

# Metrics\_Template

July 11, 2019

## 1 Model Metrics

The following graphs have been automatically generated

### 1.1 AUROC (Area Under Receiver Operating Characteristic)

```
[ ]: plt.figure(figsize=(12, 9)); plt.plot(plot_dict['ROC']['fpr'],  
      ↳plot_dict['ROC']['tpr'], linewidth = 2); plt.title('ROC Curve', size = 20);  
      ↳plt.plot([0, 1], [0, 1], 'k--'); plt.annotate('AUROC: {:.2f} ({:.2f}, {:.  
      ↳2f})'.format(plot_dict['ROC']['auc'], plot_dict['ROC']['auc_lower_bound'],  
      ↳plot_dict['ROC']['auc_upper_bound']), [0.6, 0.4], fontsize = 16); plt.  
      ↳ylabel('True Positive Rate (Sensitivity, Recall)', size = 16); plt.  
      ↳xlabel('False Positive Rate', size = 16)  
if save_figures: plt.savefig(output_dir + 'auroc.png')  
plt.show()
```

## 2 Class Probabilities

```
[ ]: plt.figure(figsize=(12, 9)); sns.distplot(results.iloc[results.iloc[:, 0].values  
      ↳== 0, 1], kde = False, norm_hist = True, label = "Negative", bins = 30); sns.  
      ↳distplot(results.iloc[results.iloc[:, 0].values == 1, 1], kde = False,  
      ↳norm_hist = True, label = "Positive", bins = 30); plt.legend(); plt.  
      ↳title("Distributions by class", size = 20); plt.xlabel("Predicted_  
      ↳Probability", size = 16); plt.ylabel("Normalized Frequency", size = 16)  
if save_figures: plt.savefig(output_dir + 'class_probabilities.png')  
plt.show()
```

### 2.1 Positive Predictive Value by Decile

```
[ ]:
```

```
plt.figure(figsize=(12, 9));sns.boxplot([1 - x for x in
    ↳plot_dict['pid']['decile_midpoint']], plot_dict['pid']['pid']);plt.
    ↳xticks(range(10), [0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.
    ↳95]);plt.title('Positive Predictive Value in Decile of Risk', size = 20);plt.
    ↳xlabel('Risk decile midpoint', size = 16);plt.ylabel('Positive Predictive_
    ↳Value (Precision)', size = 16)
if save_figures: plt.savefig(output_dir + 'ppv_decile.png')
plt.show()
```

## 2.2 Precision @ k

```
[ ]: plt.figure(figsize=(12, 9));plt.plot(plot_dict['precision_at_k']['cutpoints'],
    ↳plot_dict['precision_at_k']['precision_at_k'], linewidth = 2);plt.
    ↳title('Precision @ k', size = 20);plt.ylabel('Precision', size = 16);plt.
    ↳xlabel('Top k risk (percent)', size = 16)
if save_figures: plt.savefig(output_dir + 'precision_at_k.png')
plt.show()
```

## 3 Precision-Recall

```
[ ]: plt.figure(figsize=(12, 9));plt.plot(plot_dict['avg_precision']['recall'],
    ↳plot_dict['avg_precision']['precision'], linewidth = 2);plt.
    ↳title('Precision-Recall Curve', size = 20);plt.annotate('Average Precision:
    ↳{:.2f} ({:.2f}, {:.2f})'.format(plot_dict['avg_precision']['avg_precision'],
    ↳plot_dict['avg_precision']['avg_precision_lower_bound'],
    ↳plot_dict['avg_precision']['avg_precision_upper_bound']), [0.4, 0.8],
    ↳fontsize = 16);plt.ylabel('Positive Predictive Value (Precision)', size =
    ↳16);plt.xlabel('Recall', size = 16)
if save_figures: plt.savefig(output_dir + 'precision_recall.png')
plt.show()
```

## 4 Calibration

```
[ ]: plt.figure(figsize=(12, 9));plt.plot(plot_dict['calibration']['prob_pred'],
    ↳plot_dict['calibration']['prob_true'], linewidth = 2);plt.title('Calibration_
    ↳Curve', size = 20);plt.plot([0, 1], [0, 1], 'k--');plt.ylabel('Empirical_
    ↳Probability', size = 16);plt.xlabel('Predicted Probability', size = 16)
if save_figures: plt.savefig(output_dir + 'calibration.png')
plt.show()
```