

# 09 Amazon Fine Food Reviews Analysis\_RF

April 11, 2019

## 1 Amazon Fine Food Reviews Analysis

Data Source: <https://www.kaggle.com/snap/amazon-fine-food-reviews>

EDA: <https://nycdatascience.com/blog/student-works/amazon-fine-foods-visualization/>

The Amazon Fine Food Reviews dataset consists of reviews of fine foods from Amazon.

Number of reviews: 568,454 Number of users: 256,059 Number of products: 74,258 Timespan:

Oct 1999 - Oct 2012 Number of Attributes/Columns in data: 10

Attribute Information:

1. Id
2. ProductId - unique identifier for the product
3. UserId - unique identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or not
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - brief summary of the review
10. Text - text of the review

**Objective:** Given a review, determine whether the review is positive (rating of 4 or 5) or negative (rating of 1 or 2).

[Q] How to determine if a review is positive or negative? [Ans] We could use Score/Rating. A rating of 4 or 5 can be considered as a positive review. A rating of 1 or 2 can be considered as negative one. A review of rating 3 is considered neutral and such reviews are ignored from our analysis. This is an approximate and proxy way of determining the polarity (positivity/negativity) of a review.

## 2 Random Forest

```
<li><strong>Apply Random Forests & GBDT on these feature sets</strong>
<ul>
```

- <li><font color='red'>SET 1:</font>Review text, preprocessed one converted into vectors
- <li><font color='red'>SET 2:</font>Review text, preprocessed one converted into vectors
- <li><font color='red'>SET 3:</font>Review text, preprocessed one converted into vectors

```

<li><font color='red'>SET 4:</font>Review text, preprocessed one converted into vectors
    </ul>
</li>
<br>
<li><strong>The hyper parameter tuning (Consider two hyperparameters: n_estimators & max_depth)</strong>
    <ul>
        <li>Find the best hyper parameter which will give the maximum <a href='https://www.appliedaicourse.com/notebooks/4.3-hyperparameter-tuning-for-random-forest/>
        <li>Find the best hyper parameter using k-fold cross validation or simple cross validation data</li>
        <li>Use gridsearch cv or randomsearch cv or you can also write your own for loops to do this task</li>
    </ul>
</li>
<br>
<li><strong>Feature importance</strong>
    <ul>
        <li>Get top 20 important features and represent them in a word cloud. Do this for BOW & TFIDF.</li>
    </ul>
</li>
<br>
<li><strong>Feature engineering</strong>
    <ul>
        <li>To increase the performance of your model, you can also experiment with feature engineering
            <ul>
                <li>Taking length of reviews as another feature.</li>
                <li>Considering some features from review summary as well.</li>
            </ul>
        </ul>
    </ul>
</li>
<br>
<li><strong>Representation of results</strong>
    <ul>
        <li>You need to plot the performance of model both on train data and cross validation data for each estimator
            <img src='3d_plot.JPG' width=500px> with X-axis as <strong>n_estimators</strong>, Y-axis as <strong>accuracy</strong>
            <p style="text-align:center;font-size:30px;color:red;"><strong>(or)</strong></p> <br>
        <li>You need to plot the performance of model both on train data and cross validation data for each estimator
            <img src='heat_map.JPG' width=300px> <a href='https://seaborn.pydata.org/generated/seaborn.heatmap.html'>Heatmap</a>
        <li>You choose either of the plotting techniques out of 3d plot or heat map</li>
        <li>Once after you found the best hyper parameter, you need to train your model with it, and finally print the ROC curve</li>
            <img src='train_test_auc.JPG' width=300px></li>
        <li>Along with plotting ROC curve, you need to print the <a href='https://www.appliedaicourse.com/notebooks/4.3-hyperparameter-tuning-for-random-forest/>
            <img src='confusion_matrix.png' width=300px></li>
    </ul>
</li>
<br>
<li><strong>Conclusion</strong>
    <ul>
        <li>You need to summarize the results at the end of the notebook, summarize it in the table form
            <img src='summary.JPG' width=400px>
    </li>

```

</ul>

#

Note: Data Leakage

1. There will be an issue of data-leakage if you vectorize the entire data and then split it into train/cv/test.
2. To avoid the issue of data-leakage, make sure to split your data first and then vectorize it.
3. While vectorizing your data, apply the method fit\_transform() on your train data, and apply the method transform() on cv/test data.
4. For more details please go through this link.

## 2.1 [A] Applying RF

### 3 Import Required Packages

```
In [1]: import os
        from datetime import datetime
        import pandas as pd
        import numpy as np

        # visualization related packages
        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set()

        # data processing related
        from sklearn.preprocessing import StandardScaler

        # import model related packages and visualization of trees
        from sklearn.ensemble import RandomForestClassifier
        from xgboost import XGBClassifier
        from sklearn.tree import export_graphviz
        import graphviz

        # import model selection packages
        from sklearn.model_selection import StratifiedKFold

        # import model evaluation related packages
        from sklearn.metrics import confusion_matrix
        from sklearn.metrics import precision_recall_fscore_support
        from sklearn.metrics import auc, roc_curve
        from scipy import interp

        # calibration related package
        from sklearn.calibration import CalibratedClassifierCV
```

```

# visualization related packages
from wordcloud import WordCloud
from prettytable import PrettyTable

# for forming grid
from itertools import product

```

## 4 UTILS functions

### 4.1 Data Preprocessing related Functions

In [4]:

```

def preprocess_data(config_dict, scaling=True, dim_reduction=False):
    """
    This function does preprocessing of data such as column standardization and
    dimensionality reduction using Truncated SVD
    """

    # Read train, test data frames & truncate it as needed
    train_df = pd.read_csv(config_dict['train_csv_path'], index_col=False)
    train_df = train_df.iloc[0:config_dict['train_size']]
    test_df = pd.read_csv(config_dict['test_csv_path'], index_col=False)
    test_df = test_df.iloc[0:config_dict['test_size']]

    # print the statistics of train, test df
    print('Train df shape', train_df.shape)
    print('Class label distribution in train df:\n', train_df['Label'].value_counts())
    print('Test df shape', test_df.shape)
    print('Class label distribution in test df:\n', test_df['Label'].value_counts())

    # separate features and labels
    train_features = train_df.drop(['Label', 'Id'], axis=1)
    train_labels = train_df['Label']
    test_features = test_df.drop(['Label', 'Id'], axis=1)
    test_labels = test_df['Label']

    # create a list containing all the features
    feature_name_list = train_features.columns.values.tolist()

    # If Scaling is opted scale the train, test data
    if scaling:
        standard_scaler = StandardScaler()
        standard_scaler.fit(train_features)

        train_features = pd.DataFrame(standard_scaler.transform(train_features),
                                      columns=feature_name_list)
        test_features = pd.DataFrame(standard_scaler.transform(test_features),
                                    columns=feature_name_list)

```

```

print('Shape of -> train features :%d,%d, test features: %d,%d'%(train_features.shape[0], test_features.shape[0]))
print('Shape of -> train labels :%d, test labels: %d'%(train_labels.shape[0], test_labels.shape[0],))

# if dim reduction is opted, reduce the dimension
if dim_reduction:
    # create an SVD object
    truc_svd = TruncatedSVD(n_components=train_features.shape[1]-1, n_iter=8, algorithm='randomized')

    # fit to data
    truc_svd.fit(train_features)
    # get explained variance ratio of each component
    explained_var_ratios = truc_svd.explained_variance_ratio_
    # get cumulative ratio list for selecting the number of components
    cumulative_ratios = np.cumsum(explained_var_ratios)

    # plot the #components vs captured variance in the data
    plt.title('SVD Decomposition')
    plt.xlabel('Number of components')
    plt.ylabel('Cumulative Percentage Ratio')
    plt.plot(range(1,train_features.shape[1]), cumulative_ratios)
    plt.show()

    # set a threshold for stopping selection of components.
    svd_thresh = 0.001
    # select the number of components as the first component for which the difference
    # very less (less than svd thresh) compared with the very next component
    selected_dim = list(filter(lambda x : x[1] < svd_thresh, enumerate(np.diff(cumulative_ratios))))
    print('Num dimensions selected by SVD', selected_dim)
    print('Total variance captured:%f'%(cumulative_ratios[selected_dim]))

    # create an object for selecting the components
    truc_svd = TruncatedSVD(n_components=selected_dim, n_iter=8, algorithm='randomized')
    # refit with the desired number of components
    truc_svd.fit(train_features)

    # reduce the number of dimensions to selected number of components
    train_features = pd.DataFrame(truc_svd.transform(train_features))
    test_features = pd.DataFrame(truc_svd.transform(test_features))

    # get the shape of final data frame and print it
    size_tuple = train_features.shape + test_features.shape
    print('Shape of train df:(%d,%d), Test DF:(%d,%d)'%size_tuple)

return (train_features, train_labels, test_features, test_labels,)

```

## 4.2 Model Training and Evaluation related packages

```
In [5]: def plot_roc_curves(fold_prediction_list, plot_title):
    """
    This function helps to plot the ROC curve for a set of predictions on different fold
    """

    # set figure size
    plt.figure(figsize=(10,10))

    # reference points for X axis
    ref_points = np.linspace(0.0, 1.0, 100)

    # two lists for auc values and tpr rates
    auc_scores_list = list()
    tpr_list = list()

    # plot ROC curve for each fold
    for index, (actual_probs, predicted_probs,) in enumerate(fold_prediction_list):

        # compute ROC curve and get the AUC value for this fold
        fpr, tpr, thresholds = roc_curve(actual_probs, predicted_probs)
        # compute AUC
        auc_score = auc(fpr, tpr)

        # interpolation to approximate the curve
        tp_rates = interp(ref_points, fpr, tpr)
        tp_rates[0] = 0.0 # for setting the bottom left point

        # for plotting the individual fold and finding the average
        auc_scores_list.append(auc_score)
        tpr_list.append(tp_rates)

        # plot this fold info into a fig
        plt.plot(fpr, tpr, alpha=0.3, lw=1, label='AUC for fold %d : %f'%(index, auc_score))

    # Plot the random classifier
    plt.plot([0,1],[0,1], alpha=0.8, linestyle='--', color='red', label='Random Guess',)

    # Plot the mean performance
    mean_tpr = np.mean(tpr_list, axis=0)
    std_tprs = np.std(tpr_list, axis=0)
    # mean value of AUC and its standard deviation
    mean_auc = auc(ref_points, mean_tpr)
    std_auc = np.std(auc_scores_list)

    plt.plot(ref_points, mean_tpr, linestyle='-', color='g', lw=2,
             alpha=0.8, label='Mean AUC %f $\pm$ %f'%(mean_auc, std_auc))
```

```

# Find upper and lower bounds for shading the region around TPRs
tprs_lower_region = np.maximum(mean_tpr - std_tprs, 0)
tprs_upper_region = np.minimum(mean_tpr + std_tprs, 1)

# Fill the region between upper and lower in gray color
plt.fill_between(ref_points, tprs_lower_region, tprs_upper_region, color='gray', alpha=0.3, label='Around the mean TPRs')

# arrange the plot
plt.xlim([-0.05, 1.05])
plt.ylim([-0.05, 1.05])
plt.xlabel('False Positive Rates')
plt.ylabel('True Positive Rates')
plt.title('ROC - ' + plot_title)
plt.legend(loc='lower right')
plt.show()

return mean_auc

```

In [6]: def get\_confusion\_matrix(actual\_list, predicted\_list, cm\_title):  
 """  
*This function plots the confusion matrix given ground truth and predicted*  
 """

```

conf_matrix = confusion_matrix(actual_list, predicted_list)
col_names = ['Negative', 'Positive']
conf_df = pd.DataFrame(conf_matrix, columns=col_names)
conf_df.index = col_names

plt.figure(figsize = (5,5))

plt.title(cm_title)
sns.set(font_scale=1.4) #for label size
sns.heatmap(conf_df, annot=True, annot_kws={"size": 16})

plt.show()

```

In [7]: def create\_hyperparam\_heatmap(hyper\_param\_score\_list, hamp\_title):  
 """  
*This function accepts a list containing the hyper parameters ('max\_depth', 'n\_estimators') and the corresponding score value, such as AUC, FScore etc and plot the heatmap.*  
 """

```

hyp_score_list = [item[1] for item in hyper_param_score_list]
coord_list = [item[0] for item in hyper_param_score_list]
hyp_df = pd.DataFrame(coord_list, columns=['max_depth', 'n_estimators'])

```

```

hyp_df['AUC'] = hyp_score_list

# pivot the table for heatmap representation
pivoteds_hyp = pd.pivot_table(hyp_df, index='n_estimators', columns='max_depth',
                               values='AUC', fill_value=0)
sns.heatmap(data=pivoteds_hyp)
plt.title(hamp_title)
plt.show()

In [8]: def find_best_hyperparameter(config_dict, train_features, train_labels):
    """
    This function helps to find the best hyper parameter (alpha) for MultinomialNB algorithm.
    All set of hyper param values using which the model to be evaluated can be passed to
    list hyperparam_list.
    """
    print('='*100)

    stratified_partition = StratifiedKFold(n_splits=2)

    # read some config settings
    hyperparam_list = config_dict['hyperparam_list']
    implementation = config_dict['implementation']

    hyper_param_score_list_train = list()
    hyper_param_score_list_validation = list()

    for hyp_val in hyperparam_list:

        # declare three lists for holding prediction informations
        # for train set performance
        train_actual_labels_list = list()
        train_predicted_probs_list = list()
        train_predicted_labels_list = list()

        # for validation set performance
        val_actual_labels_list = list()
        val_predicted_probs_list = list()
        val_predicted_labels_list = list()

        # Model defined here
        if implementation == 'rf':
            classifier = RandomForestClassifier(max_depth=hyp_val[0], n_estimators=hyp_val[1],
            # Declare a calibrated classifier
            calib_classifier = CalibratedClassifierCV(base_estimator=classifier,
                                                       method='isotonic',
                                                       cv='prefit')
        else:
            classifier = XGBClassifier(max_depth=hyp_val[0], n_estimators=hyp_val[1],

```

```

        learning_rate=hyp_val[2])

# Train the model and evaluate it on the current fold data
for train_indices, val_indices in stratified_partition.split(train_features, tra

# A) train the model suing StratifiedKFold method

# get the train features, train labels for this fold
train_feat_data = train_features.iloc[train_indices, :]
train_label_data = train_labels[train_indices]

# train the classifier
classifier.fit(train_feat_data, train_label_data)

# For random forest we need to calibrate
if implementation == 'rf':
    calib_classifier.fit(train_feat_data, train_label_data)
    train_eval_y_probs = calib_classifier.predict_proba(train_feat_data)[:, :]
else:
    train_eval_y_probs = classifier.predict_proba(train_feat_data)[:, 1]

# estimate the training metrics on (train fold)
train_eval_y_value = classifier.predict(train_feat_data)

# save the results for ROC plot
train_actual_labels_list.append(train_label_data)
train_predicted_probs_list.append(train_eval_y_probs)
train_predicted_labels_list.append(train_eval_y_value)

# B) predict the labels and probability for this fold (validation fold)

# get the validation features, validation labels for this fold
validation_feat_data = train_features.iloc[val_indices, :]
validation_label_data = train_labels[val_indices]

# evaluate the classifier on validation set
val_actual_labels_list.append(validation_label_data)

# predict the probability for validation data
if implementation == 'rf':
    # validation step
    val_eval_y_probs = calib_classifier.predict_proba(validation_feat_data)[
else:
    val_eval_y_probs = classifier.predict_proba(validation_feat_data)[:, 1]

```

```

# predict the output for validation
val_eval_y_value = classifier.predict(validation_feat_data)

# save the results for ROC plot
val_predicted_probs_list.append(val_eval_y_probs)
val_predicted_labels_list.append(val_eval_y_value)

# plot the results to select best hyper params

# train data plot
train_fold_prediction_list = list(zip(train_actual_labels_list, train_predicted_))

# plot the roc curve for train data
if len(hyp_val) == 2:
    tr_mean_auc = plot_roc_curves(train_fold_prediction_list,
                                   'Train Ensemble (max_depth:%d, num_estimators:%d)'%hyp_v

# plot the roc curve for validation data
val_fold_prediction_list = list(zip(val_actual_labels_list, val_predicted_pr

ts_mean_auc = plot_roc_curves(val_fold_prediction_list,
                               'Validation Ensemble (max_depth:%d, num_estimators:%d)'%hyp

elif len(hyp_val) == 3:

    tr_mean_auc = plot_roc_curves(train_fold_prediction_list,
                                   'Train Ensemble (max_depth:%d, num_estimators:%d, learning_r

# plot the roc curve for validation data
val_fold_prediction_list = list(zip(val_actual_labels_list, val_predicted_pr

ts_mean_auc = plot_roc_curves(val_fold_prediction_list,
                               'Validation Ensemble (max_depth:%d, num_estimators:%d, learn

else:
    print('Invalid hyper param configs')

# update the list with the scores for this hyperparam
hyper_param_score_list_validation.append((hyp_val, ts_mean_auc,))
hyper_param_score_list_train.append((hyp_val, tr_mean_auc,))

print('='*100)

# plot heatmap for hyperprams for train data, validation data
if len(hyp_val) == 2:
    create_hyperparam_heatmap(hyper_param_score_list_train, 'Heatmap Hyperparams for Train Data')
    create_hyperparam_heatmap(hyper_param_score_list_validation, 'Heatmap Hyperparameters for Validation Data')

```

```

        elif len(hyp_val) > 2: # this is for xgboost where we use more than 2 hyperparam
            train_hyp_df = pd.DataFrame(hyper_param_score_list_train, columns=['Hyper Params'])
            validation_hyp_df = pd.DataFrame(hyper_param_score_list_validation, columns=['Hyper Params'])
            # print the hyper params and its score
            print('\n\n Train hyper params: \n', train_hyp_df)
            print('\n\n Validation hyper params: \n', validation_hyp_df)

        else:
            print('Invalid hyper param setting found !!!!')

    return hyper_param_score_list_validation

```

```

In [9]: def train_and_validate_model(config_dict, train_features, train_labels):
    """
    This function train a model, validate it using cross validation and return the best
    obtained during cross validation.
    """

    # find best hyperparameter by stratified cross validation
    hyper_param_scores_list = find_best_hyperparameter(config_dict, train_features, train_labels)
    implementation = config_dict['implementation']

    # decalre the optional calib classifier as None
    calib_classifier = None

    #Set the best Hyper param based on above plots
    best_hyper_param = max(hyper_param_scores_list, key=lambda x: x[1])[0]
    print('Best hyperparam value: ', best_hyper_param)

    # Final Model defined here
    if implementation == 'rf':
        classifier = RandomForestClassifier(max_depth=best_hyper_param[0],
                                             n_estimators=best_hyper_param[1])
        # Declare a calibrated classifier
        calib_classifier = CalibratedClassifierCV(base_estimator=classifier,
                                                   method='isotonic',
                                                   cv='prefit')
    else:
        classifier = XGBClassifier(max_depth=best_hyper_param[0],
                                   n_estimators=best_hyper_param[1],
                                   learning_rate=best_hyper_param[2])

    # train the classifier
    classifier.fit(train_features, train_labels)

    if implementation == 'rf':
        calib_classifier.fit(train_features, train_labels)
        train_eval_y_probs = calib_classifier.predict_proba(train_features)[:, 1]

```

```

else:
    train_eval_y_probs = classifier.predict_proba(train_features)[:, 1]

# estimate the training metrics on (train fold)
train_eval_y_value = classifier.predict(train_features)
train_info_list = list(zip([train_labels], [train_eval_y_probs]))

# get train auc value
if len(best_hyper_param) == 2:
    train_auc_val = plot_roc_curves(train_info_list,
                                    'Final Model DT (max_depth:%d, n_estimators:%d)'%best
elif len(best_hyper_param) == 3:
    train_auc_val = plot_roc_curves(train_info_list,
                                    'Final Model DT (max_depth:%d, n_estimators:%d, learning_rate:%f)'%best
else:
    print('Invalid hyper param settings!!!')

# return the final model
return (classifier, calib_classifier)

```

In [10]: def test\_and\_evaluate\_model(config\_dict, model, test\_features, test\_labels):

```

"""
This function test and evaluate the performance on unseen data.
"""

# separate out models
classifier = model[0]
calib_classifier = model[1]

# get type of model
implementation = config_dict['implementation']

# evaluate final model on test dataset
if implementation == 'xgb':
    test_eval_y_probs = classifier.predict_proba(test_features)[:, 1]
else:
    test_eval_y_probs = calib_classifier.predict_proba(test_features)[:, 1]

test_eval_y_value = classifier.predict(test_features)
test_info_list = list(zip([test_labels], [test_eval_y_probs]))

# test auc value
if implementation == 'xgb':
    best_hyp = (classifier.max_depth, classifier.n_estimators, classifier.learning_
    test_auc_val = plot_roc_curves(test_info_list,
                                    'Final Model Ensemble (max_depth:%d, n_estimators:%d, lr_ra
else:
    best_hyp = (classifier.max_depth, classifier.n_estimators,)

```

```

    test_auc_val = plot_roc_curves(test_info_list,
                                    'Final Model Ensemble (max_depth:%d, n_estimators:%d)'%best
                                    )
    print('Test auc score ', test_auc_val)

    # print the confusion matrix
    get_confusion_matrix(test_labels, test_eval_y_value, 'Ensemble Model Confusion Matrix')

    # compute precision and other metrics
    all_metrics = precision_recall_fscore_support(test_labels, test_eval_y_value)
    all_metrics_df = pd.DataFrame(list(all_metrics), columns=['Negative', 'Positive'])
    all_metrics_df.index = ['Precision', 'Recall', 'Fscore', 'Support']
    # convert fscore to percentage
    fscores = all_metrics[2] * 100.0

    print(all_metrics_df)

    # create an entry for pretty table using all the metrics

    # get the table entry
    ptabe_entry = [best_hyp] + [ '%.4f'%item for item in ([test_auc_val] + list(fscores))

    return ptabe_entry

```

#### 4.2.1 [A.1] Applying Random Forests on BOW, SET 1

```

In [9]: # form two lists
        depth_list = [3, 5, 10, 20, 50] # depends on size of dataset
        n_estimators_list = [20, 60, 120, 300, 500] # depends on size of dataset

        # declare a configuartion dictionary
        config_dict = {
            'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/BOW/train.csv',
            'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/BOW/test.csv',
            'train_size' : 50000,
            'test_size' : 20000,
            'hyperparam_list' : list(product(depth_list, n_estimators_list)),
            'implementation': 'rf' # 'xgb' or 'rf'
        }

In [10]: # read the train, test data and preprocess it
        train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                scaling=True,
                                dim_reduction=100)

        # create a list containing all the features
        feature_name_list = train_features.columns.values.tolist()

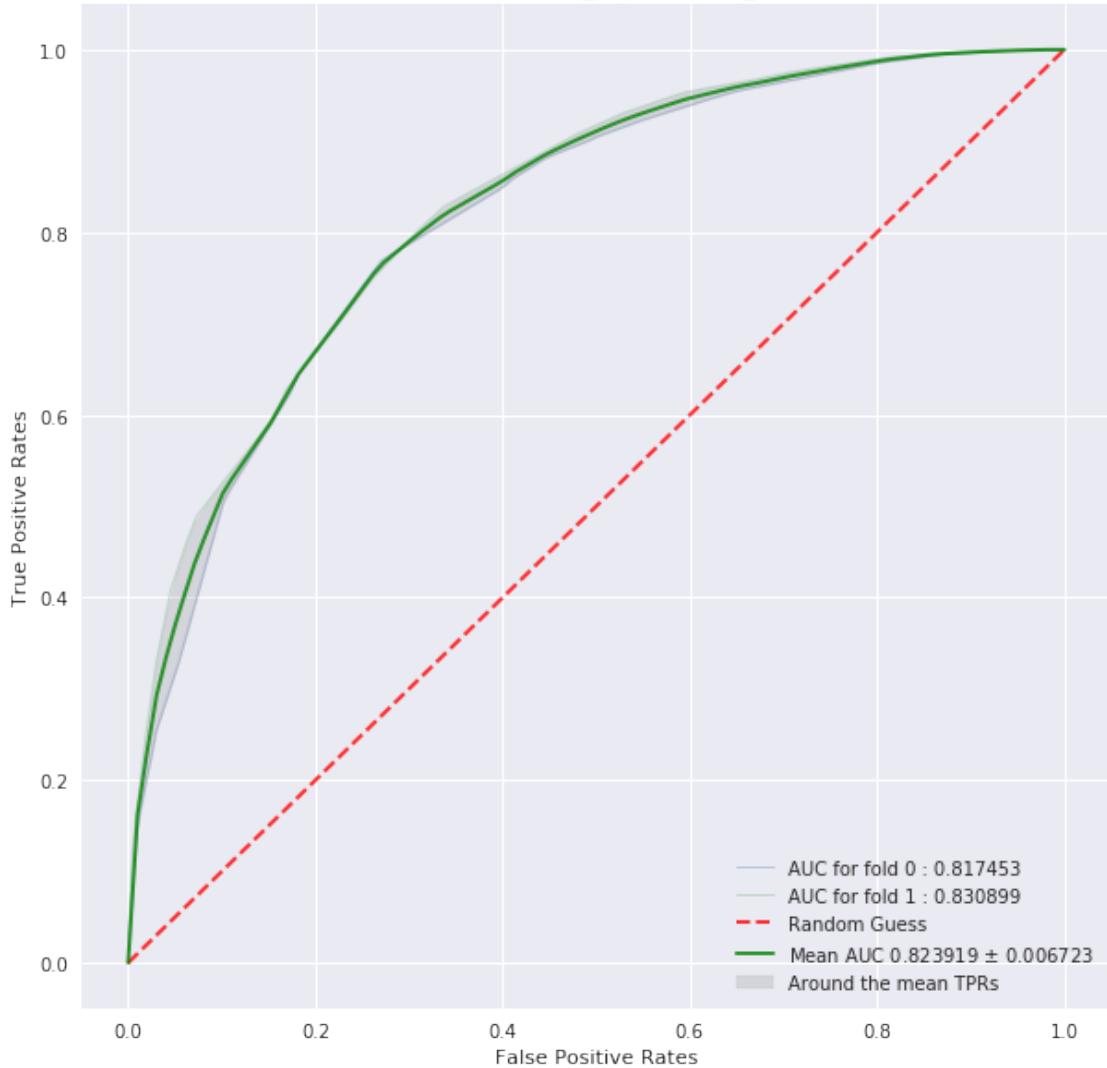
```

```
# train and validate the model
model = train_and_validate_model(config_dict, train_features, train_labels)

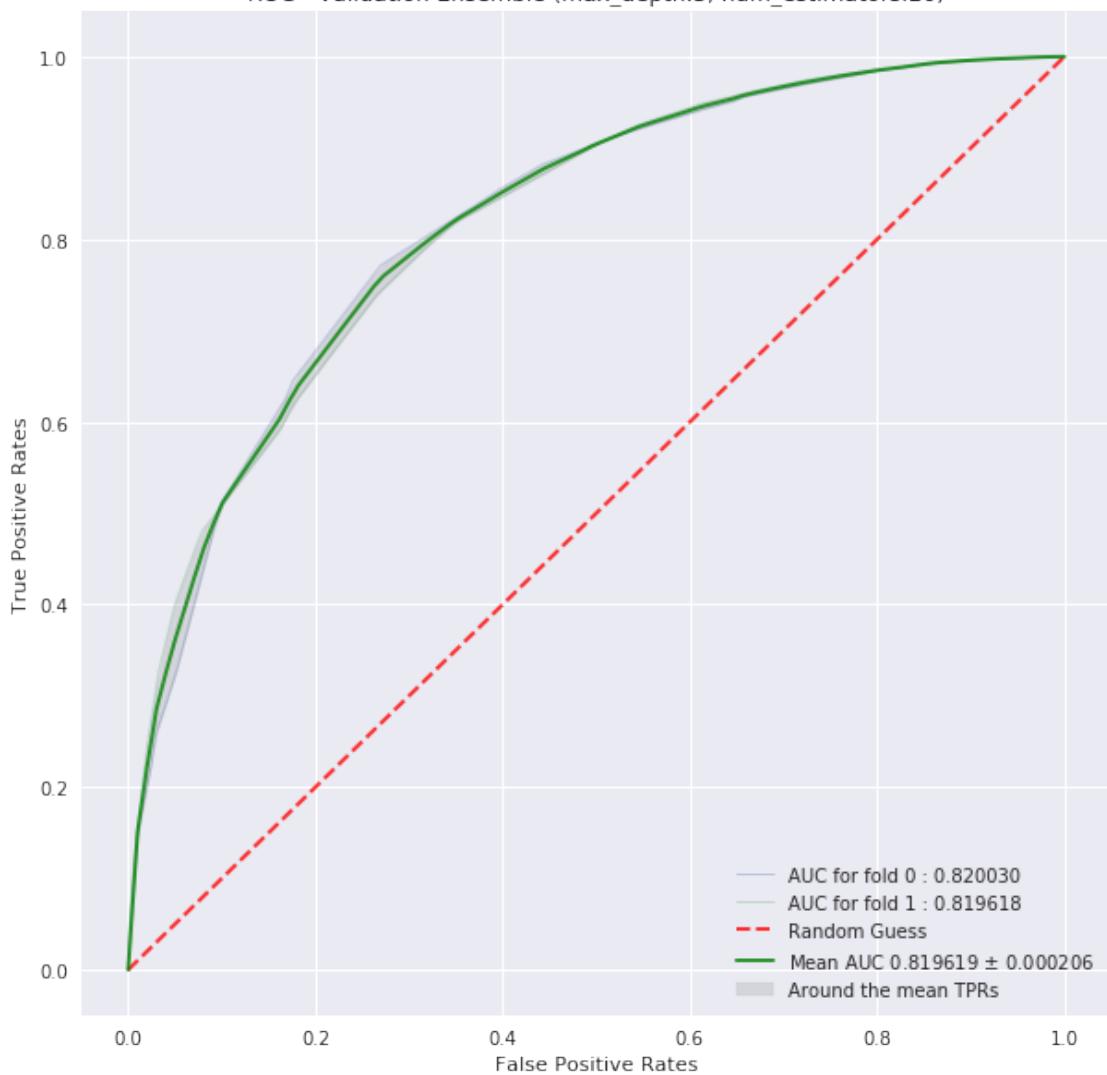
# test and evaluate the model
ptabe_entry_a1 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

Train df shape (50000, 503)
Class label distribution in train df:
0    25029
1    24971
Name: Label, dtype: int64
Test df shape (20000, 503)
Class label distribution in test df:
1    16520
0    3480
Name: Label, dtype: int64
Shape of -> train features :50000,501, test features: 20000,501
Shape of -> train labels :50000, test labels: 20000
=====
=====
```

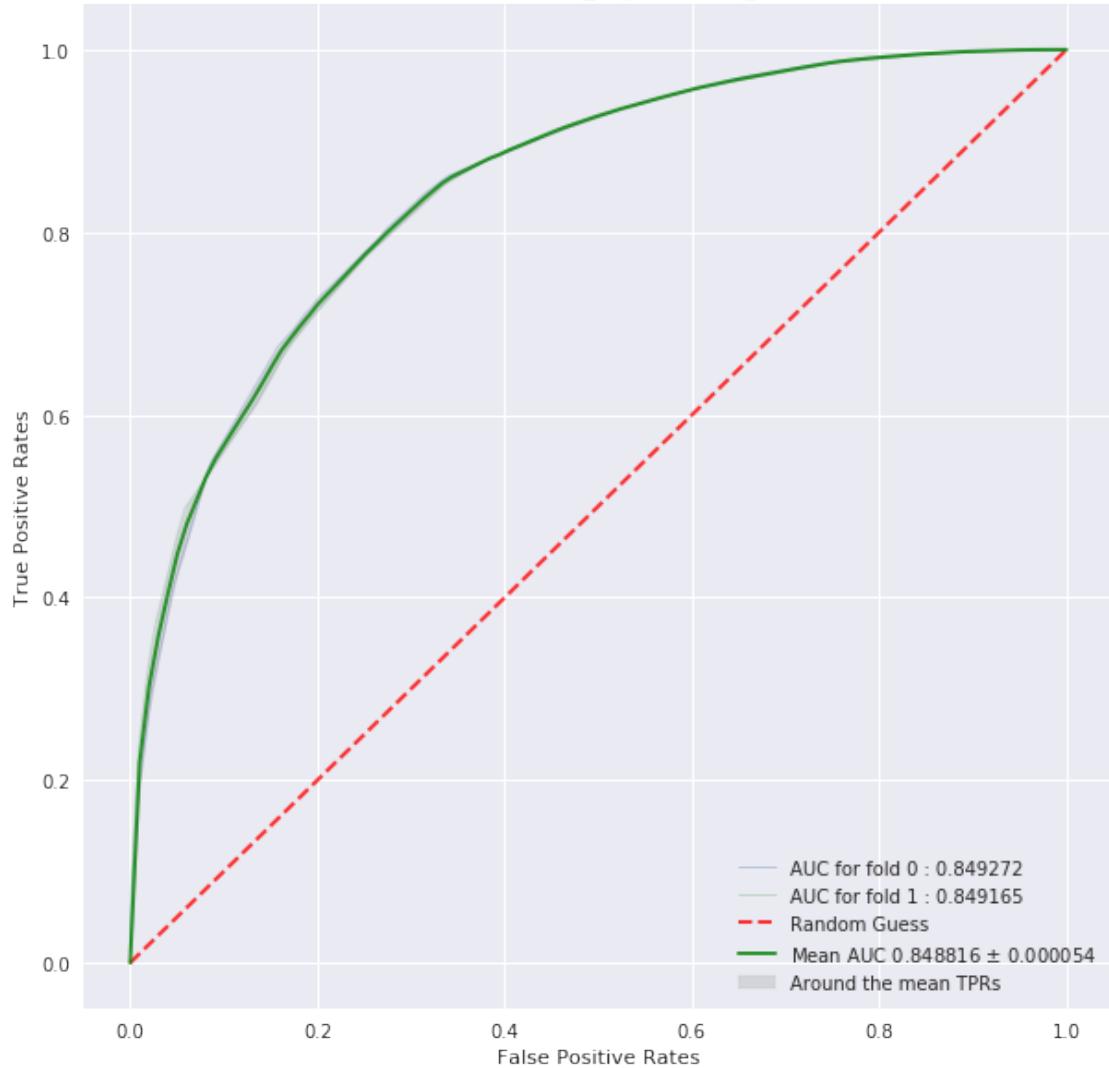
ROC - Train Ensemble (max\_depth:3, num\_estimators:20)



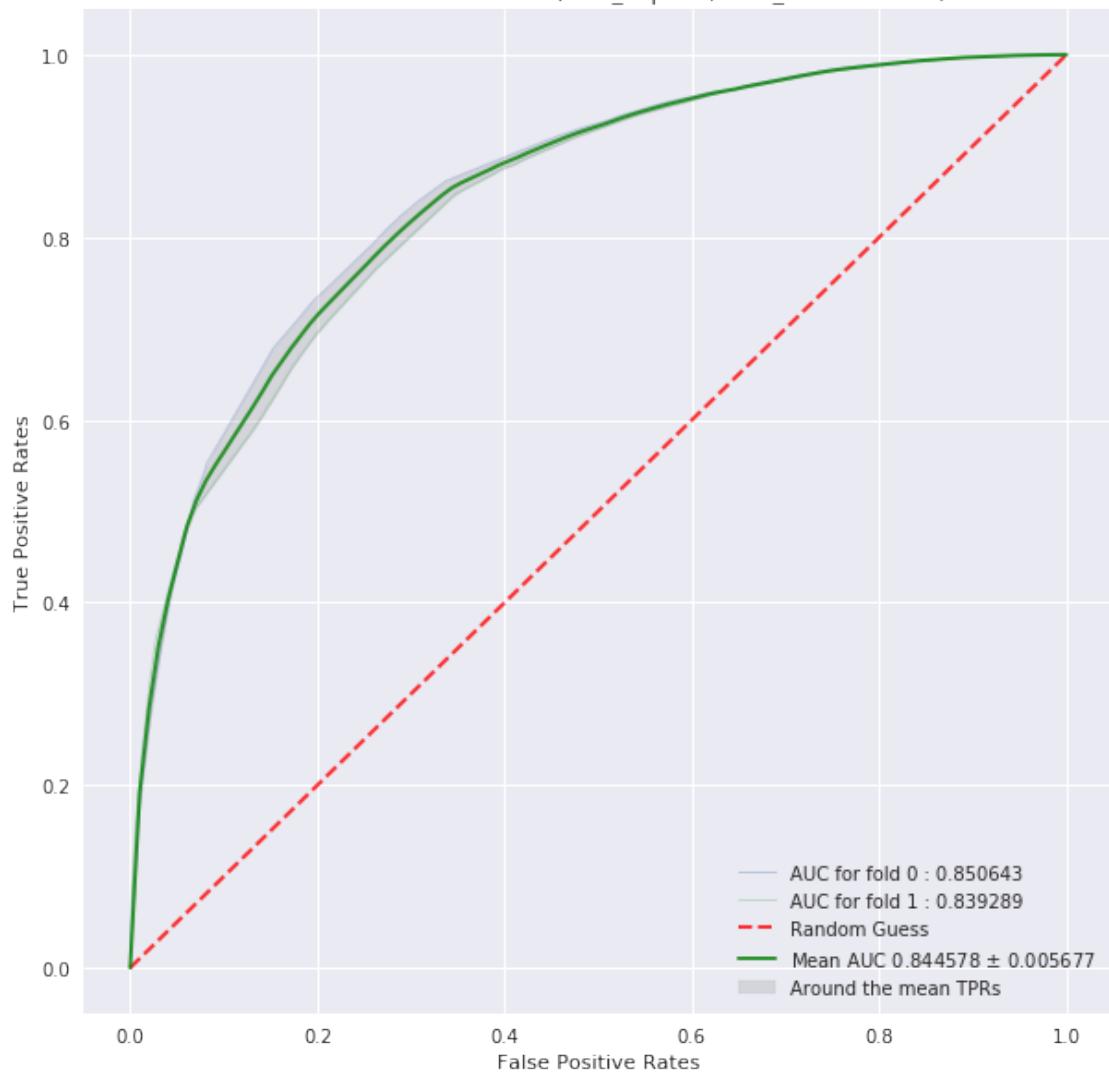
ROC - Validation Ensemble (max\_depth:3, num\_estimators:20)



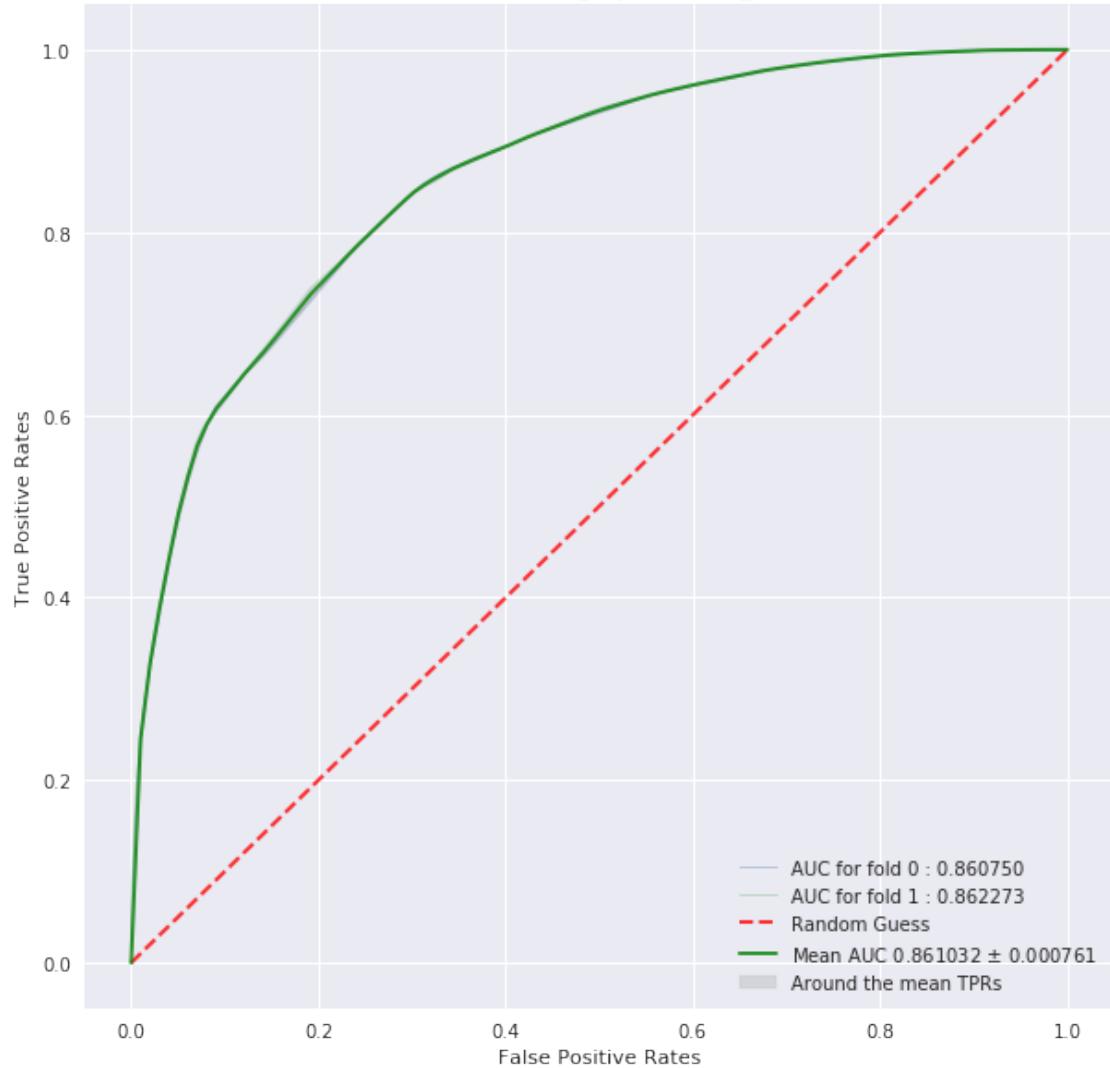
ROC - Train Ensemble (max\_depth:3, num\_estimators:60)



