Analiza Klasyfikacji Posiadaczy Kart Kredytowych

Karol Kot

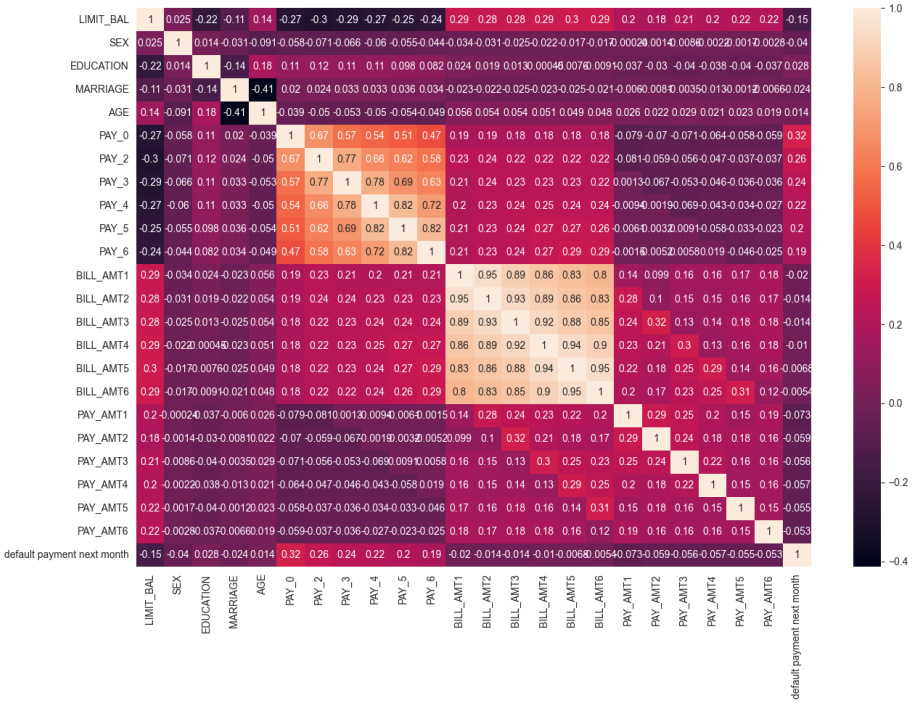
18 czerwca 2024

1 Opis Zadania

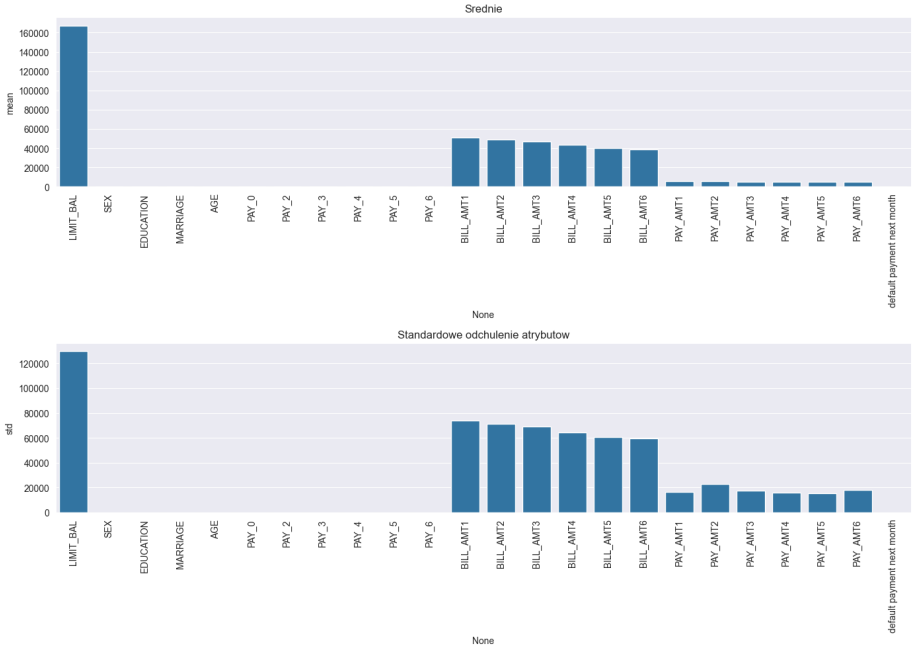
Pracowano z zestawem danych pochodzącym z banku w Tajwanie, zawierającym informacje o posiadaczach kart kredytowych i ich zwyczajach wydatkowych. Zbiór danych zawiera ponad 30,000 obserwacji, z klientami sklasyfikowanymi jako wiarygodni (credible) lub ryzykowni (uncredible), gdzie ci drudzy stanowią 20% zbioru danych. Celem było przetestowanie i porównanie wyników klasyfikatorów na danych oryginalnych, oversamplowanych, undersamplowanych oraz z SMOTE, oraz ocena wpływu selekcji cech na skuteczność klasyfikacji.

