Project Documentation

CMP461
Spring 2022

Team #7

Name	Sec	BN
Saad Eldeen Mohamed Mohamed Ahmed Badr	1	26
Omar Ibrahim Elsayed Salama	2	7
Mohamed Adel Abdelmohsen	2	21
Ali Adel Esmail	2	5

Contribution

Name	Contribution
Saad	Analysis visualizations and insights and Random forest classifier and evaluation
Omar	Random forest classifier and hyper parameters tuning and evaluation and data preprocessing
Mohamed	Adaboost classifier and hyper parameters tuning and evaluation and data preprocessing
Ali	SVM and hyper parameters tuning and evaluation and data preprocessing

Brief problem description

Problem definition:

We are looking at cold call results using a dataset from one bank in the US. Besides usual services, this bank also provides car insurance services. The bank organizes regular campaigns to attract new clients. The bank has potential customers' data, and the bank's employees call them to advertise available car insurance options. We are provided with general information about clients (age, job, etc.) as well as more specific information about the current insurance sell campaign (communication, last contact day) and previous campaigns (attributes like previous attempts, outcome).

Motivation:

The goal is to build a model which predicts for customers whether they will buy car insurance or not.

We worked with Car Insurance Cold Calls | Kaggle. In the project we are providing:

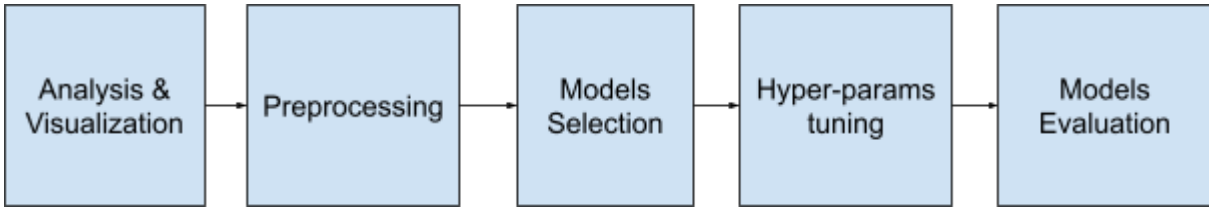
- Analysis and visualization of the features.
- Evaluation of Different Models.
- Choosing models and tuning parameters.

Evaluation Metrics:

We used the following metrics:

- Precision
- Recall
- F1-score
- Accuracy

Project pipeline



Analysis and solution of the problem

Data visualization

Please refer to the notebook attached or view all visualizations at the end of this file.

These are the columns of the dataset

Feature	Description	Example
Id	Unique ID number. Predictions file should contain this feature.	“1” … “5000”
Age	Age of the client	
Job	Job of the client.	"admin.", "blue-collar", etc.
Marital	Marital status of the client	"divorced", "married", "single"
Education	Education level of the client	"primary", "secondary", etc.
Default	Has credit in default?	"yes" - 1,"no" - 0
Balance	Average yearly balance, in USD	
HHInsurance	Is household insured	"yes" - 1,"no" - 0
CarLoan	Has the client a car loan	"yes" - 1,"no" - 0
Communication	Contact communication type	"cellular", "telephone", “NA”
LastContactMonth	Month of the last contact	"jan", "feb", etc.
LastContactDay	Day of the last contact	
CallStart	Start time of the last call (HH:MM:SS)	12:43:15
CallEnd	End time of the last call (HH:MM:SS)	12:43:15
NoOfContacts	Number of contacts performed during this campaign for this client	
DaysPassed	Number of days that passed by after the client was last contacted from a previous campaign (numeric; -1 means client was not previously contacted)	
PrevAttempts	Number of contacts performed before this campaign and for this client	
Outcome	Outcome of the previous marketing campaign	"failure", "other", "success", “NA”
CarInsurance	Has the client subscribed a CarInsurance?	"yes" - 1,"no" - 0

Extracting insights from data.

Please refer to the notebook attached.

Some insights

- Older people are more likely to buy car insurance.
- People with higher Balance are more likely to buy car insurance
- People with home insurance are less likely to buy car insurance
- People with higher NoOfContacts are less likely to buy car insurance
- People with higher DaysPassed are more likely to buy car insurance
- People with higher PrevAttempts are more likely to buy car insurance