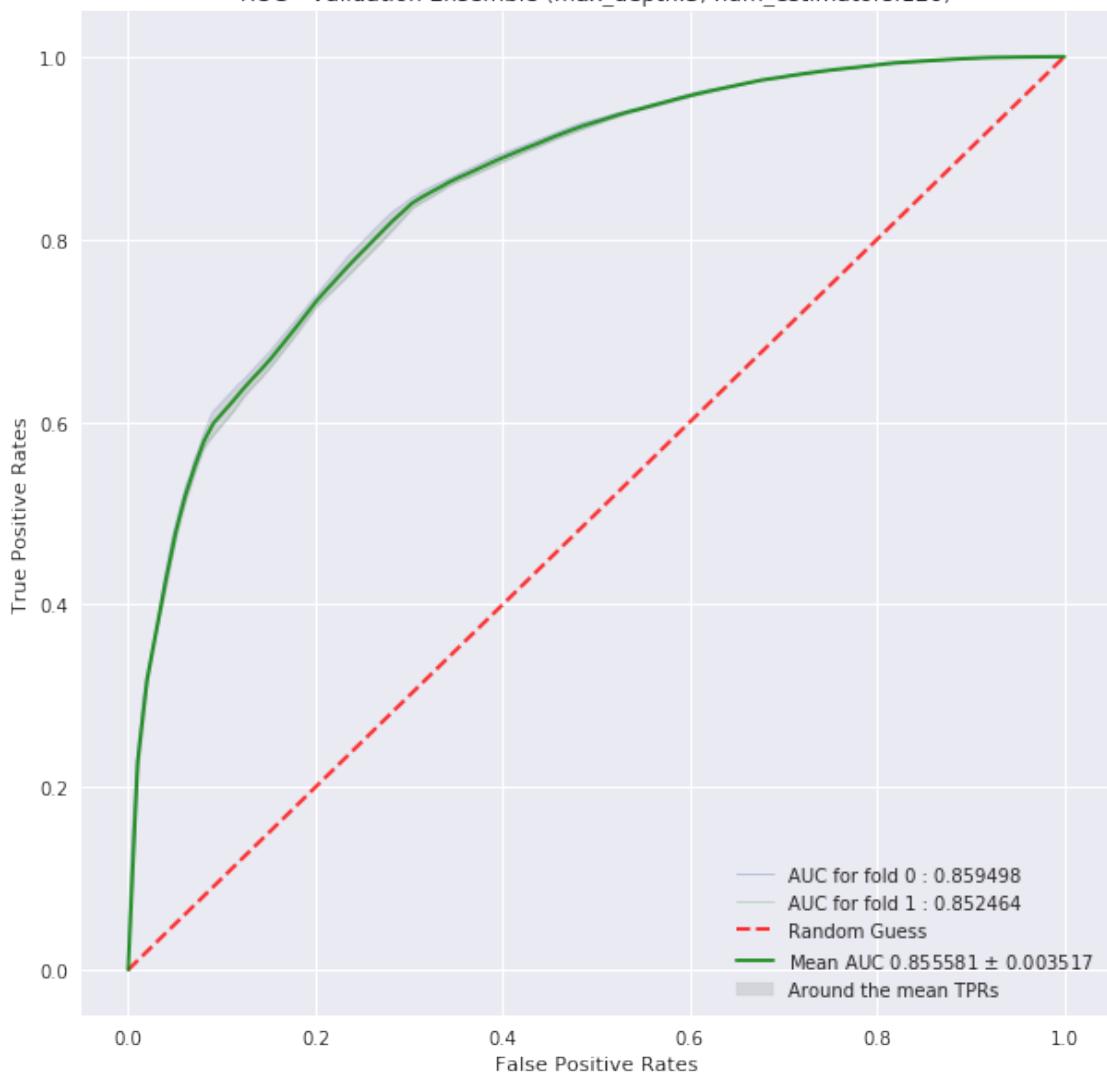
ROC - Validation Ensemble (max\_depth:3, num\_estimators:60)



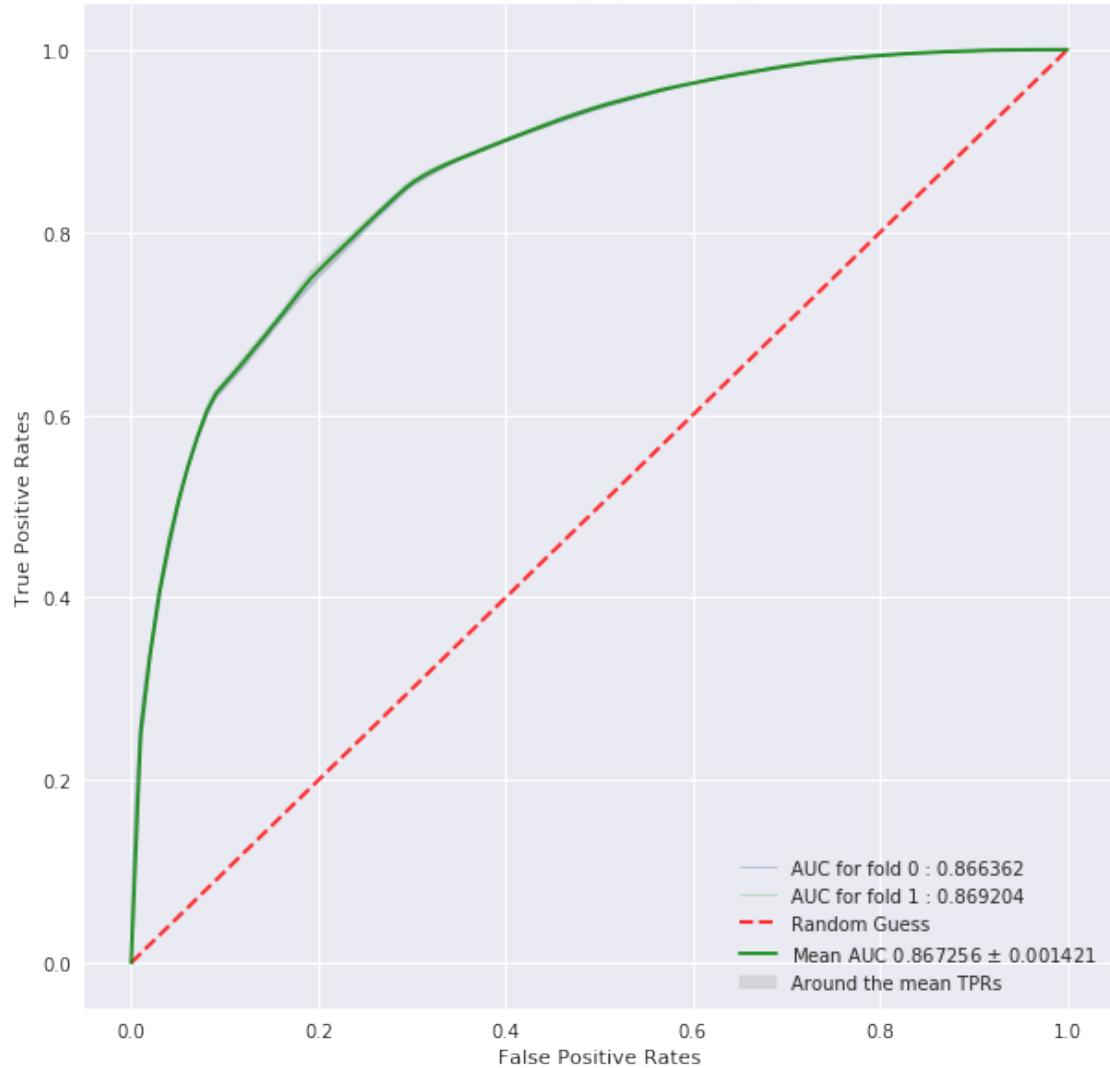
ROC - Train Ensemble (max\_depth:3, num\_estimators:120)



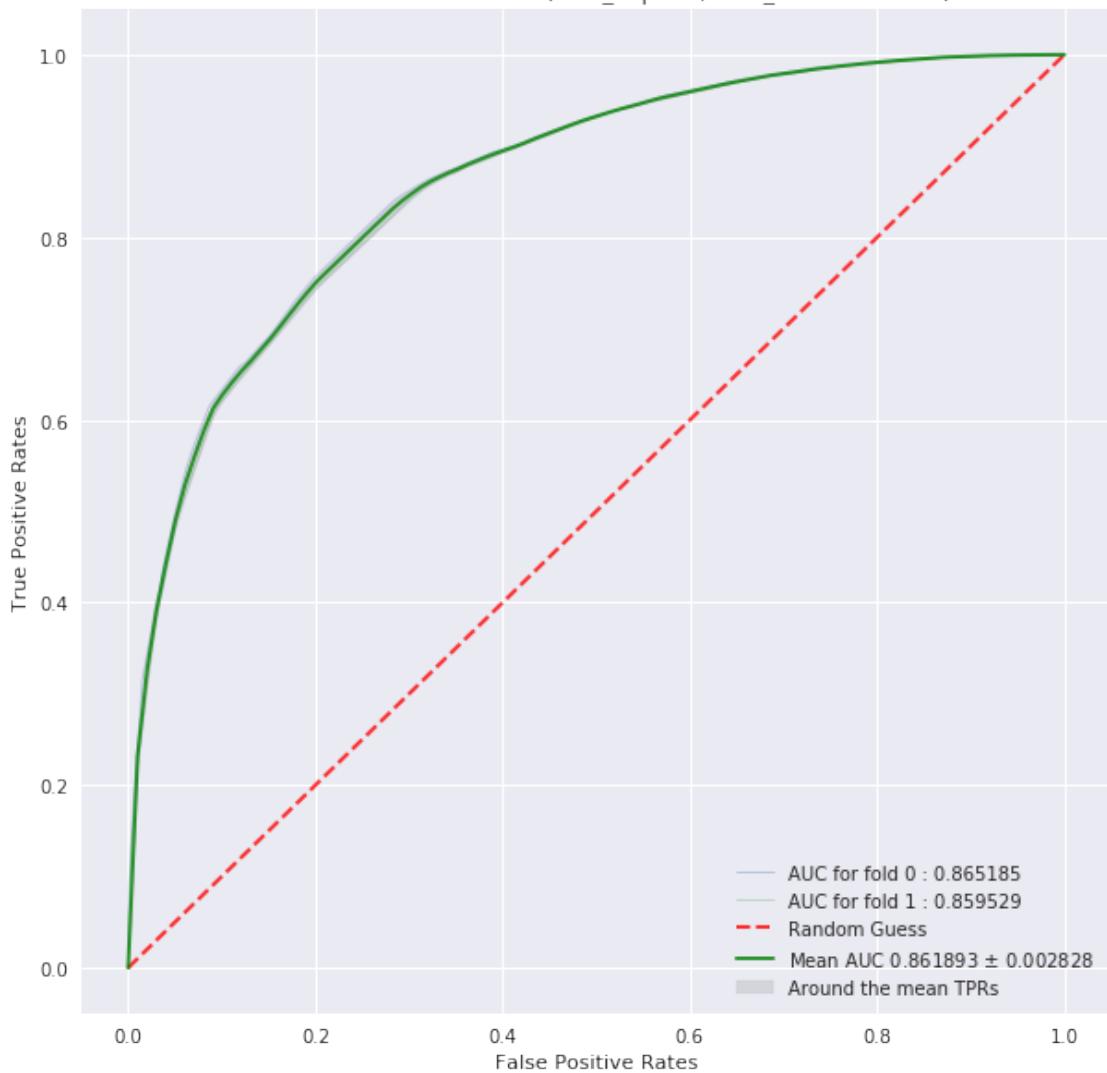
ROC - Validation Ensemble (max\_depth:3, num\_estimators:120)



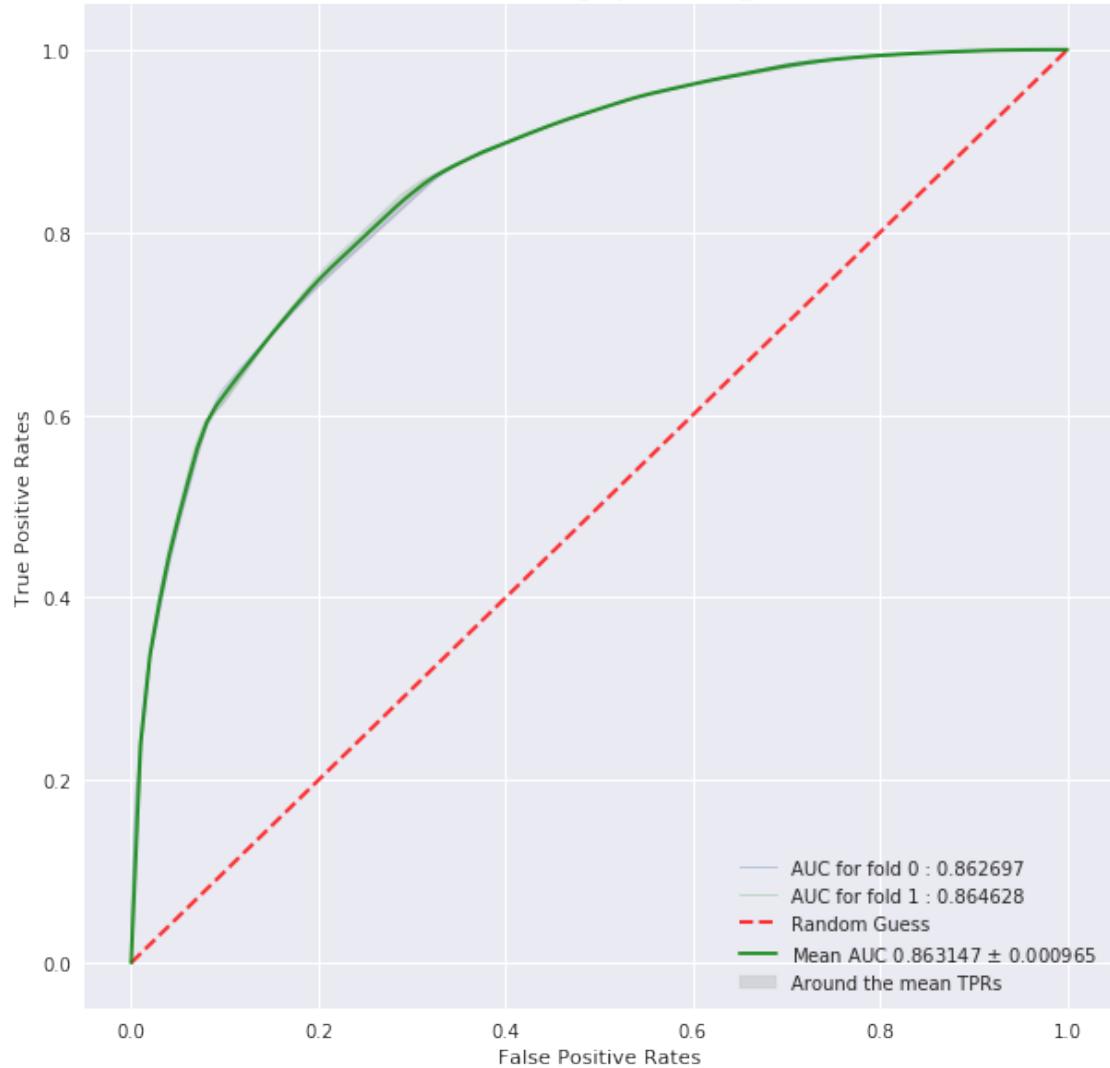
ROC - Train Ensemble (max\_depth:3, num\_estimators:300)



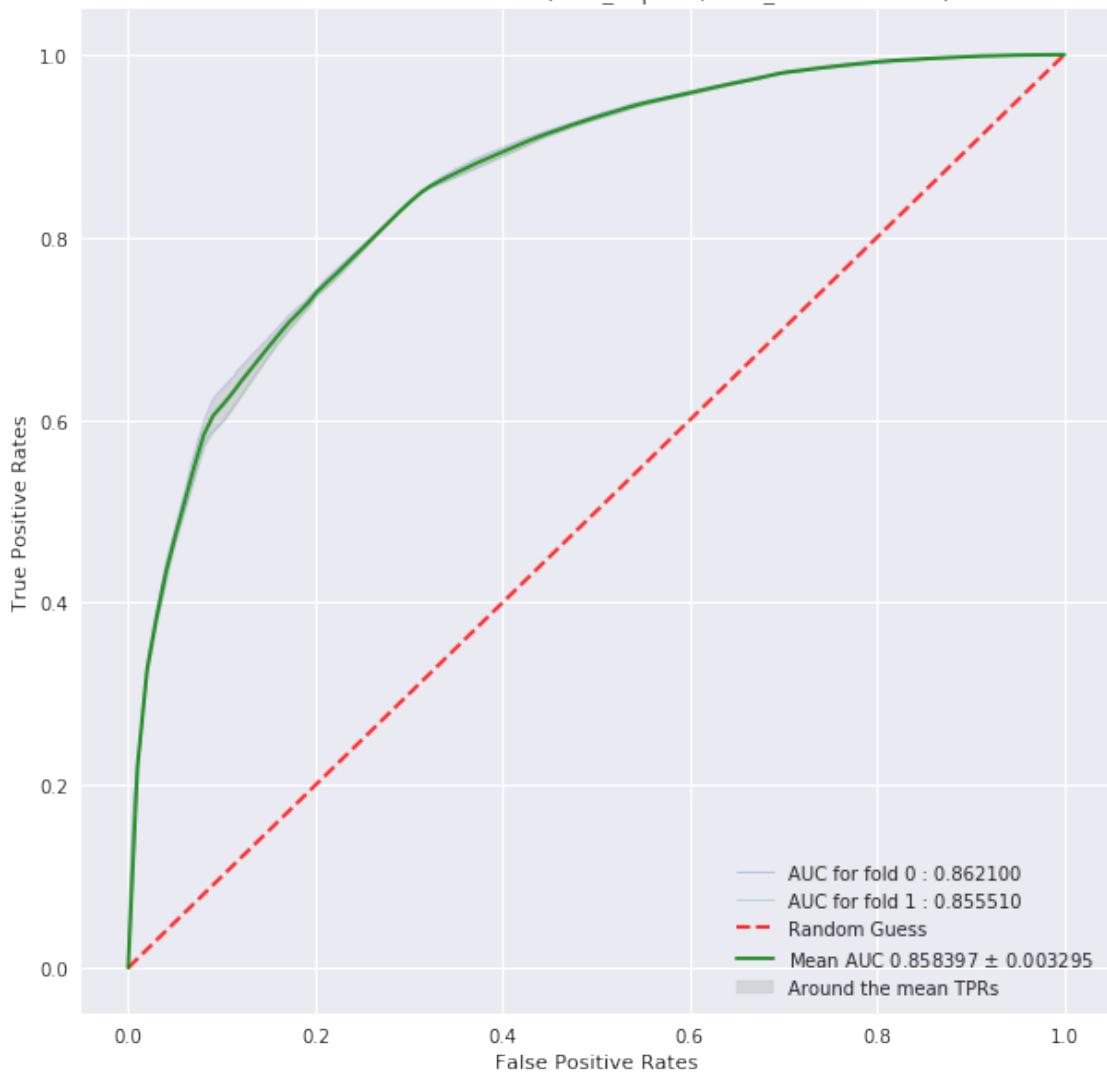
ROC - Validation Ensemble (max\_depth:3, num\_estimators:300)



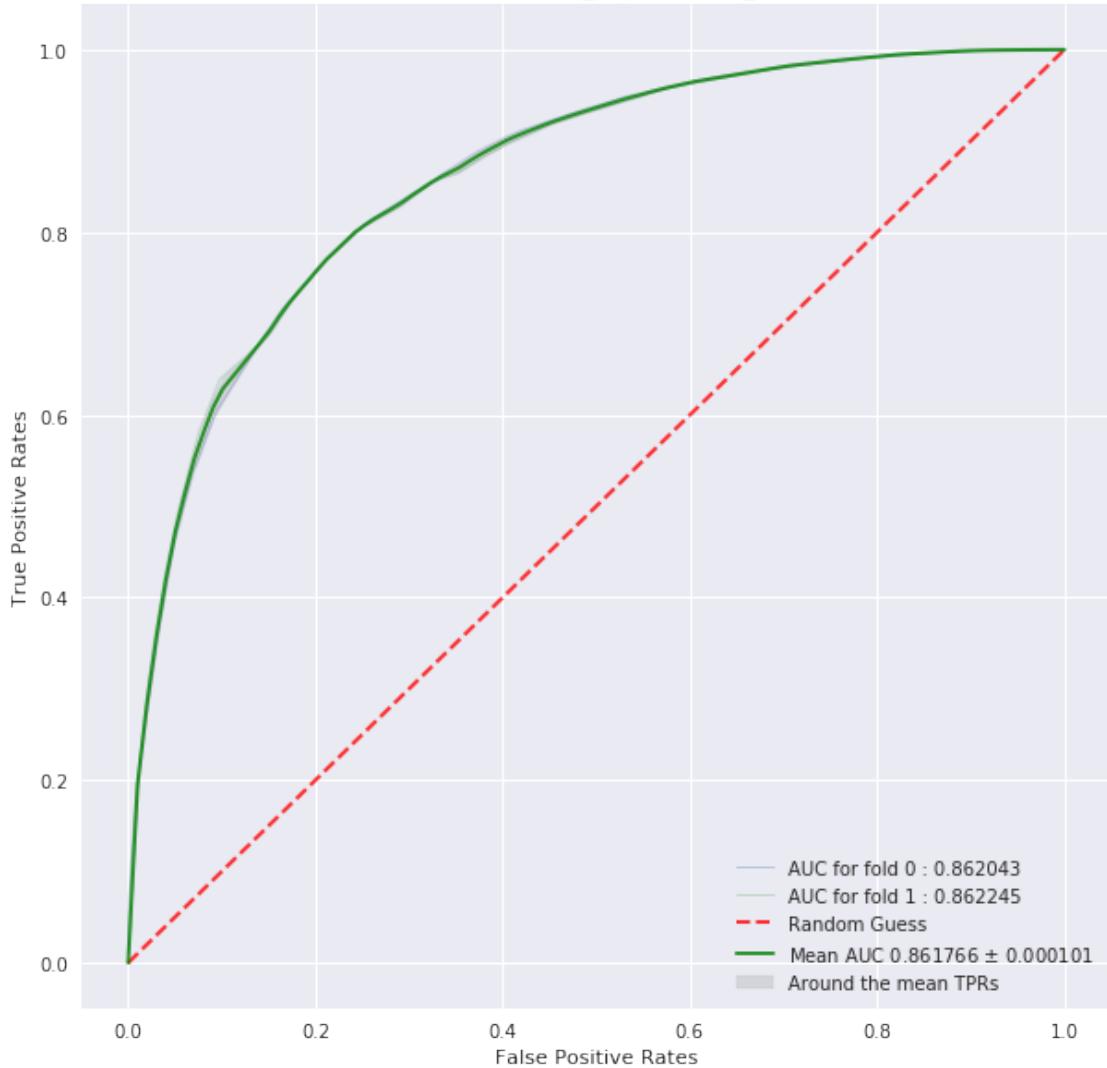
ROC - Train Ensemble (max\_depth:3, num\_estimators:500)



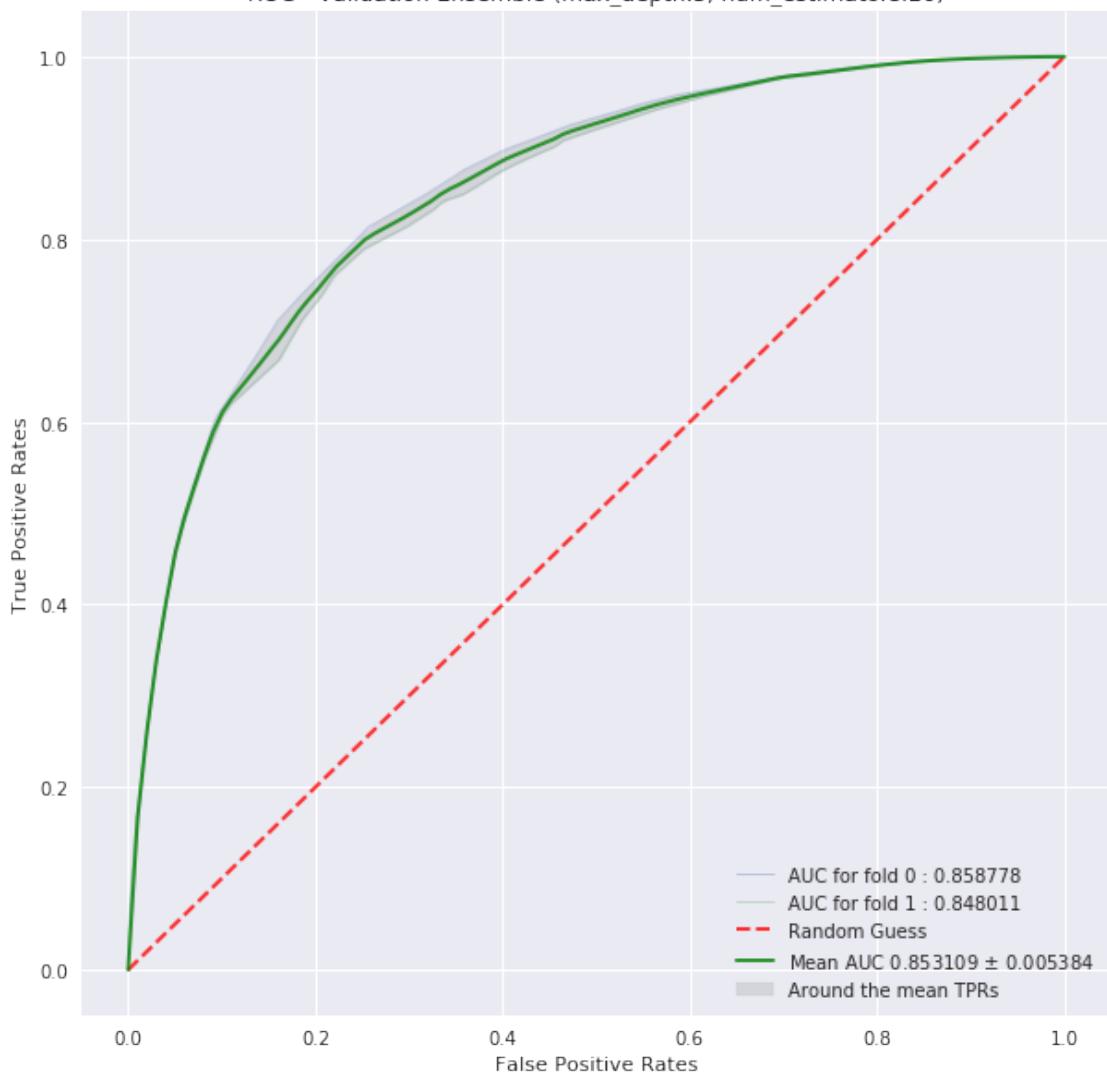
ROC - Validation Ensemble (max\_depth:3, num\_estimators:500)



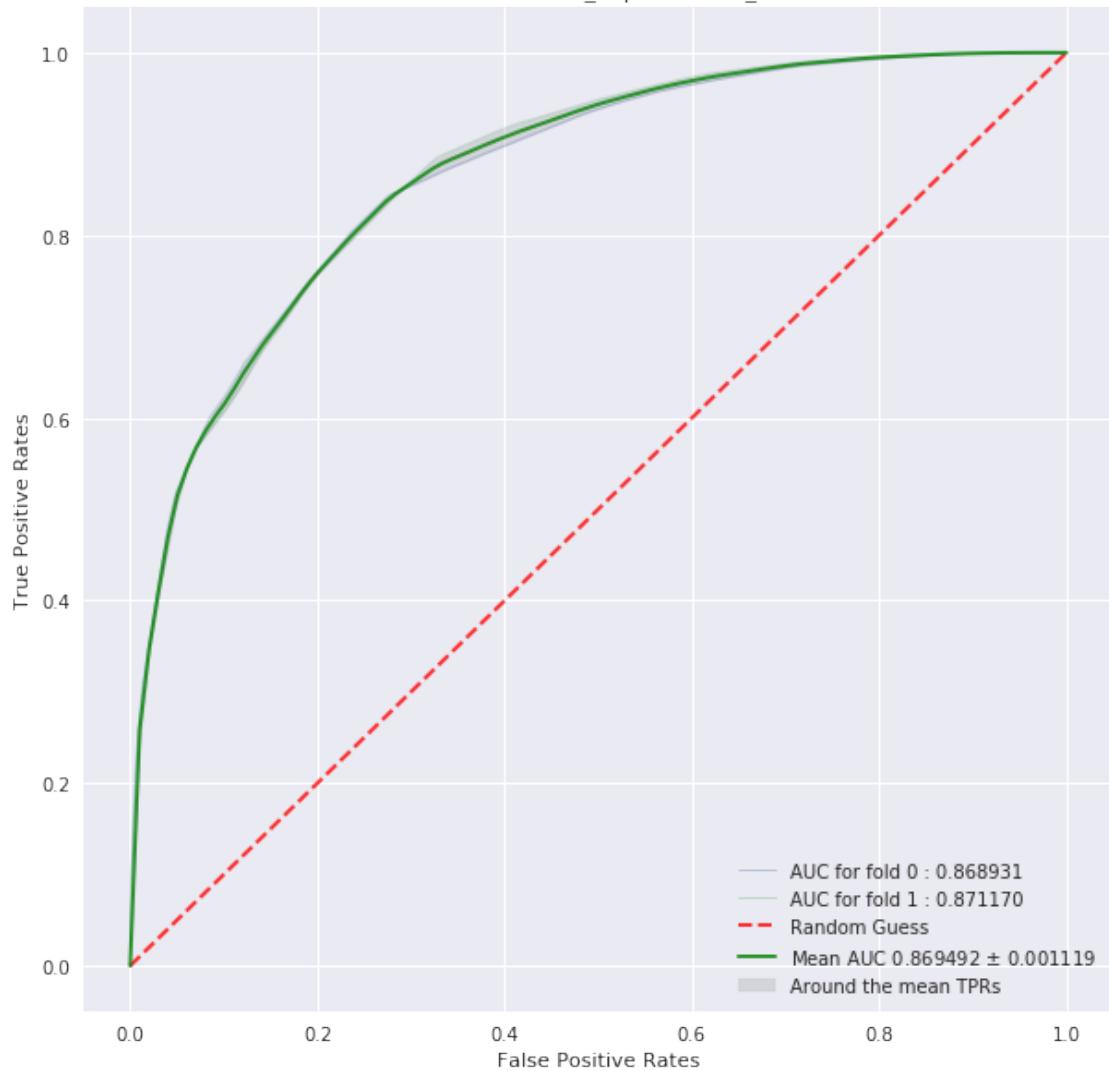
ROC - Train Ensemble (max\_depth:5, num\_estimators:20)



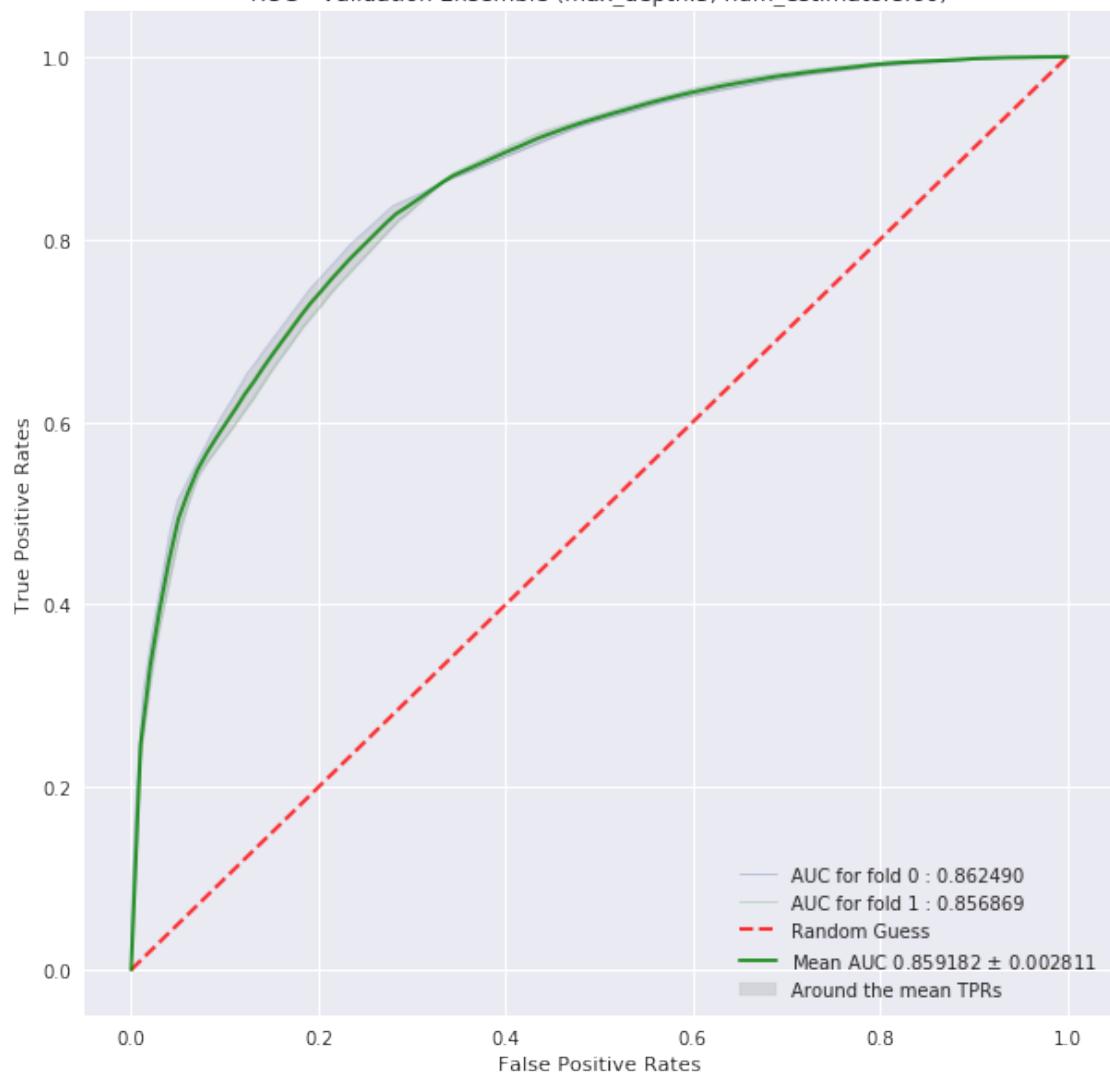
ROC - Validation Ensemble (max\_depth:5, num\_estimators:20)



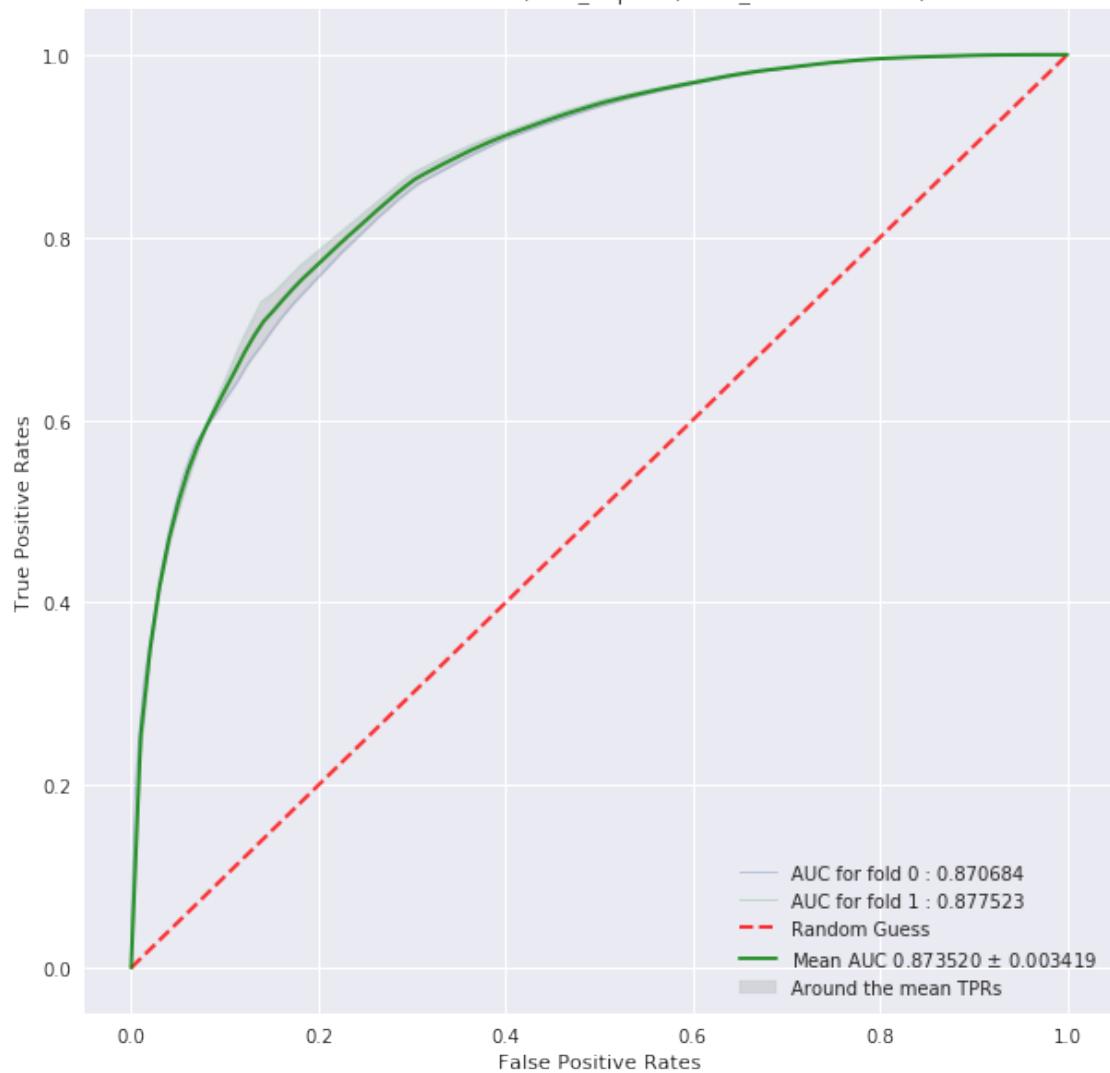
ROC - Train Ensemble (max\_depth:5, num\_estimators:60)



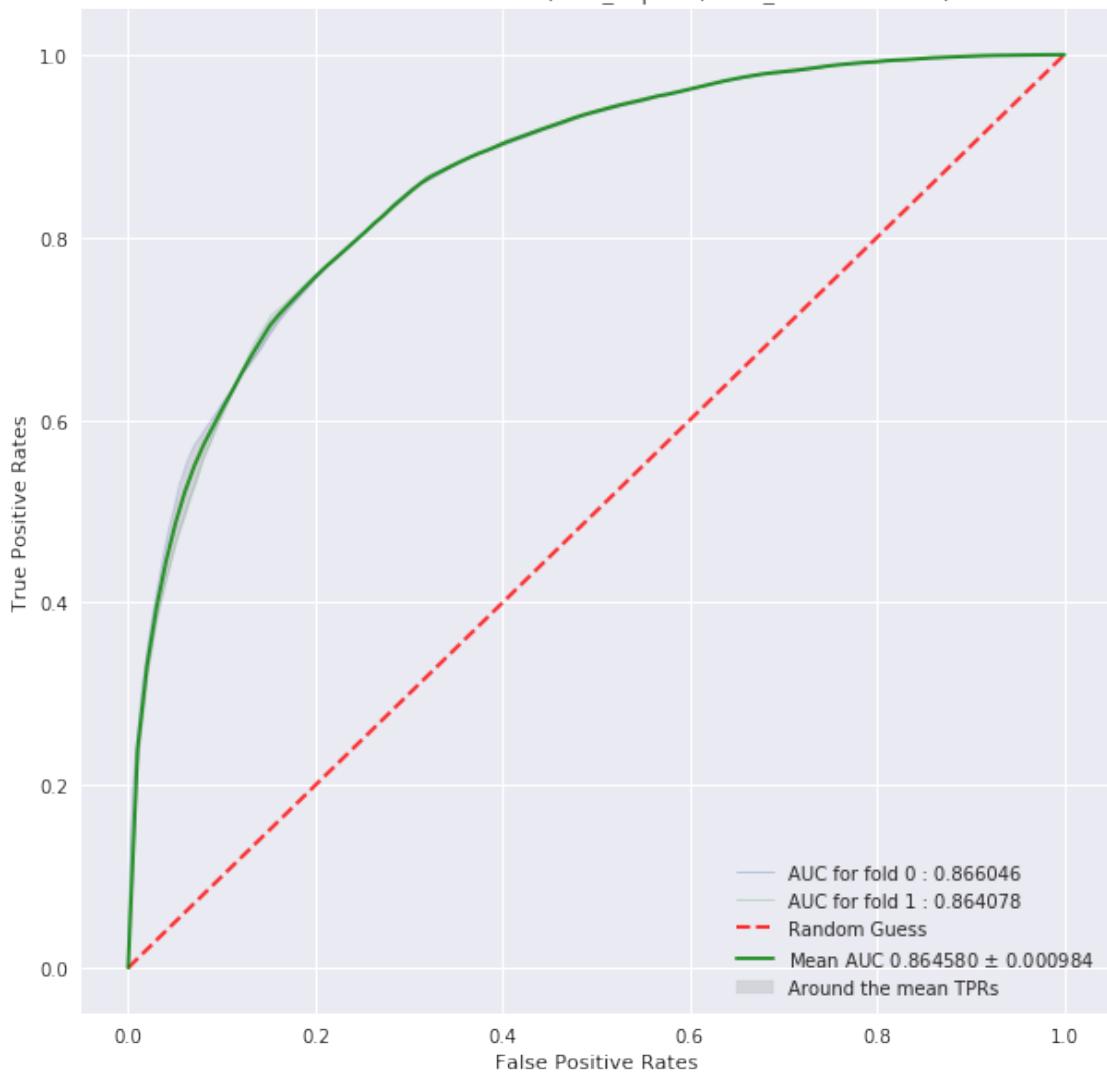
ROC - Validation Ensemble (max\_depth:5, num\_estimators:60)



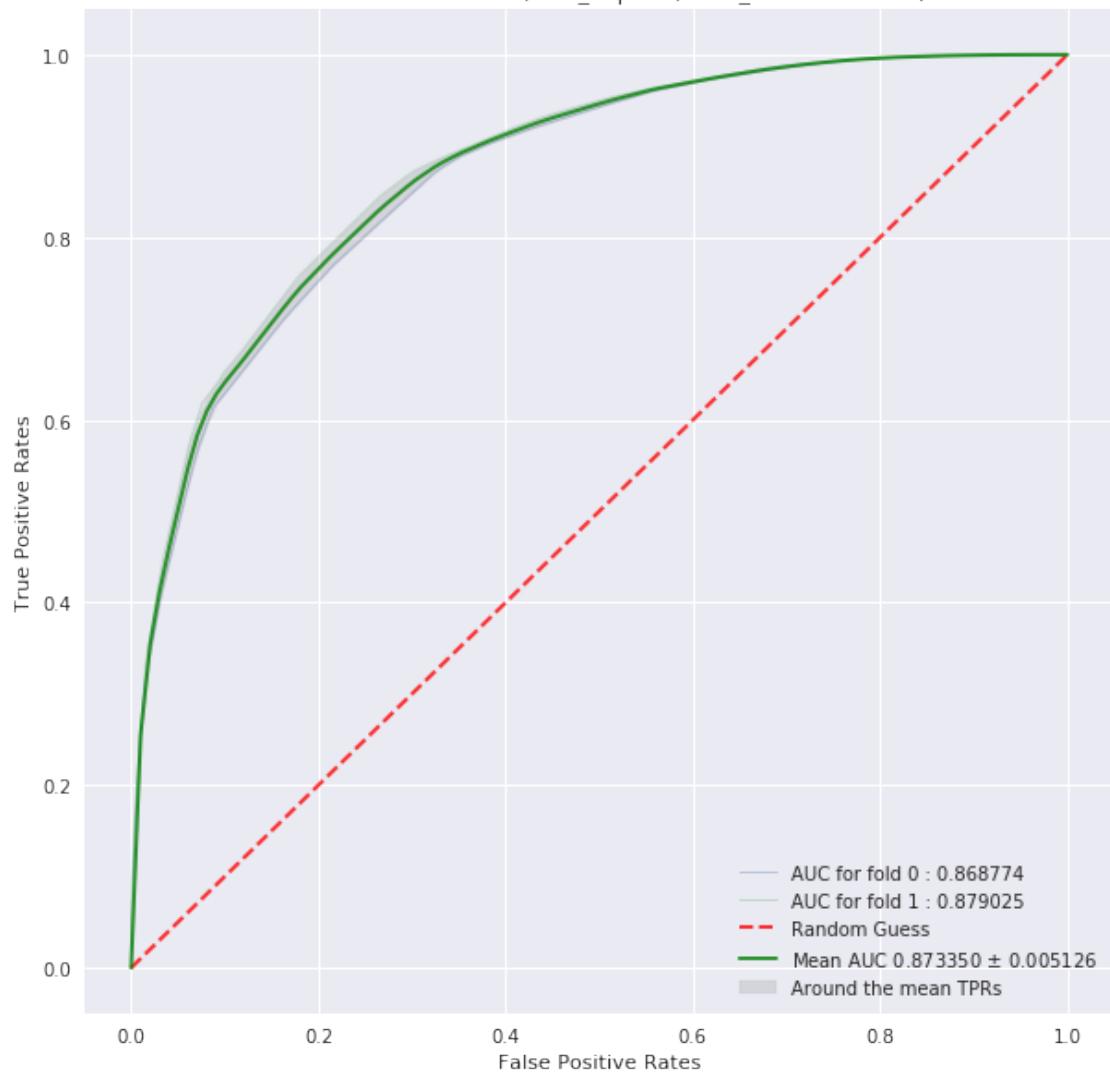
ROC - Train Ensemble (max\_depth:5, num\_estimators:120)