2 Analiza danych

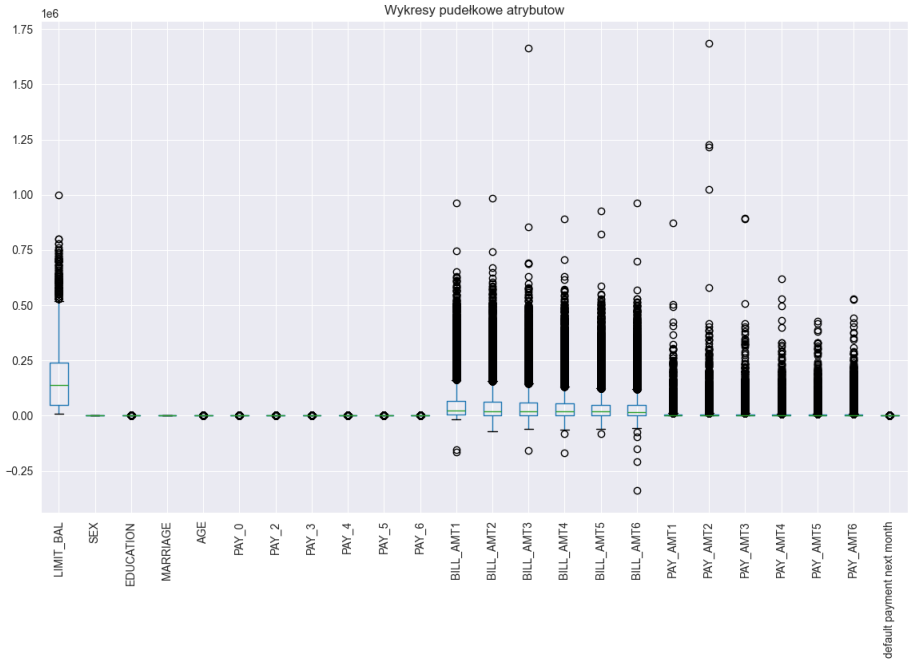
2.1 Wykres korelacji



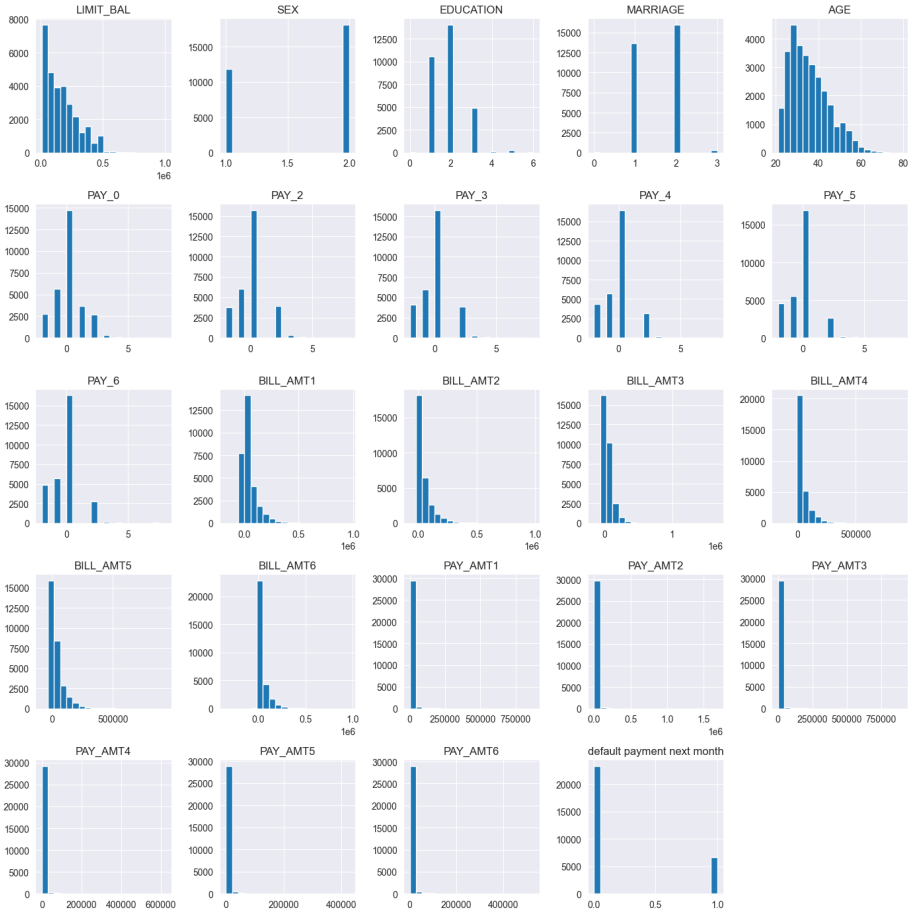
2.2 Srednie wartości atrybutów i standardowe odchylenia atrybutów