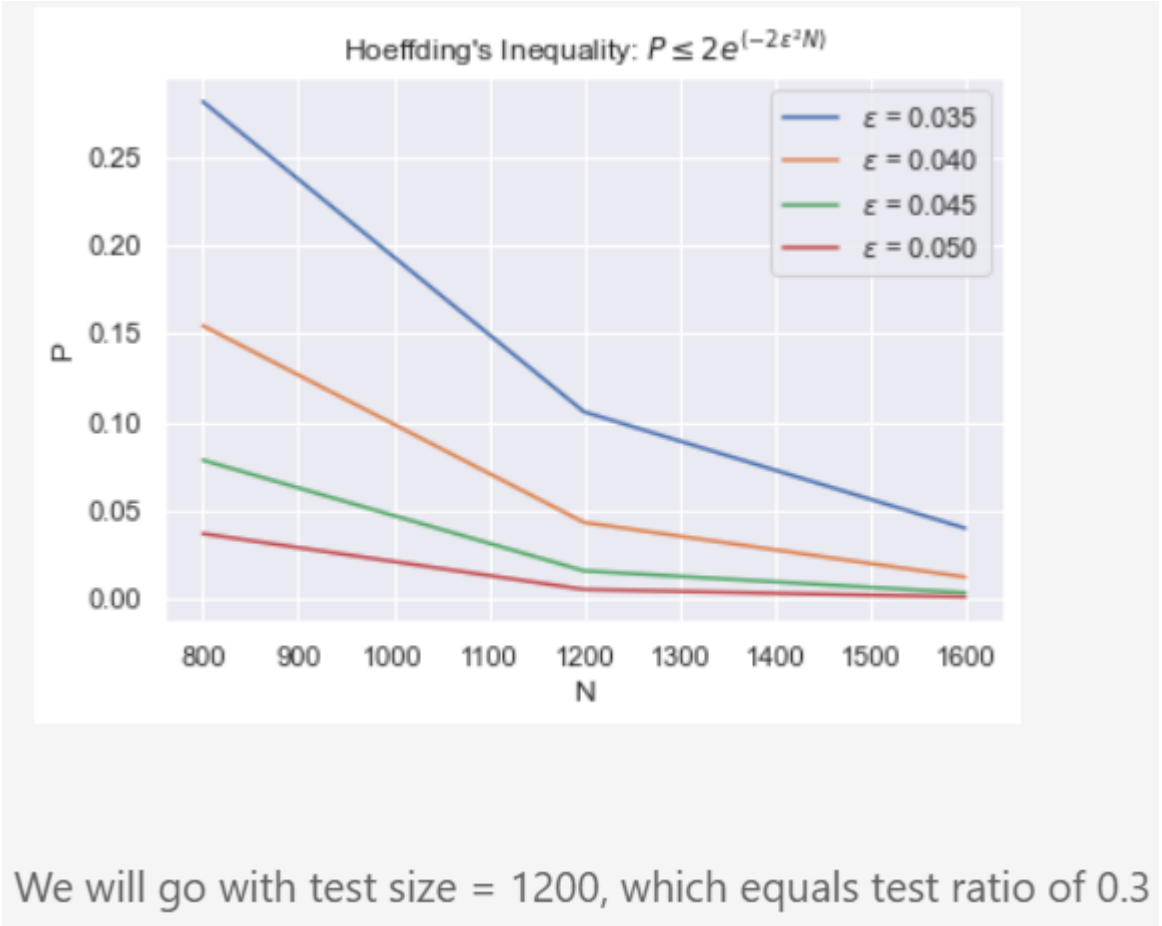
Data preprocessing

Preprocessing responsibilities are:

- Drop **Id** (not needed) and **Outcome** (75% of it is missing)
- Fill in missing data:
    - Remaining missing data are all from categorical features (Job, Communication, Education)
        - Fill them using sklearn simple imputer with strategy='most_frequent' (i.e. fill with the mode of the column values)
- Encode some categorical features as numerical levels
    - **LastContactMonth:** jan: 0, feb: 1, …, dec: 11
    - **Marital:** divorced: 0, single: 1, married: 2
    - **Communication:** telephone: 0, cellular: 1
    - **Education:** primary: 0, secondary: 1, tertiary: 2
    - **Jobs:** retired: 0, management: 1,...
        - **Note**: we tried both; ordered encoding and one hot vector encoding and this gave better results on classification
        - Ordering the jobs is done by ordering them descendingly according to the variance of the balance for each job.
- Convert **CallStart** & **CallEnd** to datetime values and compute a new feature from them which is **CallDuration** and dropping **CallEnd.**

Model/Classifier training

- We splitted the data into train and test sets with ratio (0.7/0.3) which was recommended after applying Hoeffding's Inequality



- Where P represents how close the test set can represent the out world:
  - P = 1 - P[|Eout - Etest| > ε] = P[|Eout - Etest| <= ε]

- After that we used cross validation (folds=5) on the training set to compare between different classifiers.

	Default Random Forest	Best Params Random Forest	Default Adaboost	Best Params Adaboost	Default SVM	Best Params SVM
0	0.825000	0.835714	0.816071	0.819643	0.737500	0.735714
1	0.828571	0.832143	0.828571	0.823214	0.717857	0.735714
2	0.817857	0.841071	0.810714	0.817857	0.733929	0.758929
3	0.839286	0.850000	0.798214	0.844643	0.744643	0.746429
4	0.833929	0.844643	0.825000	0.844643	0.760714	0.750000

- Then we selected **Random Forest, AdaBoost, SVM** classifiers to continue working with them further.
- For each selected classifier, we made a grid search with cross validation to tune their hyper parameters, which enhanced our three classifiers.

Hyperparameters tuning

For each classifier of the three chosen classifiers we made a grid search in a selected parameter grid:

- Random Forest Classifier
  - Tuned Hyperparameters
    - N estimators
      - Number of trees in the forest
    - Max features
      - Number of features to consider when looking for the best split (sqrt, log, auto)
    - Bootstrap
      - Use bootstrap samples are used when building trees or not
    - Criterion
      - Function to measure the quality of a split (gini, entropy, log_loss)
    - **Result**
      - n_estimators=700, max_features=auto, Bootstrap=false, criterion=entropy
- Adaboost
  - Hyperparameters:
    - N estimators:
      - The maximum number of estimators at which boosting is terminated.
    - Learning rate
      - Weight applied to each classifier at each boosting iteration
    - Algorithm:
      - The algorithm used 'SAMME OR 'SAMME.R'
    - Base estimator:
      - The base estimator from which the ensemble is grown.
    - **Result:**
      - n_estimators: 100, learning_rate: 0.5, algorithm: 'SAMME', base_estimator: RandomForestClassifier