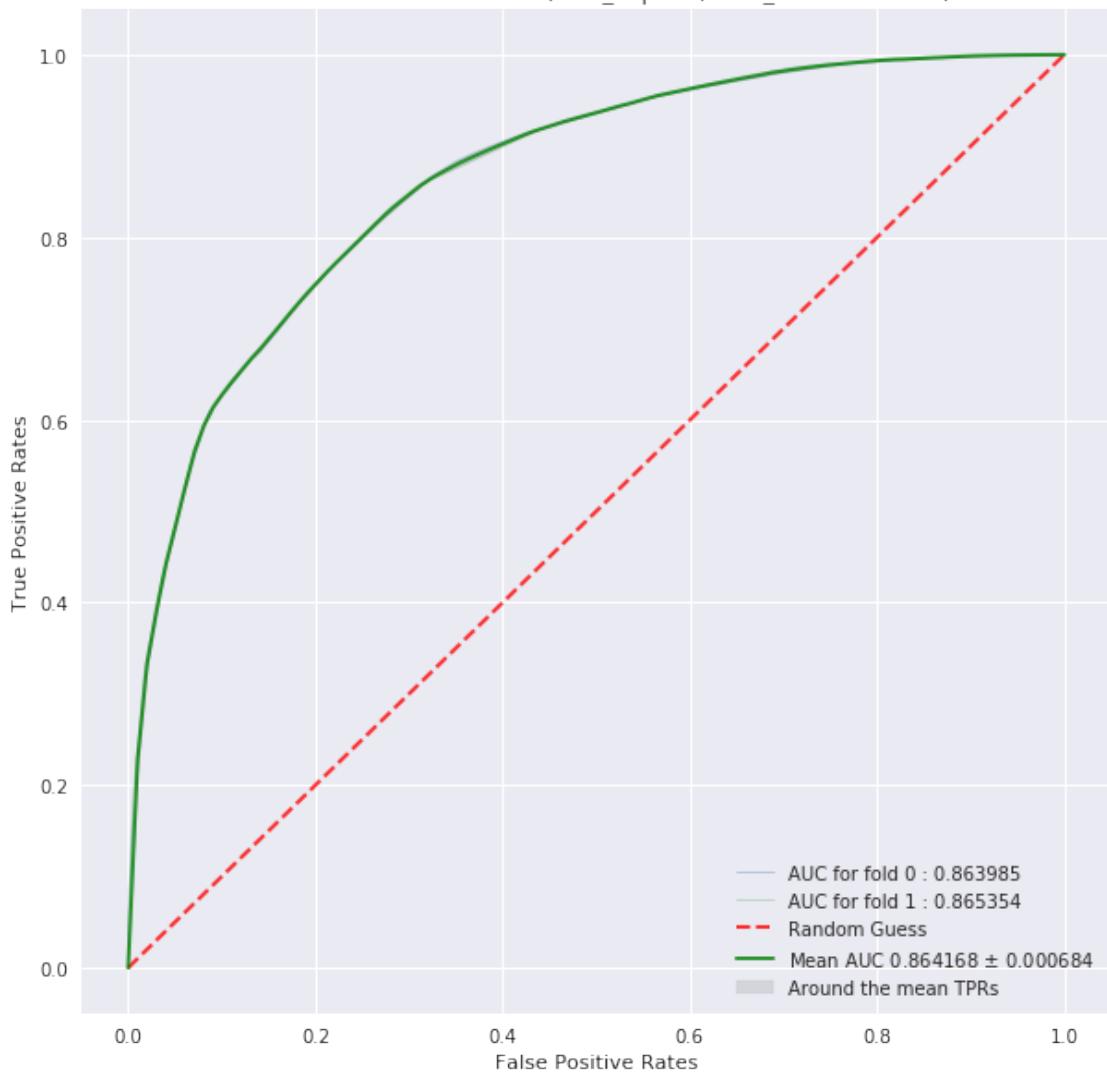
ROC - Validation Ensemble (max\_depth:5, num\_estimators:120)



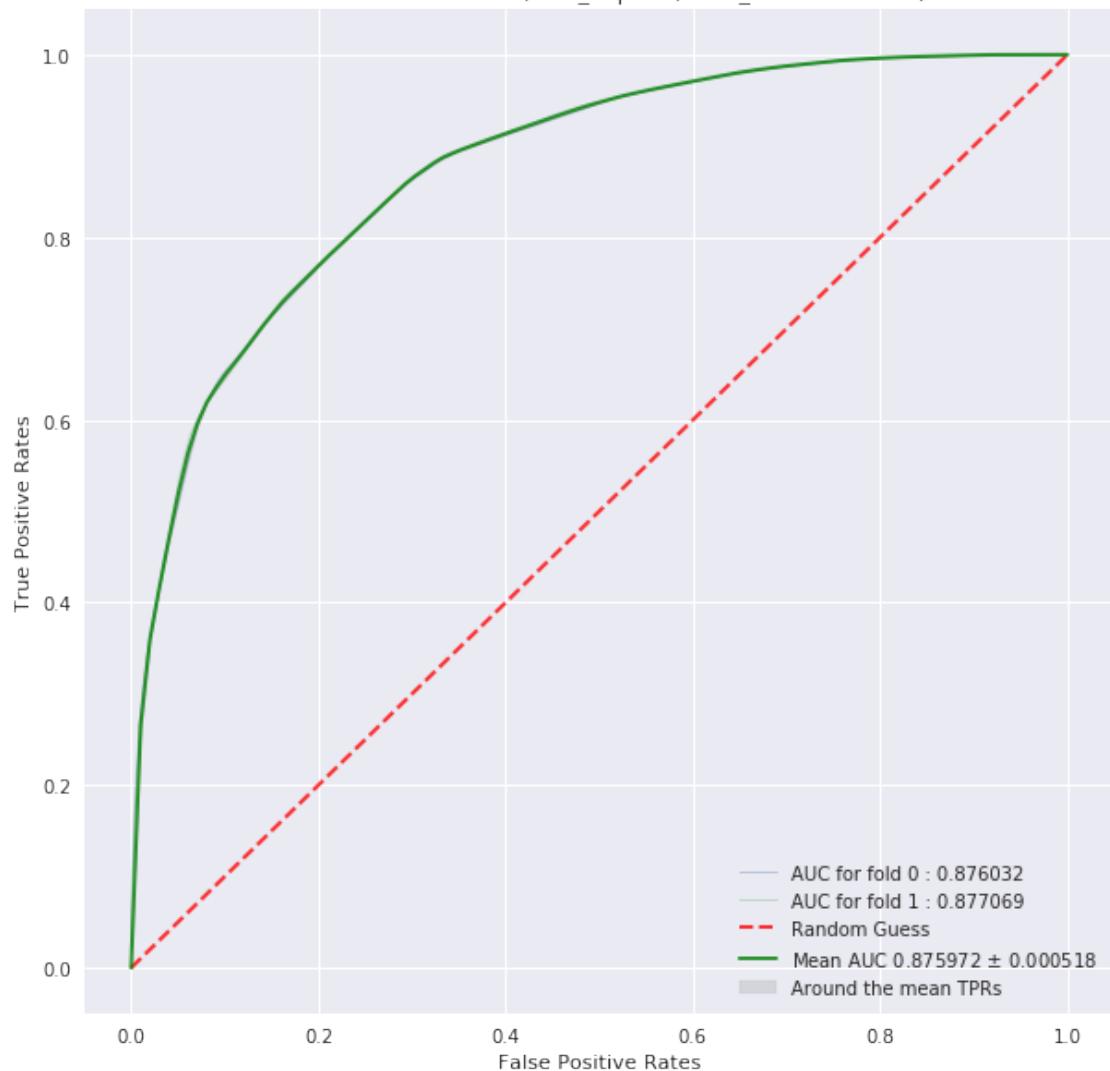
ROC - Train Ensemble (max\_depth:5, num\_estimators:300)



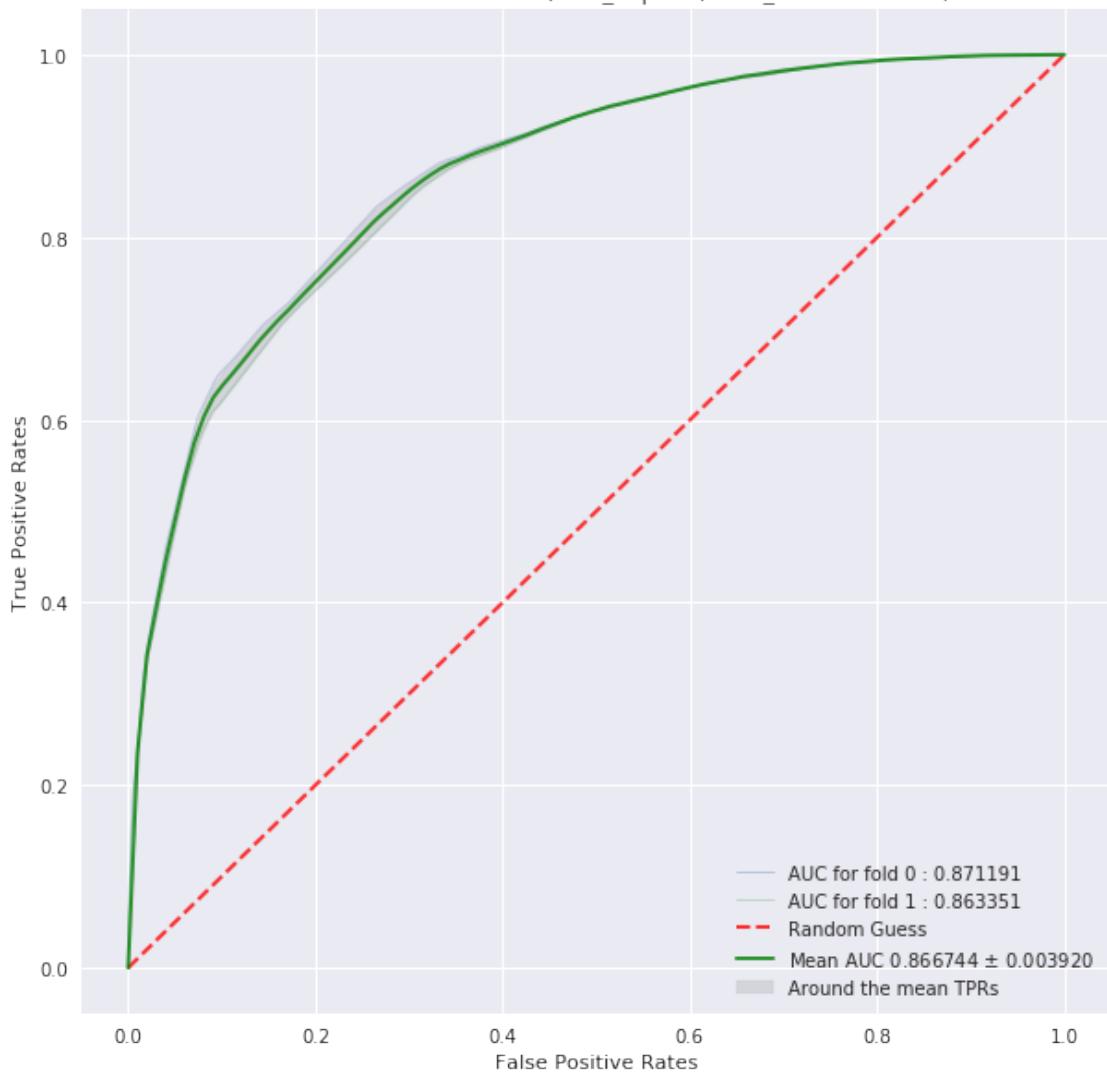
ROC - Validation Ensemble (max\_depth:5, num\_estimators:300)



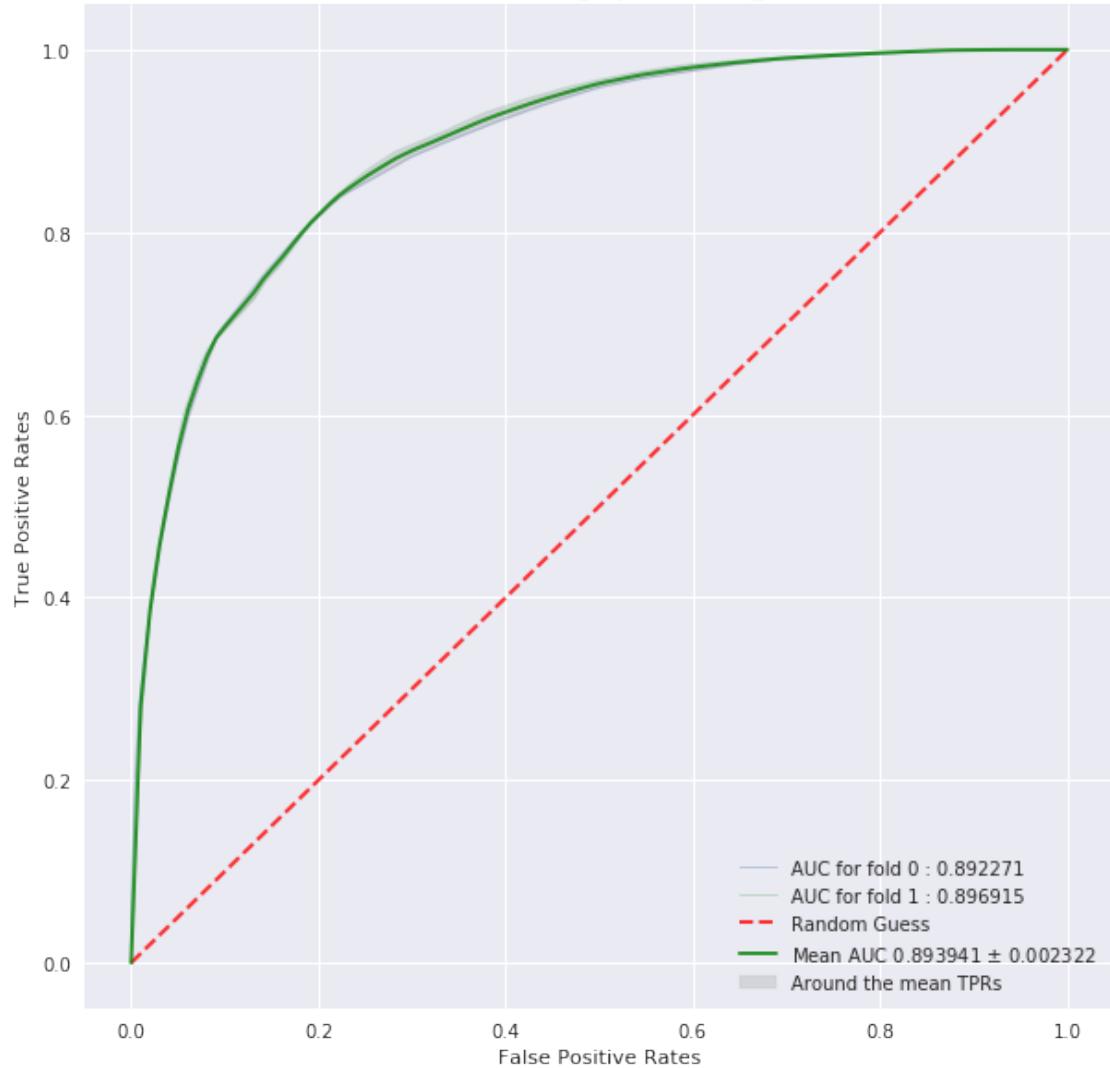
ROC - Train Ensemble (max\_depth:5, num\_estimators:500)



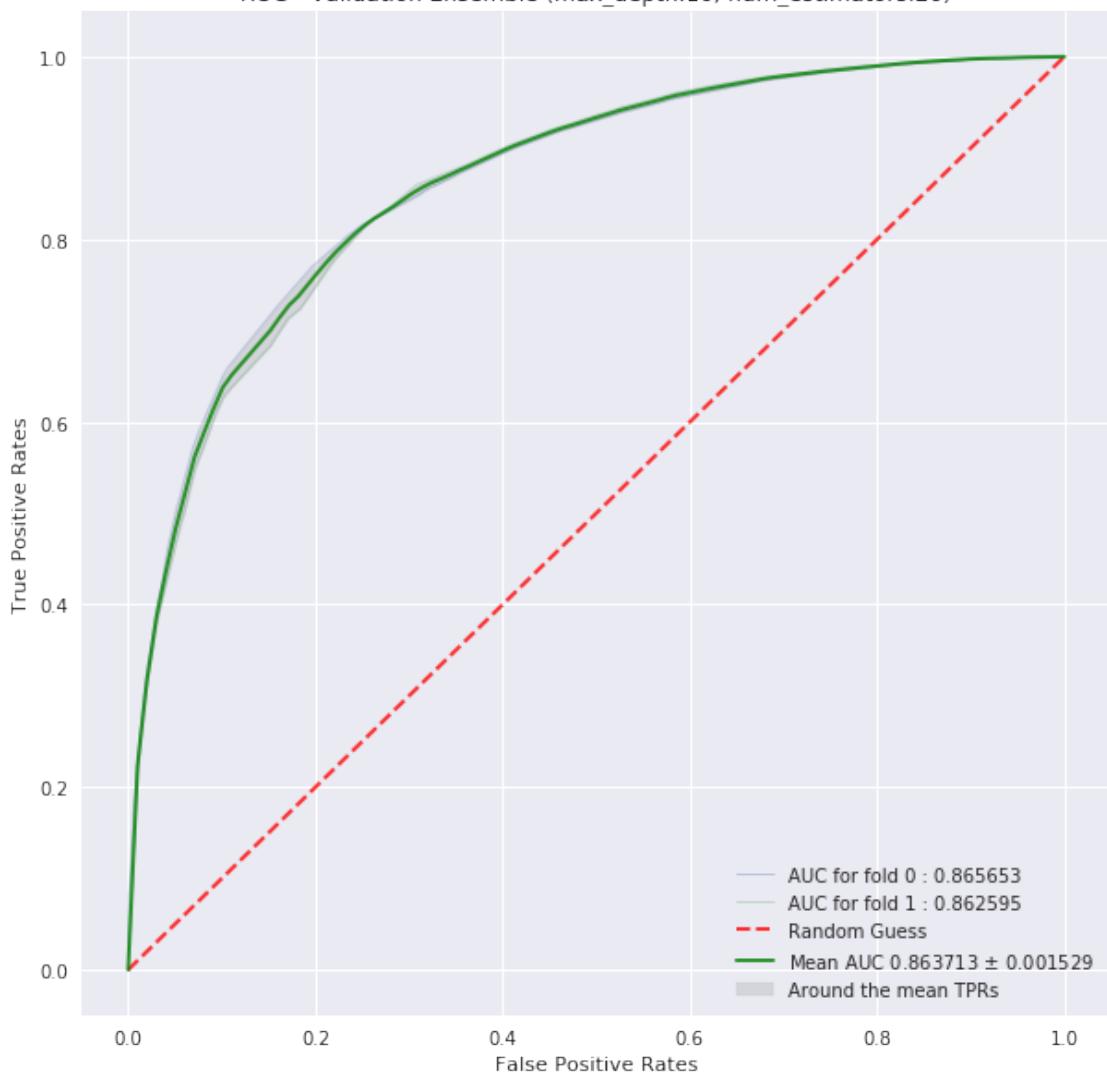
ROC - Validation Ensemble (max\_depth:5, num\_estimators:500)



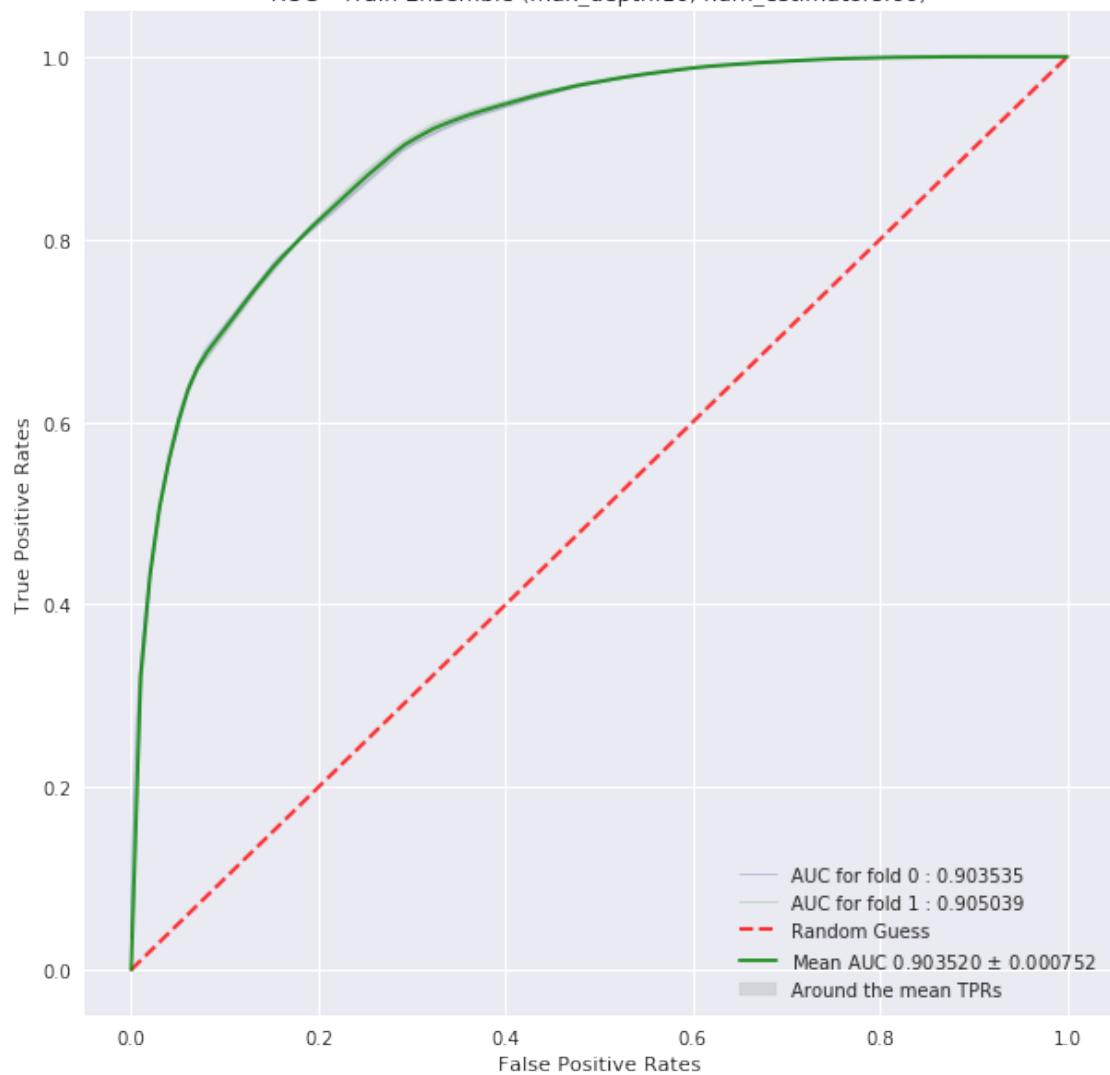
ROC - Train Ensemble (max\_depth:10, num\_estimators:20)



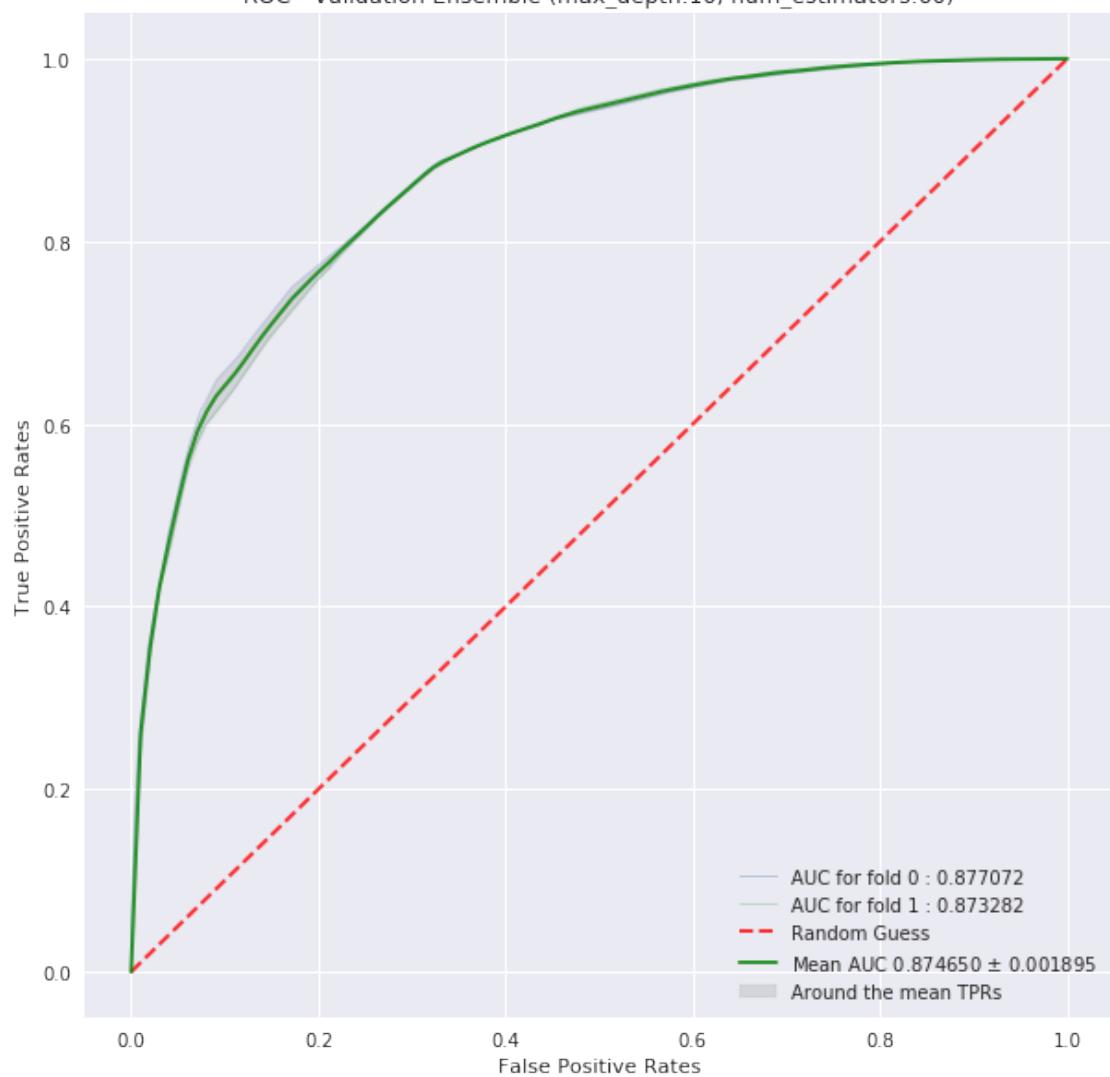
ROC - Validation Ensemble (max\_depth:10, num\_estimators:20)



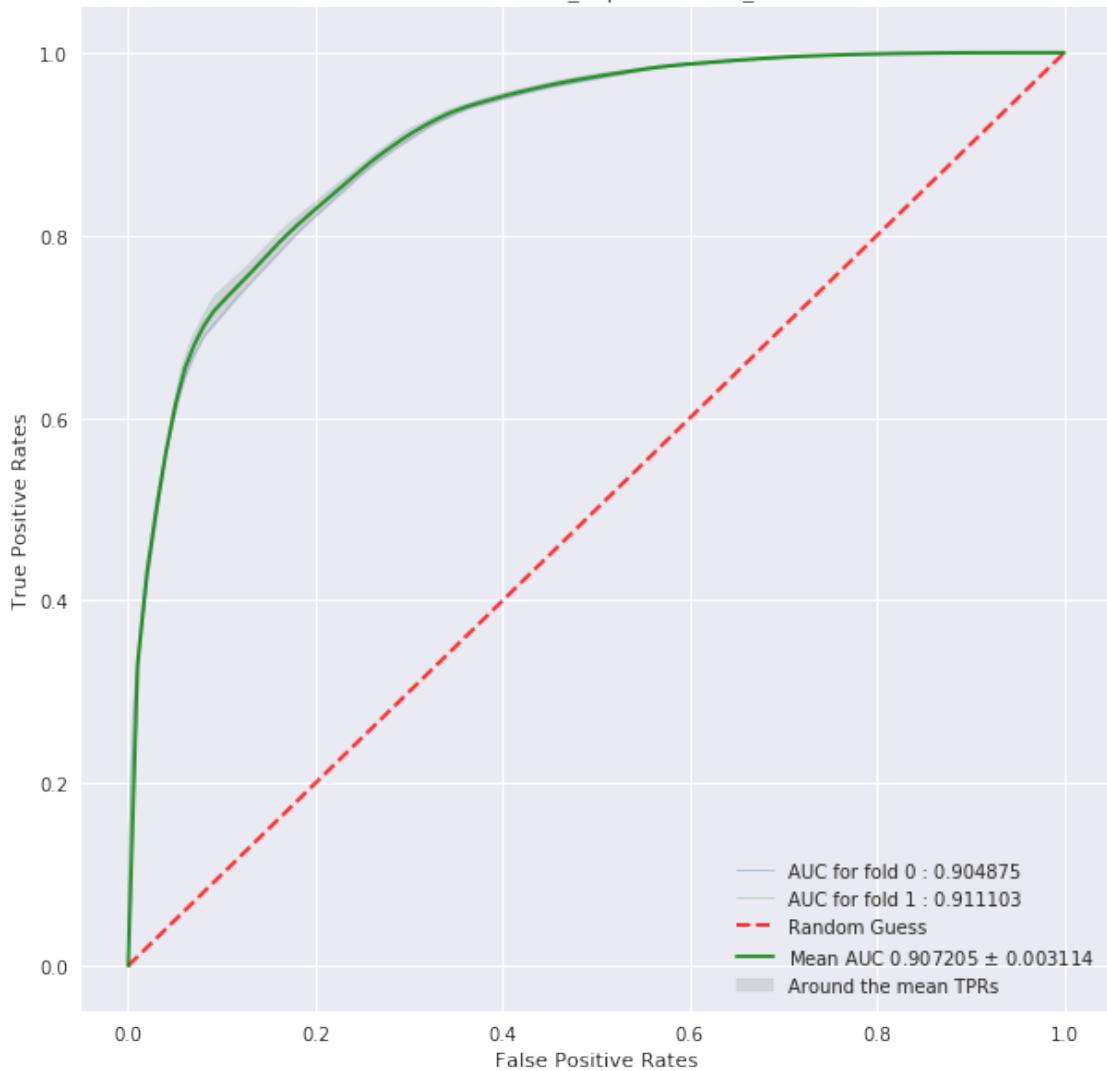
ROC - Train Ensemble (max\_depth:10, num\_estimators:60)



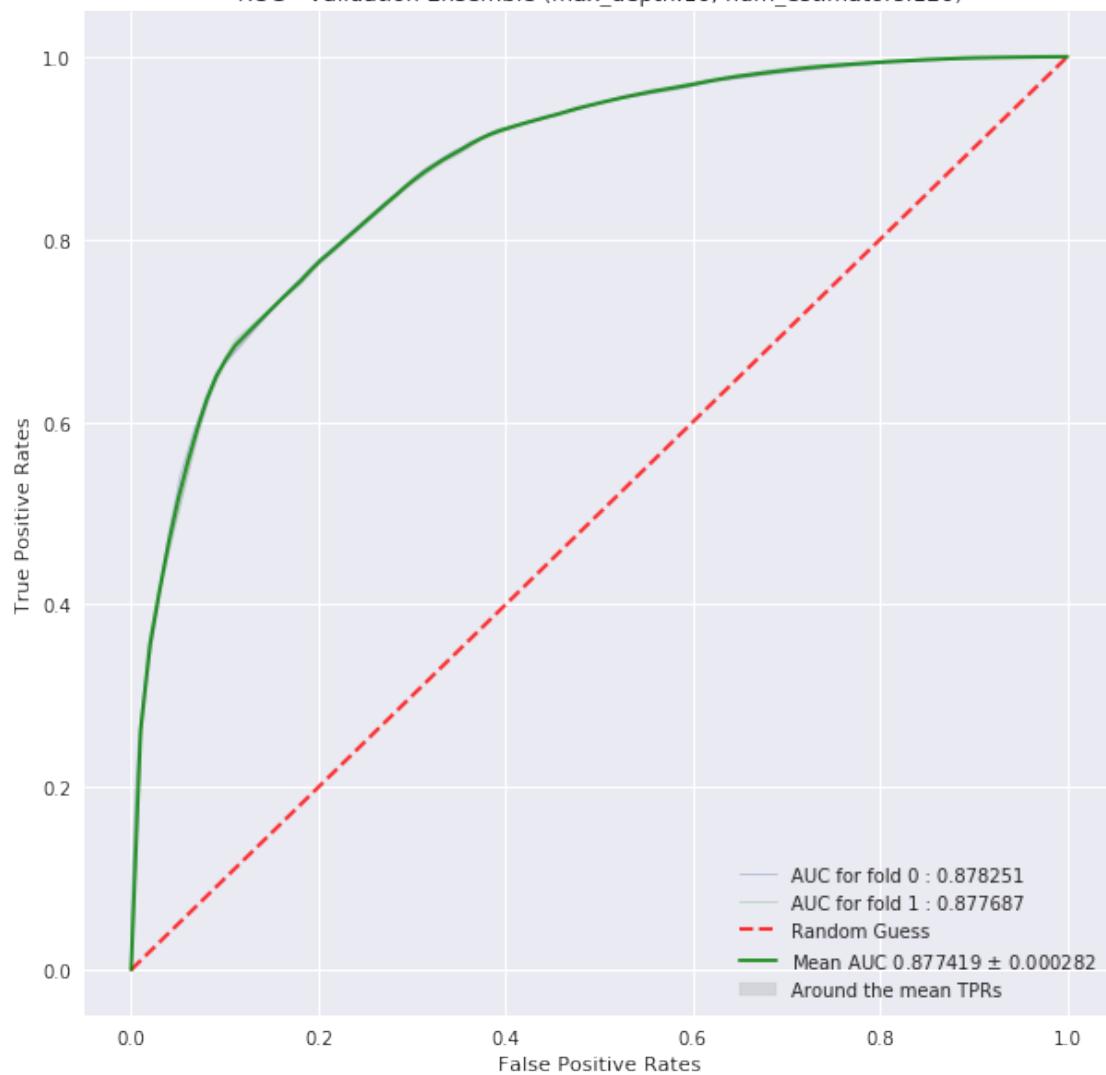
ROC - Validation Ensemble (max\_depth:10, num\_estimators:60)



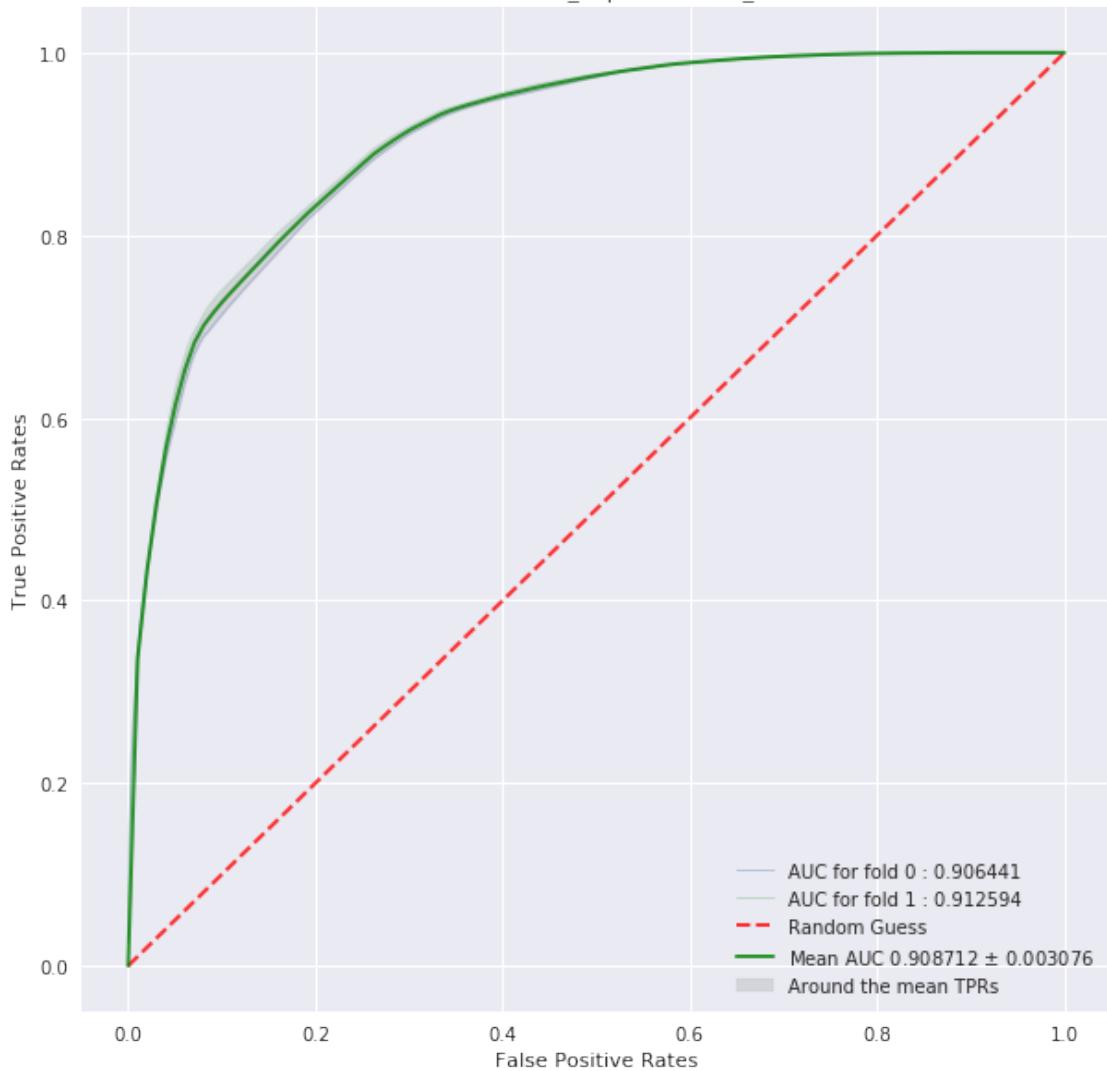
ROC - Train Ensemble (max\_depth:10, num\_estimators:120)



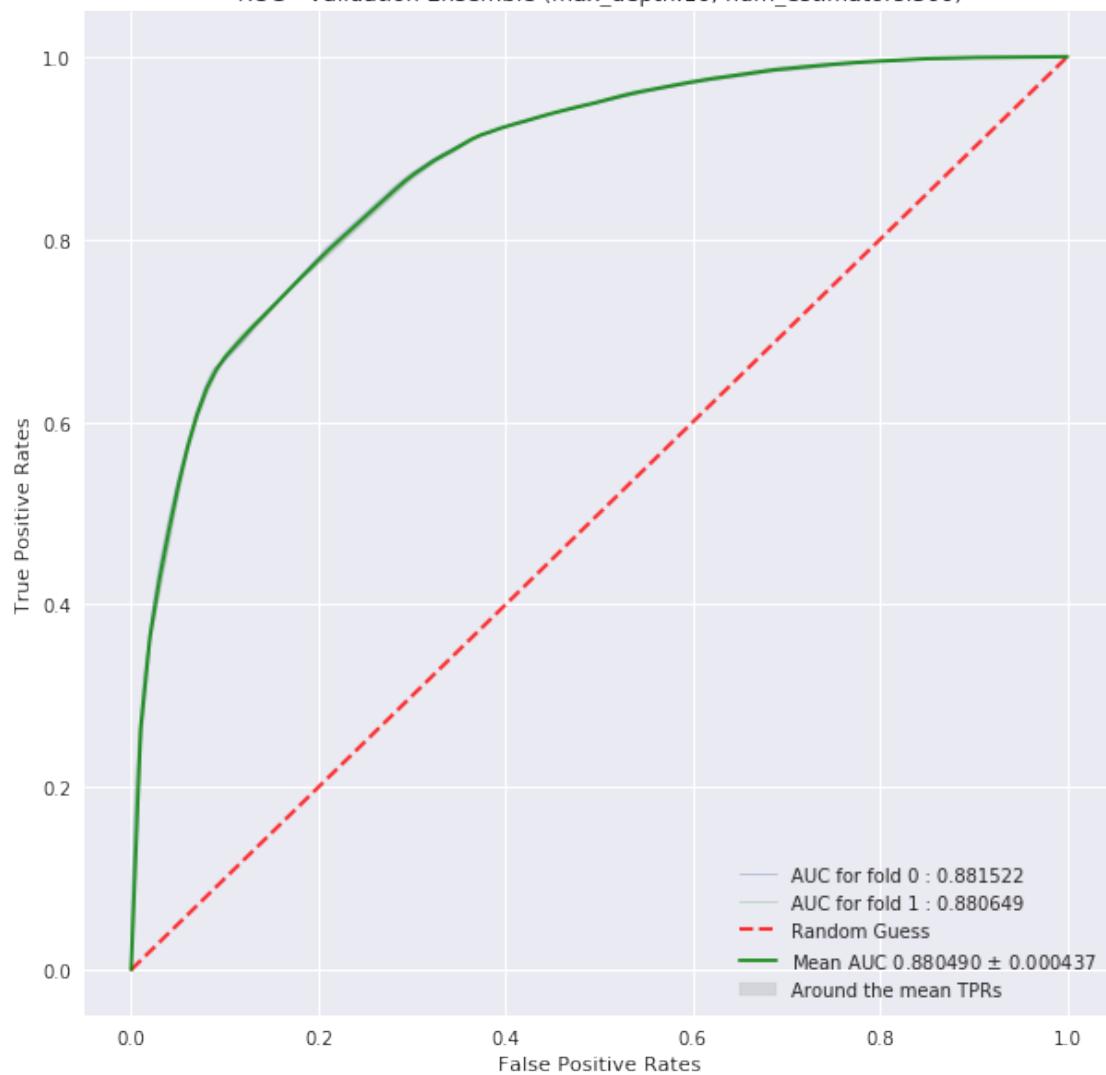
ROC - Validation Ensemble (max\_depth:10, num\_estimators:120)



