Data: csv file with transactions in online store

12,330 records, 10 numerical columns, 8 categorical columns
target column : ‘Revenue’ - boolean with 10,422 (84%) negative

Task:

Build a machine learning model to predict whether a customer will buy a product or not.

Approach:

- Check if there are missing values in each column
- Split data into features and target
- Identify highly correlated features
- Remove records that contain outliers
- Separately order names of categorical and numerical columns in descending order of relevance to target
- Prepare train and test data
- Use l_1 regularization and grid search to find the most accurate Logistic Regressor
- Consider alternative models - LR without penalty with i most relevant numeric and j most relevant categorical features
- Build neural network estimator using Bayesian Optimization to tune hyperparameters

Results:

- No missing values found in data
- 4 pairs of highly correlated features with Pearson's coefficient

$$\rho_{X,Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} > 0.6$$

Administrative	**Administrative_Duration**
Informational	**Informational_Duration**
ProductRelated	**ProductRelated_Duration**
ExitRates	**BounceRates**

- For each column 99 quantile was computed and records that contain outliers (values > 99 quantile) were removed. 11,923 records are kept with 1,719 positive.
- For each categorical column mutual information with target was computed:

$$I(X,Y) = E\left(\ln \frac{p(x,y)}{p(x)p(y)}\right) = \sum_{x,y} p(x,y)\left[\ln p(x,y) - \ln p(x)p(y)\right]$$

feature name	mutual_info_w_target	num_categories
TrafficType	0.0158	20
Month	0.01542	10
VisitorType	0.00491	3
OperatingSystems	0.00321	8
Browser	0.0012	13
Weekend	0.00042	2
Region	0.00038	9

- Numerical columns ordered by absolute value of Pearson's coefficient

feature name	corr_coeff w target
PageValues	0.56
ExitRates	-0.199
ProductRelated	0.171
Administrative	0.147
Informational	0.105
SpecialDay	-0.076

- Split data into train and test
- Transform categorical columns using OneHotEncoder and numerical features using StandardScaler

- Logistic Regression with l_1 regularization

$$\hat{y} = p(x) = \frac{1}{1+e^{-t}} \qquad t = b_0 + b \cdot x$$

Objective function that is minimized:

$$-\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right] \leftarrow \text{BCE}$$

$$+ \quad C^{-1} \Sigma_i |b_i| \leftarrow \text{penalty}$$

Grid search cross-validation: scoring='accuracy',
cv=RepeatedStratifiedKfold (n_splits=10, n_repeats=3)

class_weight =['none', 'balanced'] C=[[0.01,0.001,0.0005,0.0003,0.0001] solver='saga'

chosen model: class_weight='balanced', C=0.0005, single feature='PageValues'
validation accuracy= 89.697% test accuracy= 89.308%

Alternative way to find best LR model: (i,j) LR models with class-weight='balanced' without penalty - where i number of most relevant numerical features and j number of most relevant categorical features:

	(1, 0)	(1, 1)	(2, 0)	(2, 1)
validation accuracy	89.39	89.26	89.32	88.89
test accuracy	89.27	89.6	89.18	88.89

Neural Network model allows more complex parametrization of p(x):

```
Sequential([
Dense(k, activation ='elu'),
BatchNormalization(),
Dropout(r),
Dense(1,activation='sigmoid')
])

optimizer= Adam(learning_rate)
metrics='accuracy'
loss=BCE
```

Bayesian Optimization over hyperparameters:
k=[4,8,12,16]

r=[0, 0.1,0.2,0.3,0.4,0.5]

learning_rate=[1e-4,1e-3,1e-2]

model chosen: k=12, r=0.1, learning_rate=0.01

validation accuracy = 90.41%
test accuracy = 89.98%