- SVM
  - Hyperparameters:
    - C
      - Regularization parameter
    - Gamma
      - Kernel coefficient
    - Kernel
      - Kernel type to be used in the algorithm
    - **Result:**
      - C=1, gamma=0.0001, kernel=rbf

Model Evaluation

I. **Train set**

A. Random Forest

	0	1	accuracy	macro avg	weighted avg
precision	1.0	1.0	1.0	1.0	1.0
recall	1.0	1.0	1.0	1.0	1.0
f1-score	1.0	1.0	1.0	1.0	1.0
support	1670.0	1130.0	1.0	2800.0	2800.0

B. Adaboost

	0	1	accuracy	macro avg	weighted avg
precision	1.0	1.0	1.0	1.0	1.0
recall	1.0	1.0	1.0	1.0	1.0
f1-score	1.0	1.0	1.0	1.0	1.0
support	1670.0	1130.0	1.0	2800.0	2800.0

C. SVM

	0	1	accuracy	macro avg	weighted avg
precision	0.862557	0.859496	0.861429	0.861026	0.861321
recall	0.913174	0.784956	0.861429	0.849065	0.861429
f1-score	0.887144	0.820537	0.861429	0.853840	0.860263
support	1670.000000	1130.000000	0.861429	2800.000000	2800.000000

II. **Test set**

A. Random Forest

	0	1	accuracy	macro avg	weighted avg
precision	0.870345	0.800000	0.8425	0.835172	0.842559
recall	0.869146	0.801688	0.8425	0.835417	0.842500
f1-score	0.869745	0.800843	0.8425	0.835294	0.842529
support	726.000000	474.000000	0.8425	1200.000000	1200.000000

B. Adaboost

	0	1	accuracy	macro avg	weighted avg
precision	0.868785	0.796218	0.84	0.832502	0.840121
recall	0.866391	0.799578	0.84	0.832985	0.840000
f1-score	0.867586	0.797895	0.84	0.832740	0.840058
support	726.000000	474.000000	0.84	1200.000000	1200.000000

C. SVM

	0	1	accuracy	macro avg	weighted avg
precision	0.801664	0.691023	0.7575	0.746344	0.757961
recall	0.796143	0.698312	0.7575	0.747228	0.757500
f1-score	0.798894	0.694648	0.7575	0.746771	0.757717
support	726.000000	474.000000	0.7575	1200.000000	1200.000000

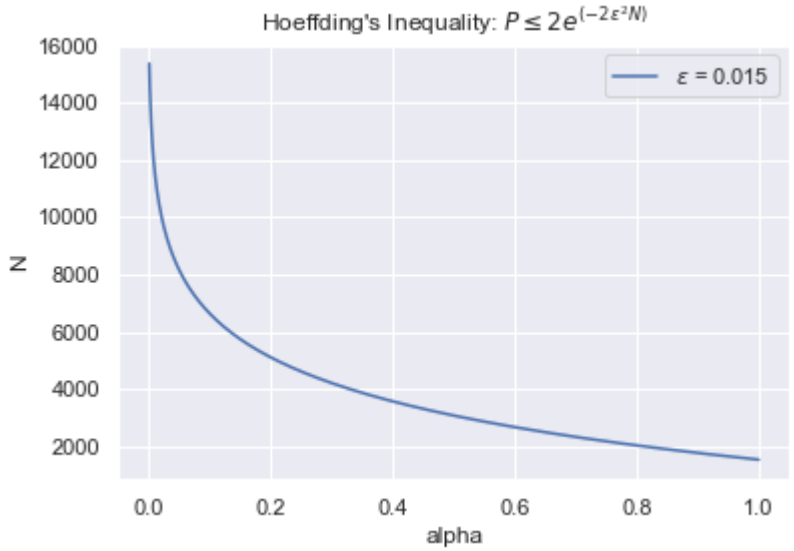
- We made a baseline ZeroR dummy classifier which got accuracy = 0.605

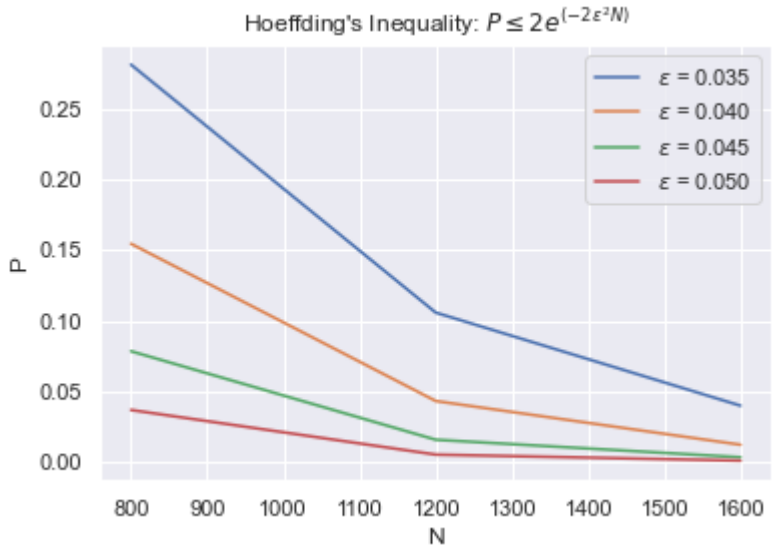
Conclusion:

- We analyzed the data and found some important insights *(as mentioned in its section)*
- Tried several classifiers
- Selected the best three classifiers and tuned their hyper parameters
- And our work resulted in three robust classifiers which are obviously not dummy or random.