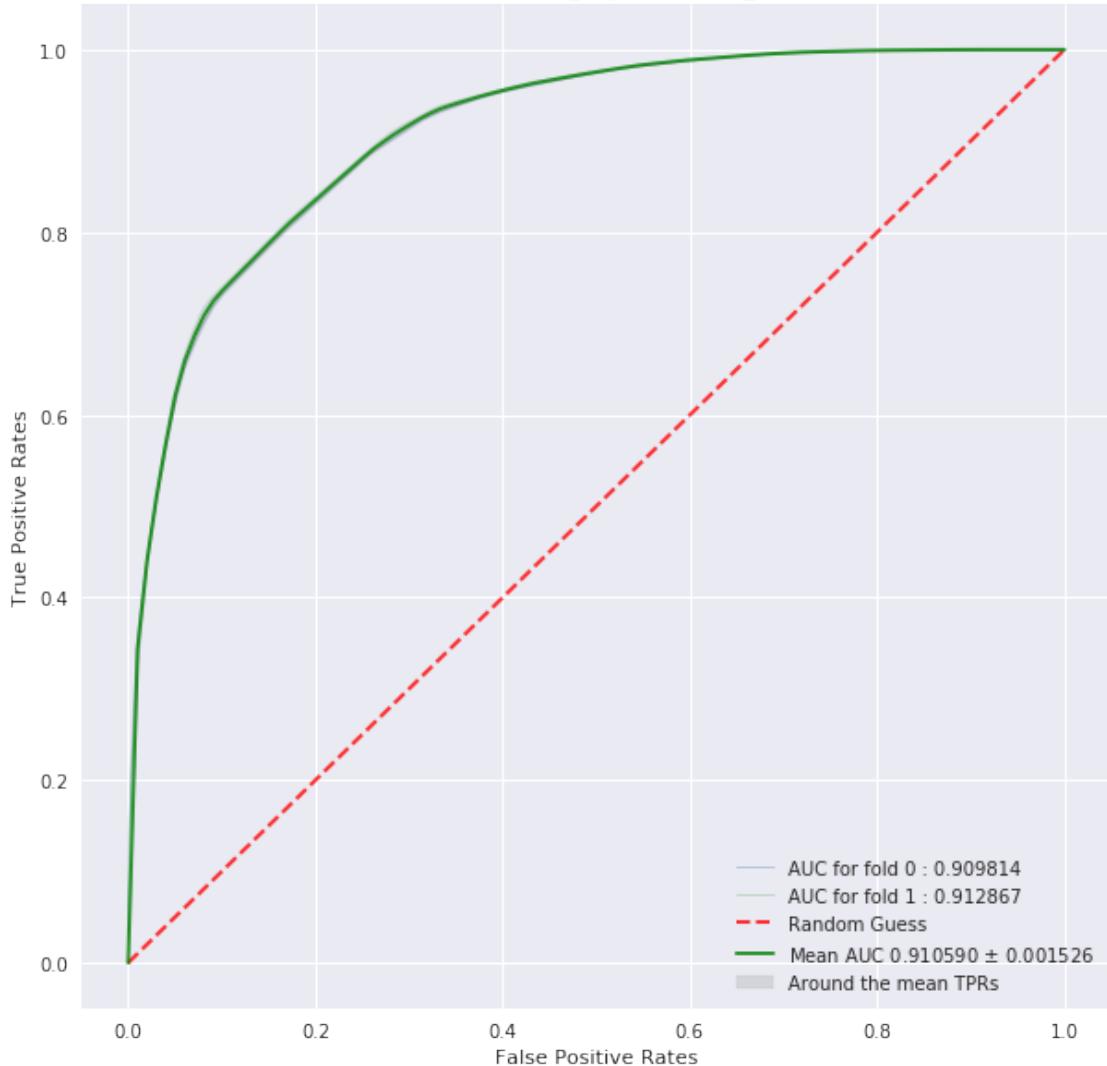
ROC - Train Ensemble (max\_depth:10, num\_estimators:300)



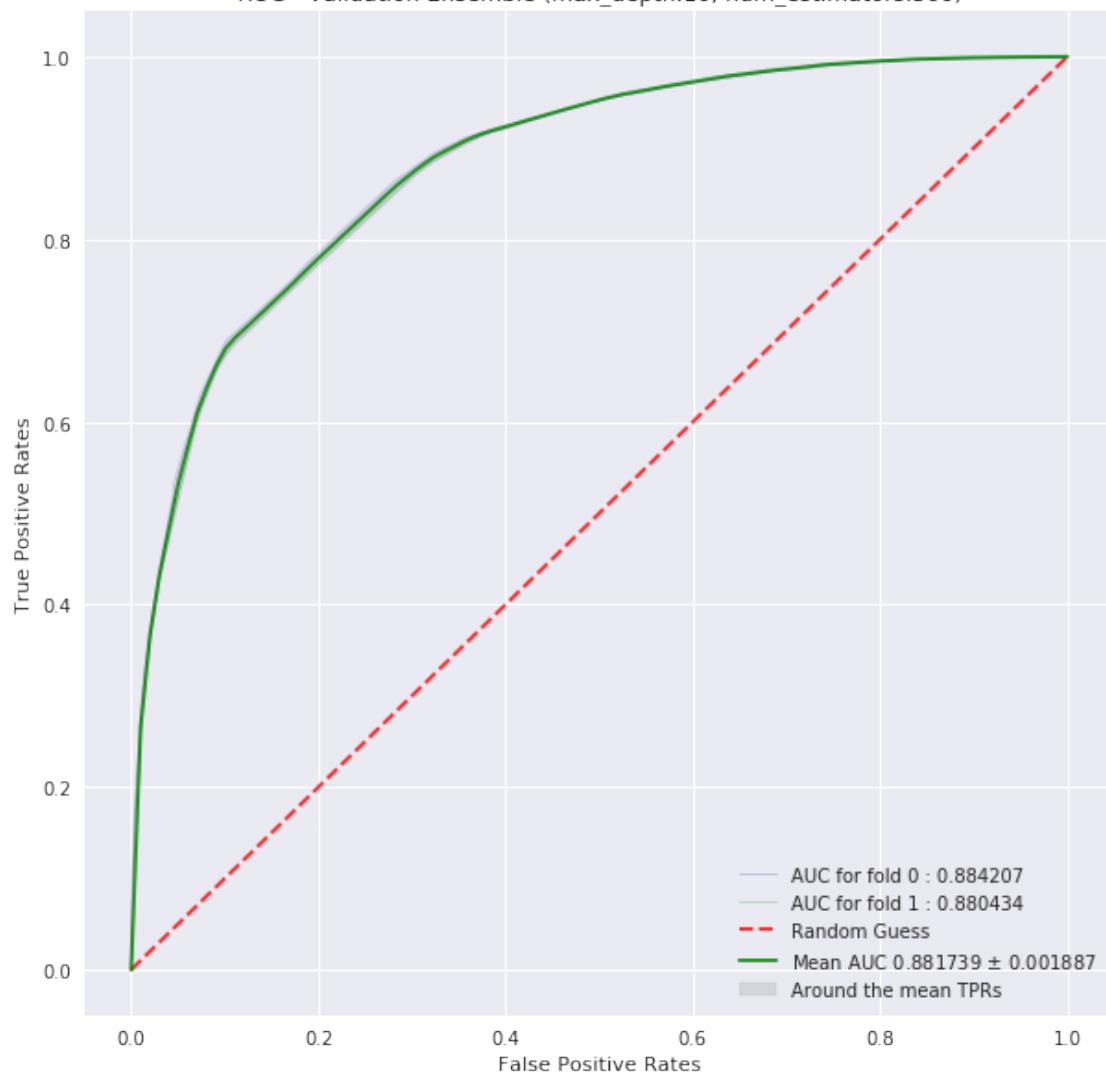
ROC - Validation Ensemble (max\_depth:10, num\_estimators:300)

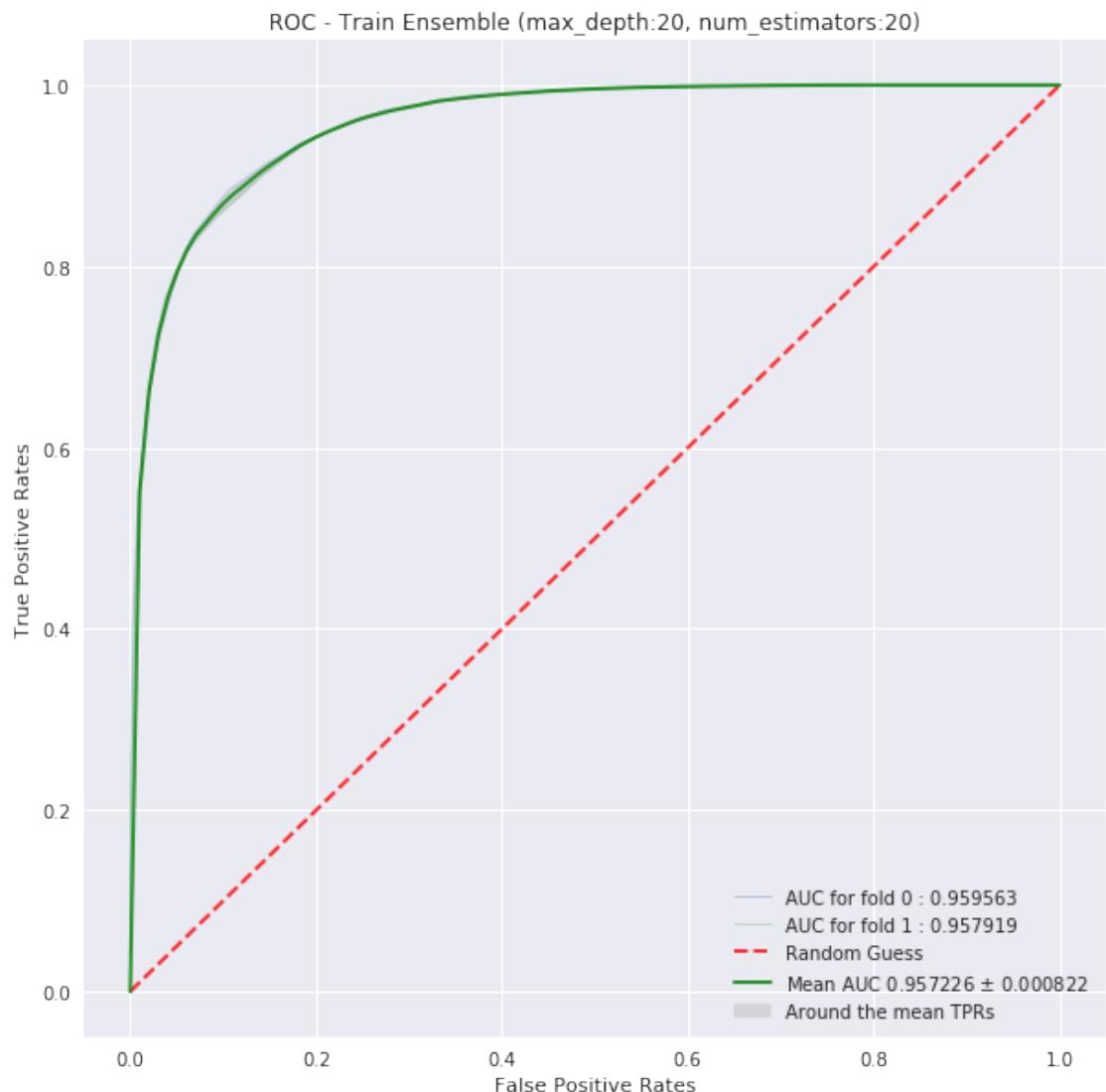


ROC - Train Ensemble (max\_depth:10, num\_estimators:500)

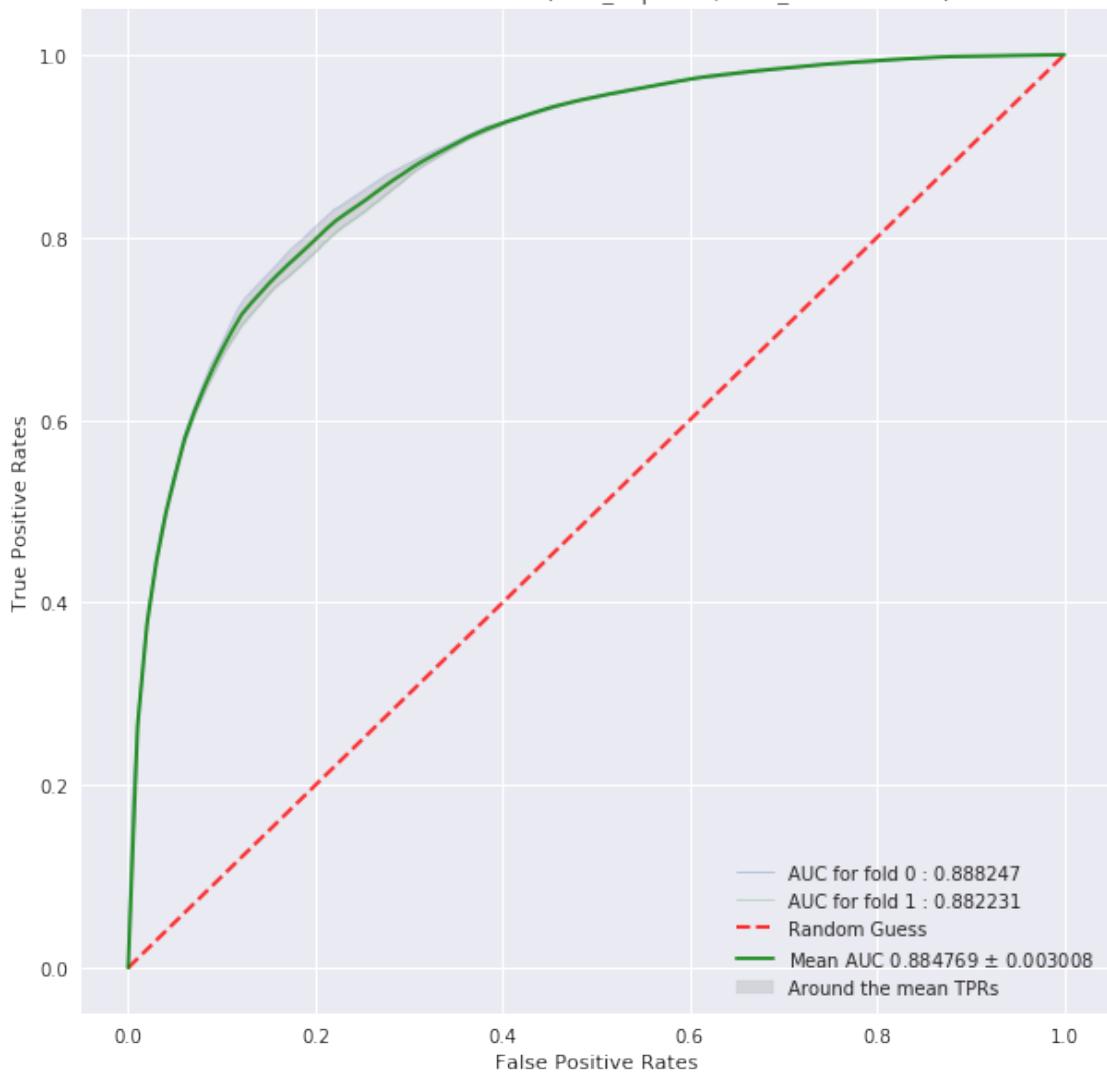


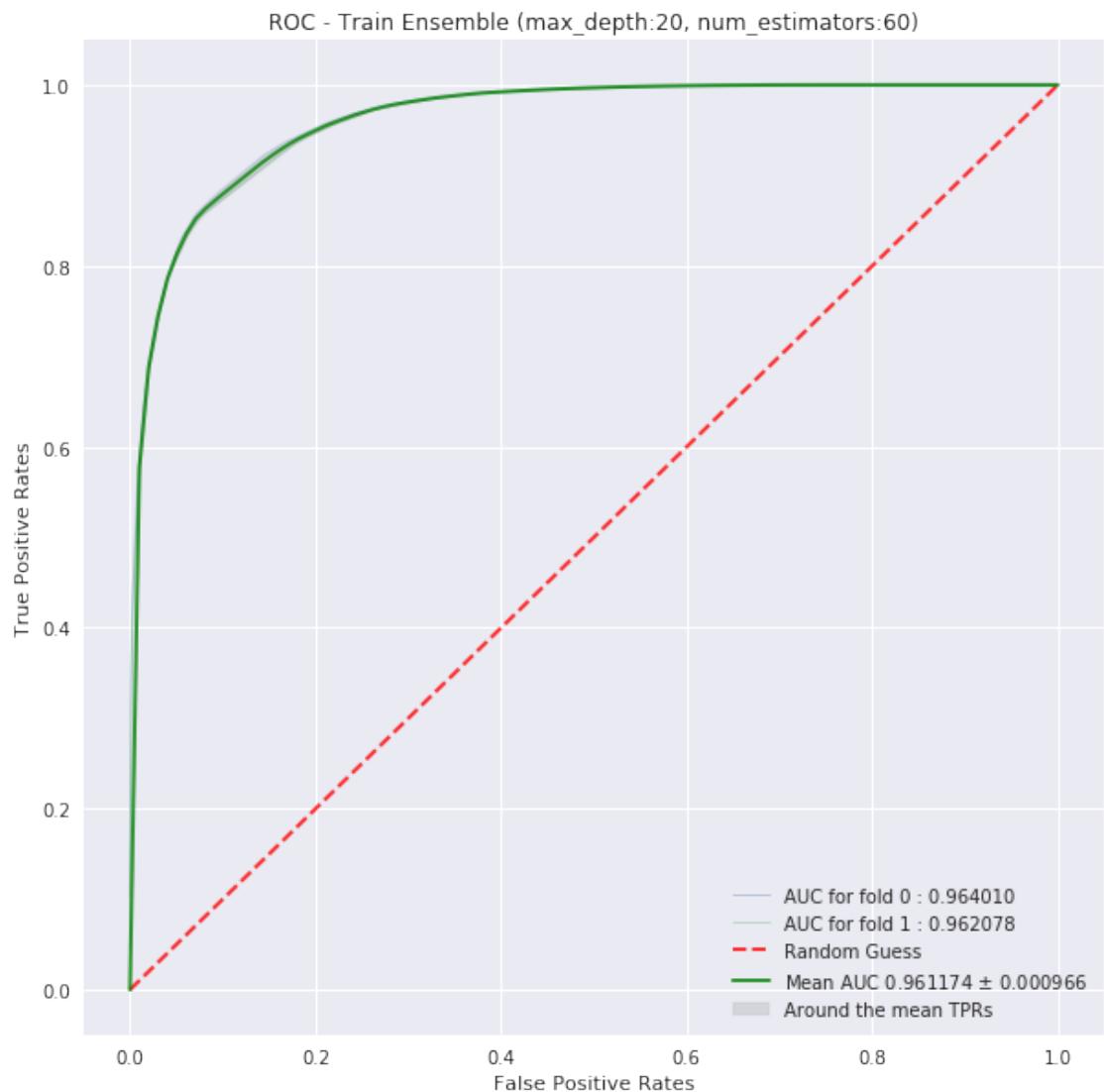
ROC - Validation Ensemble (max\_depth:10, num\_estimators:500)



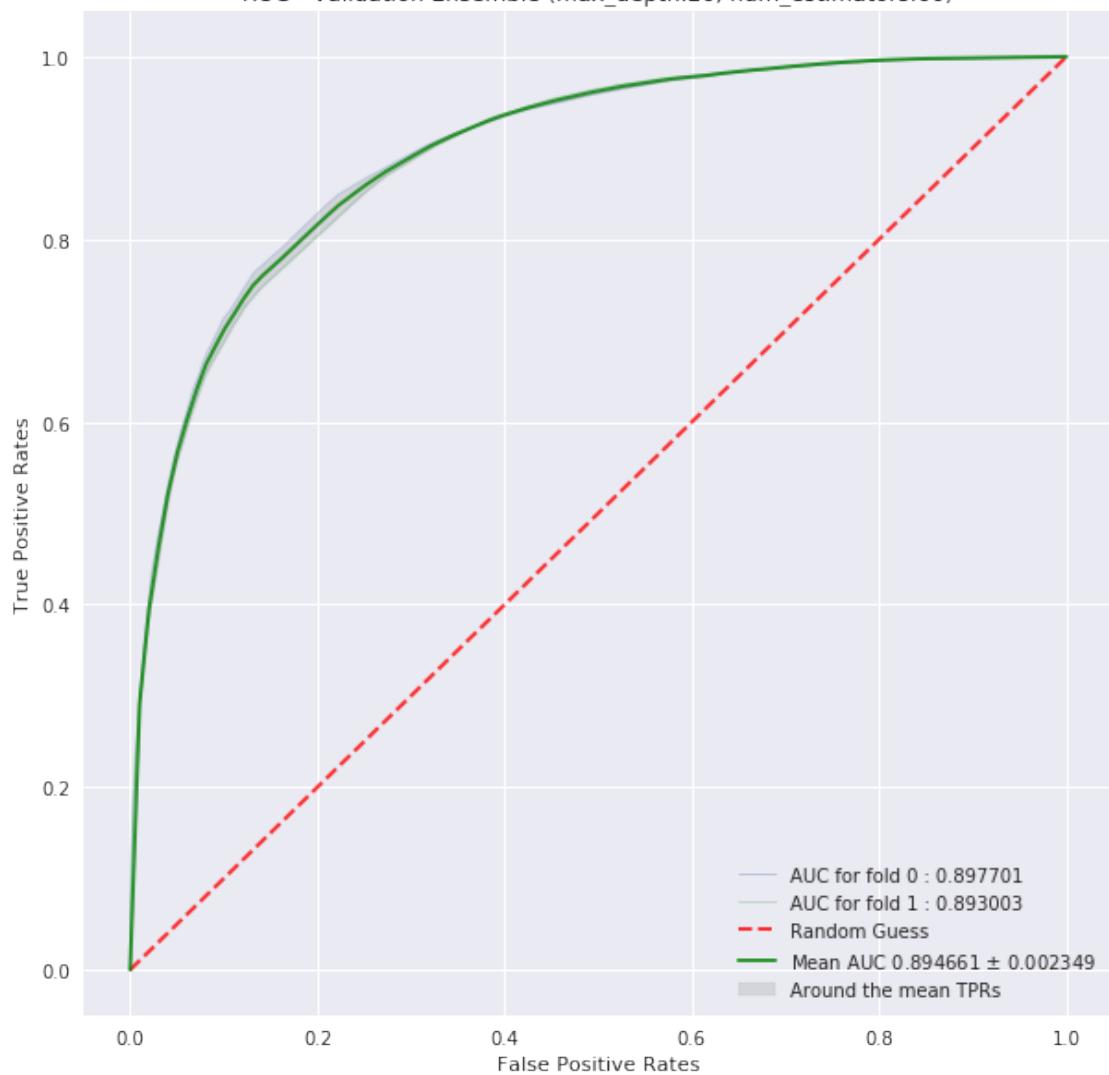


ROC - Validation Ensemble (max\_depth:20, num\_estimators:20)

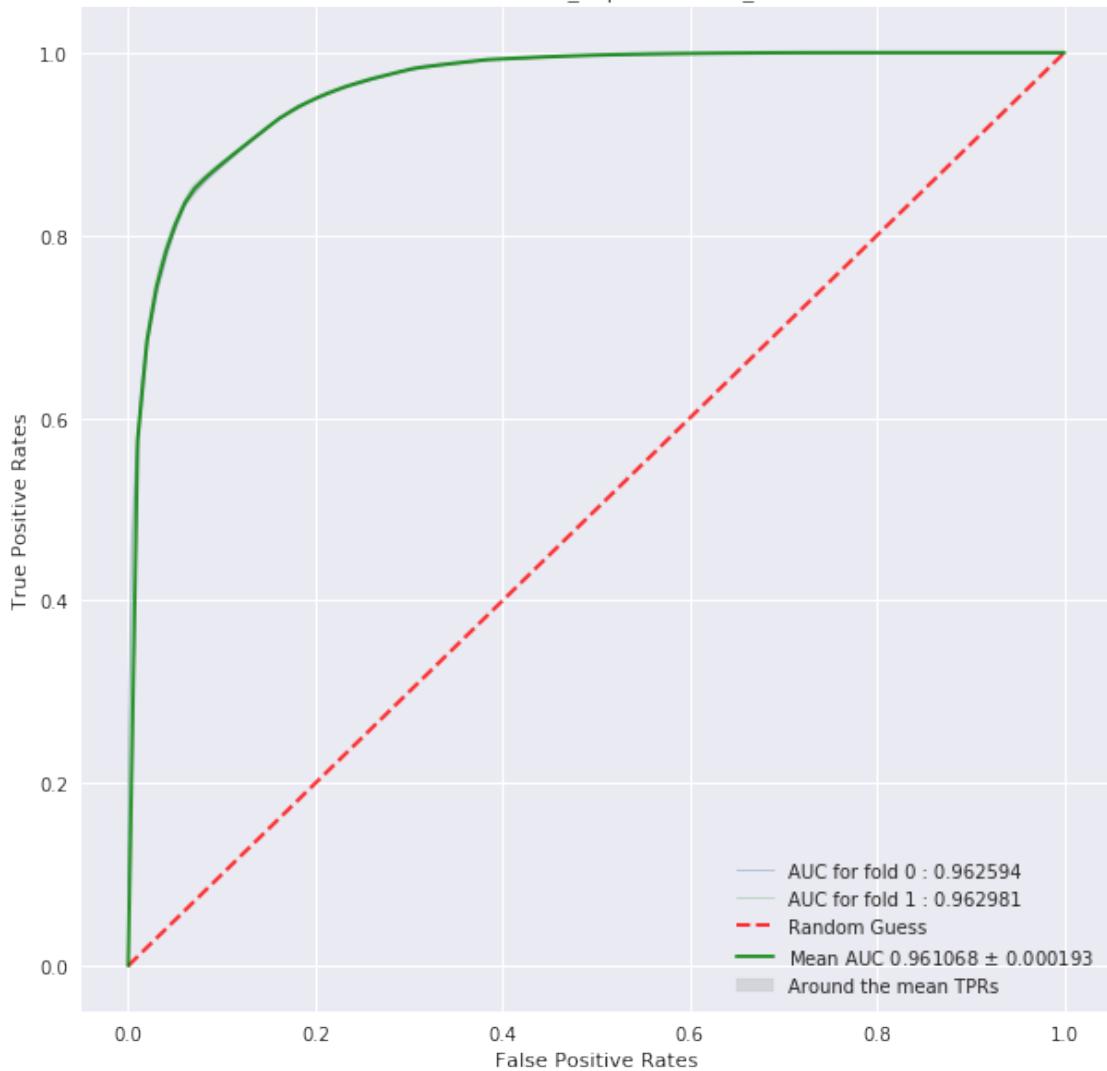




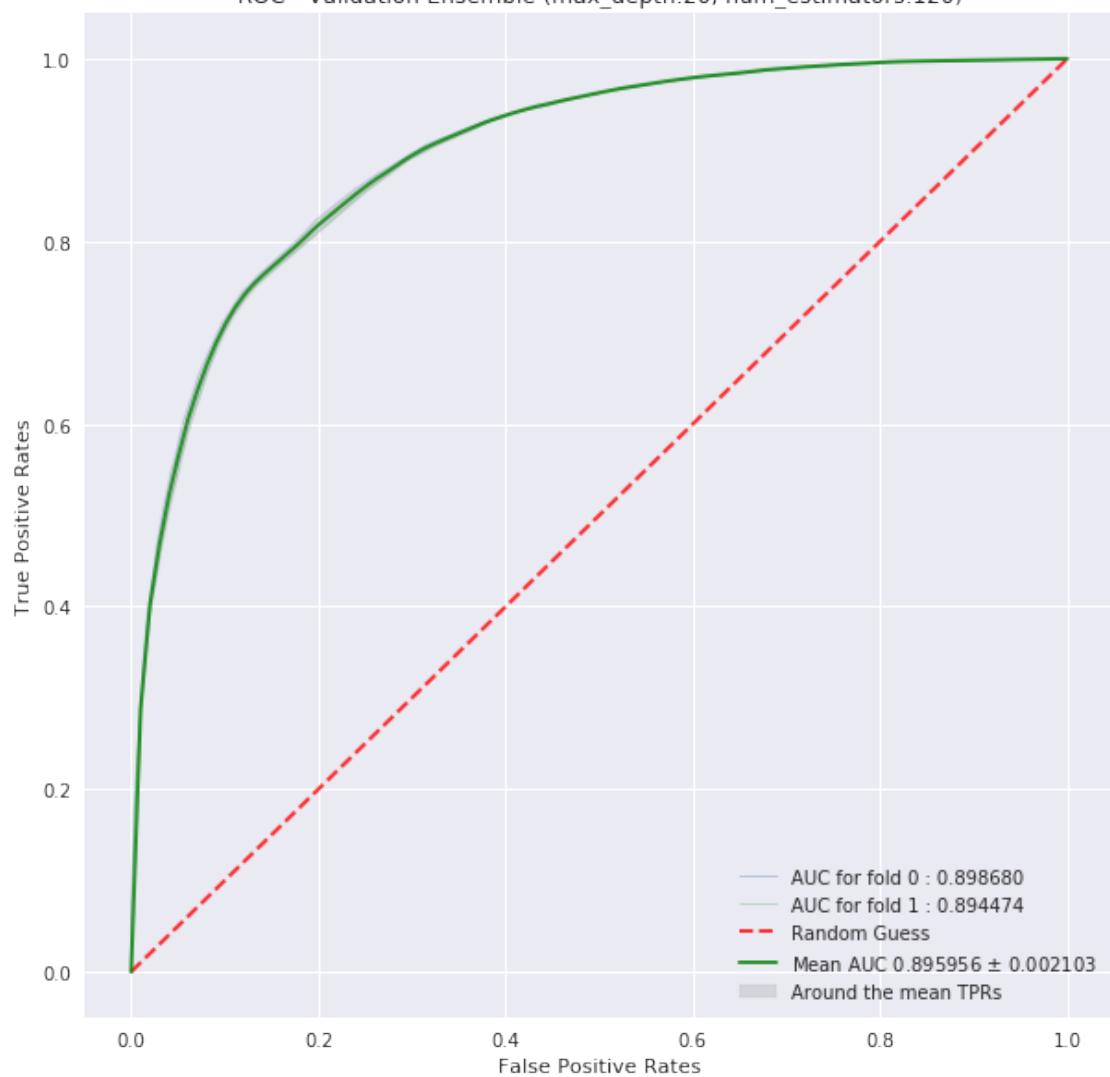
ROC - Validation Ensemble (max\_depth:20, num\_estimators:60)



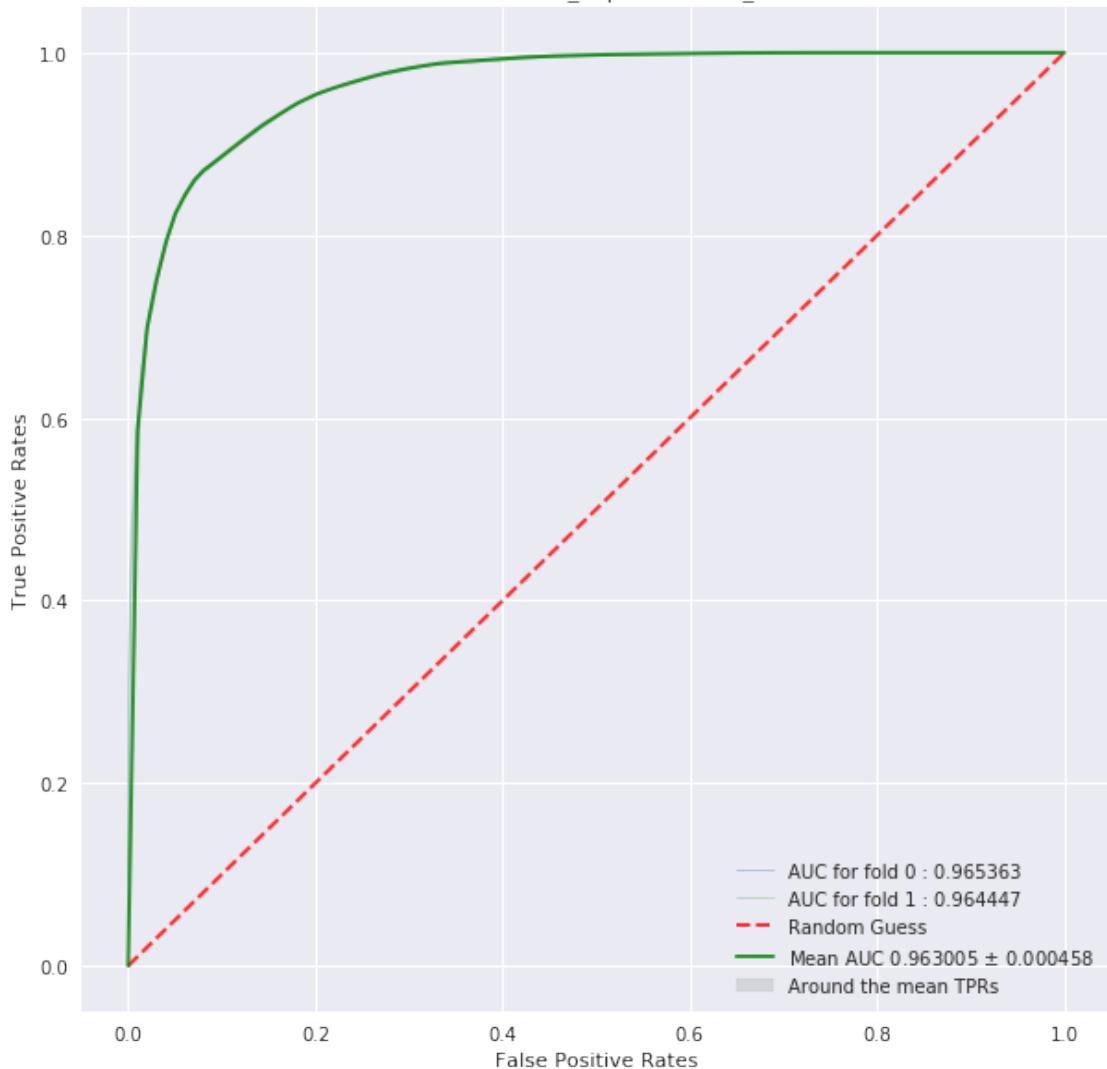
ROC - Train Ensemble (max\_depth:20, num\_estimators:120)



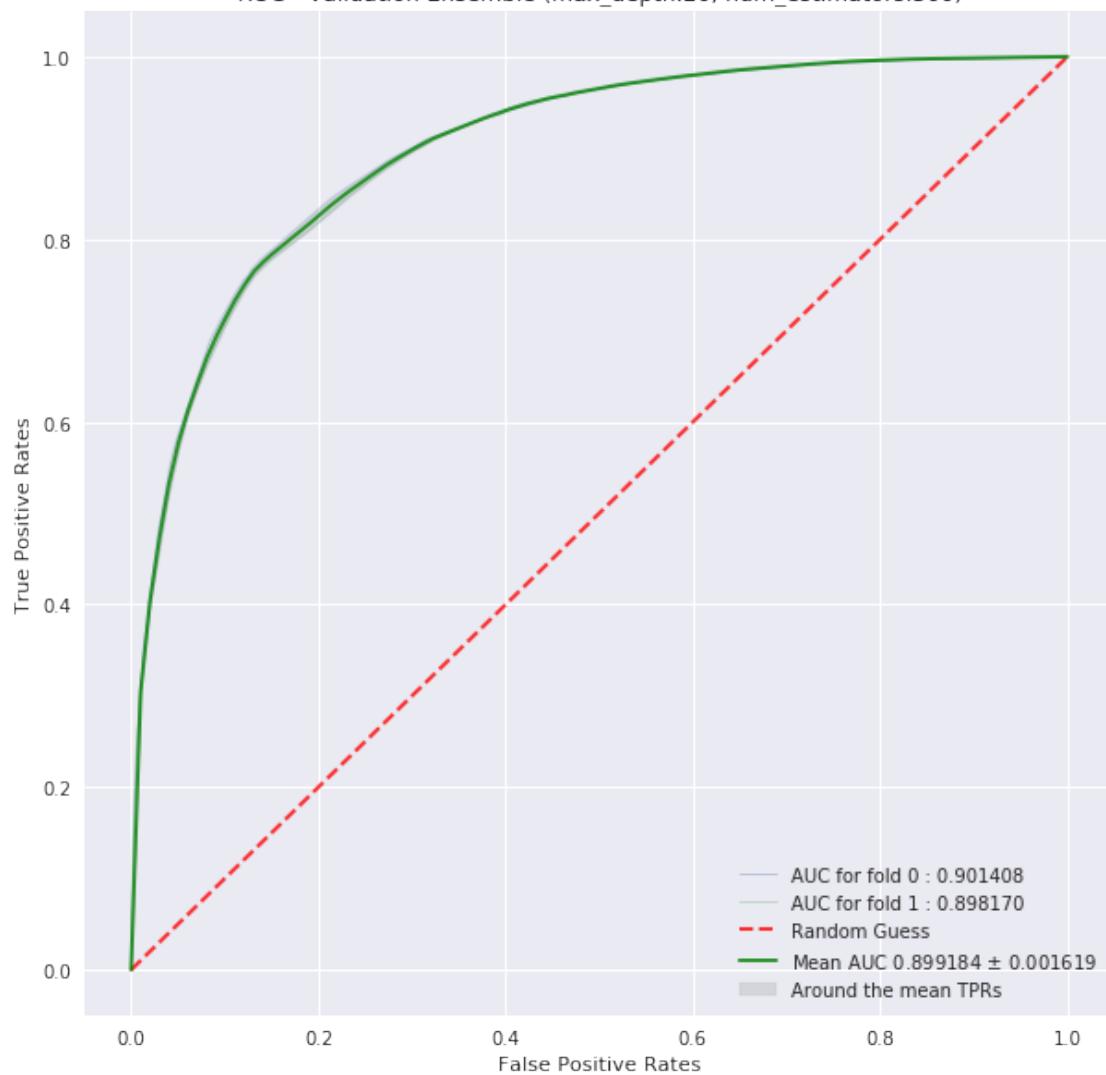
ROC - Validation Ensemble (max\_depth:20, num\_estimators:120)



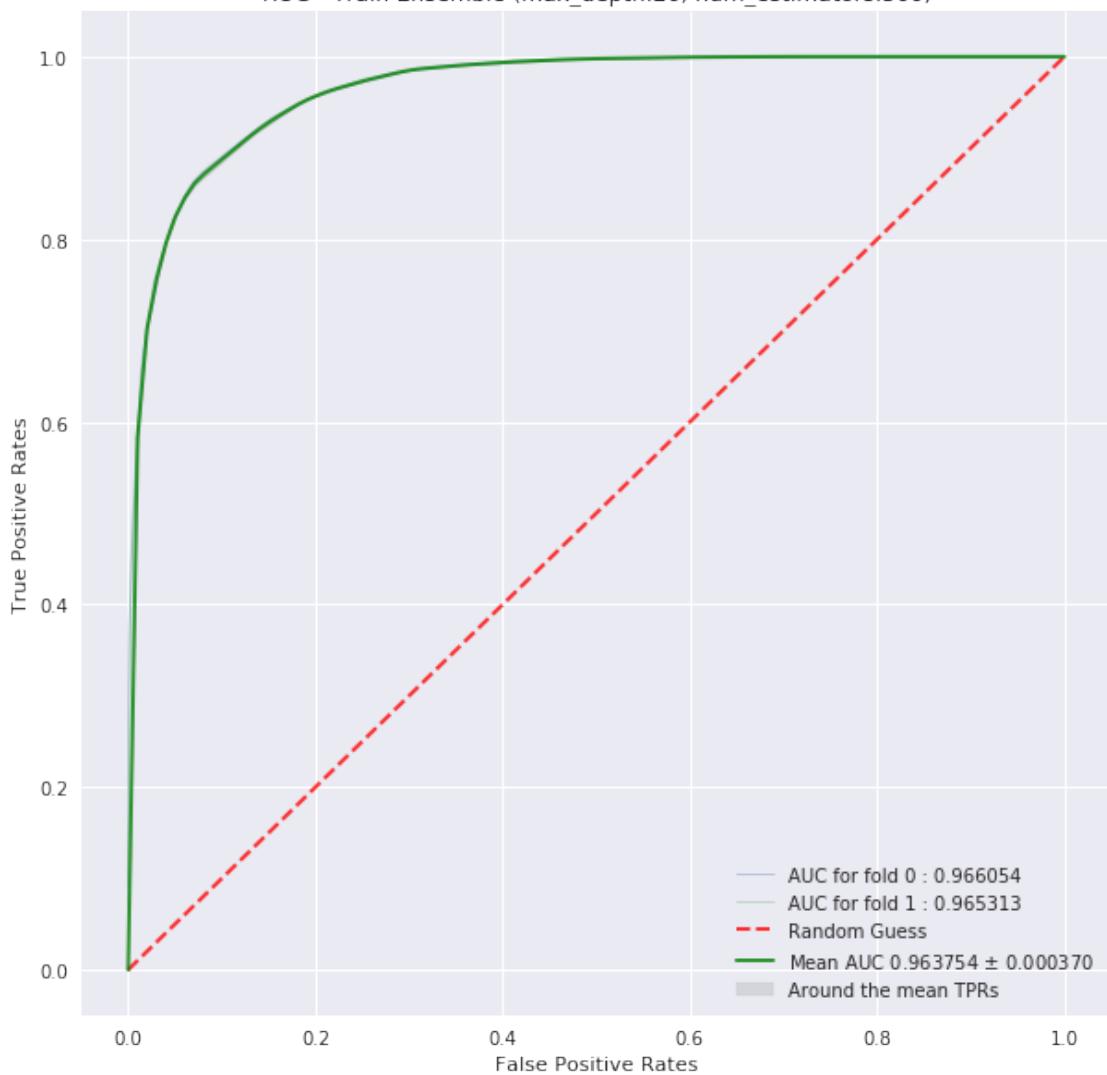
ROC - Train Ensemble (max\_depth:20, num\_estimators:300)



