier (w Optuna)	70,240	73,595
AdaBoostClassifier (w Optuna)	70,066	73,519

Craigslist Used Cars Dataset Regression

Prediction of car prices based on 26 features, including Price, Year, Manufacturer, Model, Odometer, and more, from Craigslist, the world's largest collection of used vehicles for sale.

Because this is real data, there is a lot of data preparation and cleaning to deal with.

Data Cleaning and Analysis

00 Loaded the dataset and performed initial data exploration

Dropped irrelevant columns.

01 Removed rows with missing values.
Eliminated duplicates based on key columns.
Converted relevant columns to numeric types.
Removed outliers in 'price' and 'odometer.'

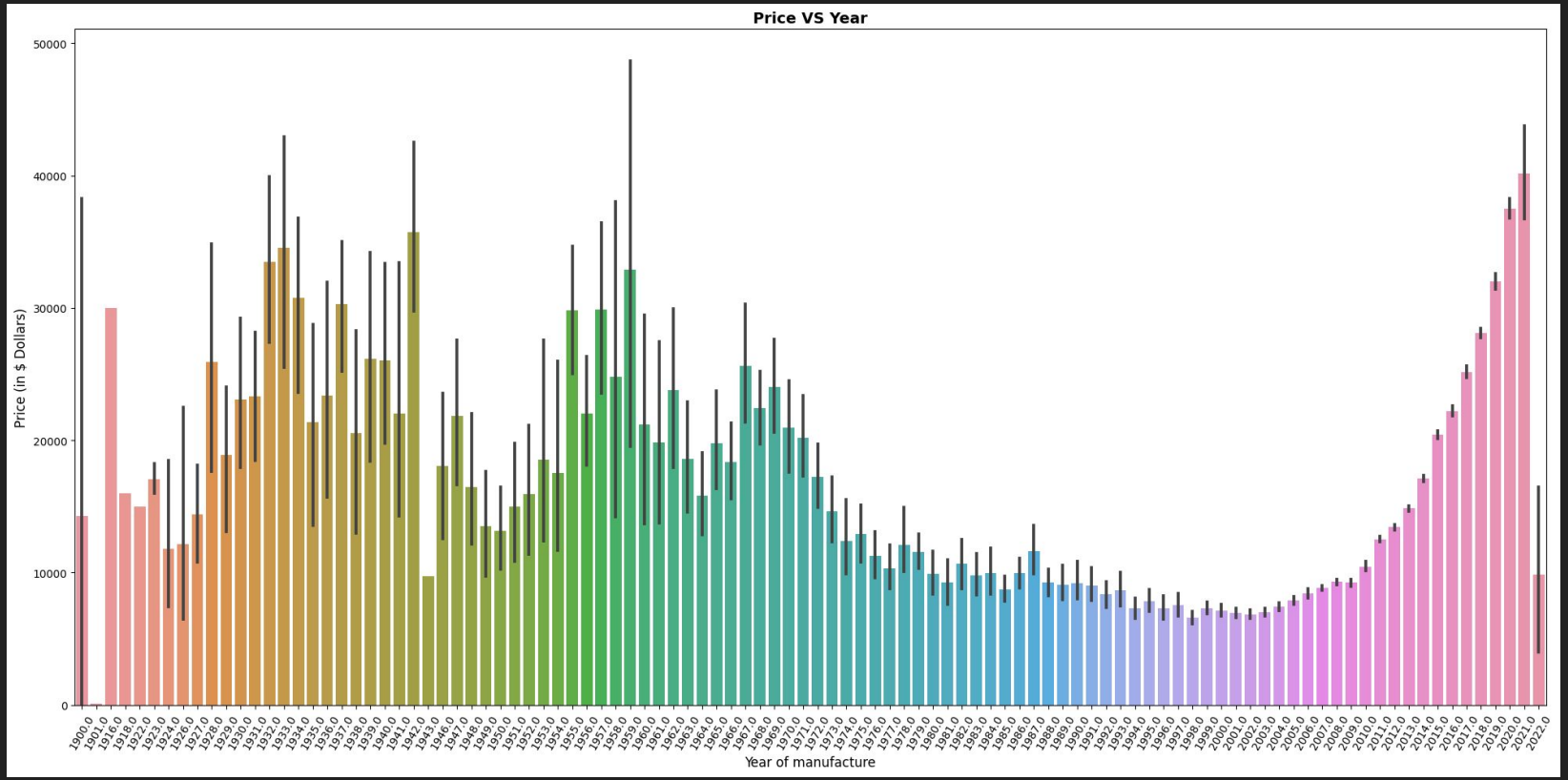
02 Feature Scaling: Explored scaling techniques (Robust Scaler).