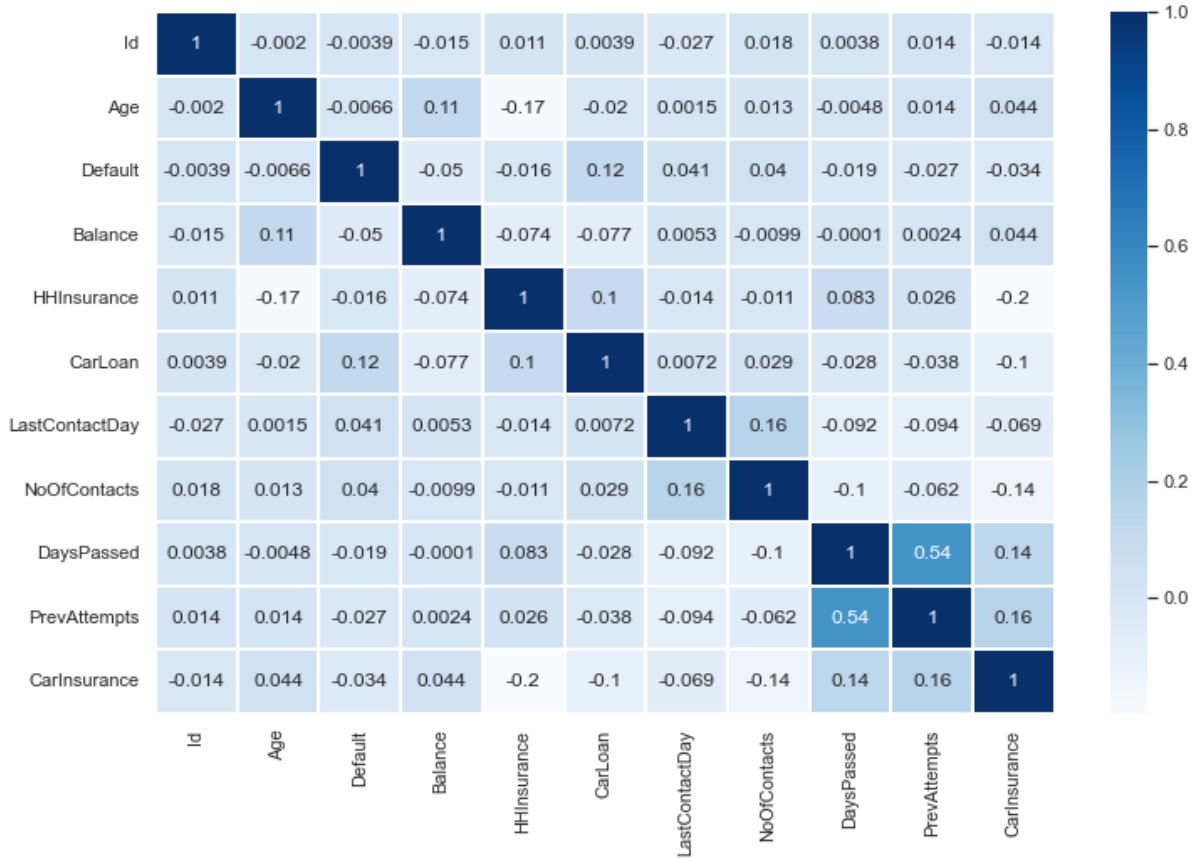
Visualizations

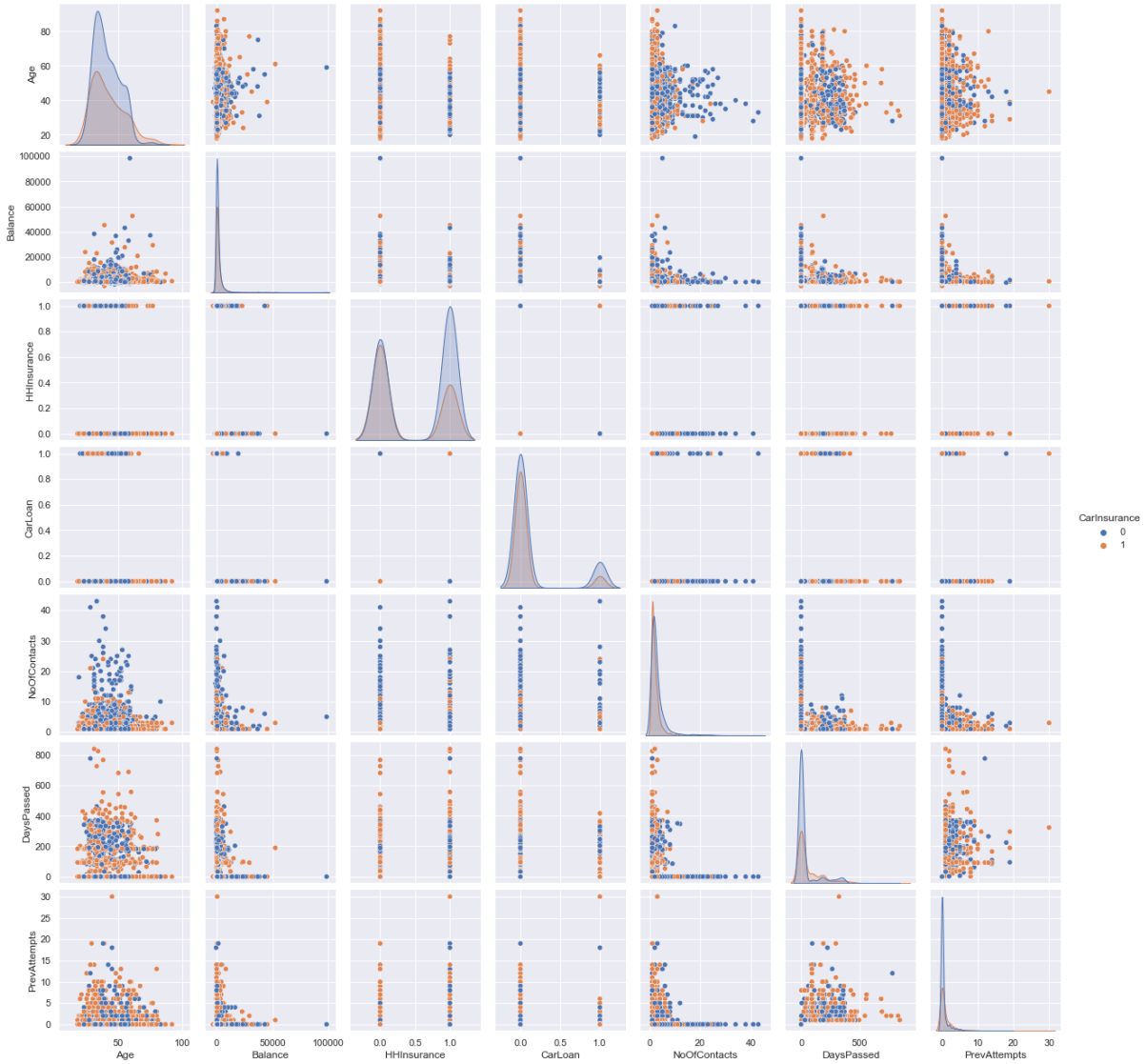
I- Hoeffding's inequality

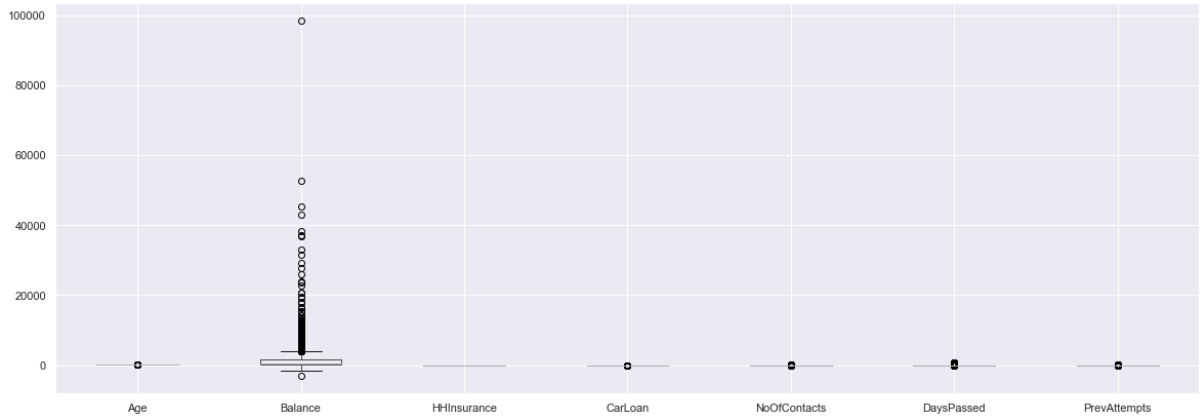




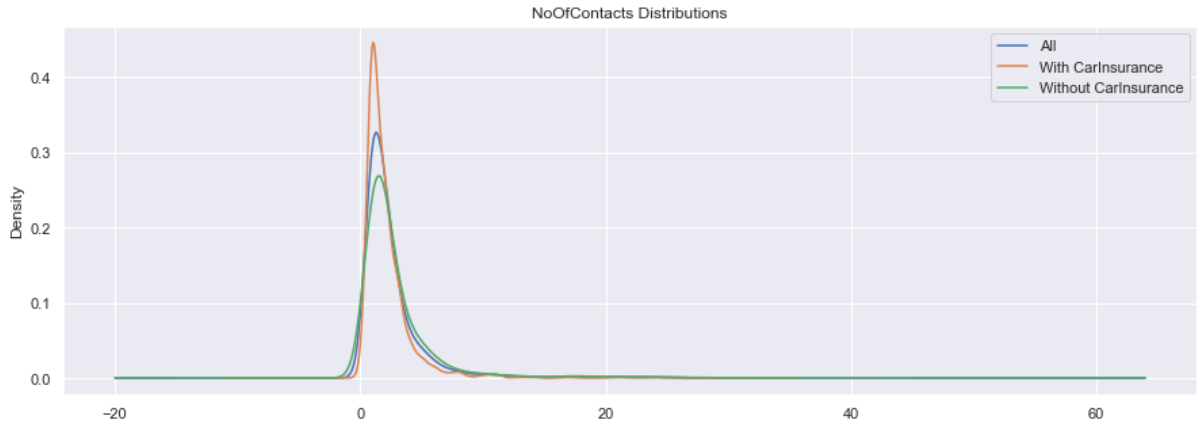
II- Features analysis