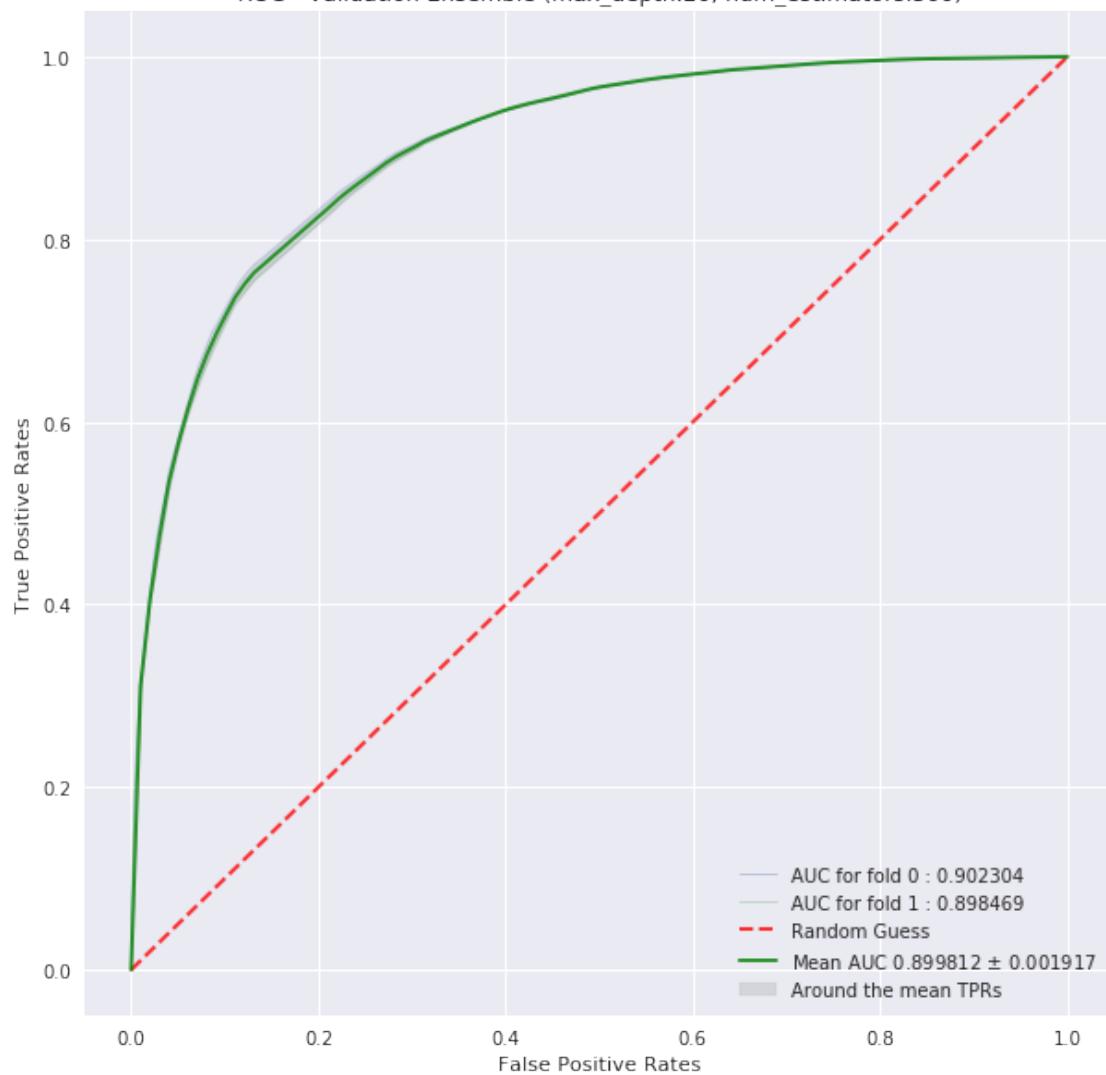
ROC - Validation Ensemble (max\_depth:20, num\_estimators:300)

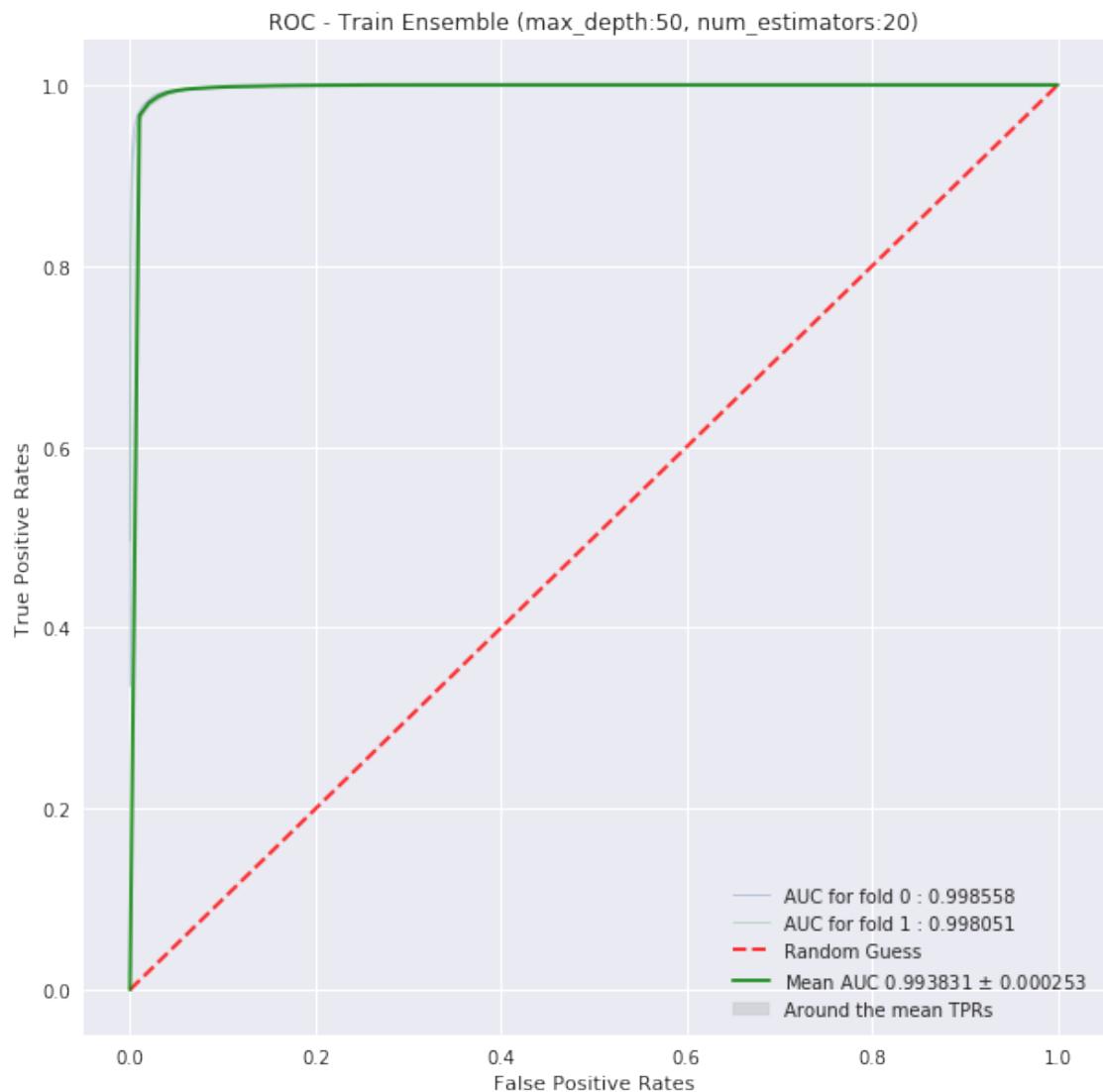


ROC - Train Ensemble (max\_depth:20, num\_estimators:500)

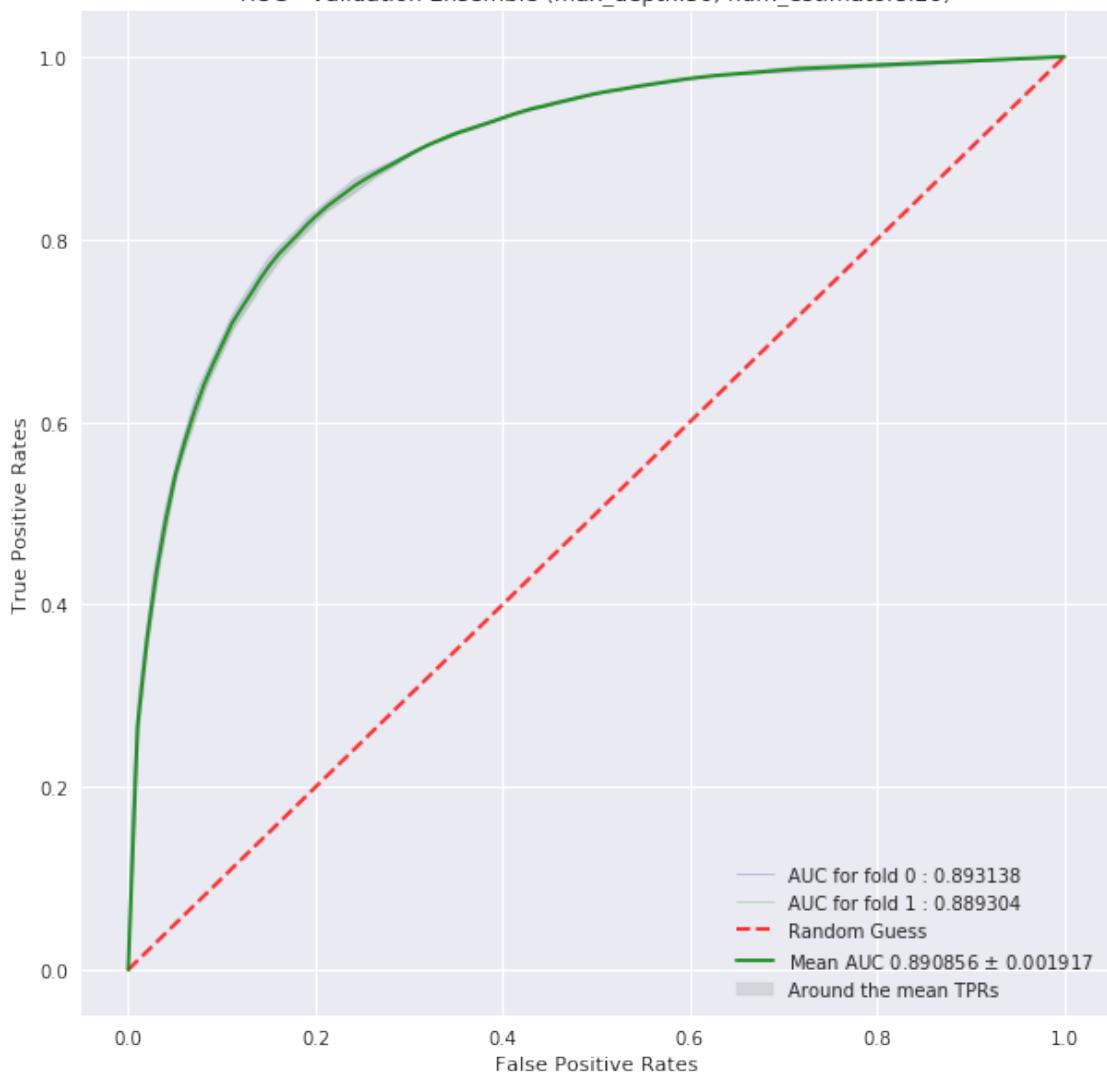


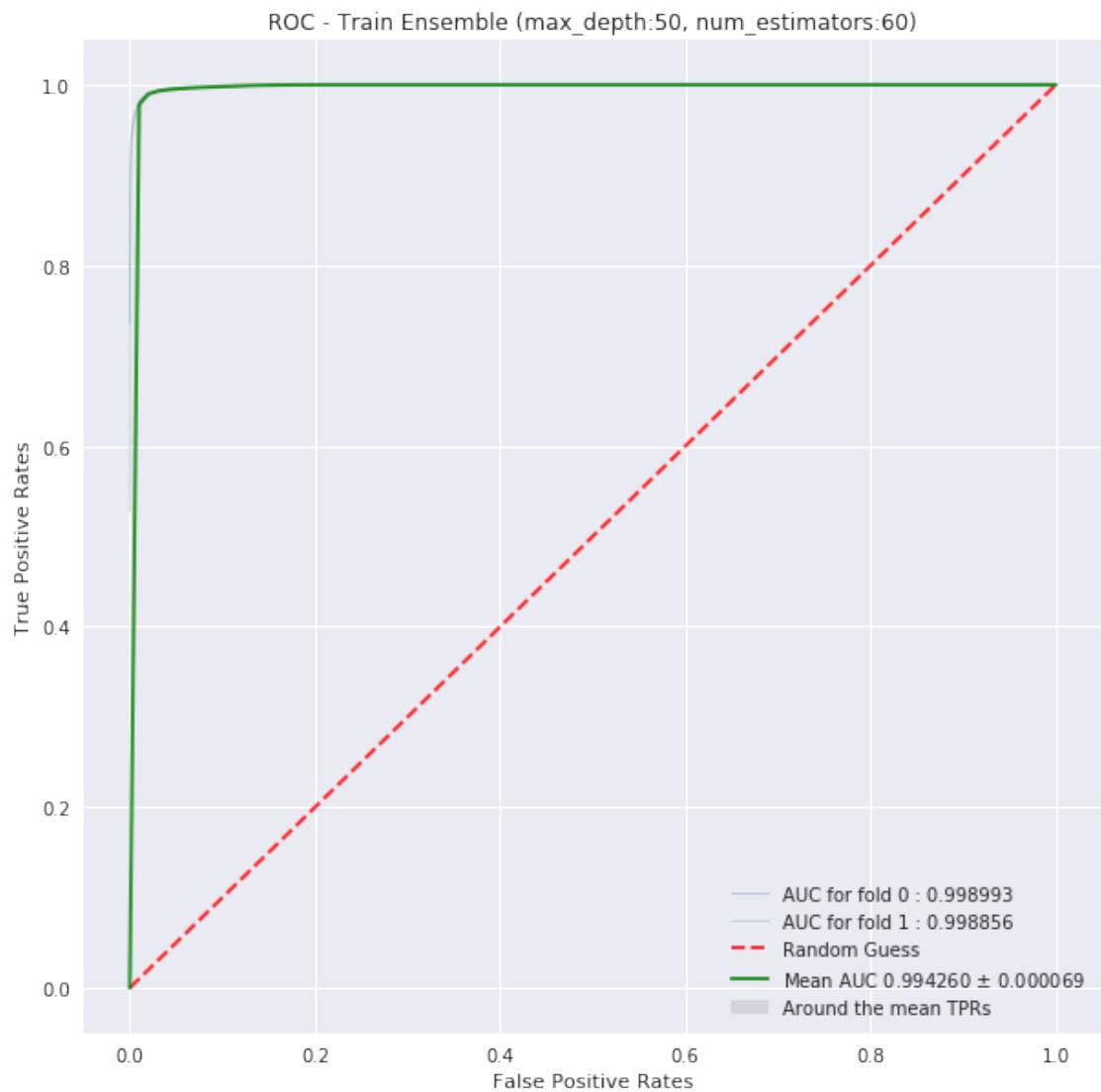
ROC - Validation Ensemble (max\_depth:20, num\_estimators:500)

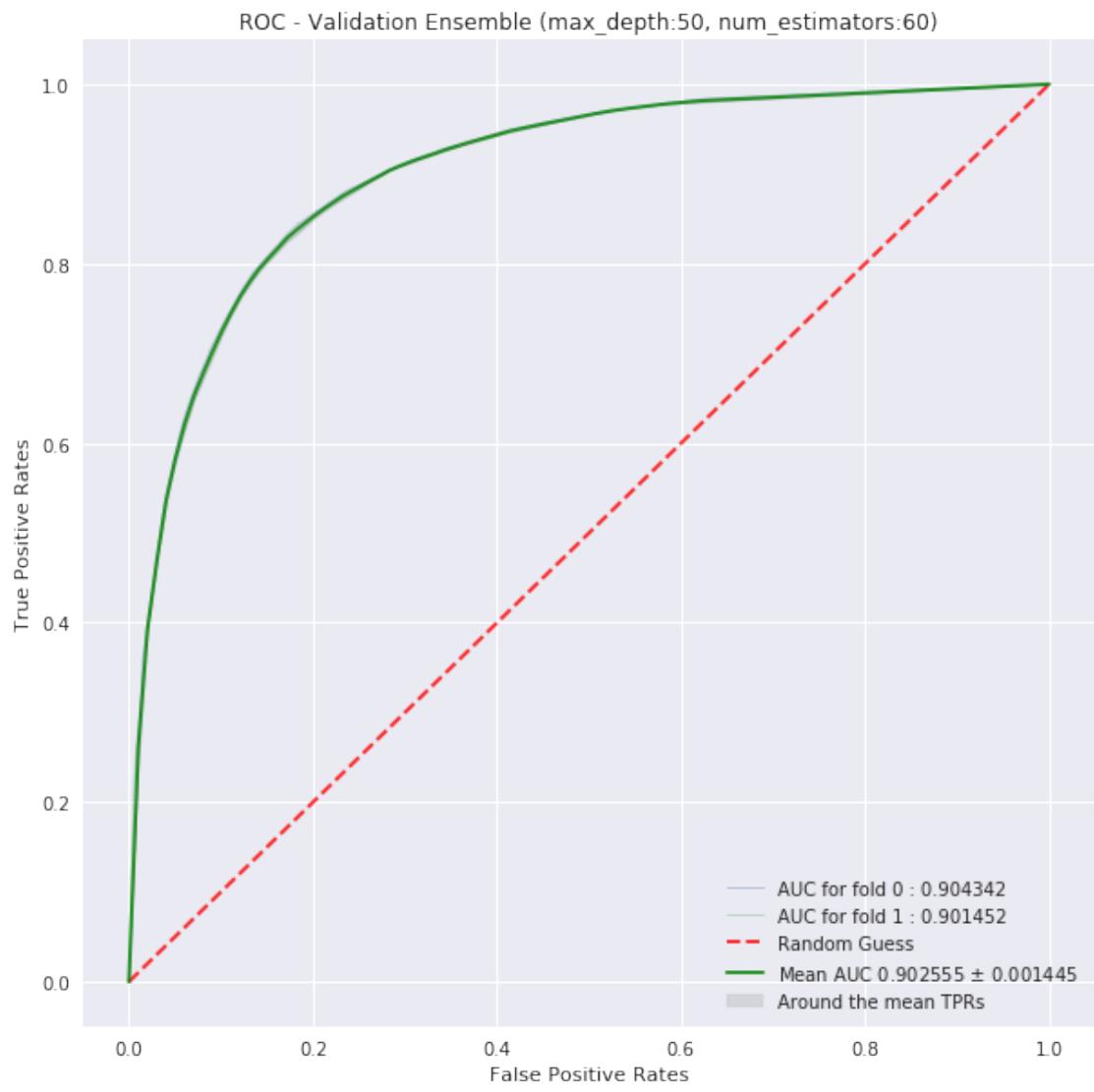




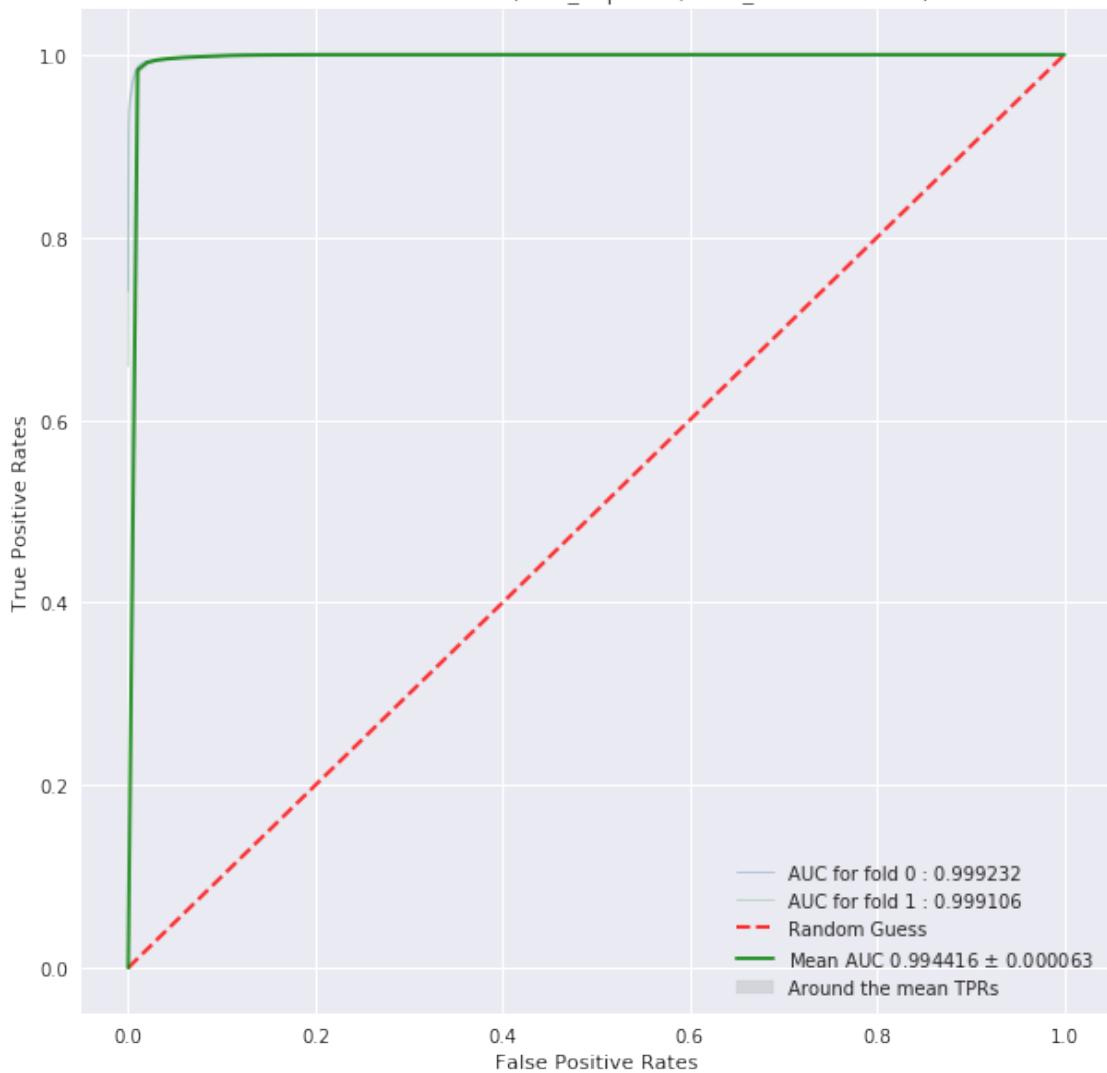
ROC - Validation Ensemble (max\_depth:50, num\_estimators:20)



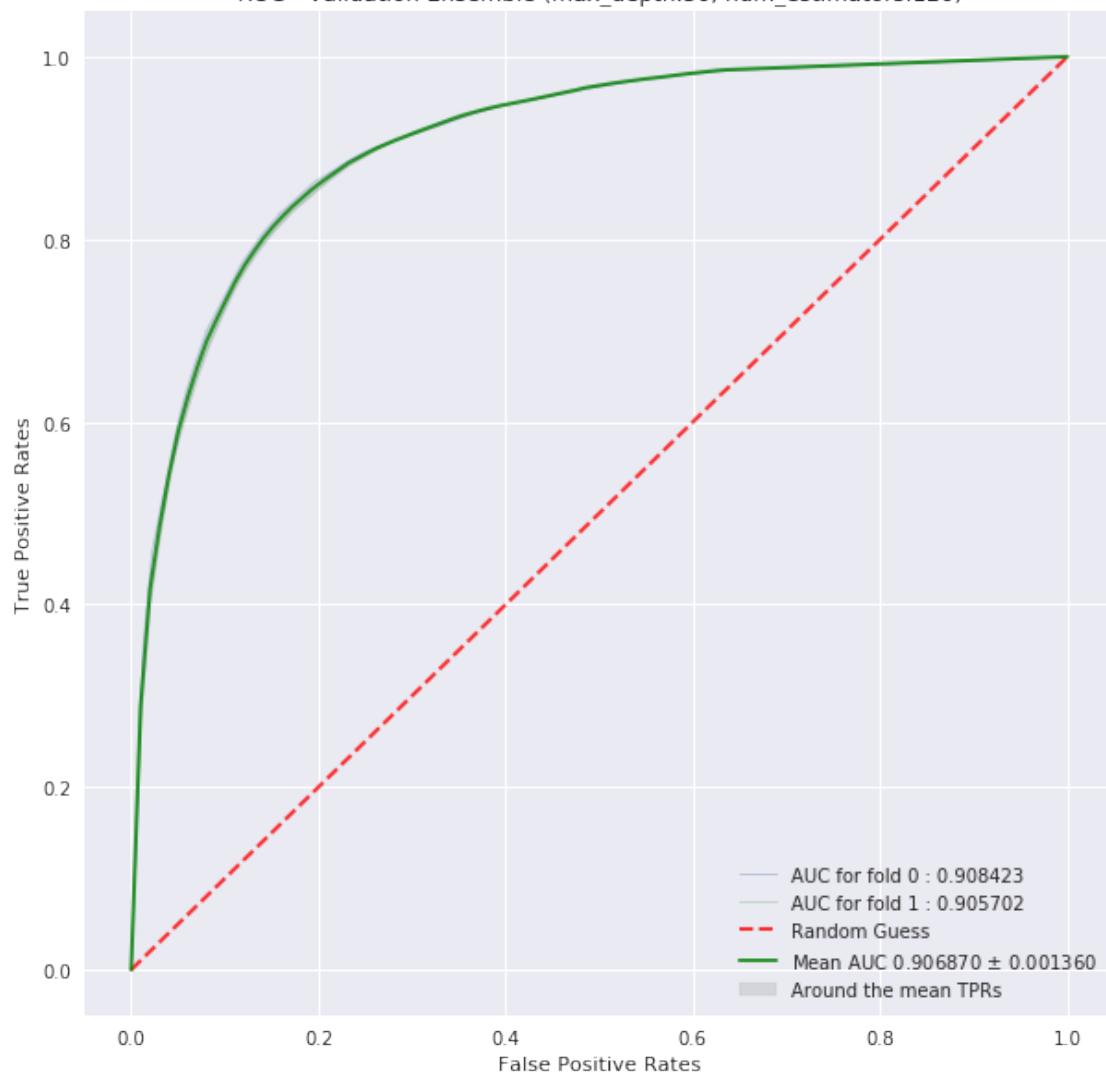




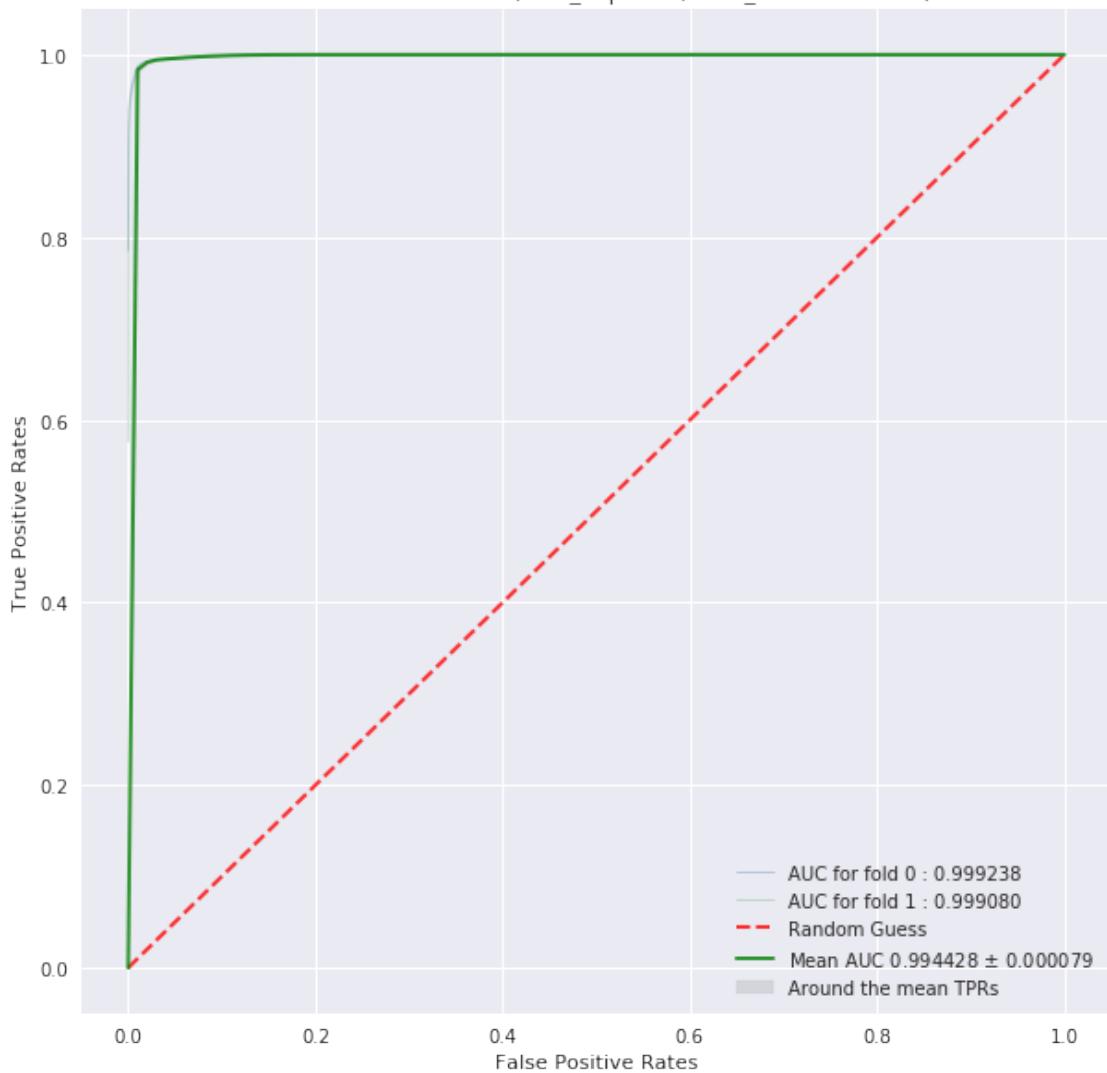
ROC - Train Ensemble (max\_depth:50, num\_estimators:120)



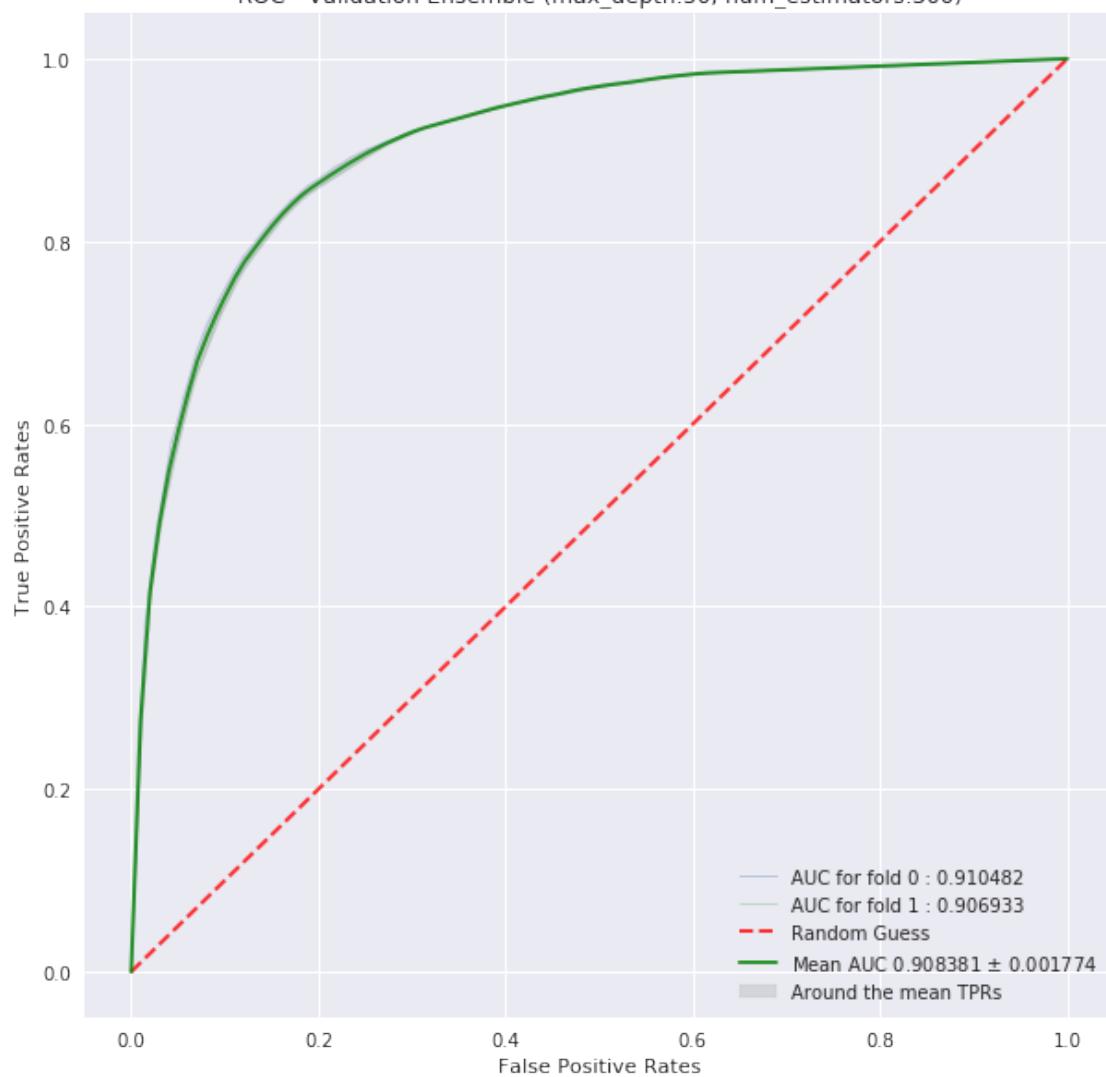
ROC - Validation Ensemble (max\_depth:50, num\_estimators:120)



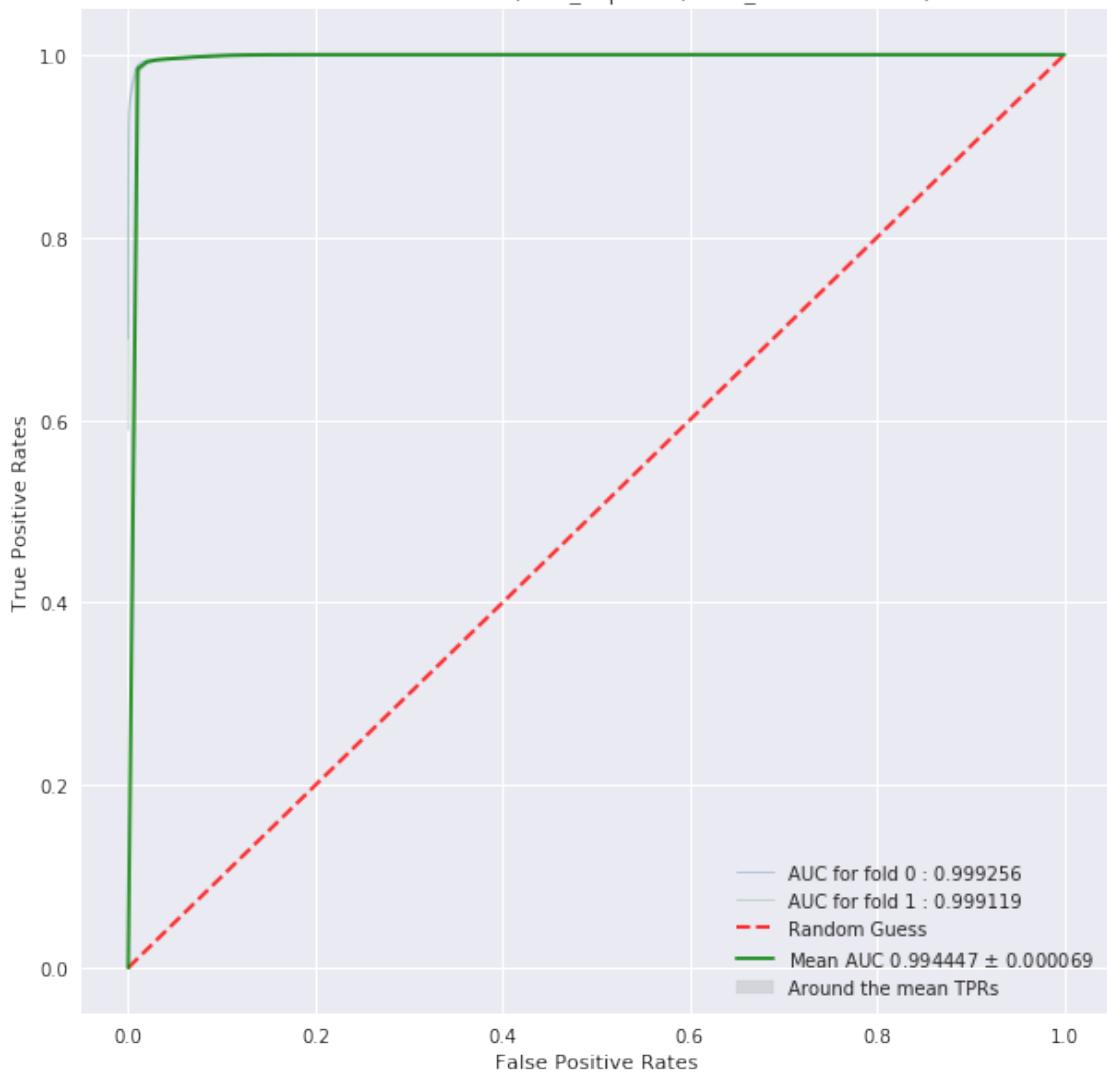
ROC - Train Ensemble (max\_depth:50, num\_estimators:300)



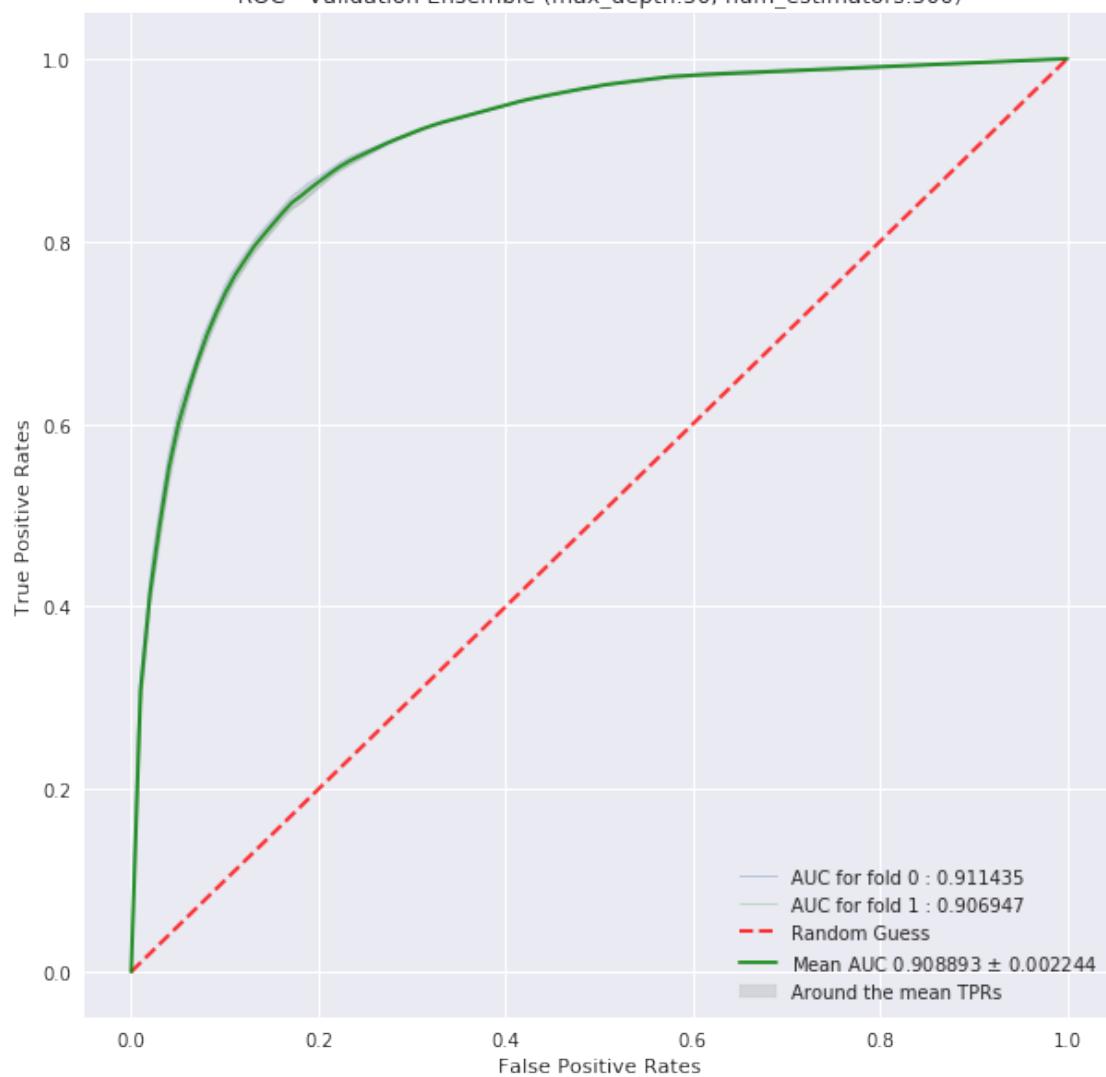
ROC - Validation Ensemble (max\_depth:50, num\_estimators:300)



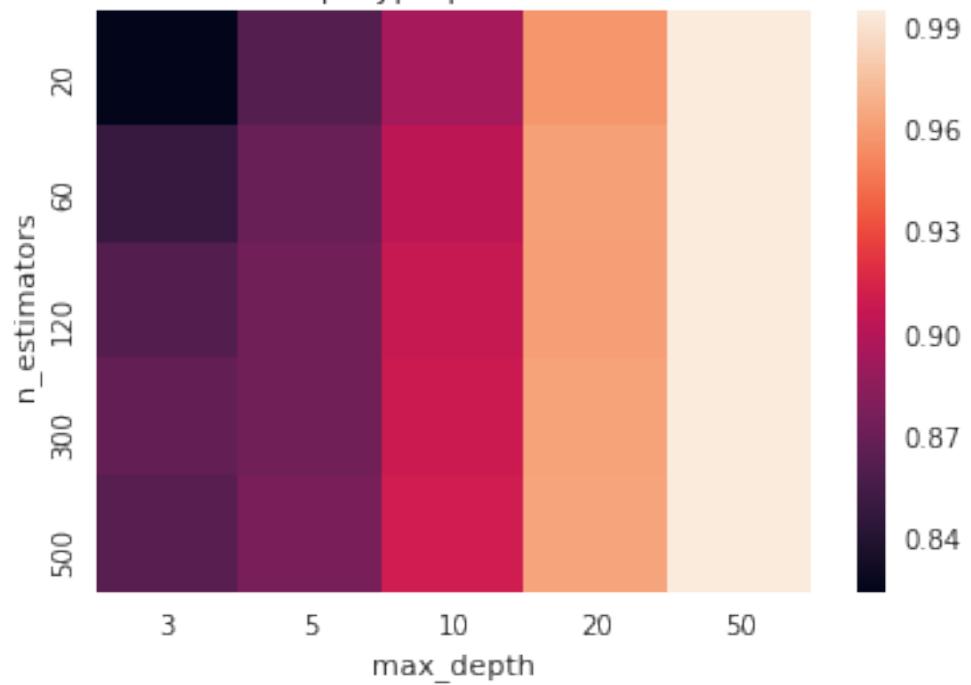
ROC - Train Ensemble (max\_depth:50, num\_estimators:500)