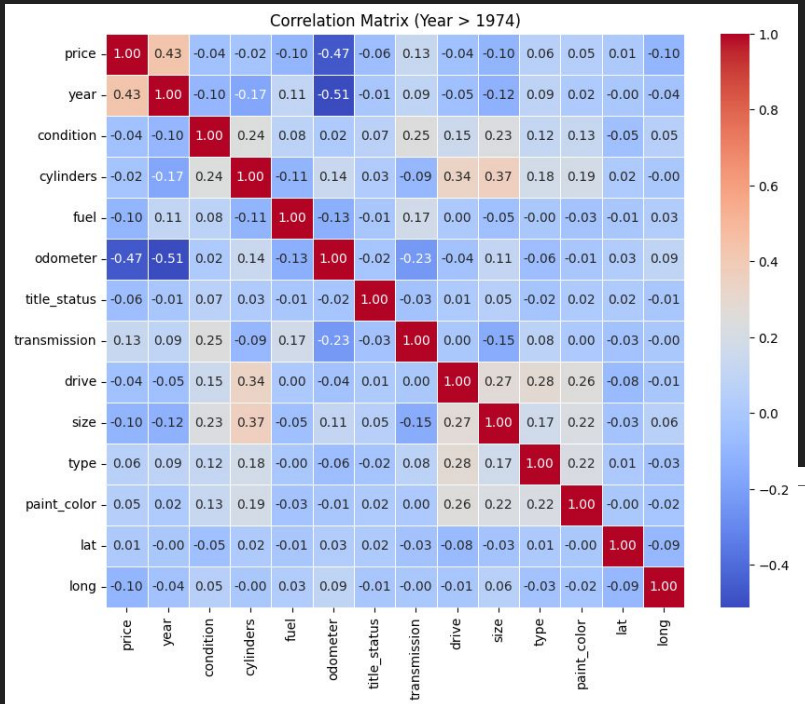
03 Log Transformation: Applied log transformation for skewed variables.

	region	price	year	manufacturer	model	condition	cylinders	fuel	odometer	title_status	transmission	drive	size	type	paint_color	state
27	auburn	33590	2014.0	gmc	sierra 1500 crew cab slt	good	8 cylinders	gas	57923.0	clean	other			pickup	white	al
28	auburn	22590	2010.0	chevrolet	silverado 1500	good	8 cylinders	gas	71229.0	clean	other			pickup	blue	al
29	auburn	39590	2020.0	chevrolet	silverado 1500 crew	good	8 cylinders	gas	19160.0	clean	other			pickup	red	al
30	auburn	30990	2017.0	toyota	tundra double cab sr	good	8 cylinders	gas	41124.0	clean	other			pickup	red	al
31	auburn	15000	2013.0	ford	f-150 xlt	excellent	6 cylinders	gas	128000.0	clean	automatic	rwd	full-size	truck	black	al
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
202699	grand rapids	14900	2012.0	chevrolet	camaro	excellent	6 cylinders	gas	94000.0	clean	automatic	rwd		convertible	black	mi
202700	grand rapids	10500	2013.0	bmw	x1 awd 4dr xdrive28i	excellent	4 cylinders	gas	107763.0	clean	automatic	4wd		hatchback	orange	mi
202705	grand rapids	16000	2013.0	toyota	venza limited awd	excellent	6 cylinders	gas	91534.0	clean	automatic	fwd	mid-size	SUV	black	mi
202706	grand rapids	2000	1984.0	ford	f-150	good		gas	100000.0	clean	manual					mi
202707	grand rapids	15155	2011.0	dodge	challenger			gas	129601.0	clean	automatic	rwd	compact	coupe	black	mi

Important decision

We divided the dataset into vintage (year < 1974) and nowadays cars (year >= 1974)