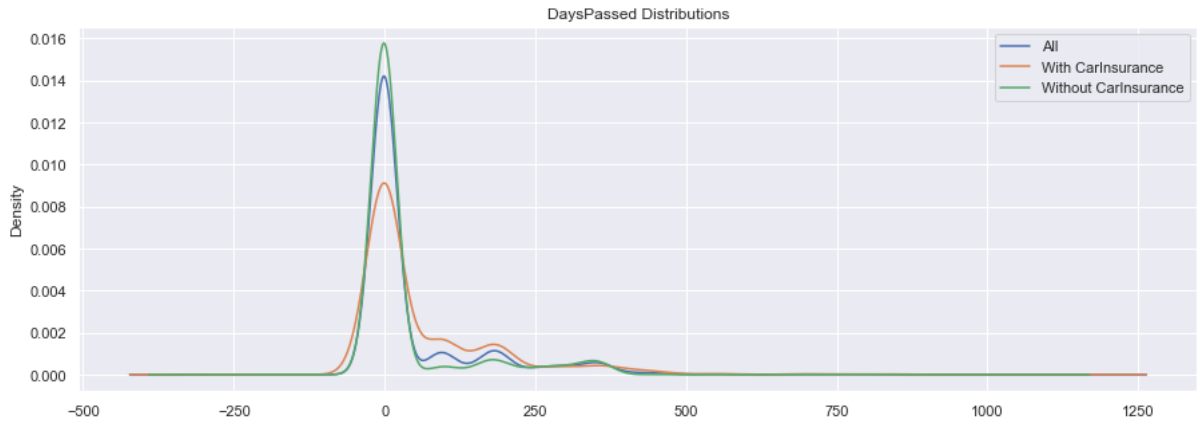

Age
Balance
HHInsurance
CarLoan
NoOfContacts
DaysPassed
PrevAttempts
CarInsurance
0
1


Age
Balance
HHInsurance
CarLoan
NoOfContacts
DaysPassed
PrevAttempts

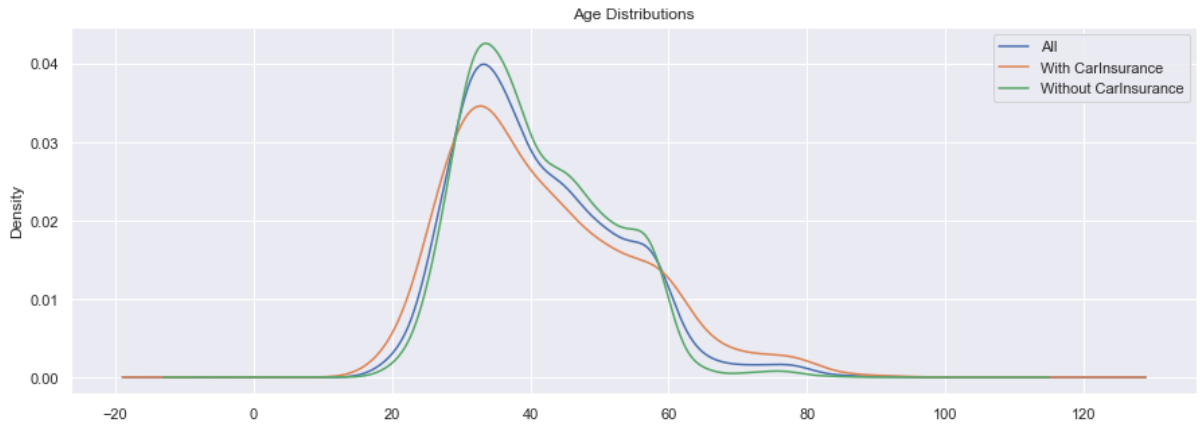