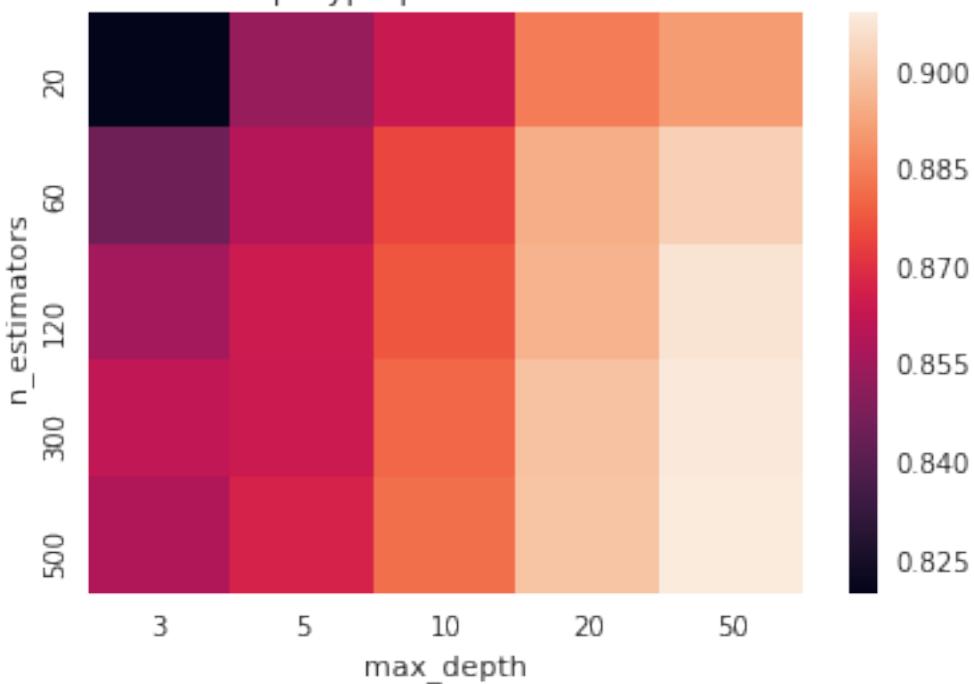
ROC - Validation Ensemble (max\_depth:50, num\_estimators:500)



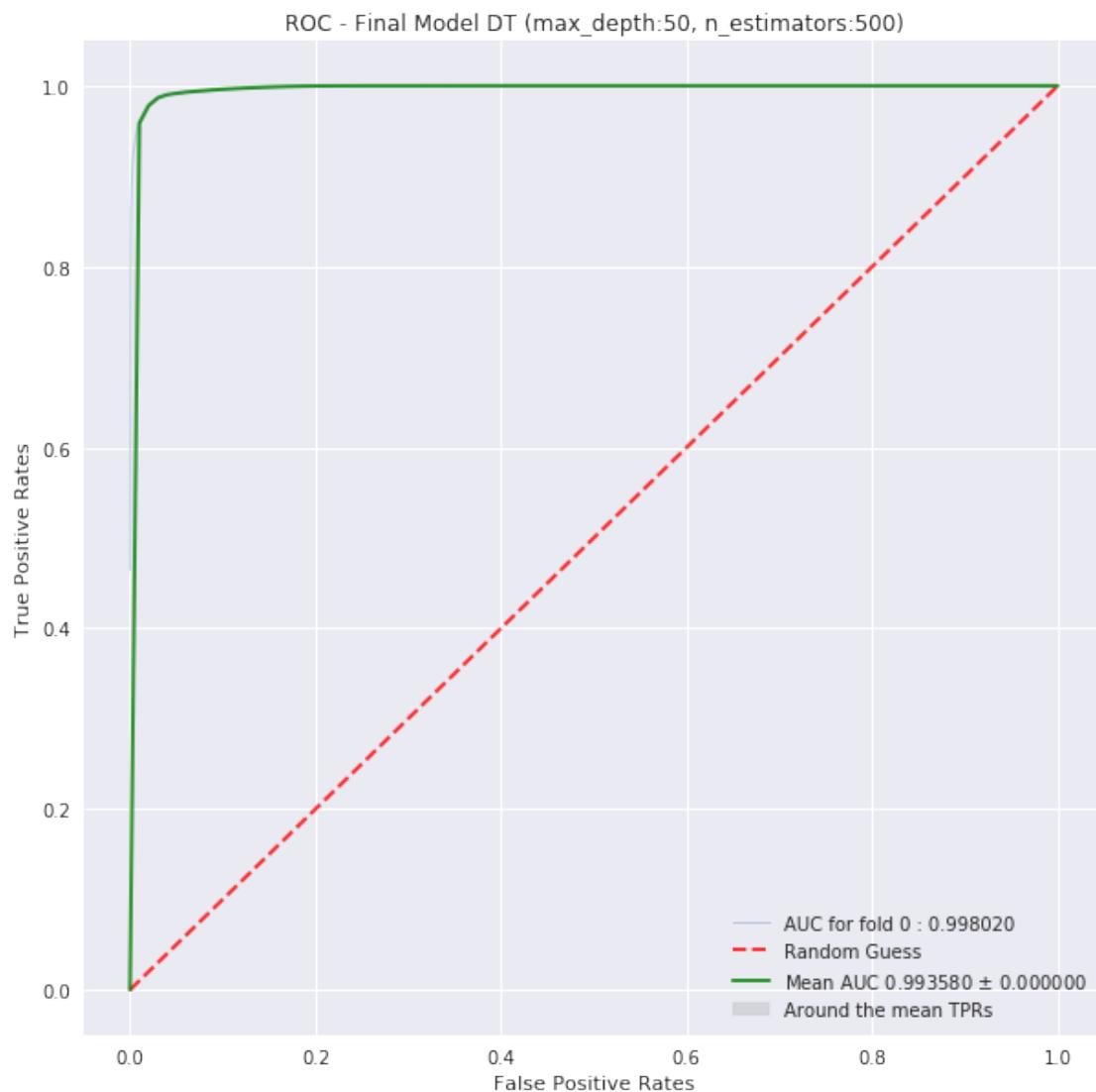
Heatmap Hyperparams for Train

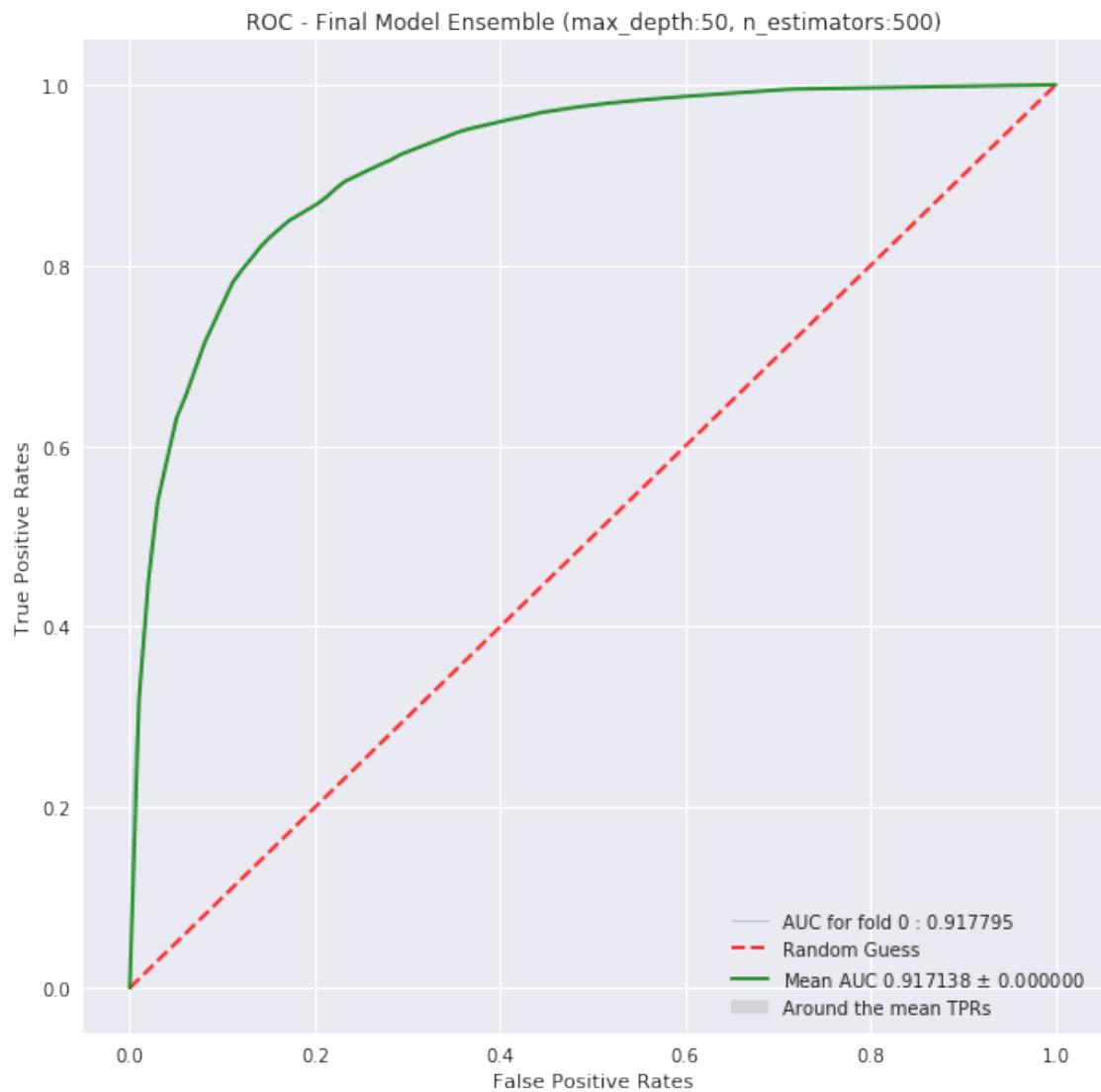


Heatmap Hyperparams for Validation

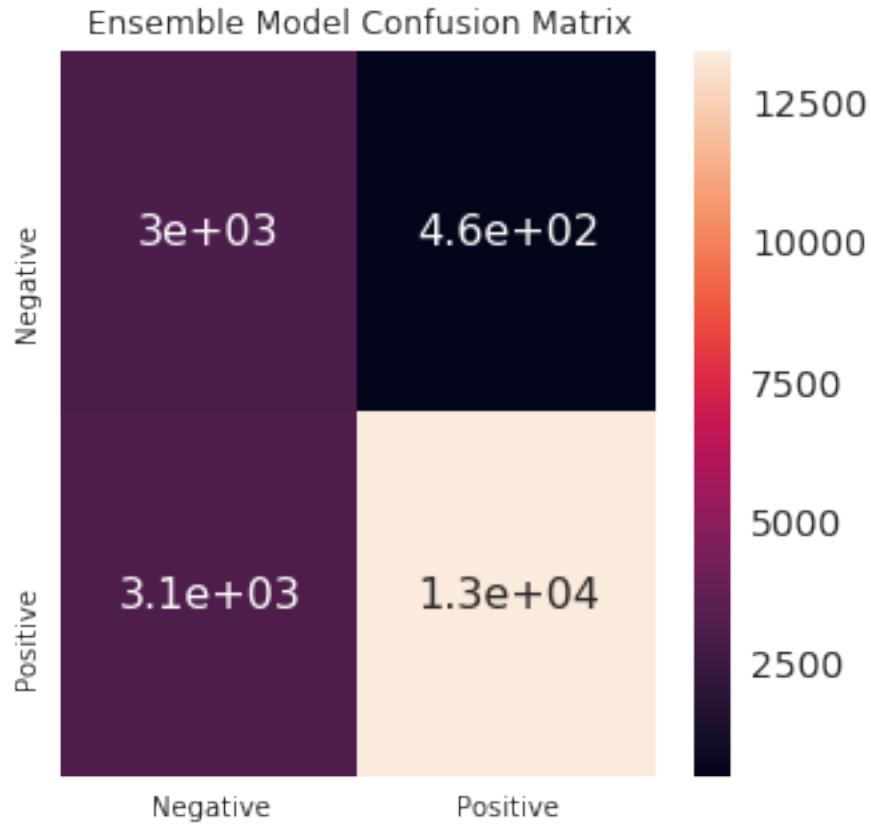


Best hyperparam value: (50, 500)





Test auc score 0.917138478904123



	Negative	Positive
Precision	0.491362	0.966460
Recall	0.866379	0.811077
Fscore	0.627080	0.881977
Support	3480.000000	16520.000000

#### 4.2.2 [A.2] Wordcloud of top 20 important features from SET 1

```
In [11]: # get feature and its importance as tuple
    feature_imp_info = list(zip(feature_name_list, model[0].feature_importances_))

    # filter only those features which have a value greater than zero
    feature_imp_dict = dict(list(filter(lambda x: x[1] > 0.0, feature_imp_info)))

    # create word cloud object for displaying the output
    wc = WordCloud(background_color='white', width=800, height=800)
    wc_output = wc.generate_from_frequencies(feature_imp_dict)

In [12]: plt.figure(figsize=(8,8))
    plt.imshow(wc_output)
```

```
plt.axis('off')
plt.tight_layout(pad=0.0)
plt.title('RF Feature Importances')
plt.show()
```



### 4.3 Observation

best hyper param identified is max\_depth = 50, and n\_estimators=500

Review length is identified as one of the important feature

Positive words such as 'great', 'best' & negative words such as 'disappoint', 'not buy' are recognized as important features by random forest

#### 4.3.1 [A.3] Applying Random Forests on TFIDF, SET 2

```
In [13]: # form two lists
    depth_list = [3, 5, 10, 20, 50] # depends on size of dataset
    n_estimators_list = [20, 60, 120, 300, 500] # depends on size of dataset

    # create a configuartion dictionary
    config_dict = {
        'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF/',
        'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF/test.csv',
        'train_size' : 50000,
        'test_size' : 20000,
        'hyperparam_list' : list(product(depth_list, n_estimators_list)),
        'implementation': 'rf' # 'xgb' or 'rf'
    }

In [14]: # read the train, test data and preprocess it
    train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                                                           scaling=True,
                                                                           dim_reduction=True)

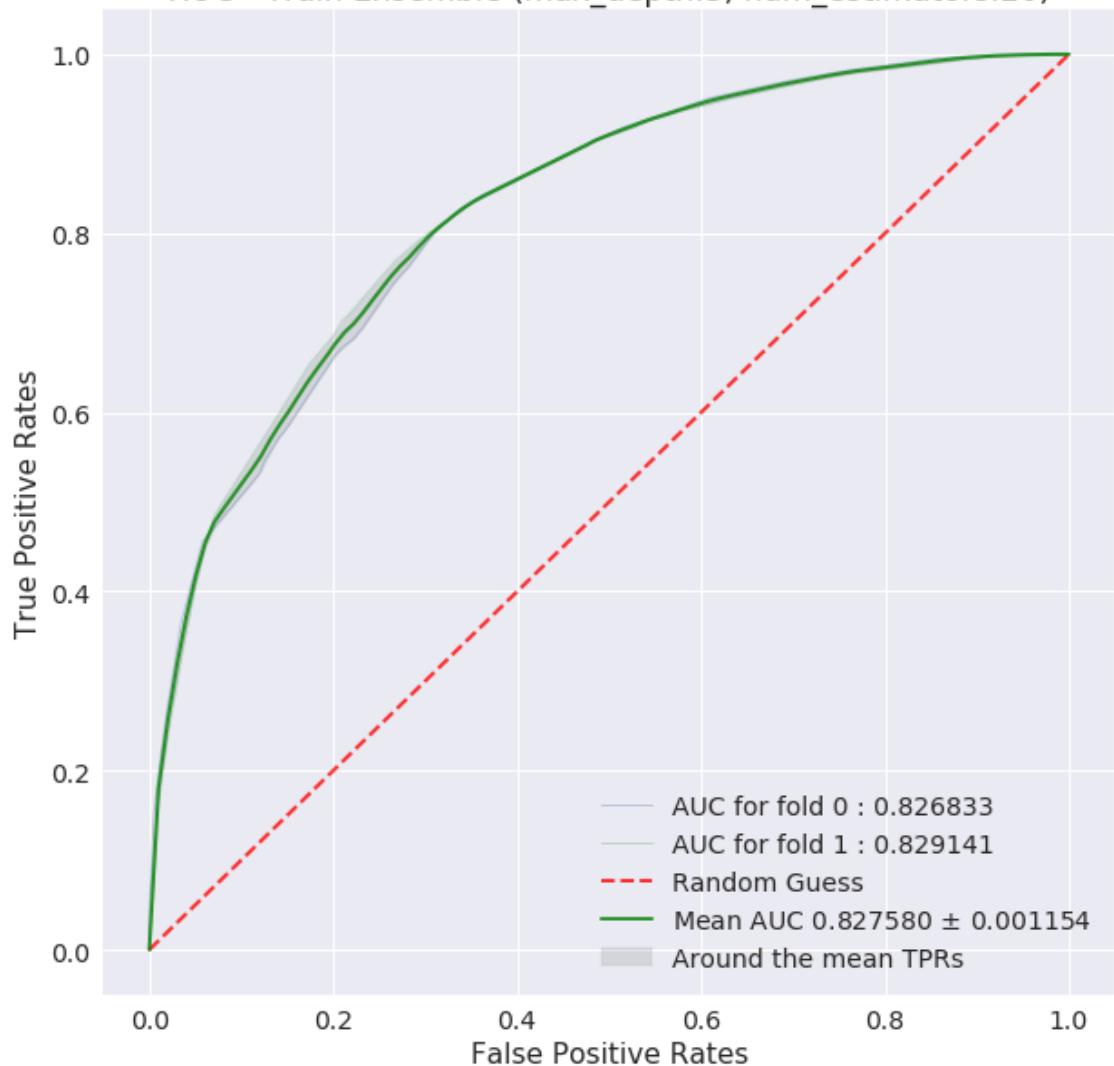
    # create a list containing all the features
    feature_name_list = train_features.columns.values.tolist()

    # train and validate the model
    model = train_and_validate_model(config_dict, train_features, train_labels)

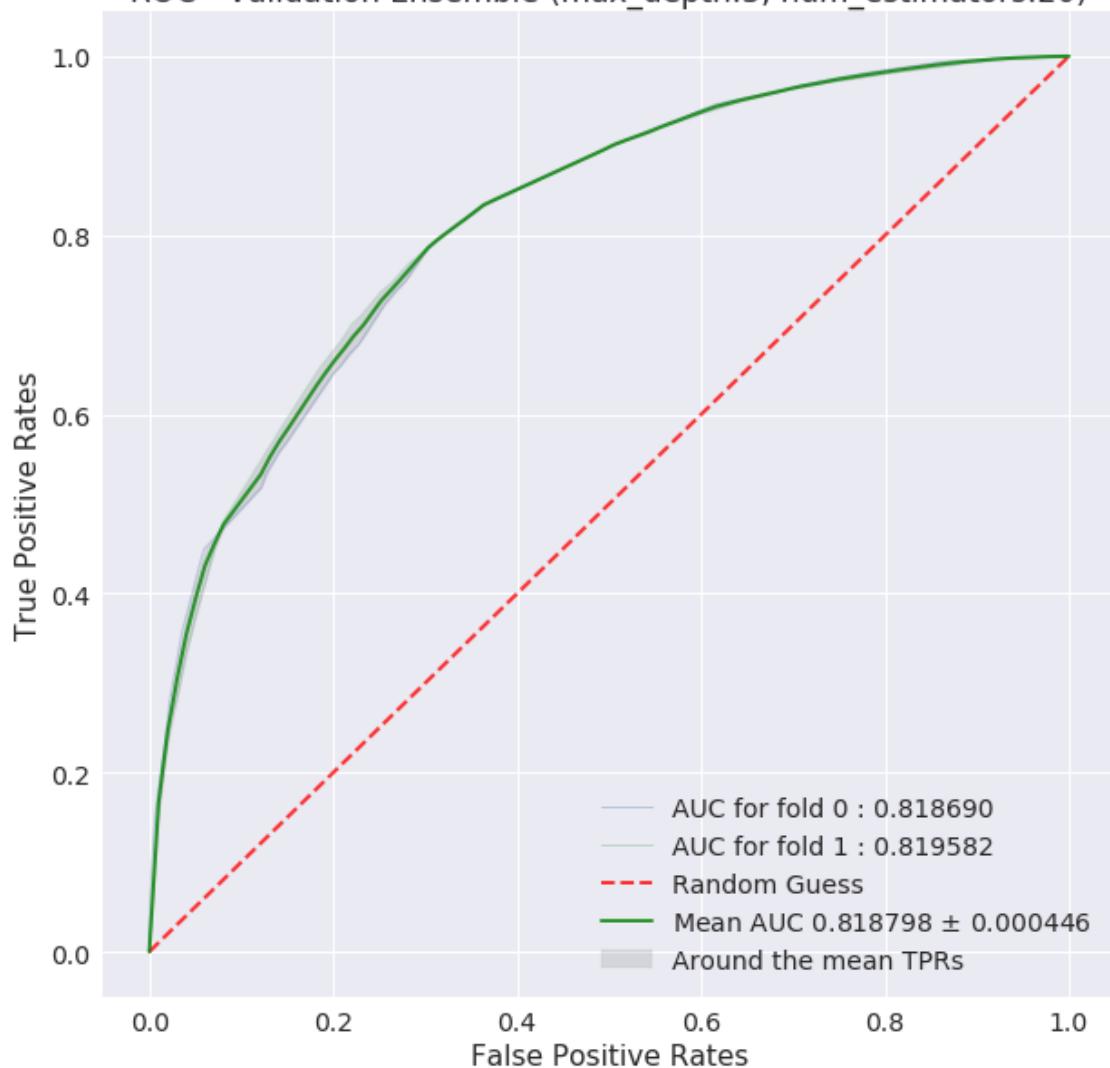
    # test and evaluate the model
    ptabe_entry_a2 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

Train df shape (50000, 503)
Class label distribution in train df:
0    25029
1    24971
Name: Label, dtype: int64
Test df shape (20000, 503)
Class label distribution in test df:
1    16520
0    3480
Name: Label, dtype: int64
Shape of -> train features :50000,501, test features: 20000,501
Shape of -> train labels :50000, test labels: 20000
=====
```

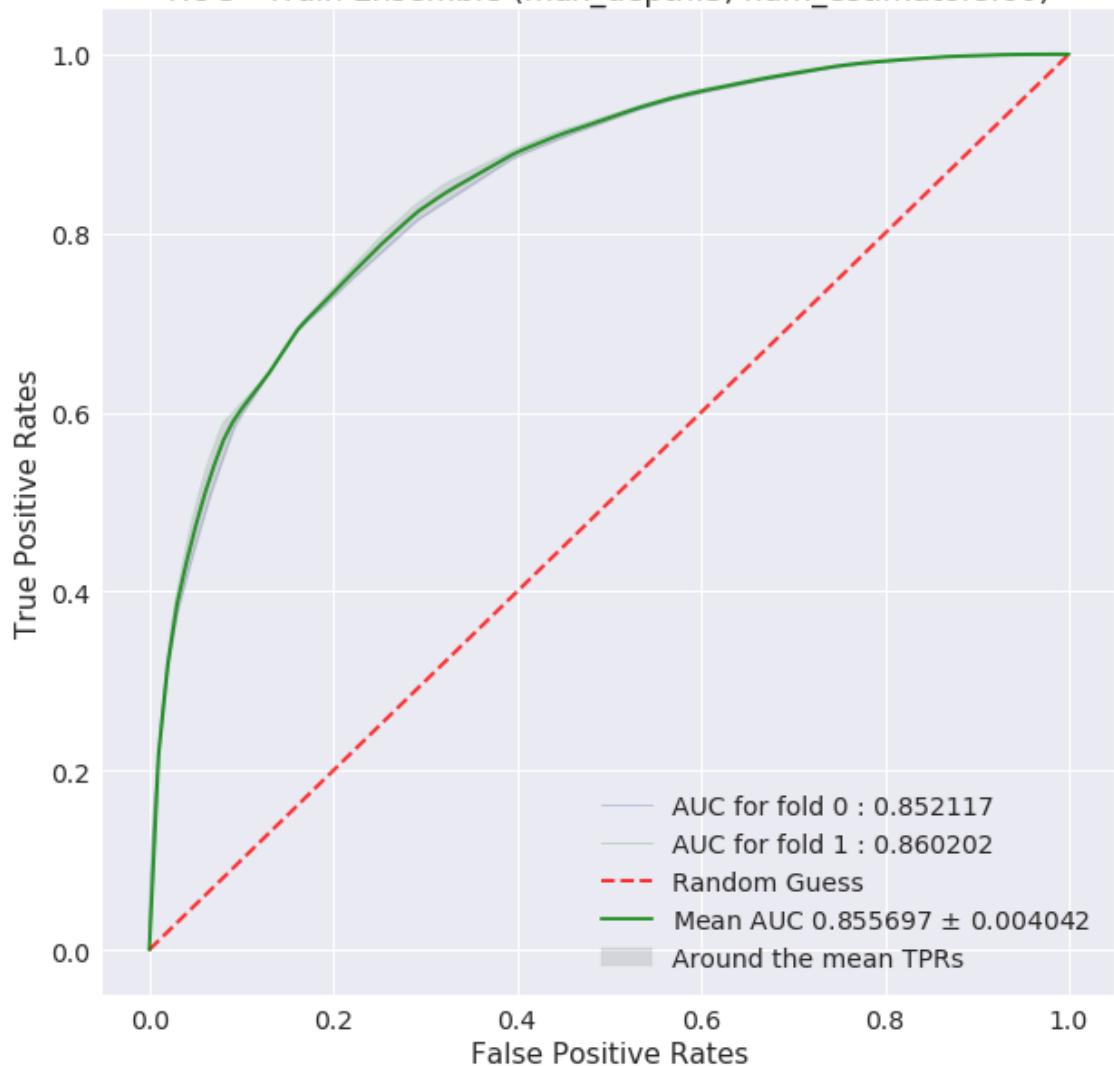
ROC - Train Ensemble (max\_depth:3, num\_estimators:20)



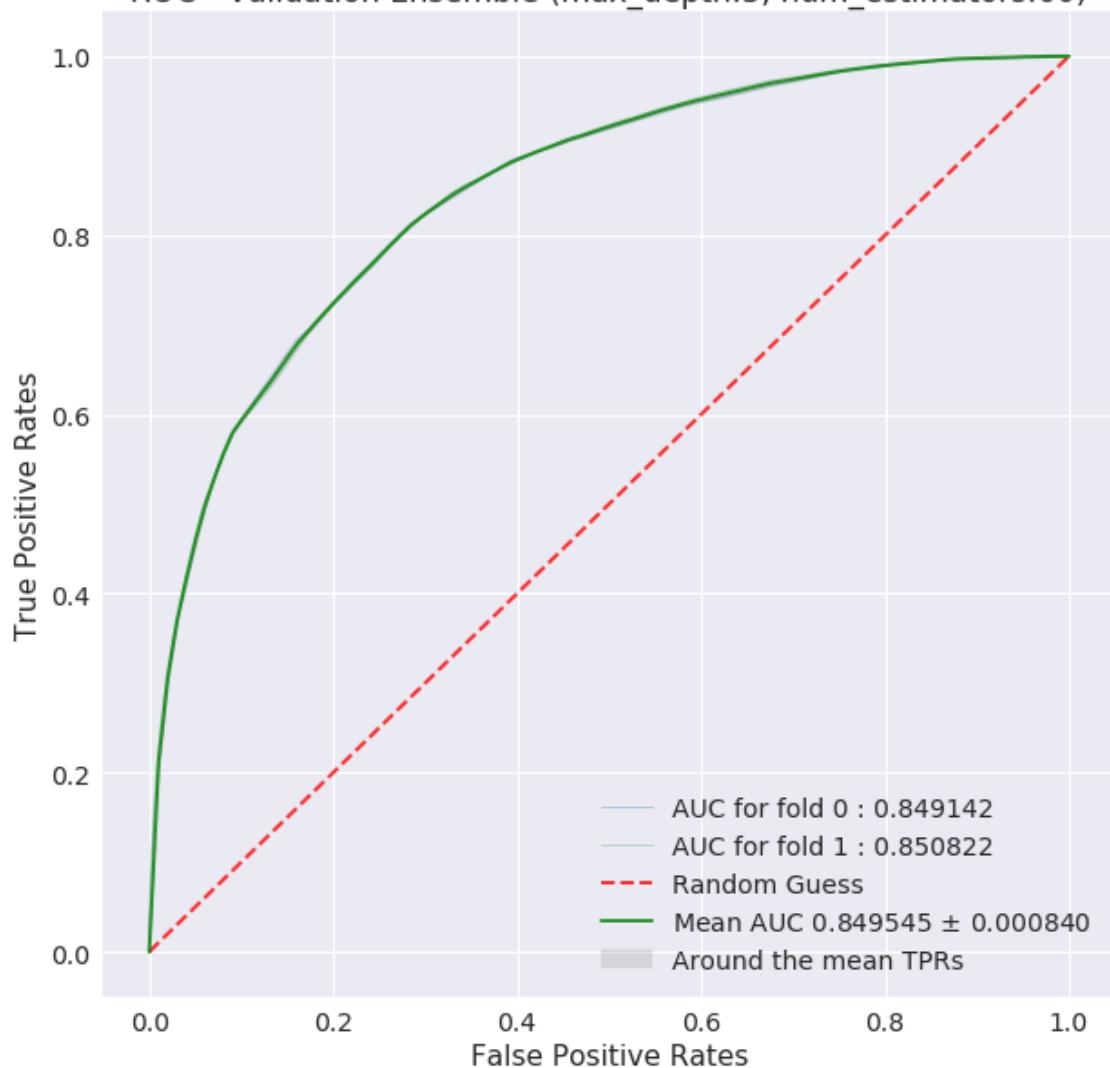
ROC - Validation Ensemble (max\_depth:3, num\_estimators:20)



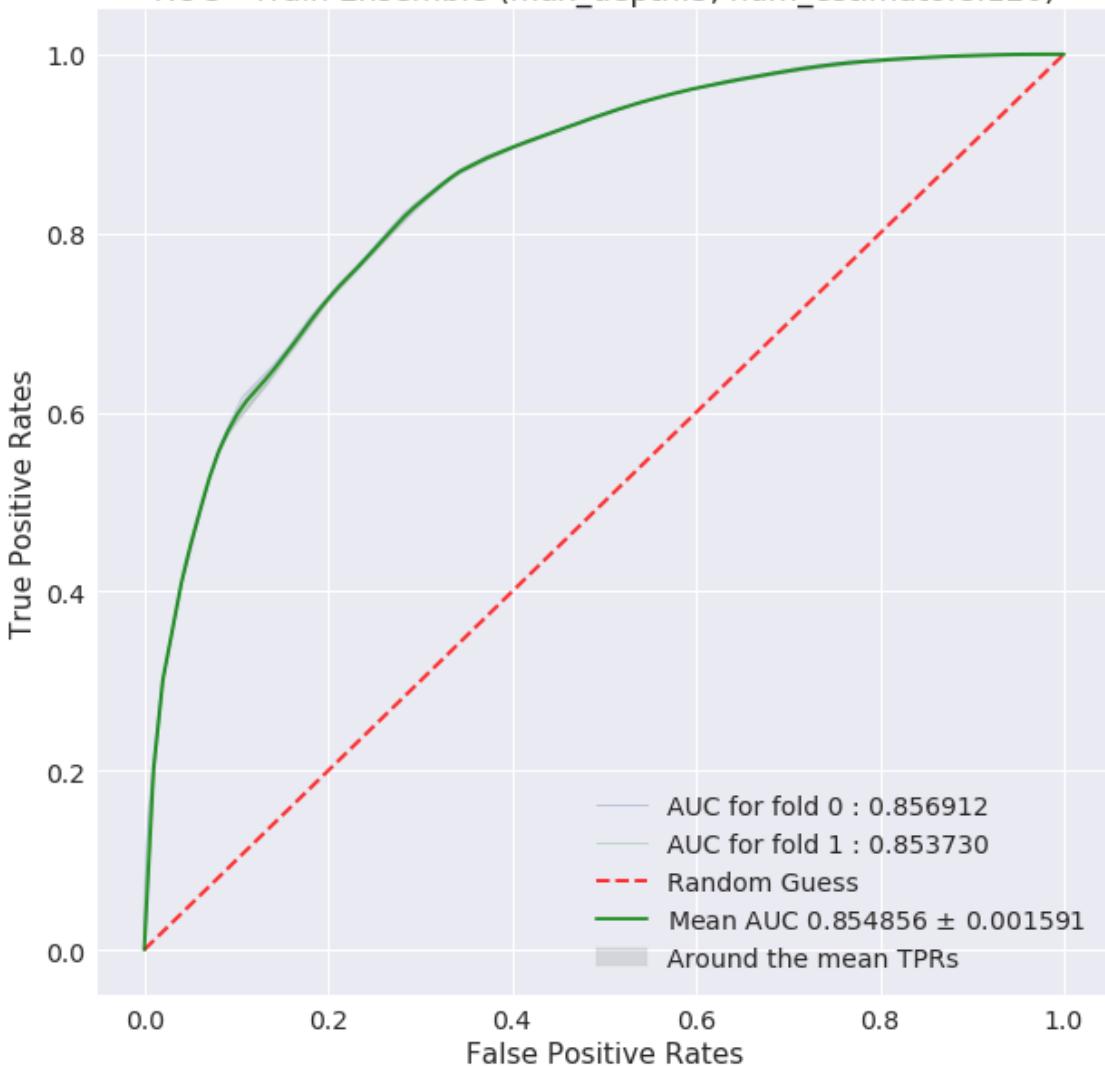
ROC - Train Ensemble (max\_depth:3, num\_estimators:60)



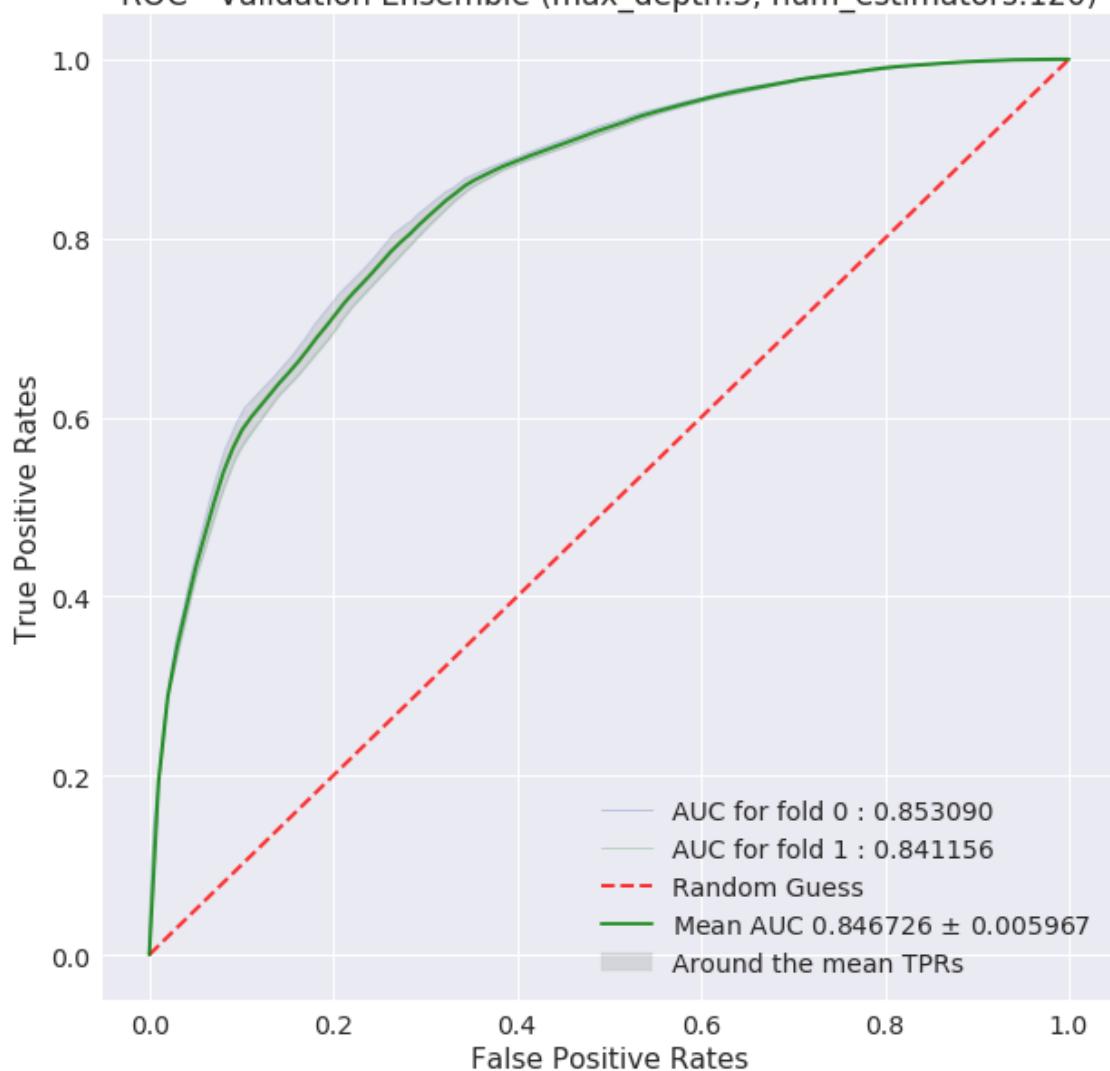
ROC - Validation Ensemble (max\_depth:3, num\_estimators:60)



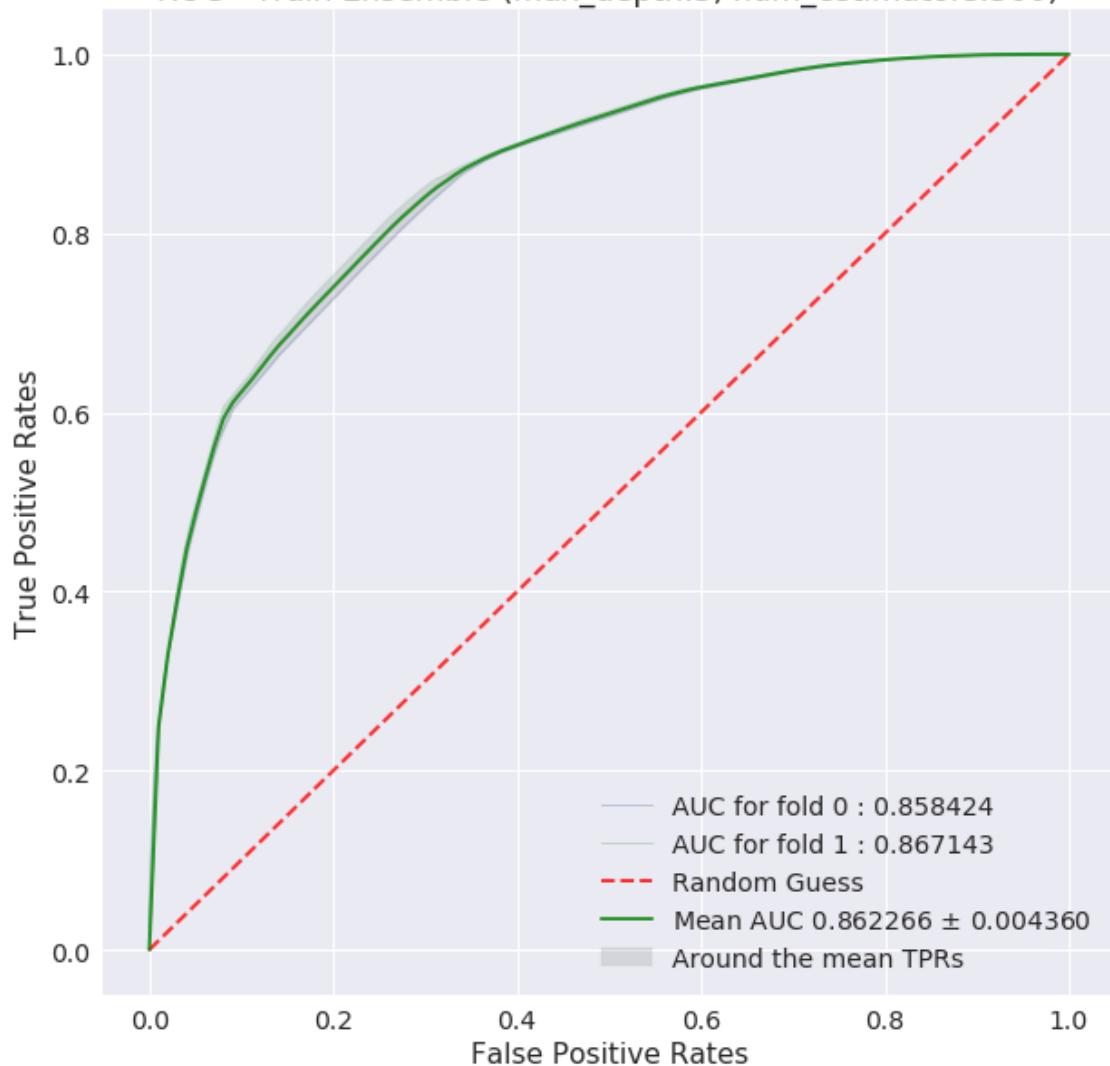
ROC - Train Ensemble (max\_depth:3, num\_estimators:120)



