Modeling Exercise for Data Science Candidates – Executive Summary

Amudha Giridharan

Objective:

The goal of this binary classification modeling assignment is to predict the individuals who are likely to have an income >=60K from their demographic, employment, socio-economic and financial information.

Recall is considered as the determining metric instead of accuracy, as the focus is to predict the positive cases rightly while balancing the precision and accuracy.

Data insights:

Initial analysis on the data provided showed that

1. Data is highly imbalanced (90% of the cases are negative)– this calls for over/under sampling and/or choice of tree based classifiers to avoid overfitting
2. All the predictors looked weak, and this was another reason to go for tree based ensemble-boosting models
3. Data is biased – for example we can see a bias in education. The % of master degree holder is around 9 to 13 as per Wikipedia. Other websites also have a close % value but the provided train data has only 4%. This indicates that the provided data might not be a true representation of the population. We also observe a slight gender bias with 64% Males.
4. The data has equal number of categorical and numeric variables -encoding is needed for optimal prediction

Approach:

1. **Data pre-processing**
   a. Csv read was updated to handle '?' (treat is as nulls) and remove init space in all strings in final model. The ? and inti space were identified based on initial data analysis.
   b. Response variable coded to 0/1
   c. Train/Test split was 80/20
   d. As the data has numerical and categorical variables, we had to separate them and impute.
      i. Around 4K of records has NaN values and it was not advisable to remove these, especially the positive cases. So, the decision was to impute.
      ii. KNN imputer was chosen instead of mean for numerical variables. For example, We had around 29 cases which the income was >60K and using mean to impute would have imputed 20 for age, which we didn't want. For one of the scenarios, where occupation was professional-specialty, Marital status was Married, KNN imputed 41.4, which is more reasonable.
      iii. KNN was also the choice for categorical variables too. Onehotencoding was leveraged. The test experiments with label encoding didn't show performance improvement in final model.
2. **Feature selection:**
   a. Correlation coefficient and chi-square tests were used to evaluate feature importance of numerical and categorical features
   b. Based on this analysis, features LotSize, Suburban, OwnHouse , WorkClass, Country were found to have low correlation coeff compared to other features and could be dropped.
   c. Experiments adding in feature Country didn't improve model performance
3. **Feature Scaling:**
   a. Numerical features were scaled using standard scaler. This is to make sure high values in features like capital gain doesn't influence the models compared to features like age.

4. **Data Imbalance:**
   a. SMOTE was used to oversample the positive cases but didn't show any improvement in model performance. This makes sense as the tree based models handle the imbalance well. So, this step wasn't included in final model
5. **Model Selection :**
   a. 10 different classifiers were run with default parameters on the processed data.
   Below are the metrics

Model	Train Accuracy	Test Accuracy	Precision	Recall	AUC	TP	FN	TN	FP
LogisticRegression	91.20%	91.12%	0.74	0.21	0.79	118	446	4888	42
RandomForest	100.00%	92.68%	0.78	0.40	0.90	225	339	4867	63
GradientBoosting	93.22%	92.54%	0.78	0.38	0.92	216	348	4868	62
SVC	91.61%	91.34%	0.89	0.18	0.78	100	464	4918	12
KNN	93.92%	91.32%	0.62	0.39	0.81	222	342	4795	135
Naïve Bayes	89.63%	89.15%	0.46	0.31	0.85	175	389	4723	207
MLP	91.55%	91.32%	0.69	0.28	0.85	156	408	4861	69
XGBoost	95.74%	92.46%	0.72	0.43	0.92	242	322	4838	92
Light GBM	94.53%	92.61%	0.75	0.42	0.92	237	327	4851	79
CatBoost	95.22%	92.83%	0.76	0.44	0.93	249	315	4851	79

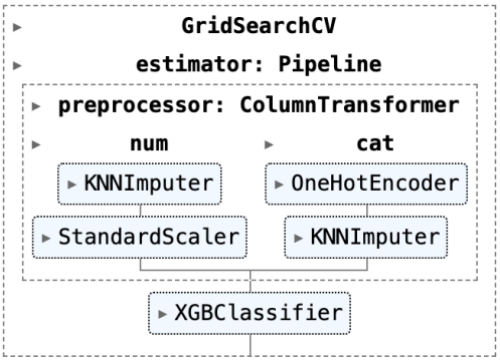
   b. Based on the above metrics, the last 3 models (XGBoost, Light GBM and CatBoost) were taken for further experiments. Scoring was done based on Recall.
   c. Below are the results of the three models with hyper param tuning and cross validation

HyperParam Tuned model	Test Accuracy	Precision	Recall	TP	FN	TN	FP
CatBoost	92.70%	0.72	0.48	270	294	4823	107
Light GBM	70.40%	0.25	0.95	533	31	3335	1595
XGBoost	77.81%	0.30	0.88	495	69	3780	1150

   d. Based on the above results, XGBoost was chosen as the preferred model for further experiments
6. Stacking approach was tried with CatBoost, LightGBM and XGBoost models with meta model as Logistic Regression. But even with hyperparameter tuning, the performance deteriorated compared to XGB.
7. Further hyper parameter tuning , feature selection and threshold tuning was done for performance improvement on XGB model. The final XGB model has marginal improvement compared to the earlier ones.

Model	Train Accuracy	Test Accuracy	Precision	Recall	AUC	TP	FN	TN	FP
XGBoost	76.14%	75.66%	0.28	0.90	0.82	508	56	3648	1281

8. A pipeline was built with preprocessor and classifier



9. The flexibility of inputting threshold has been added for prediction, in case user wants to go for higher recall at the expense of precision or vice versa.
10. Precision-Recall curve is also plotted to observe the trade off at various thresholds to help choose the appropriate threshold value.

Potential improvement space:

1. The hyperparameter choice was very restricted due to long run in public notebooks. High performance environment will provide flexibility to tune the model with additional parameters and range.
2. Ensemble models with neural networks is potential analysis space for this project. Again, this couldn't be done due to system and time constraints.