III- Continuous Features only



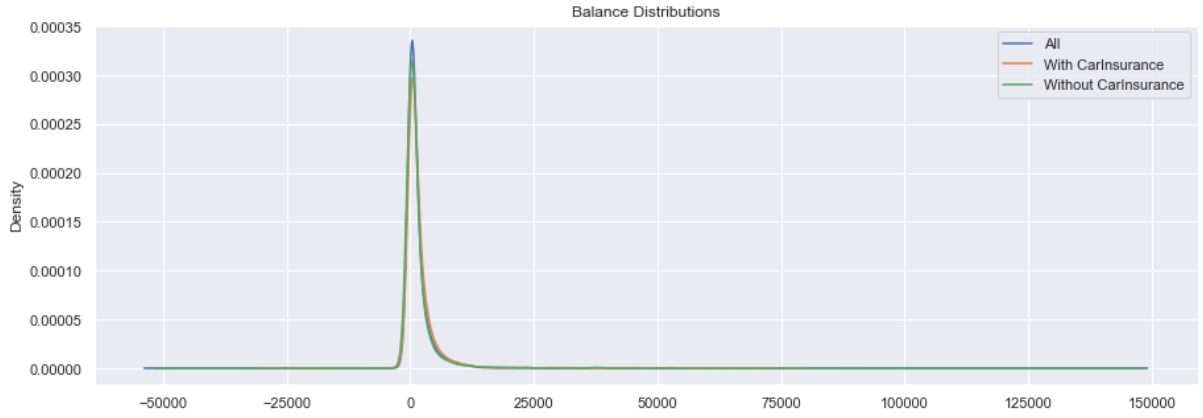
People with higher number of contacts are less likely to buy car insurance



People with higher DaysPassed since the last contact are more likely to buy car insurance