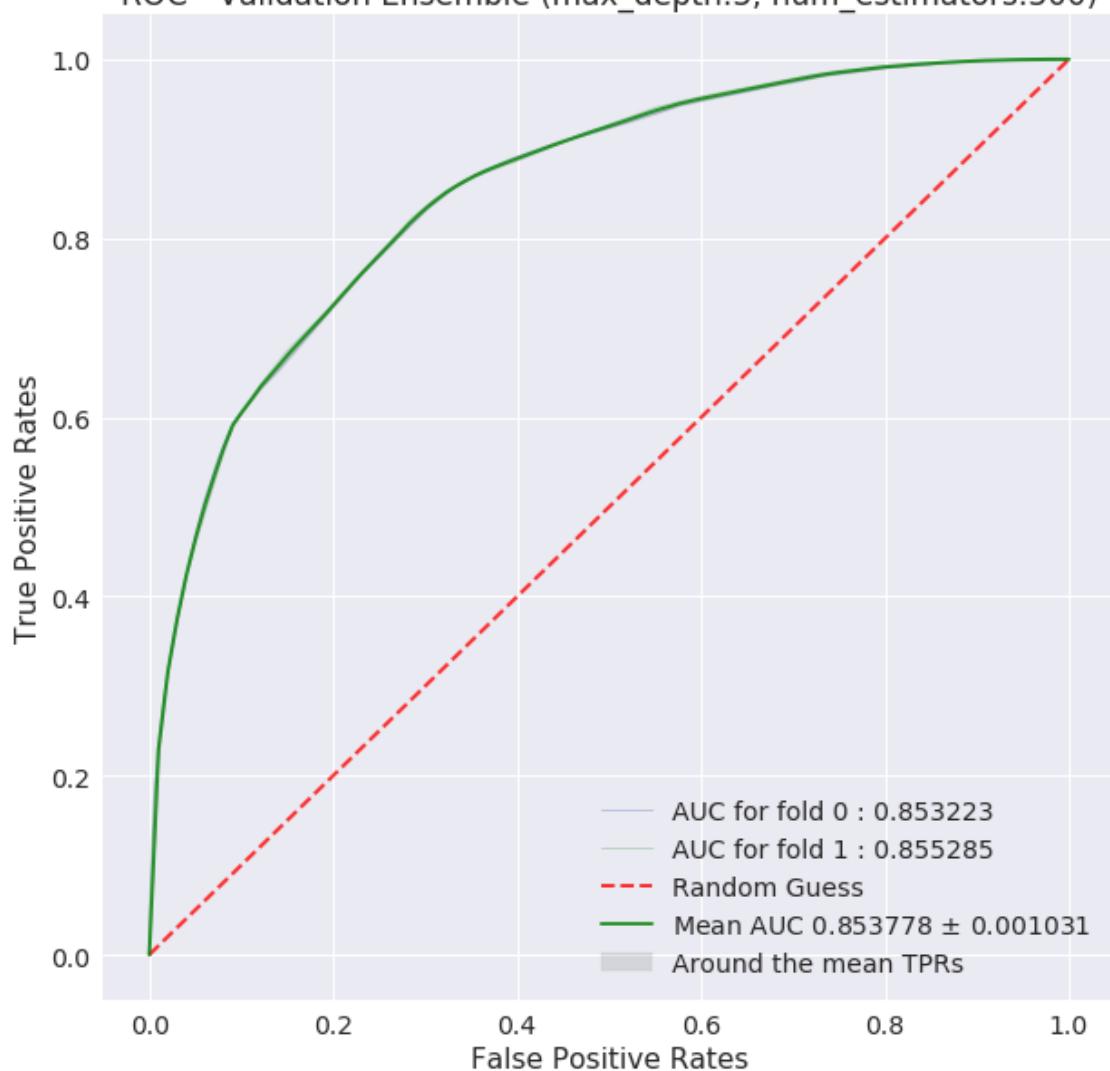
ROC - Validation Ensemble (max\_depth:3, num\_estimators:120)



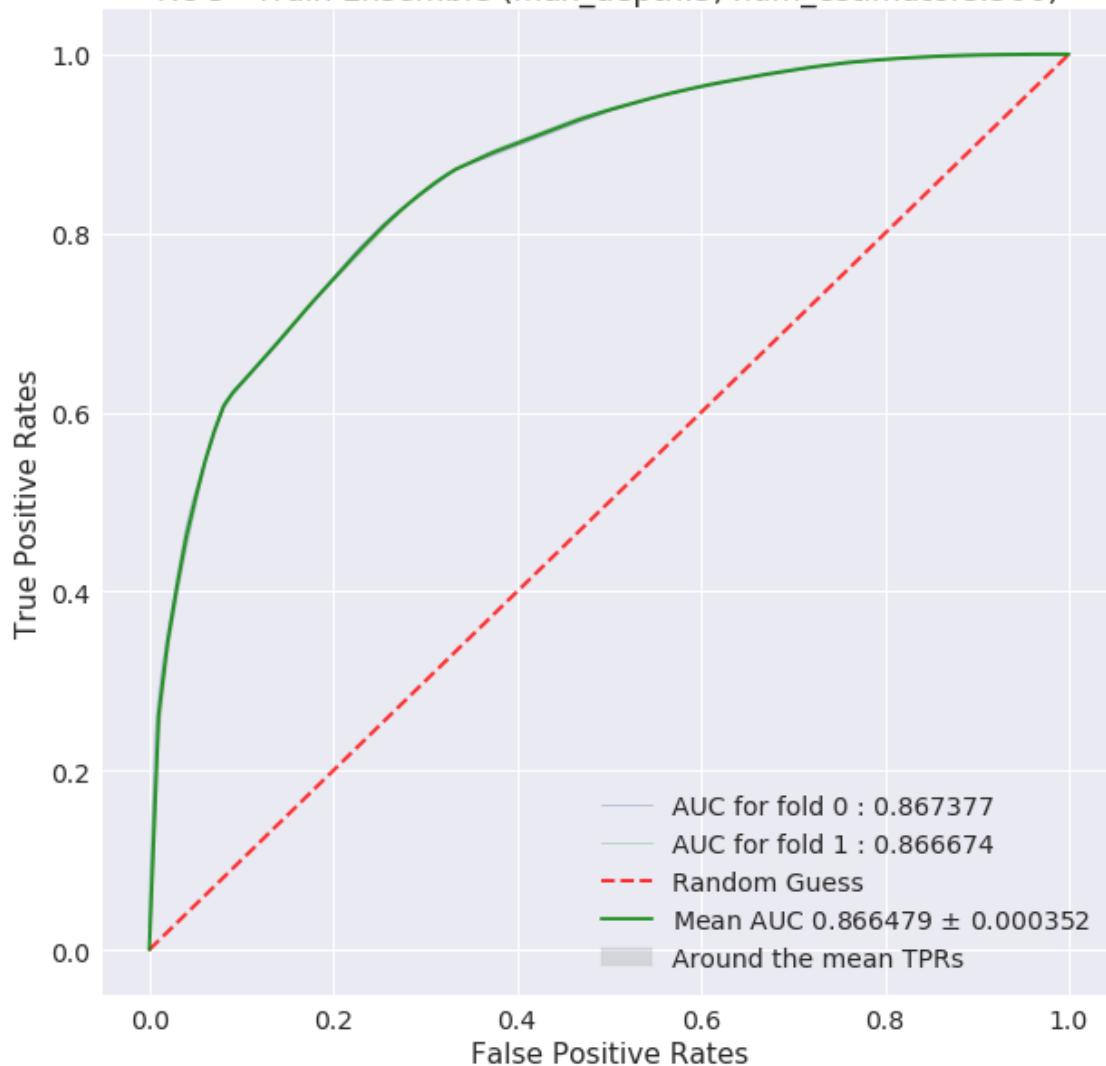
ROC - Train Ensemble (max\_depth:3, num\_estimators:300)



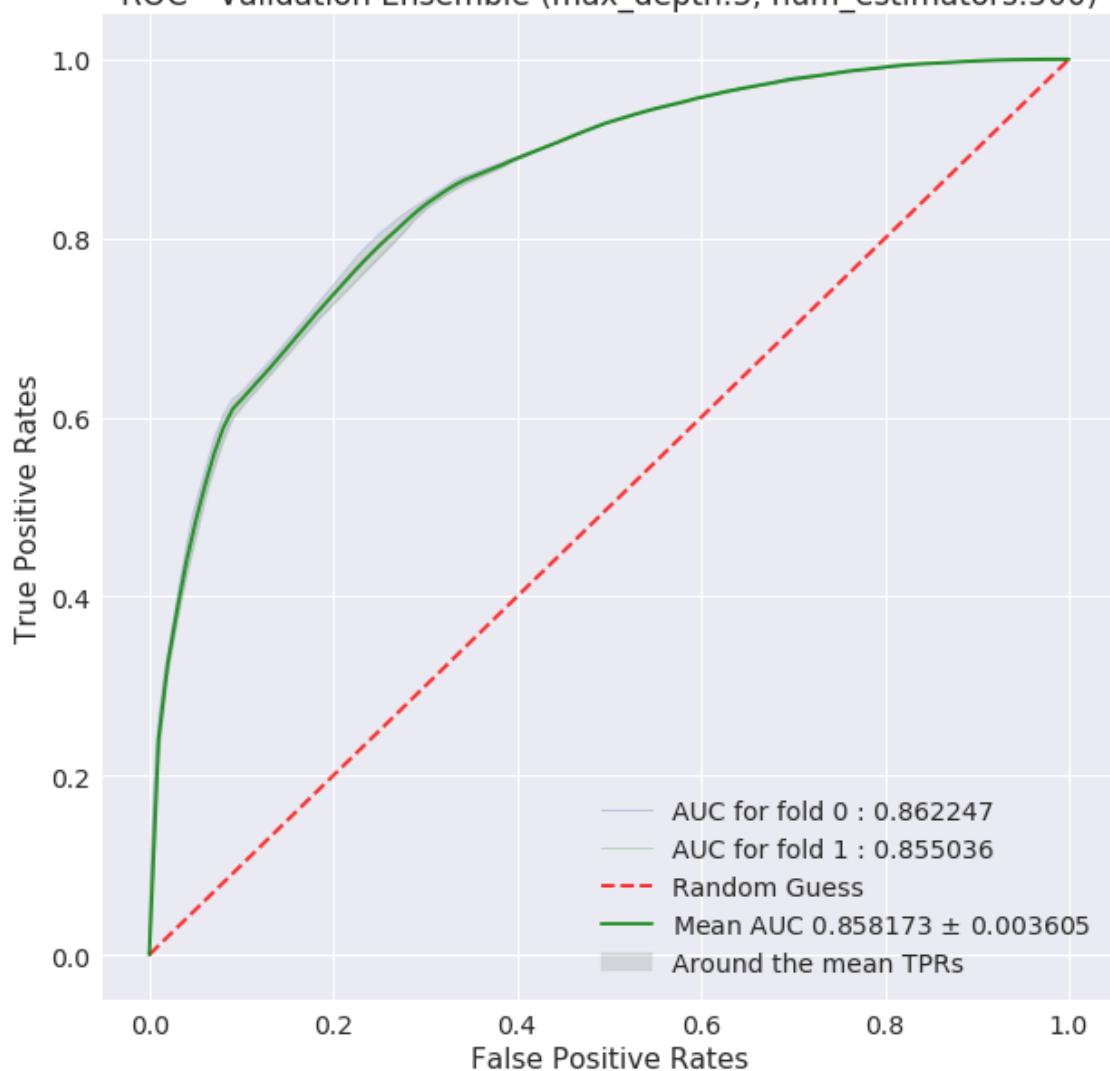
ROC - Validation Ensemble (max\_depth:3, num\_estimators:300)



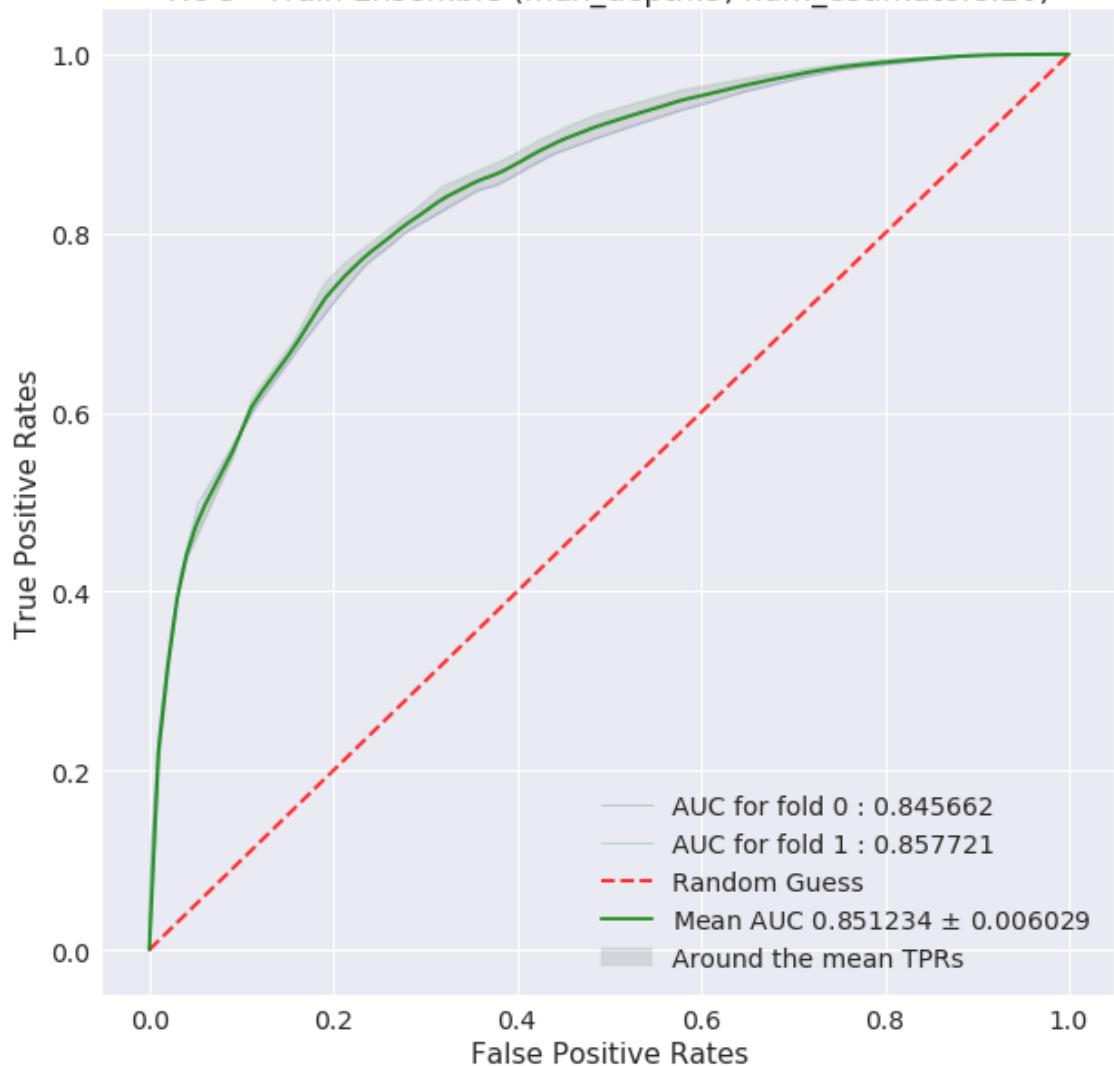
ROC - Train Ensemble (max\_depth:3, num\_estimators:500)



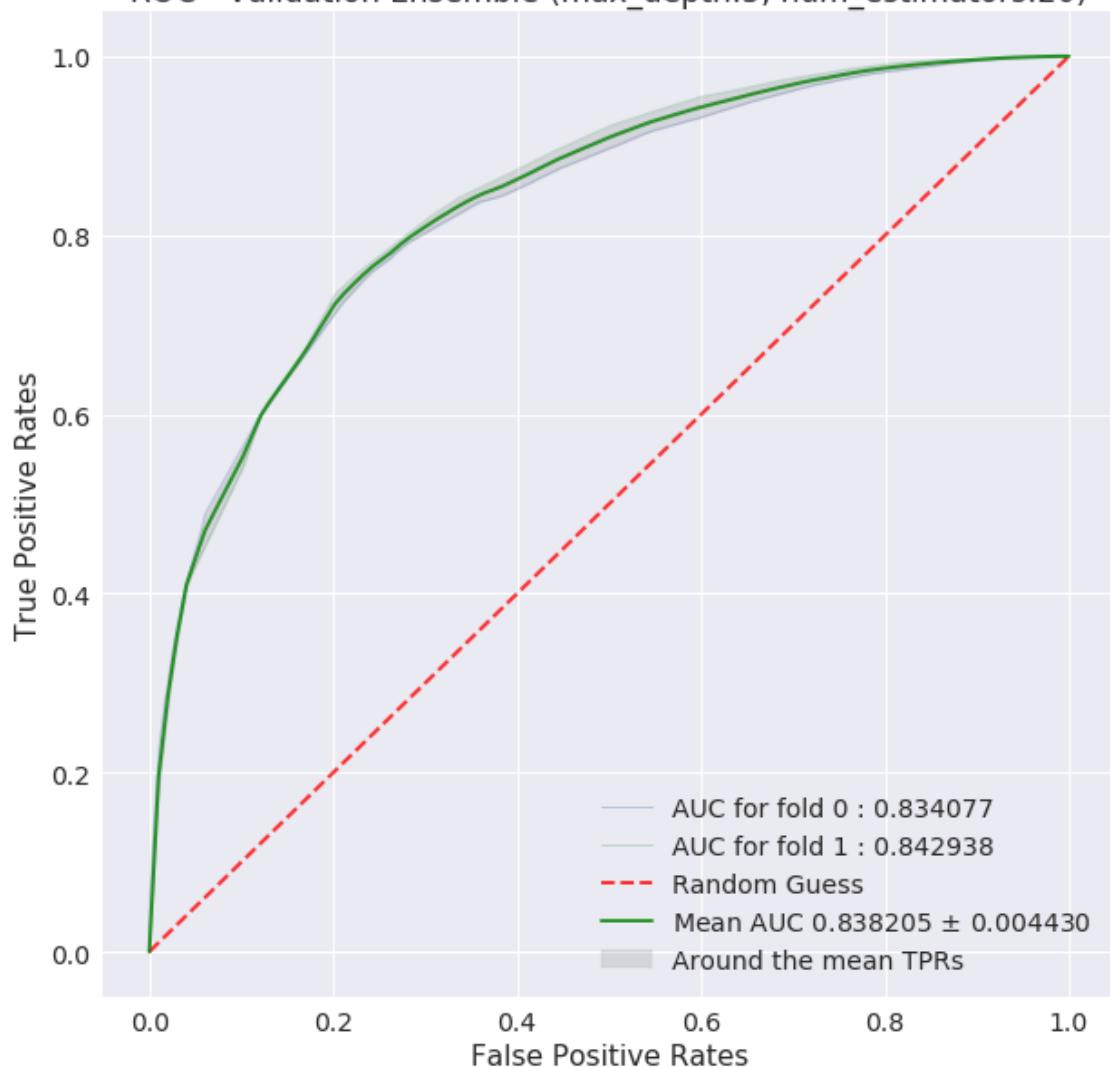
ROC - Validation Ensemble (max\_depth:3, num\_estimators:500)



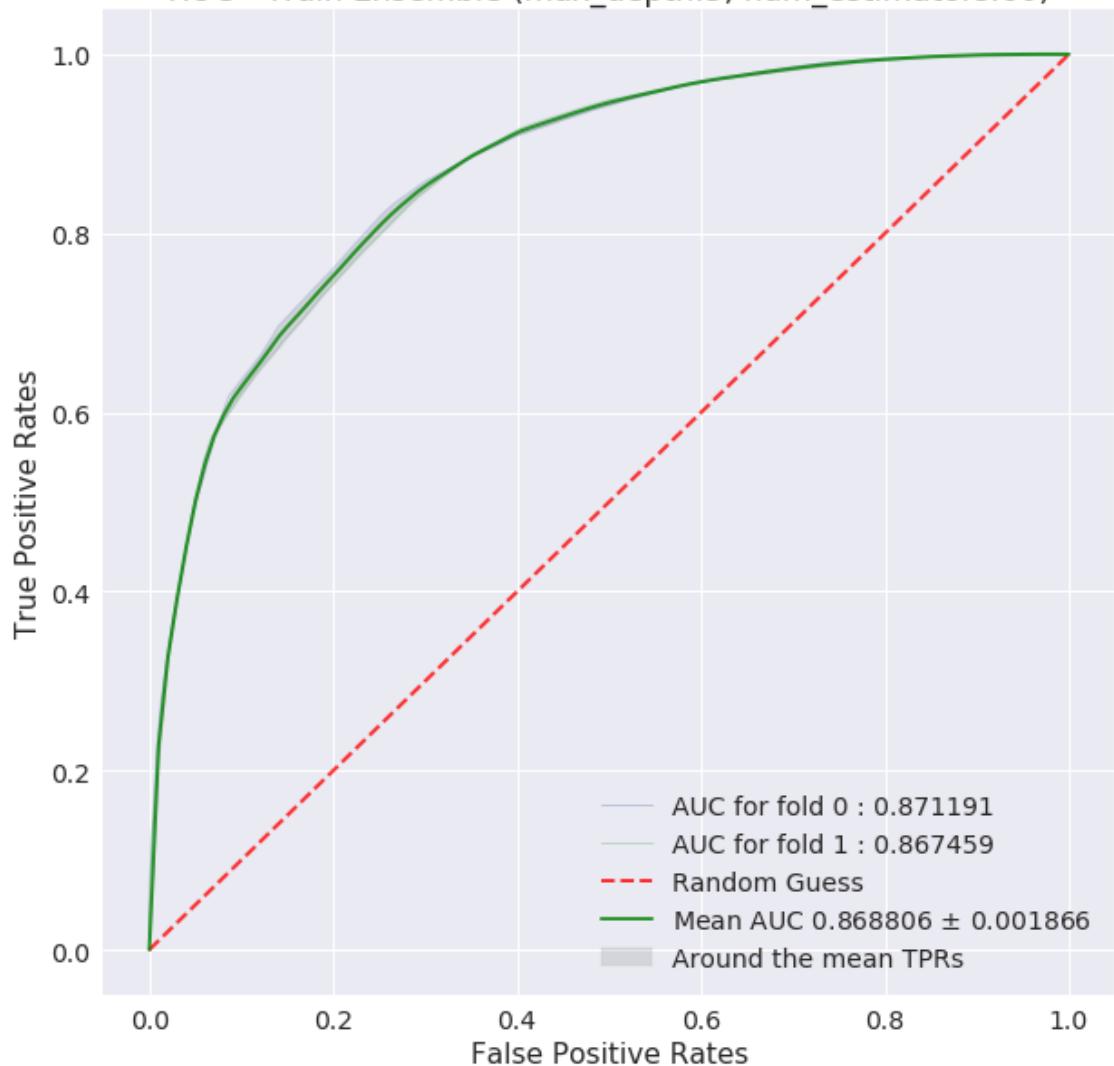
ROC - Train Ensemble (max\_depth:5, num\_estimators:20)



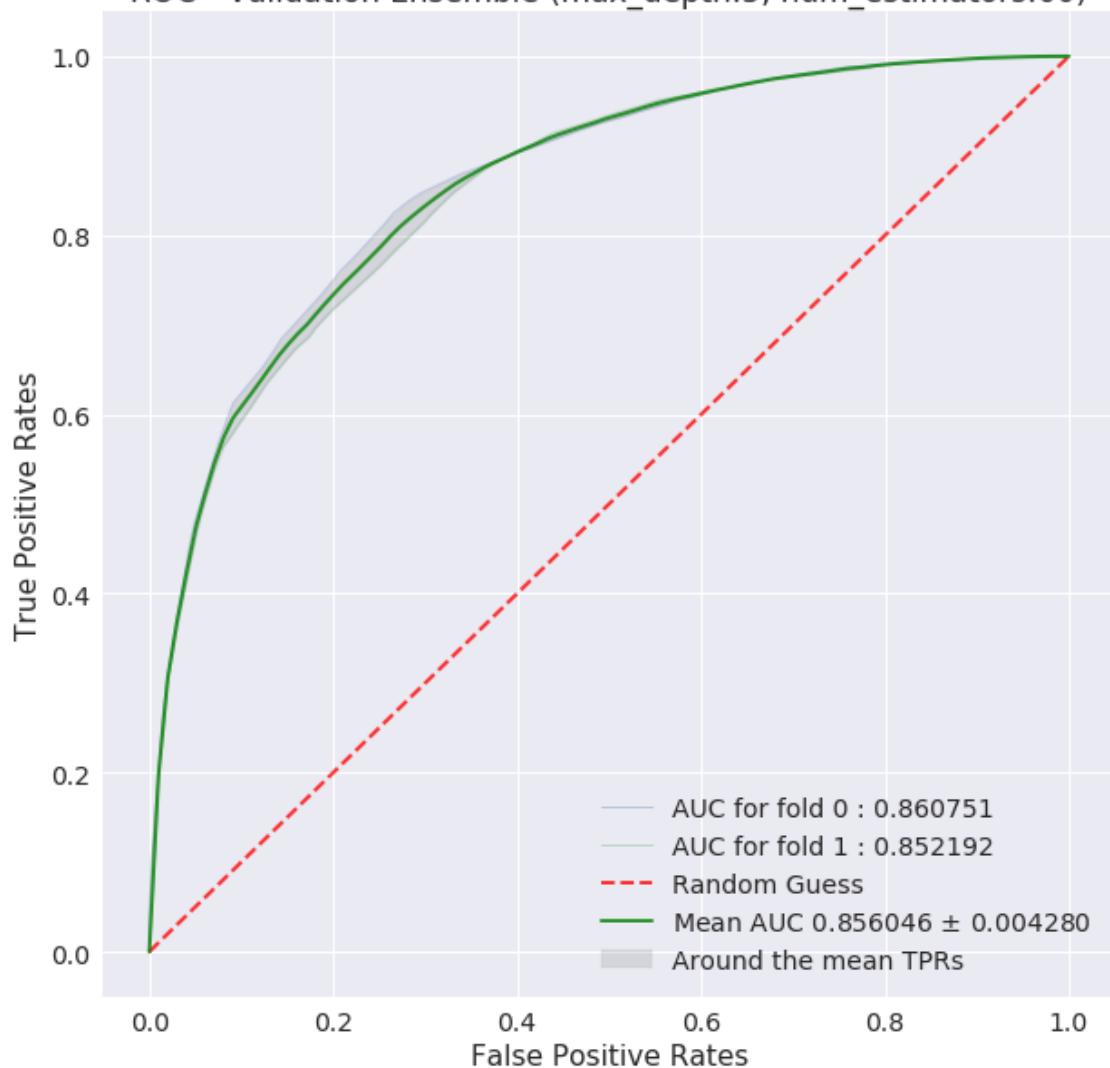
ROC - Validation Ensemble (max\_depth:5, num\_estimators:20)



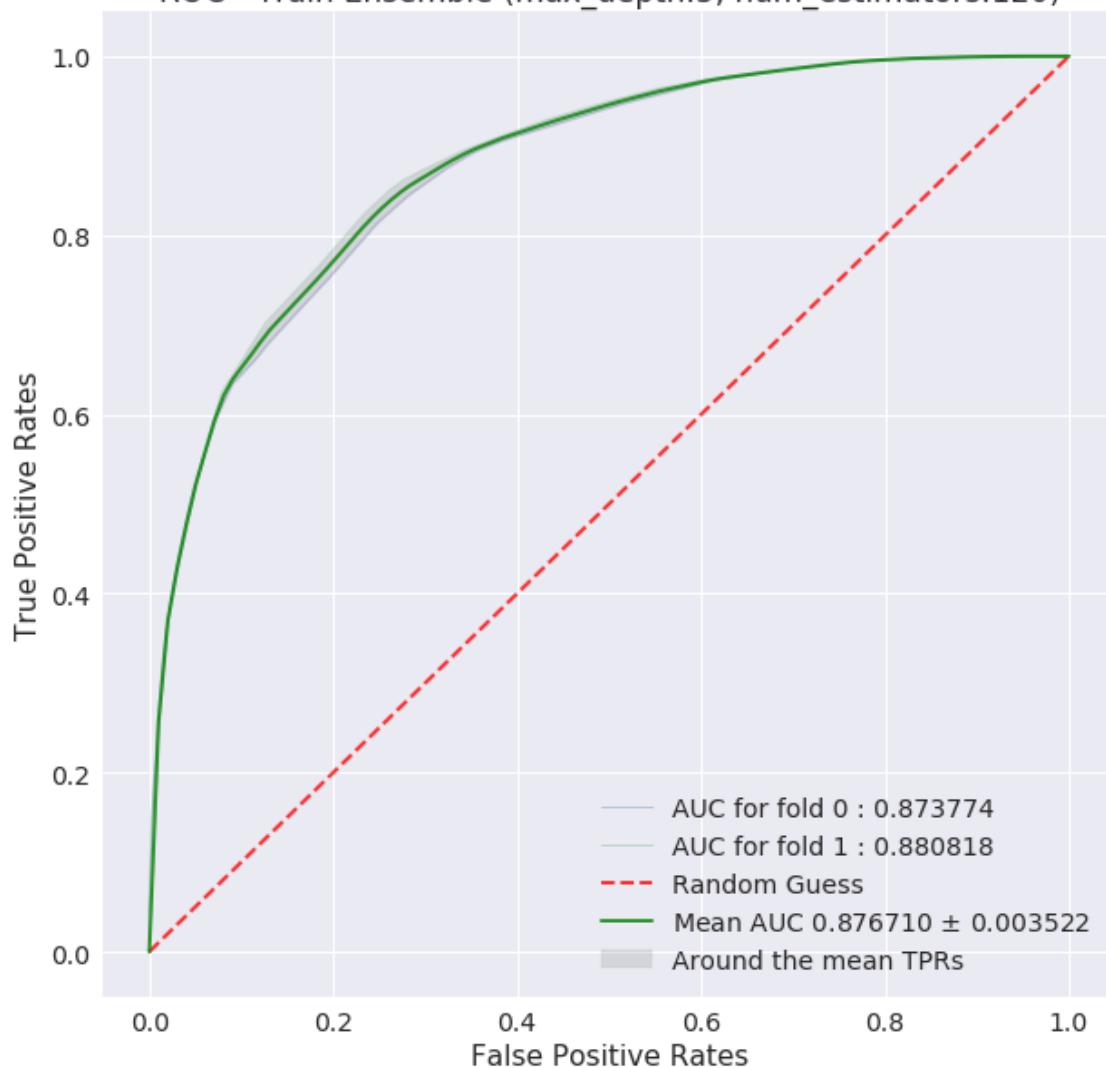
ROC - Train Ensemble (max\_depth:5, num\_estimators:60)



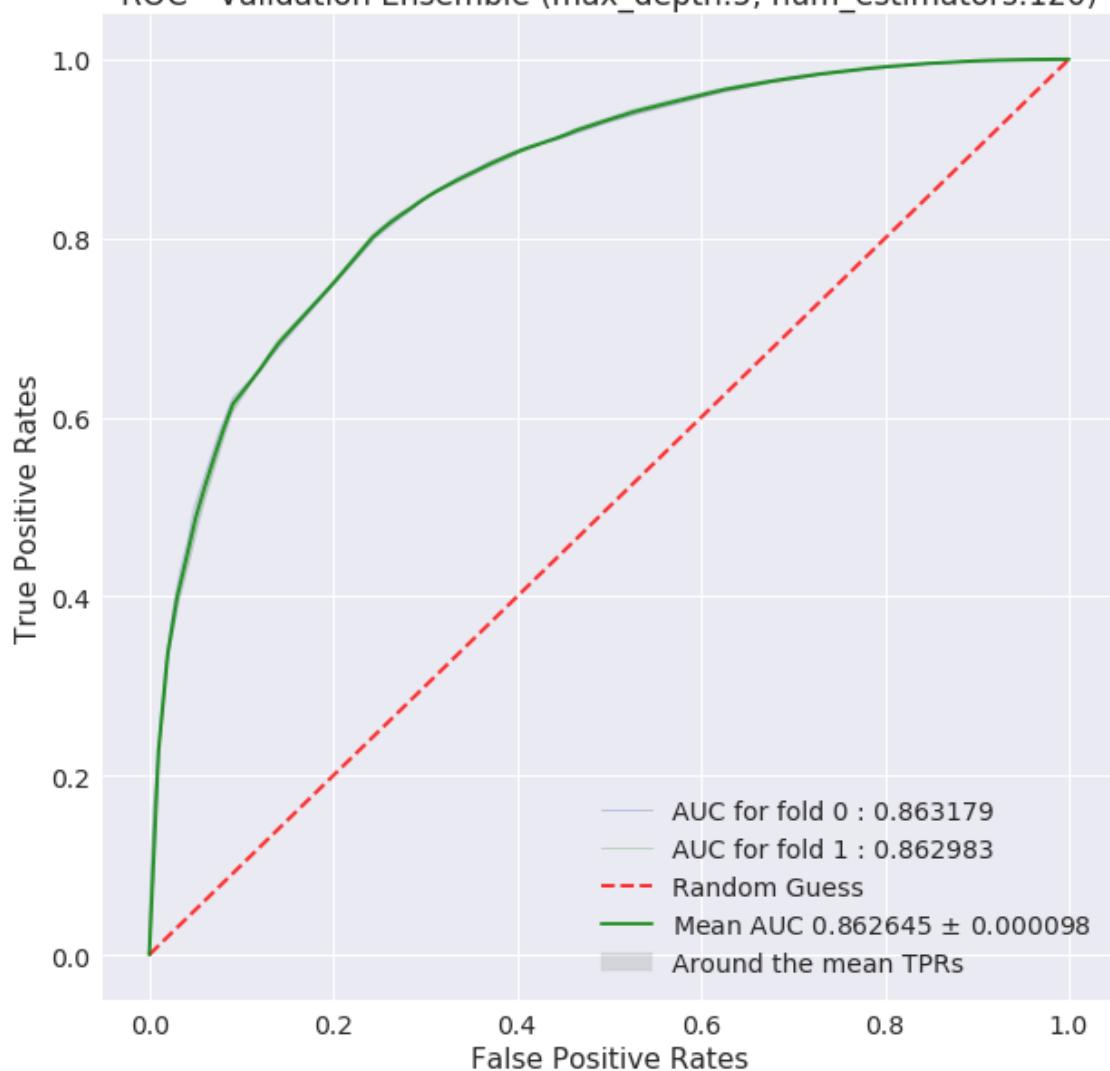
ROC - Validation Ensemble (max\_depth:5, num\_estimators:60)



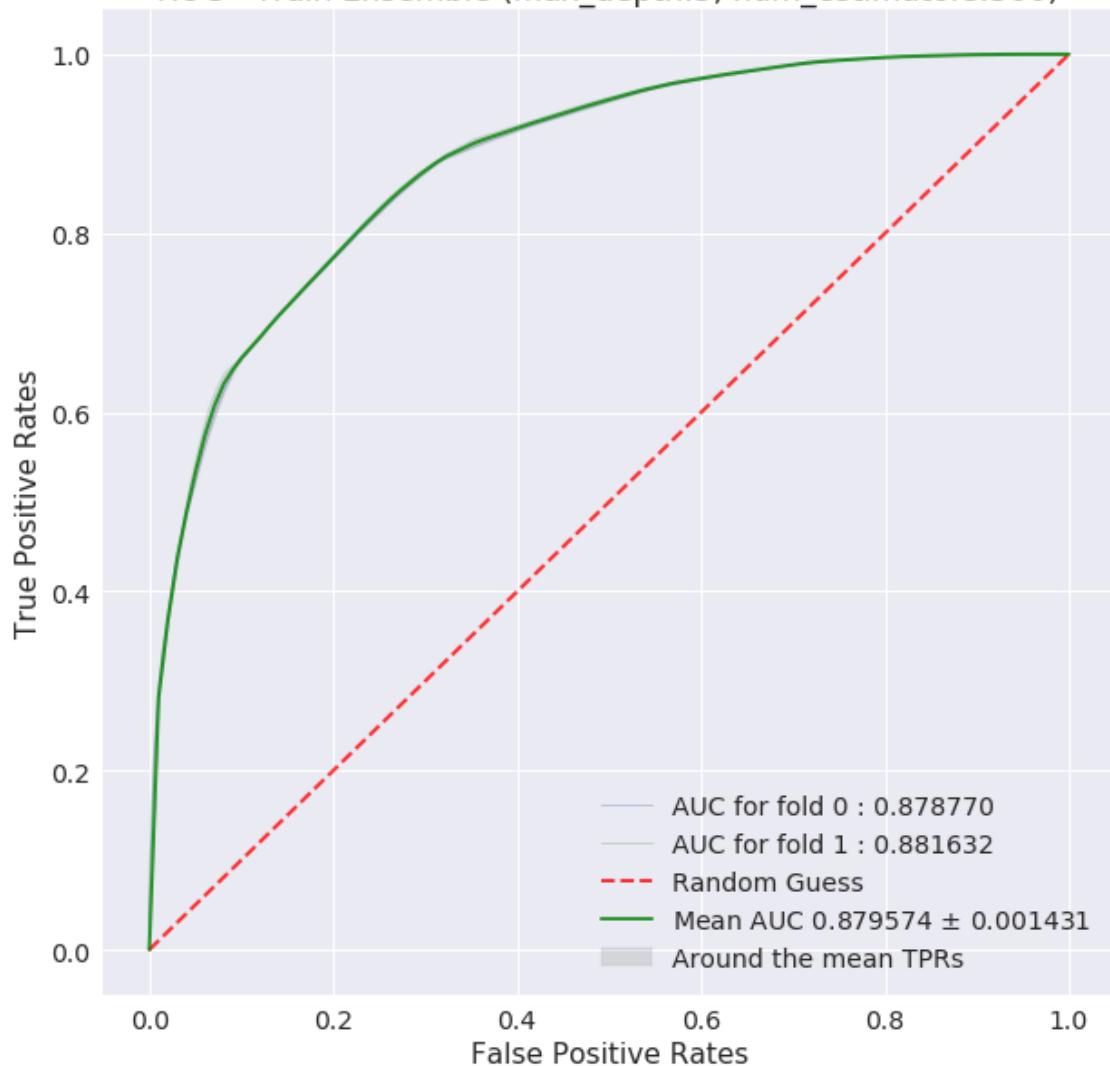
ROC - Train Ensemble (max\_depth:5, num\_estimators:120)

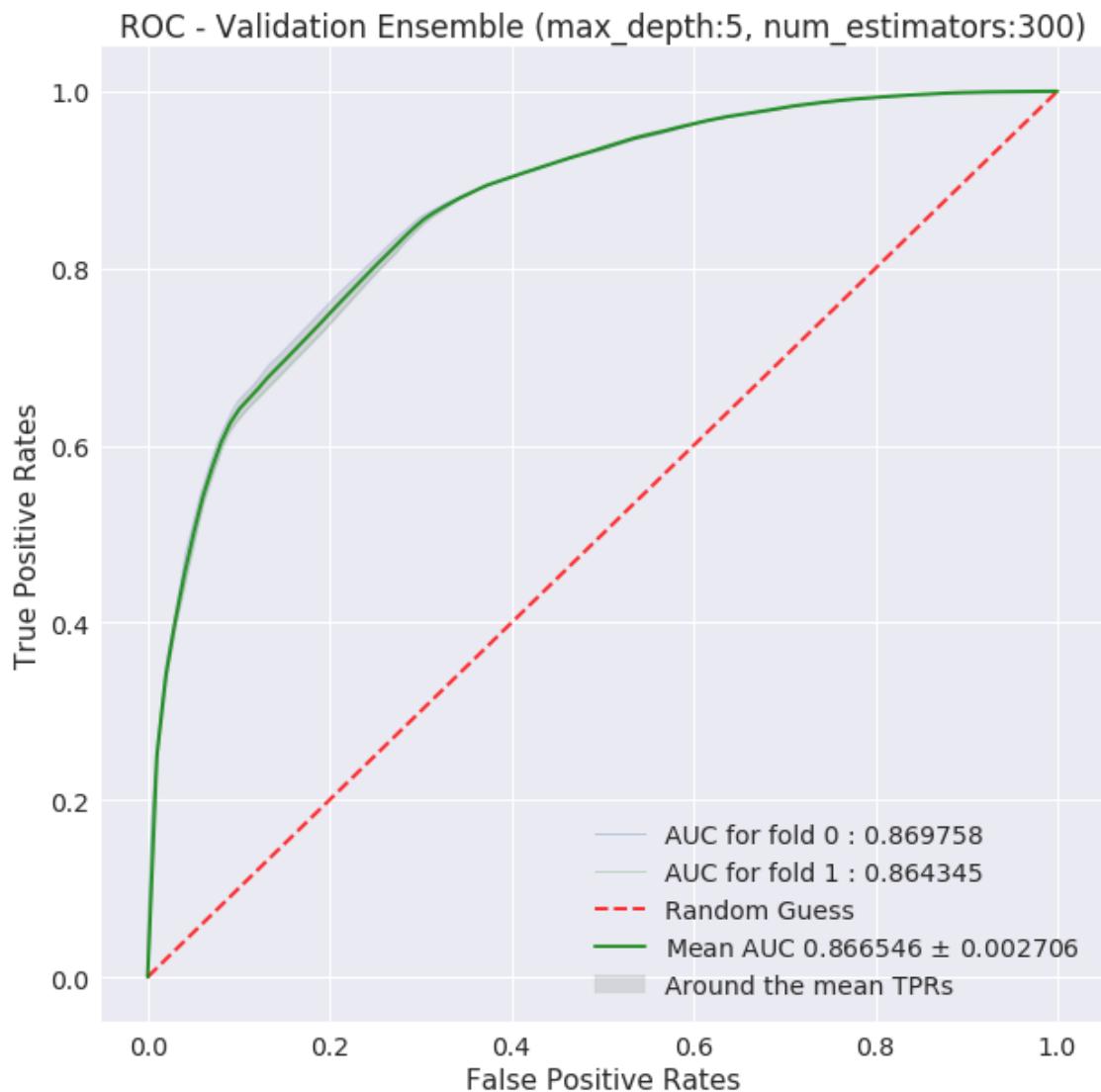


ROC - Validation Ensemble (max\_depth:5, num\_estimators:120)

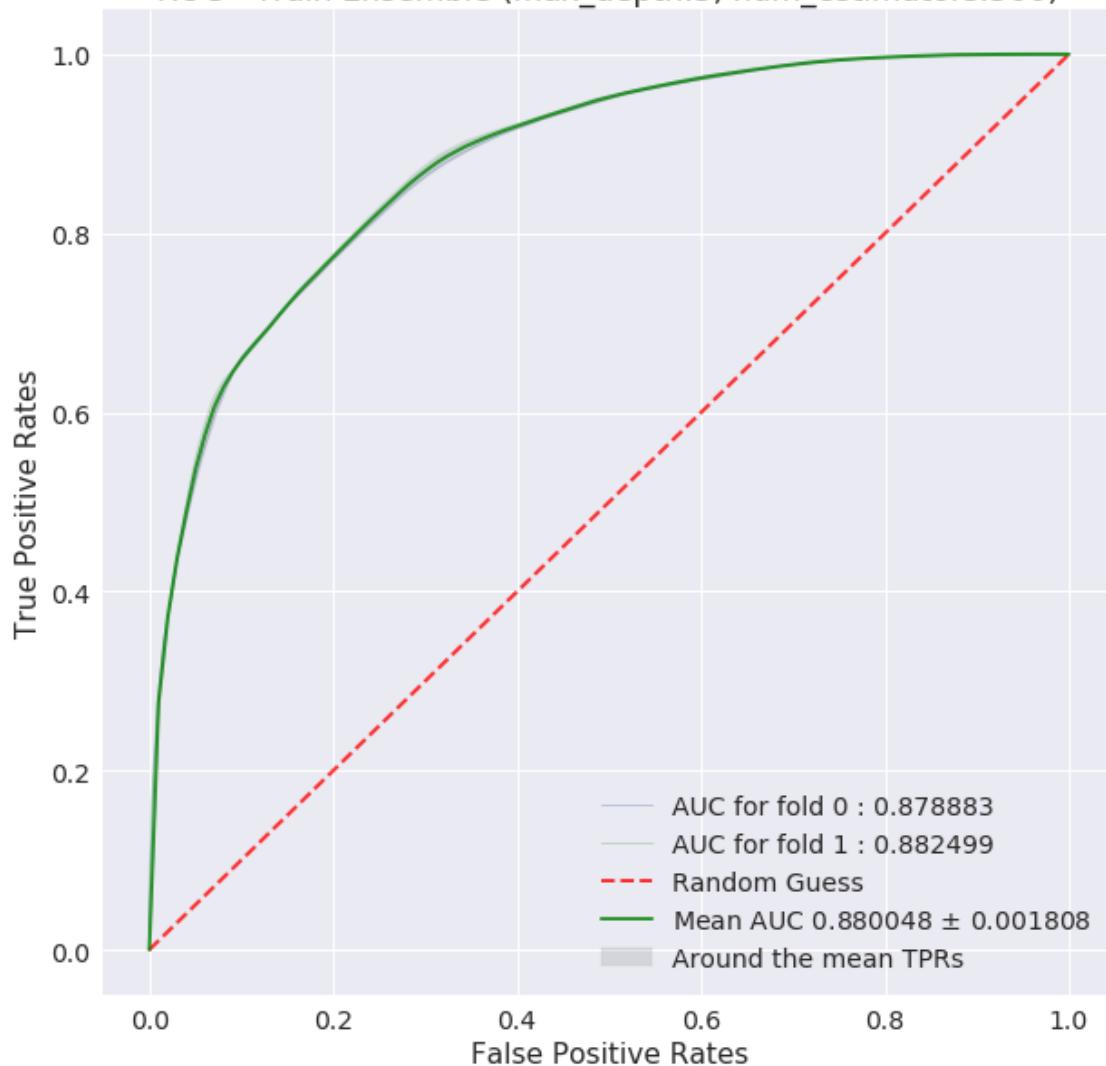


ROC - Train Ensemble (max\_depth:5, num\_estimators:300)