2.3 Wykresy pudełkowe



2.4 Histogramy



3 Odpowiedzi na Pytania

3.1 Testowanie klasyfikatora na danych oryginalnych

Wyniki klasyfikatorów na danych oryginalnych:

- **Regresja Logistyczna**:
  - Dokładność: 0.8086
  - AUC: 0.7149
- **XGBoost**:
  - Dokładność: 0.8098
  - AUC: 0.7578

Wyniki pokazują, że XGBoost ma nieco wyższą dokładność i znacznie wyższe AUC w porównaniu do regresji logistycznej na danych oryginalnych.

3.2 Testowanie klasyfikatora na danych oversamplowanych, undersamplowanych i z SMOTE

Wyniki klasyfikatorów na różnych metodach próbkowania:

- **Dane oversamplowane**:
  - Regresja Logistyczna:
    - Dokładność: 0.6836
    - AUC: 0.7155
  - XGBoost:
    - Dokładność: 0.7606
    - AUC: 0.7570
- **Dane undersamplowane**:
  - Regresja Logistyczna:
    - Dokładność: 0.6807
    - AUC: 0.7166
  - XGBoost:
    - Dokładność: 0.7053
    - AUC: 0.7522
- **Dane z SMOTE**:
  - Regresja Logistyczna:
    - Dokładność: 0.6713
    - AUC: 0.7176
  - XGBoost:
    - Dokładność: 0.8058
    - AUC: 0.7557

Wyniki wskazują, że XGBoost konsekwentnie osiąga lepsze wyniki zarówno pod względem dokładności, jak i AUC we wszystkich metodach próbkowania w porównaniu do regresji logistycznej. Oversampling i SMOTE przynoszą większe korzyści dla modelu XGBoost niż undersampling.

3.3 Porównanie wyników oraz obserwacje i wnioski

3.3.1 Obserwacje

- **Wydajność na danych oryginalnych**: XGBoost przewyższa regresję logistyczną pod względem AUC, co wskazuje na lepszą separację klas.
- **Wpływ metod próbkowania**:
  - **Oversampling**: Poprawia AUC dla XGBoost, ale nie tak bardzo dla regresji logistycznej.
  - **Undersampling**: Powoduje niższą dokładność dla obu modeli, ale AUC pozostaje stosunkowo stabilne.
  - **SMOTE**: Zapewnia zrównoważoną poprawę AUC dla obu modeli, przy czym XGBoost znacząco zyskuje na dokładności.

3.3.2 Wnioski

- **Wybór modelu**: XGBoost jest ogólnie bardziej efektywny niż regresja logistyczna dla tego zbioru danych, szczególnie pod względem AUC.
- **Metody próbkowania**: SMOTE i oversampling są bardziej efektywne niż undersampling w poprawie wydajności modeli, szczególnie dla XGBoost.
- **AUC jako metryka**: AUC jest kluczową metryką w tym niezbalansowanym zbiorze danych, ponieważ lepiej odzwierciedla zdolność modelu do rozróżniania między klasami wiarygodnymi i niewiarygodnymi niż sama dokładność.

3.4 Czy selekcja cech zwiększa skuteczność klasyfikacji?

3.4.1 Wyniki selekcji cech

- **Wybrane cechy przy użyciu RFE**:
  - Regresja Logistyczna:
    - Dokładność: 0.8087
    - AUC: 0.6975
  - XGBoost:
    - Dokładność: 0.8140
    - AUC: 0.7389
- **Wybrane cechy przy użyciu SFS**:
  - Regresja Logistyczna:
    - Dokładność: 0.8118
    - AUC: 0.7062

- XGBoost:
  - Dokładność: 0.8128
  - AUC: 0.7429

3.5 Wybór cech przy użyciu SFS i RFE

Wybrane cechy przy użyciu RFE:

- SEX
- EDUCATION
- MARRIAGE
- AGE
- PAY_0
- PAY_2
- PAY_3
- PAY_4
- PAY_5
- PAY_6

Wybrane cechy przy użyciu SFS:

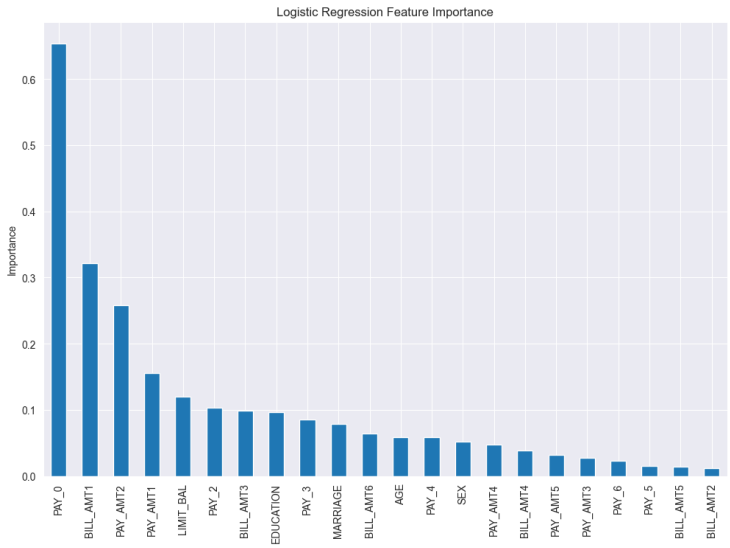
- SEX
- EDUCATION
- MARRIAGE
- AGE
- PAY_0
- PAY_AMT1
- PAY_AMT2
- PAY_AMT3
- PAY_AMT4
- PAY_AMT5

3.5.1 Wykresy ważności cech

W celu oceny ważności cech dla regresji logistycznej oraz XGBoost, zastosowano następujące funkcje - metody zostaly uzyte obok RFE i SFS:

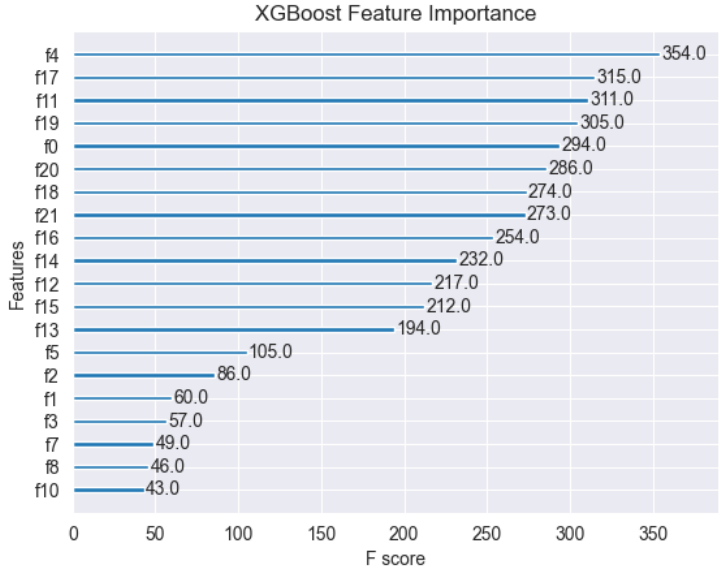
```
def plot_lr_feature_importance(model, feature_names):
    importance = np.abs(model.coef_[0])
    feature_importance = pd.Series(importance, index=feature_names).sort_values(ascending=Fa
    feature_importance.plot(kind='bar', figsize=(12, 8))
    plt.title('Logistic Regression Feature Importance')
    plt.ylabel('Importance')
    plt.show()
    return feature_importance

def plot_xgb_feature_importance(model, feature_names):
    plt.figure(figsize=(12, 8))
    plot_importance(model, max_num_features=20, importance_type='weight')
    plt.title('XGBoost Feature Importance')
    plt.show()
    importance = model.get_booster().get_score(importance_type='weight')
    importance_mapped = {feature_names[int(k[1:])]: v for k, v in importance.items()}
    return importance_mapped
```

Wyniki regresji logistycznej:



Rysunek 1: Wykres ważności cech - Regresja Logistyczna

Wyniki XGBoost:



Rysunek 2: Wykres ważności cech - XGBoost

3.5.2 Obserwacje

- **Regresja Logistyczna**: Selekcja cech nieznacznie poprawia dokładność, ale ma mieszany wpływ na AUC.
- **XGBoost**: Selekcja cech znacząco poprawia zarówno dokładność, jak i AUC, przy czym SFS daje nieco lepsze wyniki niż RFE.

3.5.3 Wnioski

- **Efektywność selekcji cech**: Selekcja cech, szczególnie przy użyciu SFS, poprawia wydajność klasyfikacji dla XGBoost. W przypadku regresji logistycznej wpływ jest mniej wyraźny, ale nadal korzystny pod względem dokładności.
- **Rekomendacja**: Stosowanie metod selekcji cech, takich jak RFE i SFS, może znacząco poprawić wydajność modelu, zwłaszcza w przypadku bardziej złożonych modeli jak XGBoost. W szczególności funkcje oceny ważności cech pomagają zidentyfikować kluczowe atrybuty, które wpływają na skuteczność klasyfikacji.