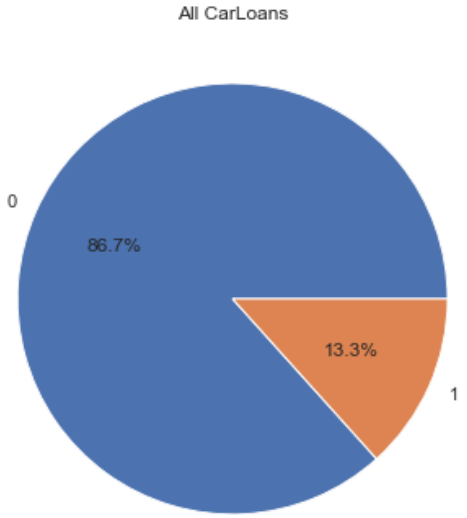


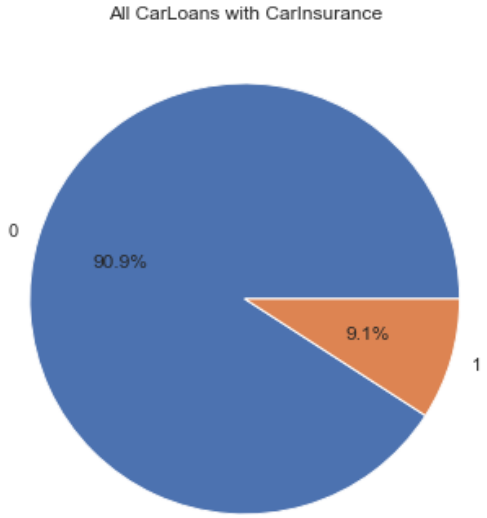
Older people (above 60) and young people (below 30) are more likely to buy car insurance.

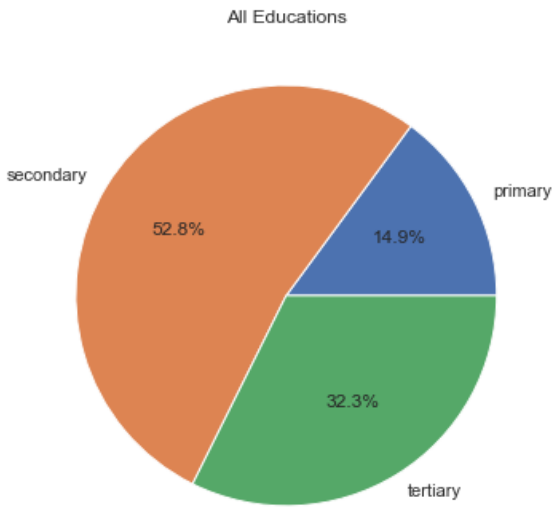


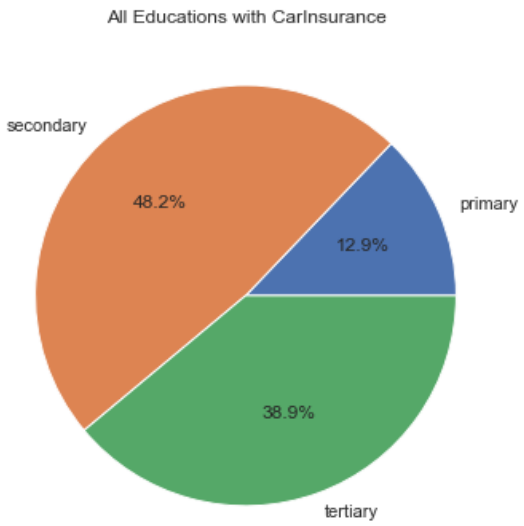
People with higher Balance are more likely to buy car insurance