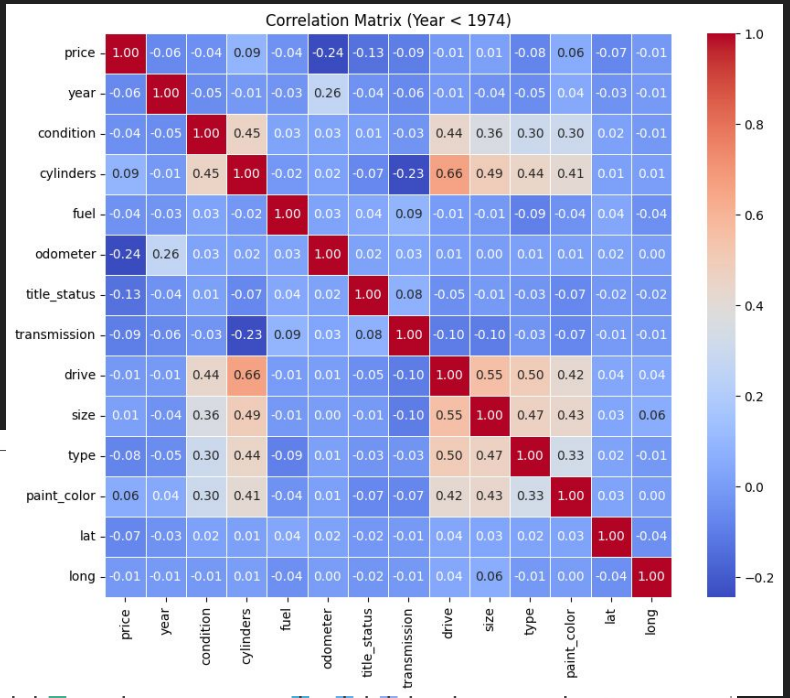
Nowadays Cars (>1974)

We can see how the price of the Car is mainly related to the year and the odometer variable.

Vintage Cars (<1974)

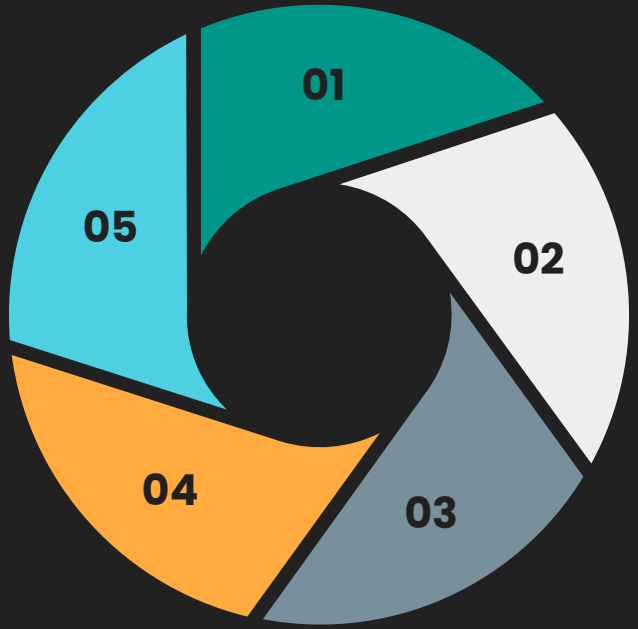
In this case, different car characteristics matter.

Decision was made to eliminate this portion from the dataset to avoid confusing our model during the training process.



Model Examined

5 Stochastic Gradient Descent



1 Linear Regression

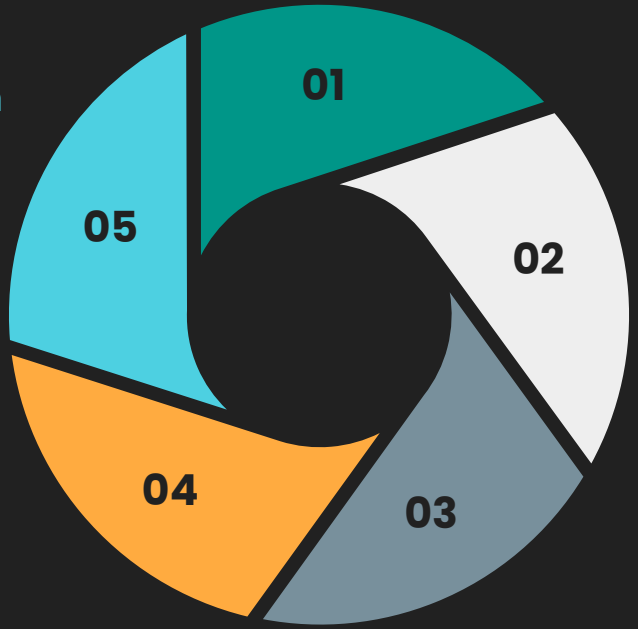
2 Random Forest Regressor

3 Polynomial Regression

4 Support Vector Machines:

Utilizes different kernels (e.g., RBF, polynomial, linear).