IV- Categorical Features only

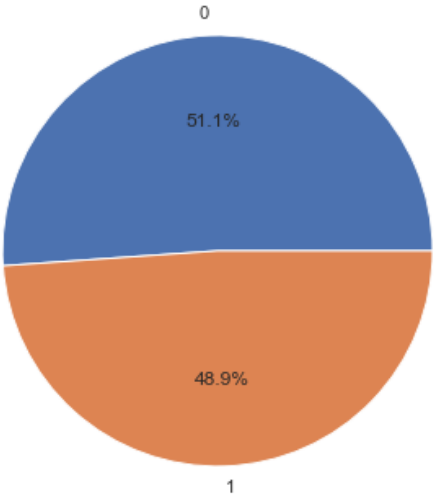




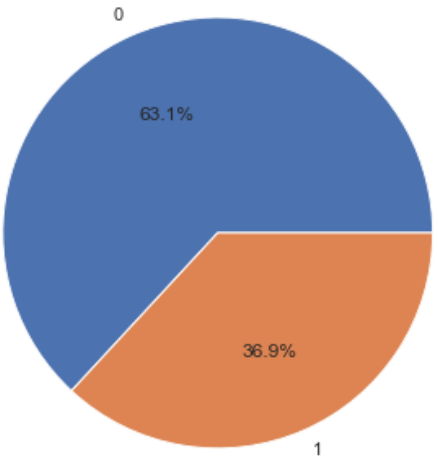




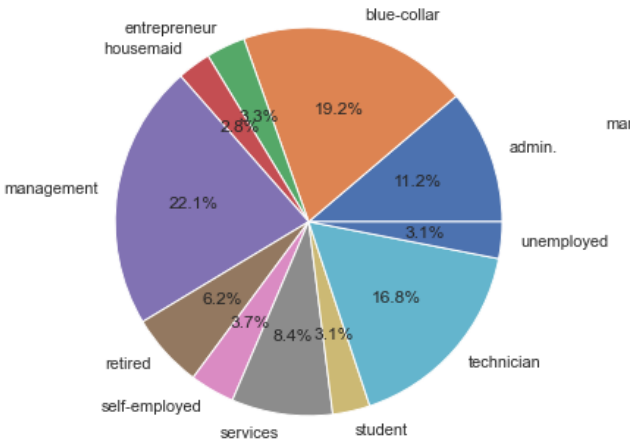
All HHInsurances



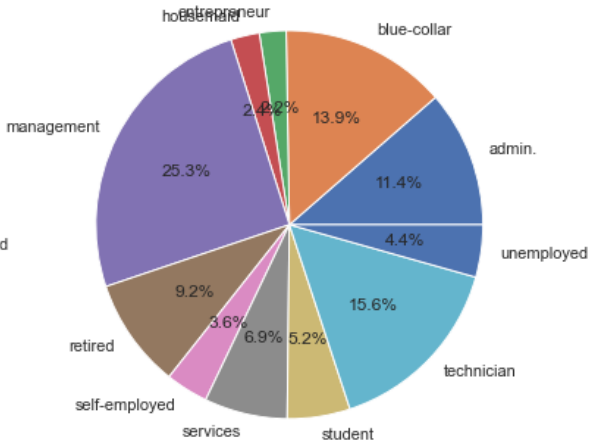
All HHInsurances with CarInsurance

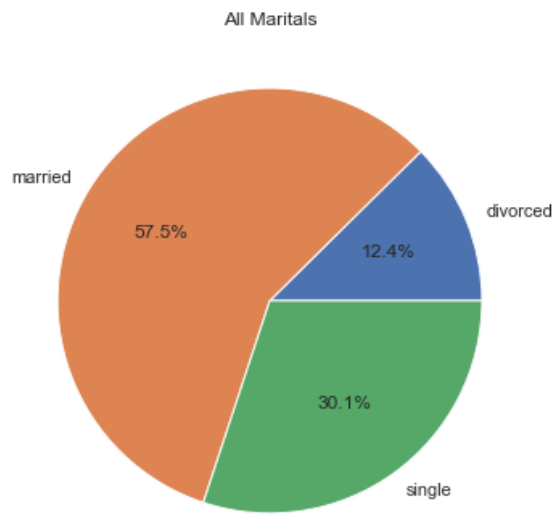


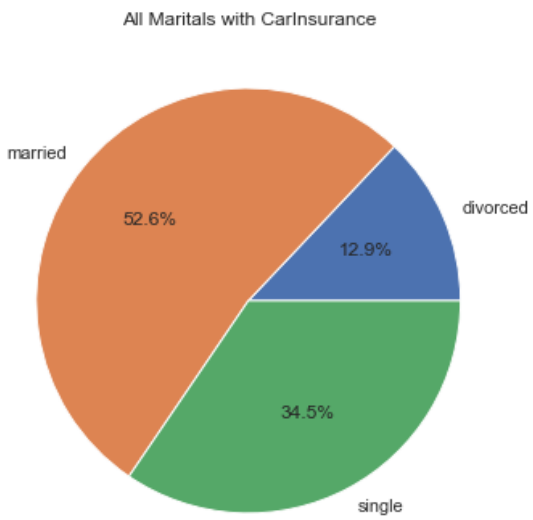
All Jobs

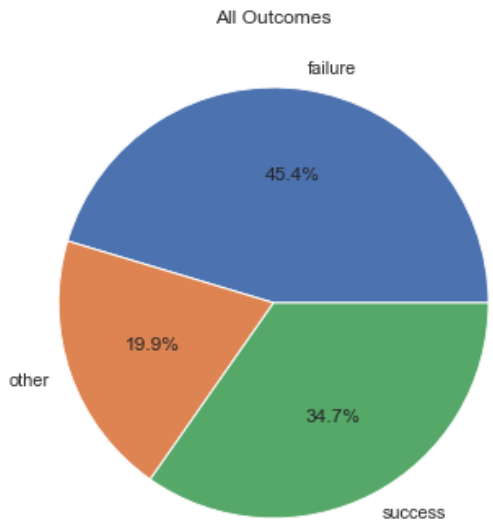


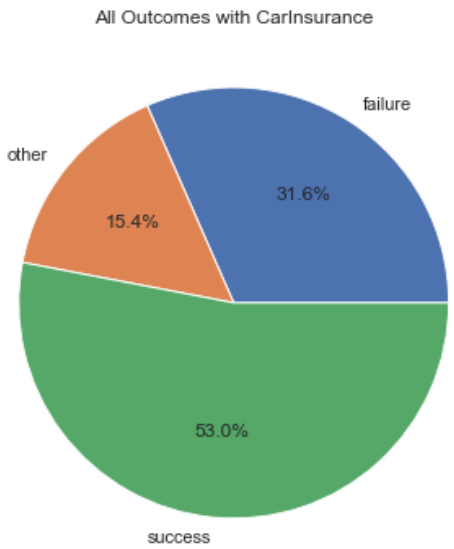
All Jobs with CarInsurance




All Maritals
married
57.5%
divorced
12.4%
30.1%
single


All Maritals with CarInsurance
married
52.6%
divorced
12.9%
34.5%
single


All Outcomes
failure
45.4%
other
19.9%
34.7%
success


All Outcomes with CarInsurance
failure
other
31.6%
15.4%
53.0%
success