Model Examined with Optuna



1 Random Forest Regression

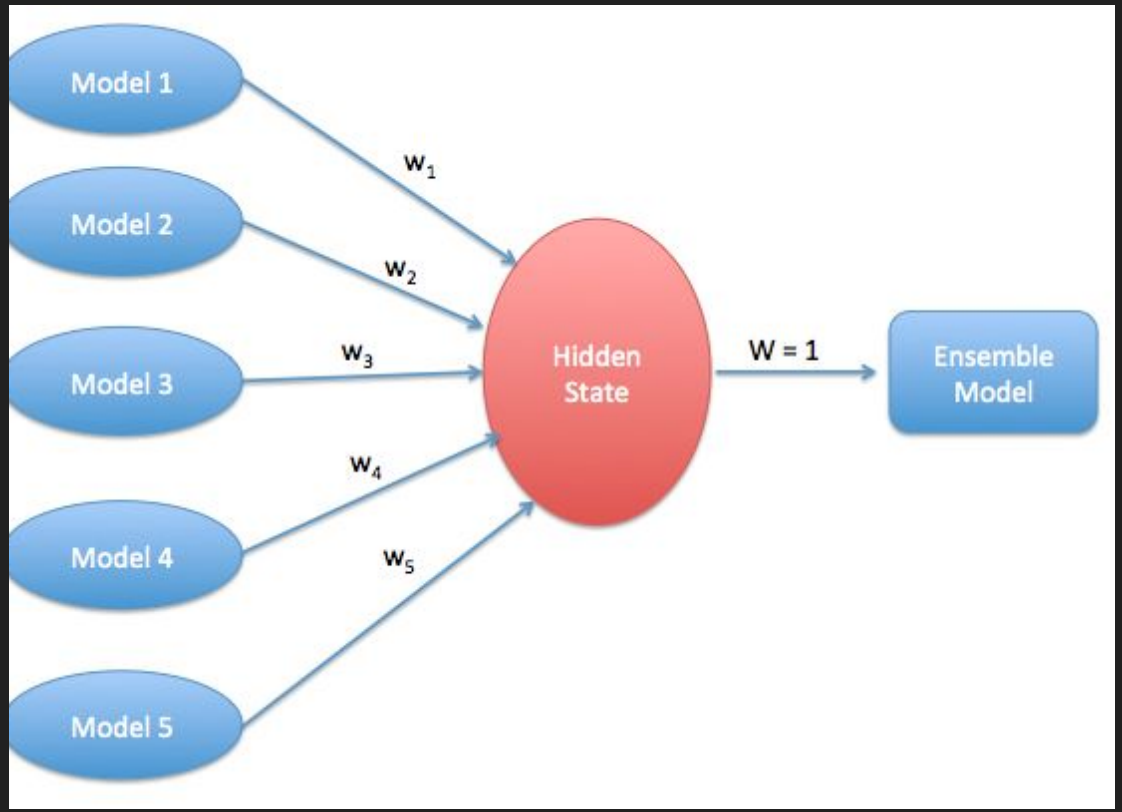
2 XGBoost Regression

3 LightGBM Regression

4 AdaBoost Regression

5 Ensemble Weight Optimization

Ensembling Weight Optimization



Determines the optimal contribution of individual models within an ensemble to improve overall predictive performance. Through iterative adjustments (using Optuna) we assign the appropriate influence to each model. The goal is to achieve a balanced ensemble that leverages the strengths of its components while mitigating potential weaknesses, leading to improved overall predictive power.

```
Weights:
{'weight xgb': 0.91415321595424,
'weight rf': 0.04076301949321206,
'weight_lgbm':0.045083764}
```

Table of results

Classifier	MSE	R^2	MAE	RMSE
Linear Regression	2.043478e+08	0.276194	7630.385498	14295.028211
RandomForestRegressor	1.356227e+08	0.519620	4396.569466	11645.718053
Polynomial Regression	1.698260e+08	0.398472	5920.404338	13031.729350
SVR (RBF Kernel)	2.735569e+08	0.031054	9098.544764	16539.555373
SVR (Poly Kernel)	2.749914e+08	0.025972	9159.062673	16582.865941
SVR (Linear Kernel)	2.140665e+08	0.241770	7351.838316	14631.010697
SGDRegressor	2.046871e+08	0.274992	7589.830868	14306.891497
XGBoost	1.191112e+08	0.578105	3509.487886	10913.808988
LightGBM	1.144392e+08	0.594653	3693.570106	10697.623599
Ensembling Method		0.58172115888		

Plotting our Results