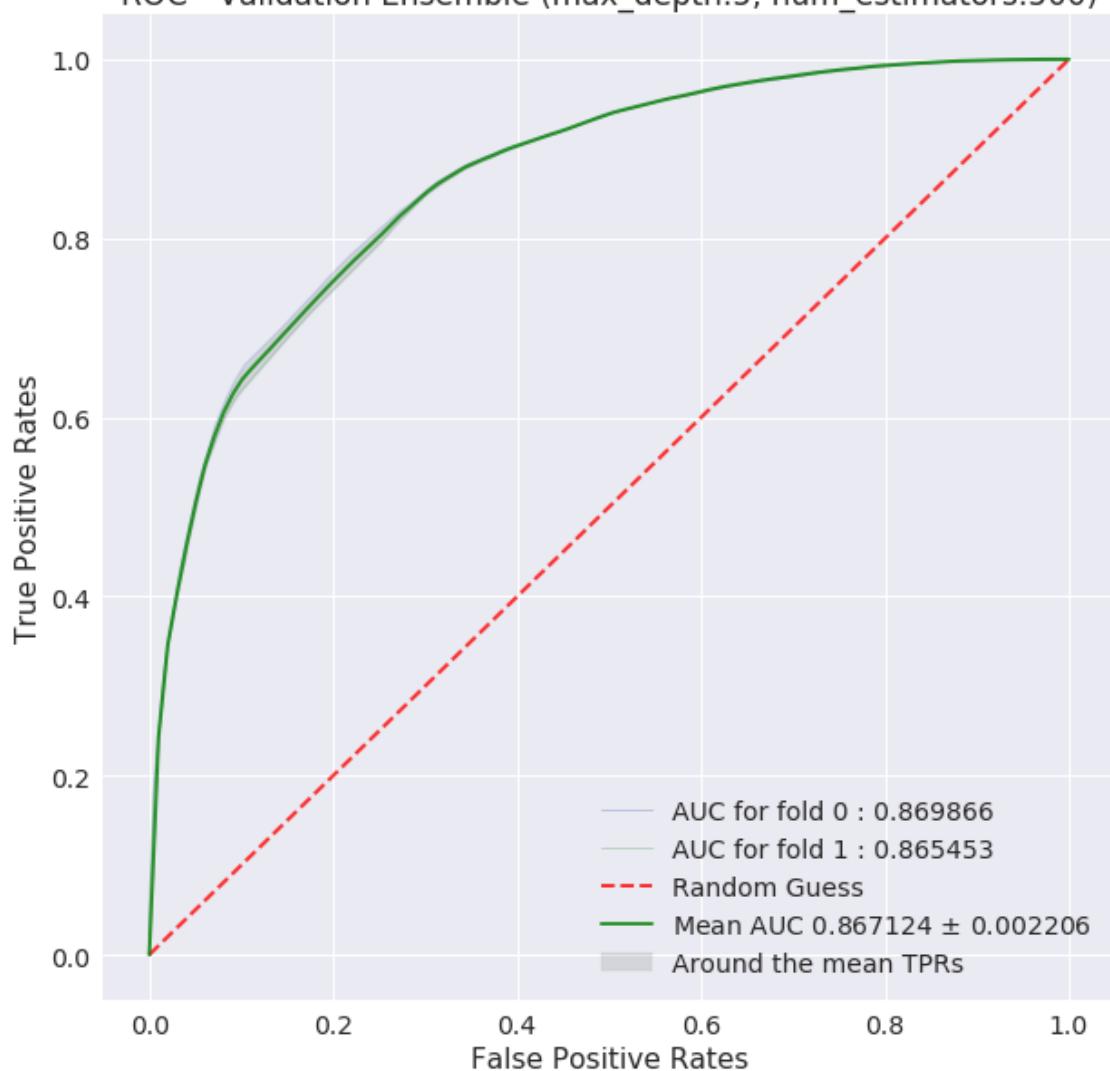




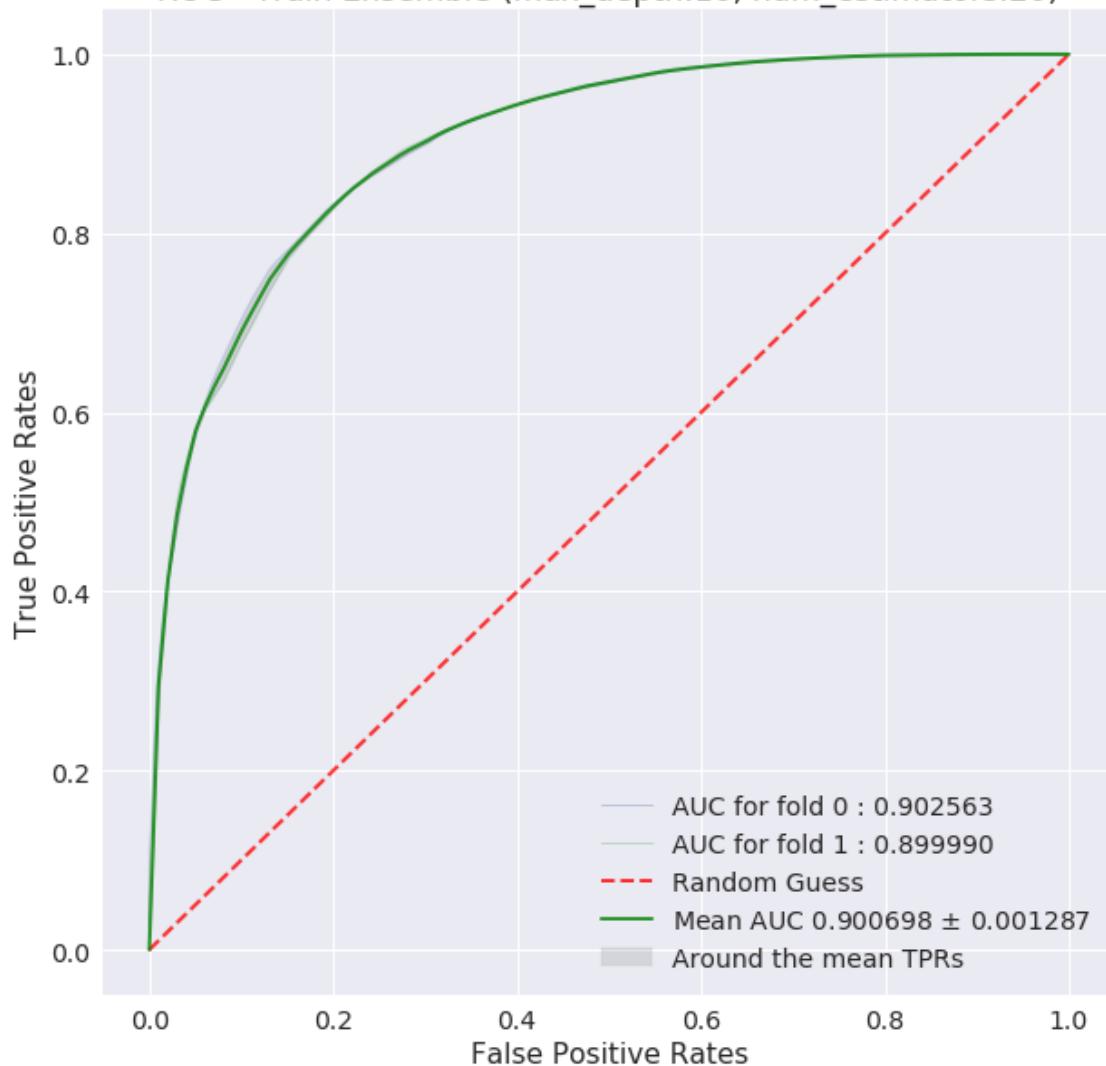
ROC - Train Ensemble (max\_depth:5, num\_estimators:500)



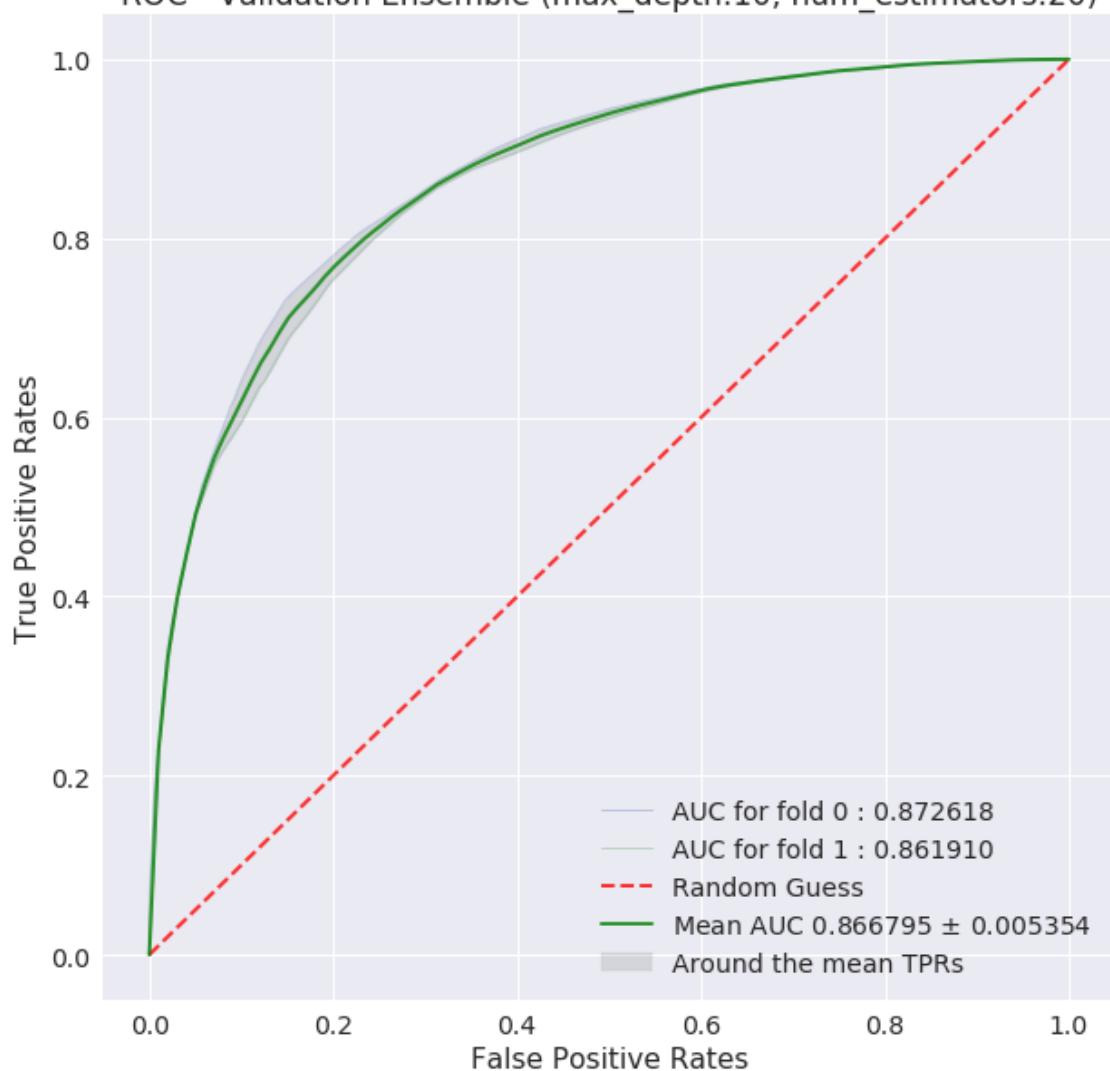
ROC - Validation Ensemble (max\_depth:5, num\_estimators:500)



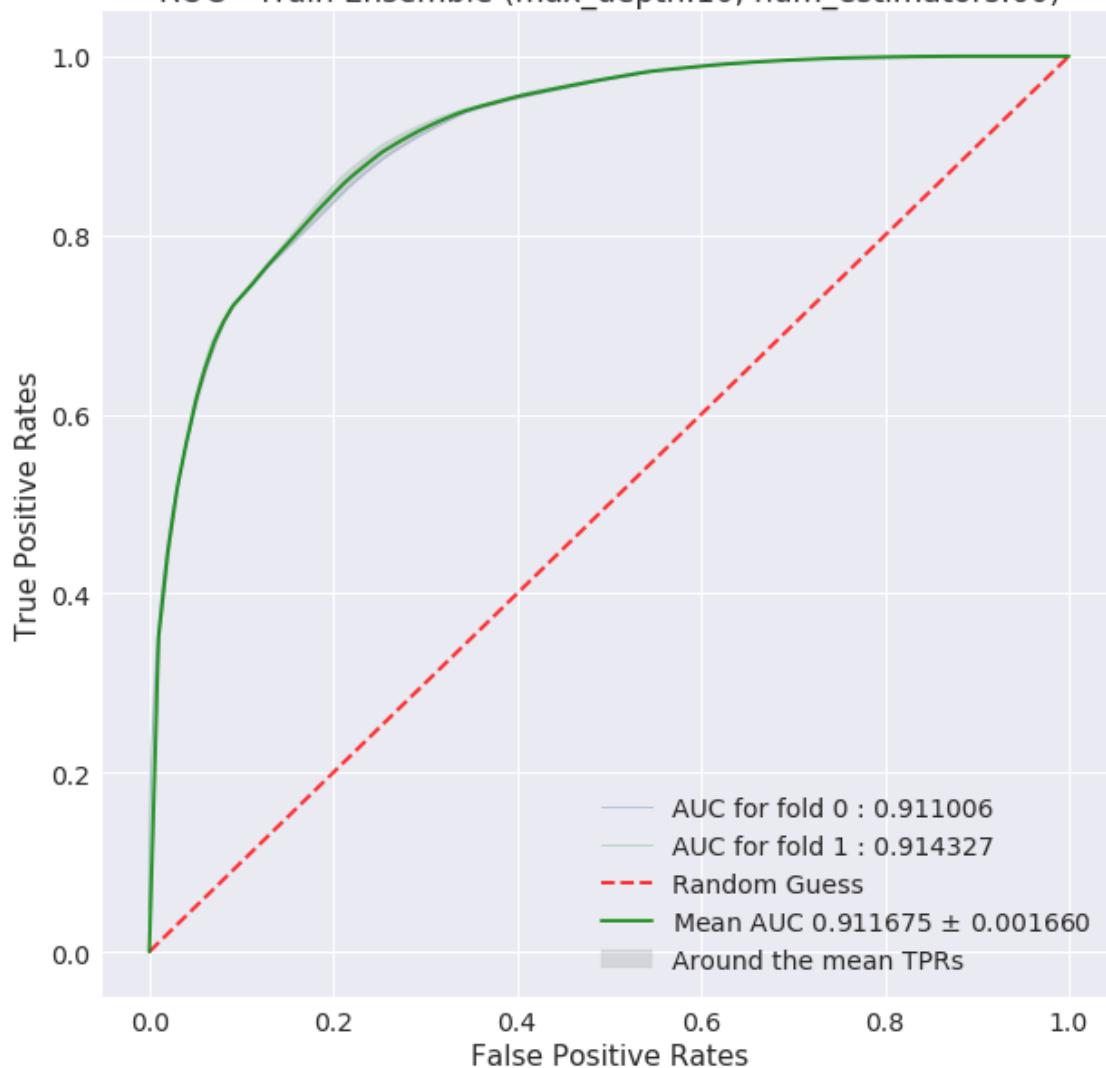
ROC - Train Ensemble (max\_depth:10, num\_estimators:20)



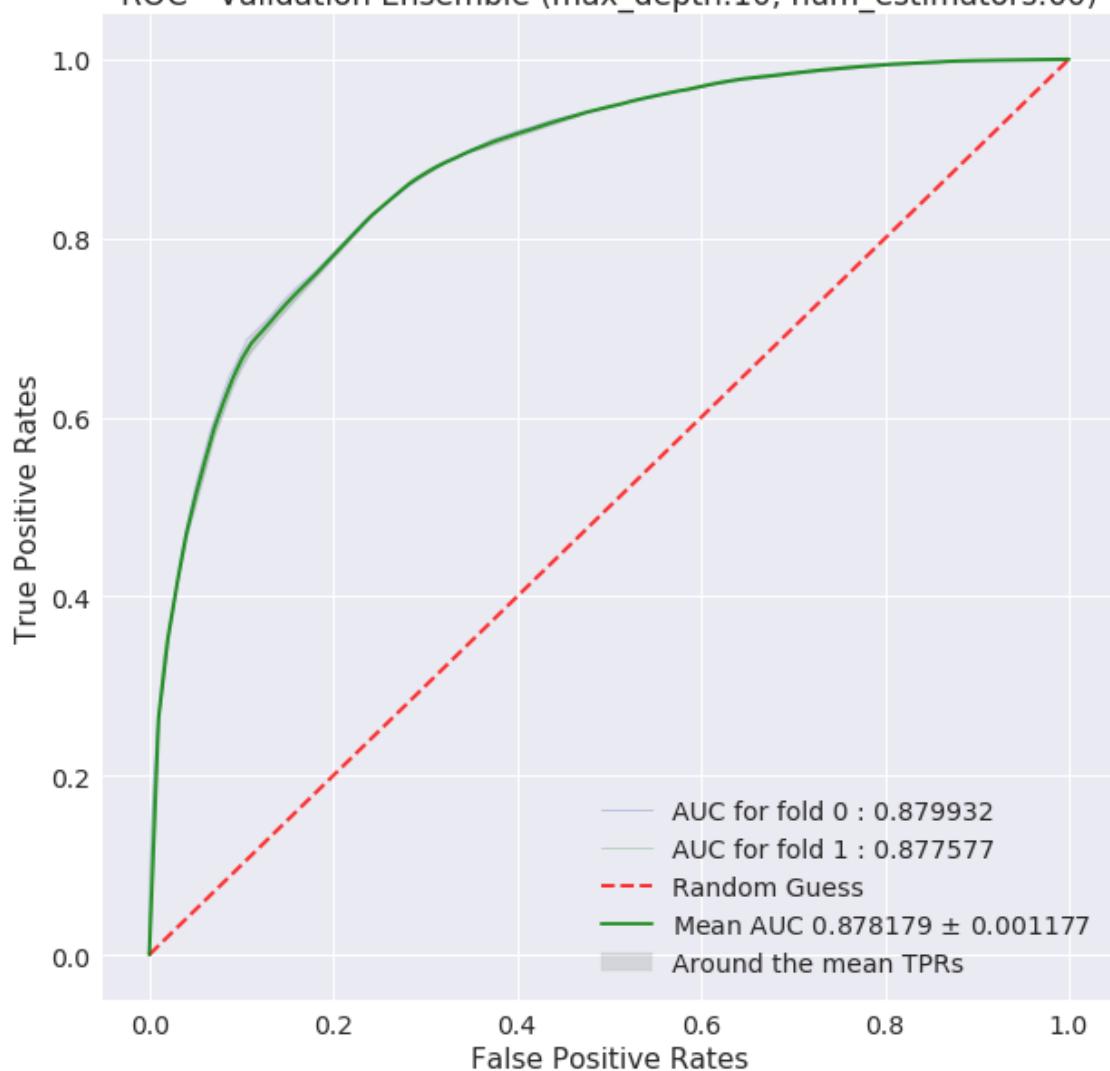
ROC - Validation Ensemble (max\_depth:10, num\_estimators:20)



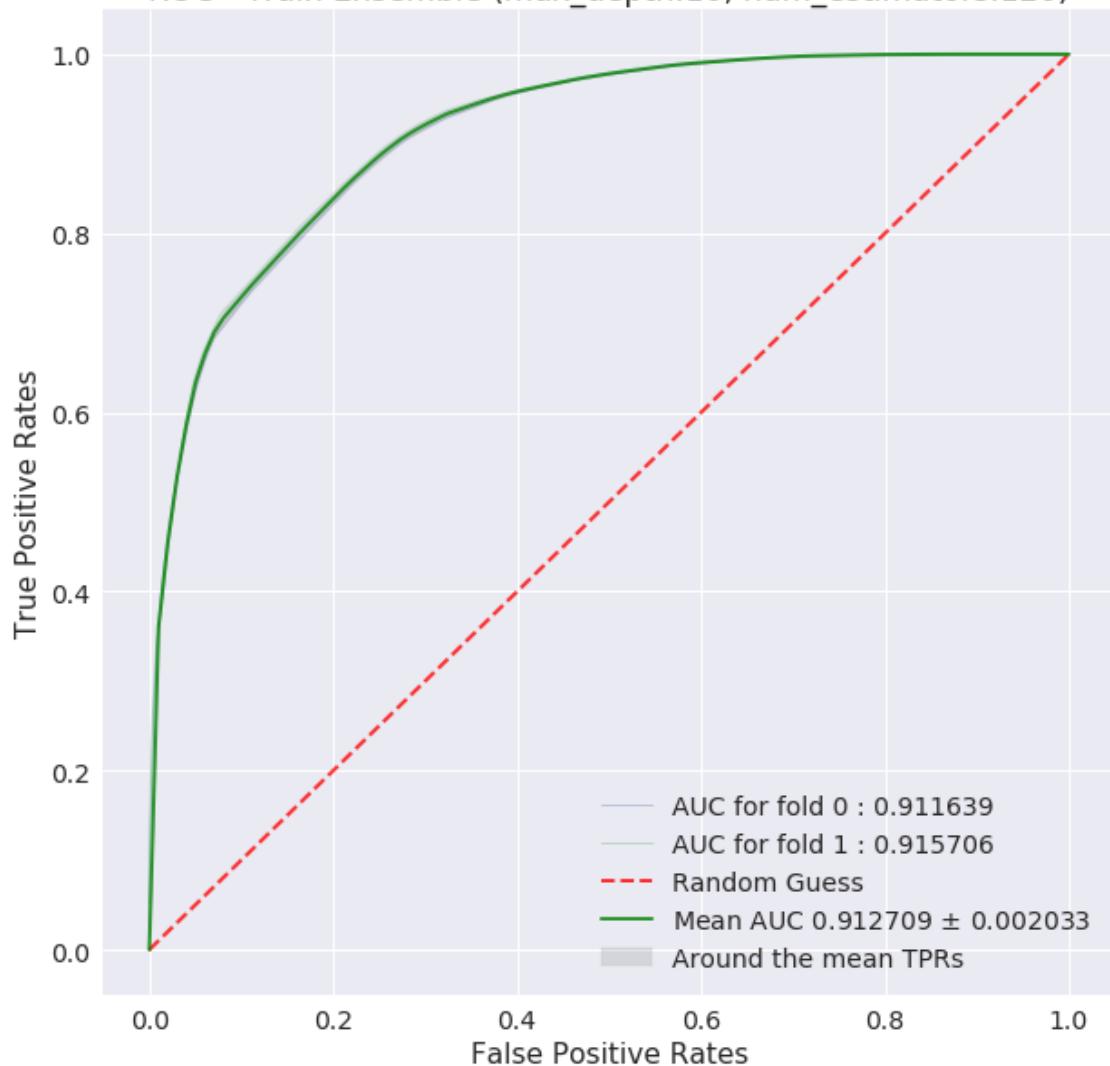
ROC - Train Ensemble (max\_depth:10, num\_estimators:60)



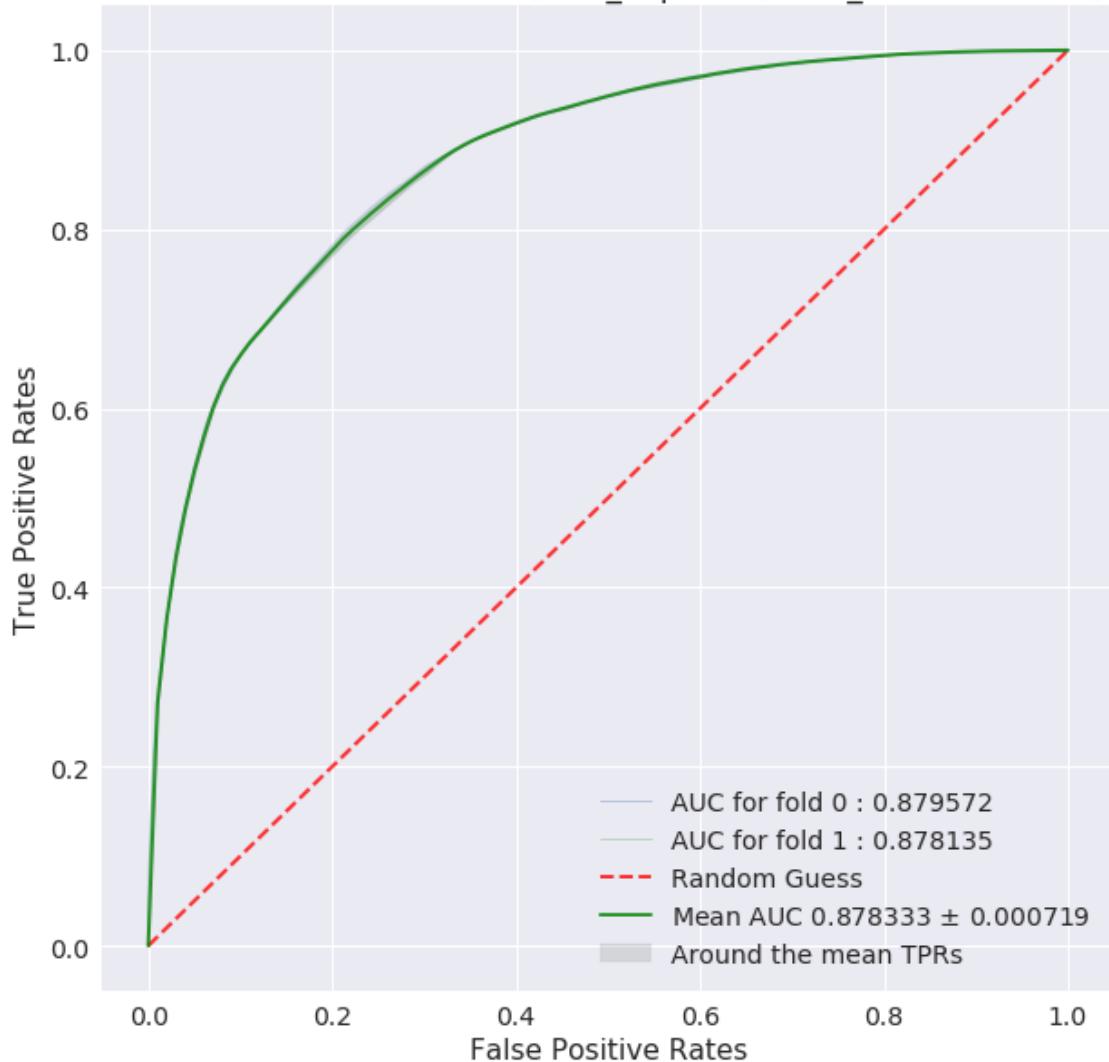
ROC - Validation Ensemble (max\_depth:10, num\_estimators:60)



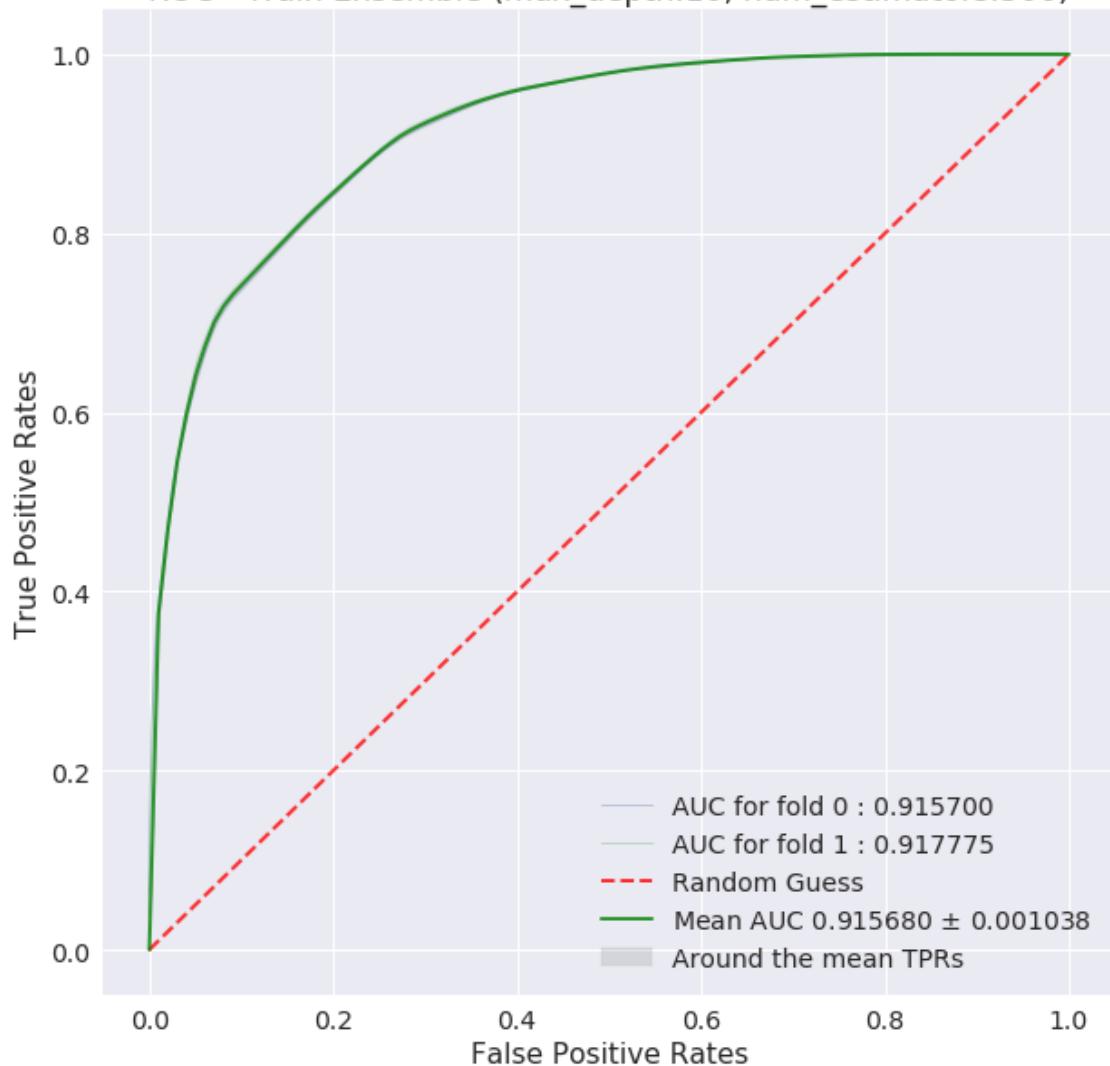
ROC - Train Ensemble (max\_depth:10, num\_estimators:120)



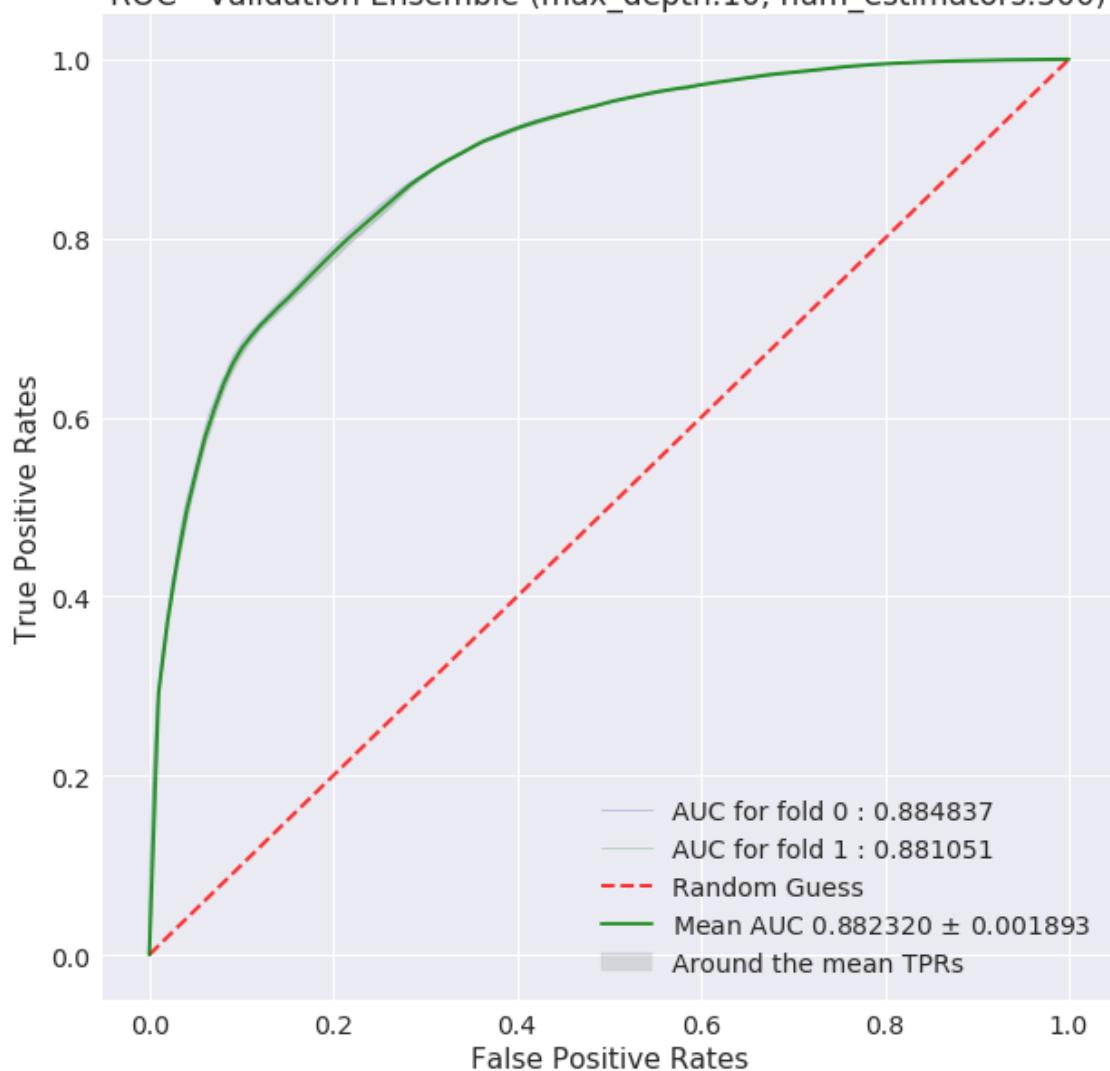
ROC - Validation Ensemble (max\_depth:10, num\_estimators:120)



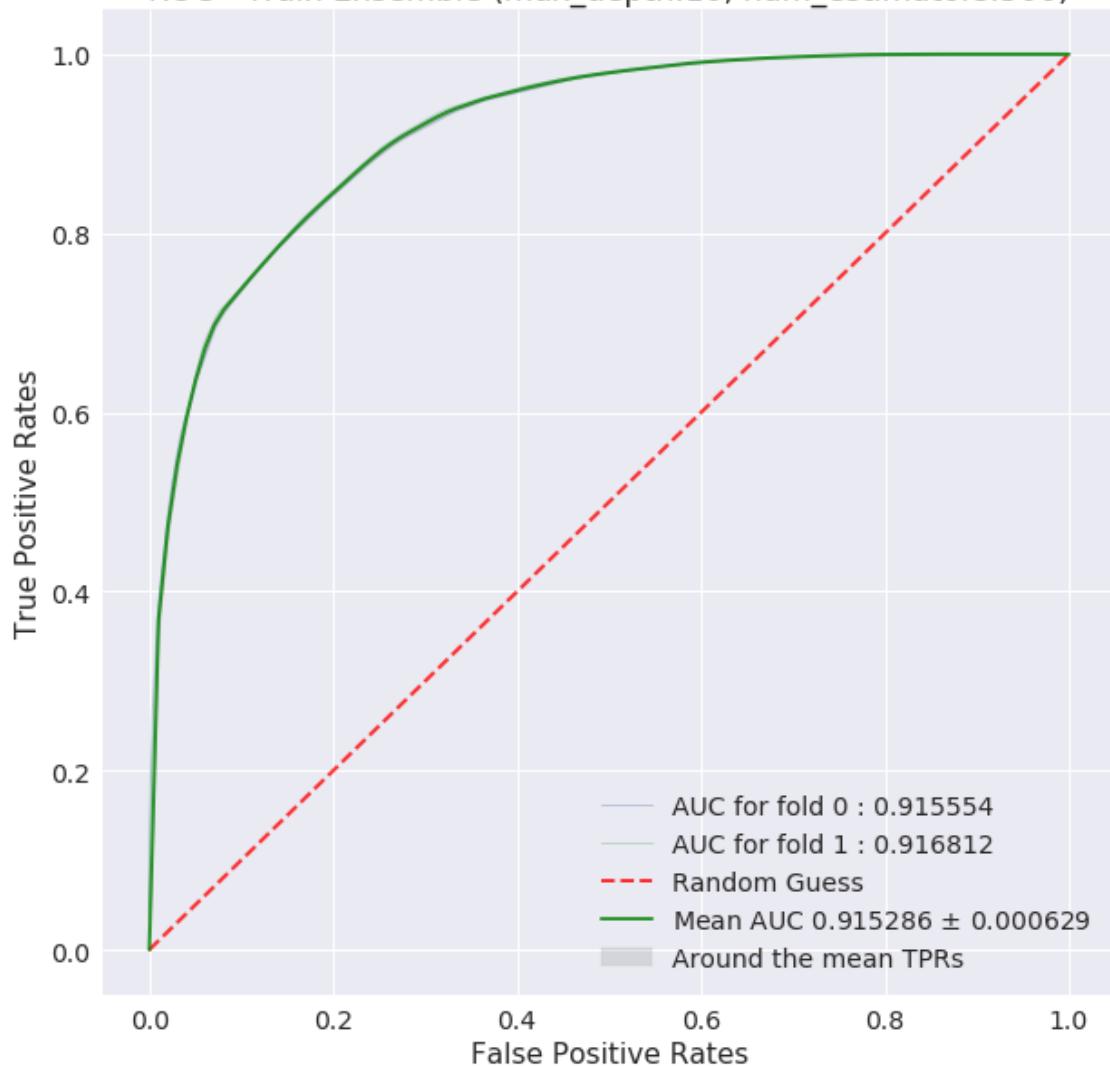
ROC - Train Ensemble (max\_depth:10, num\_estimators:300)



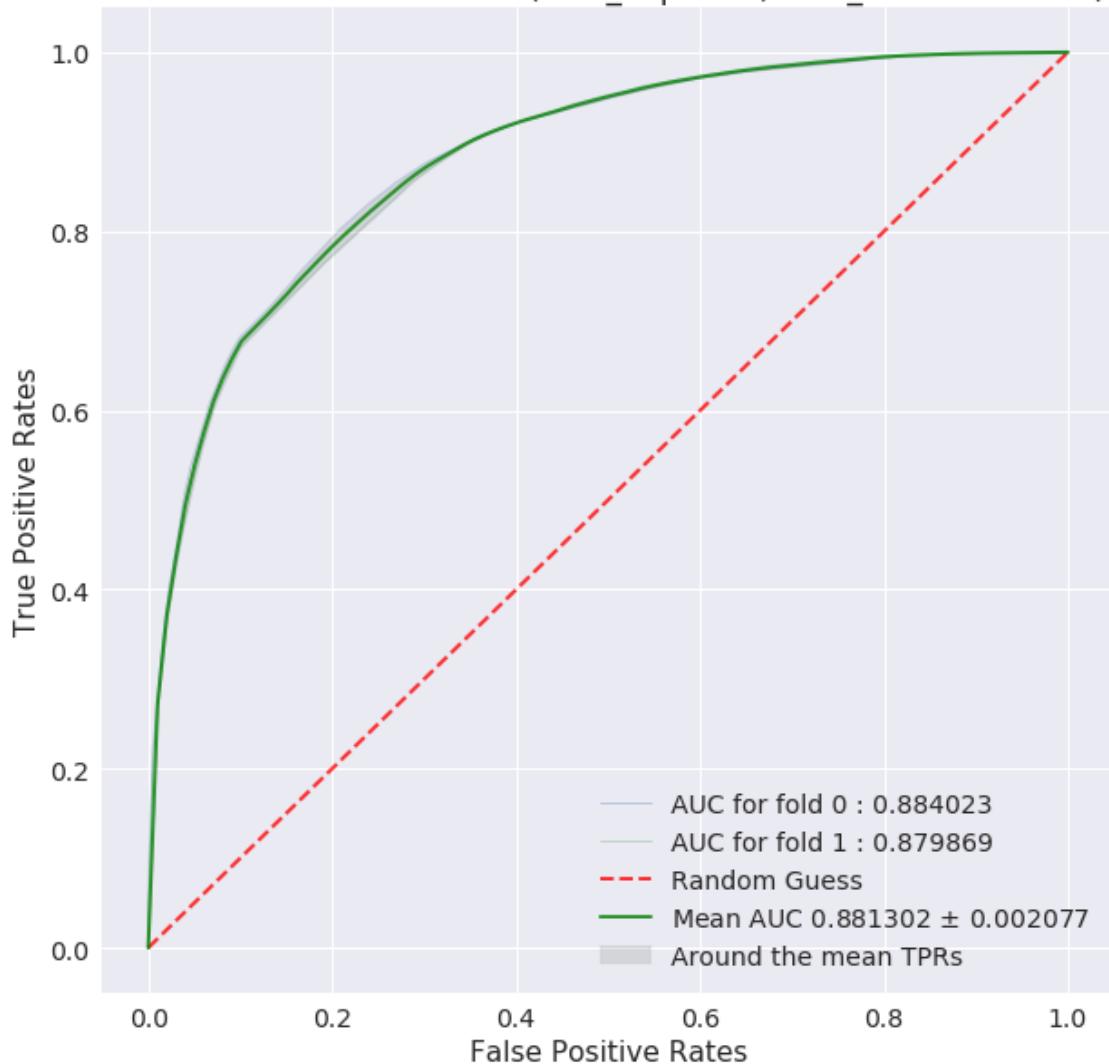
ROC - Validation Ensemble (max\_depth:10, num\_estimators:300)



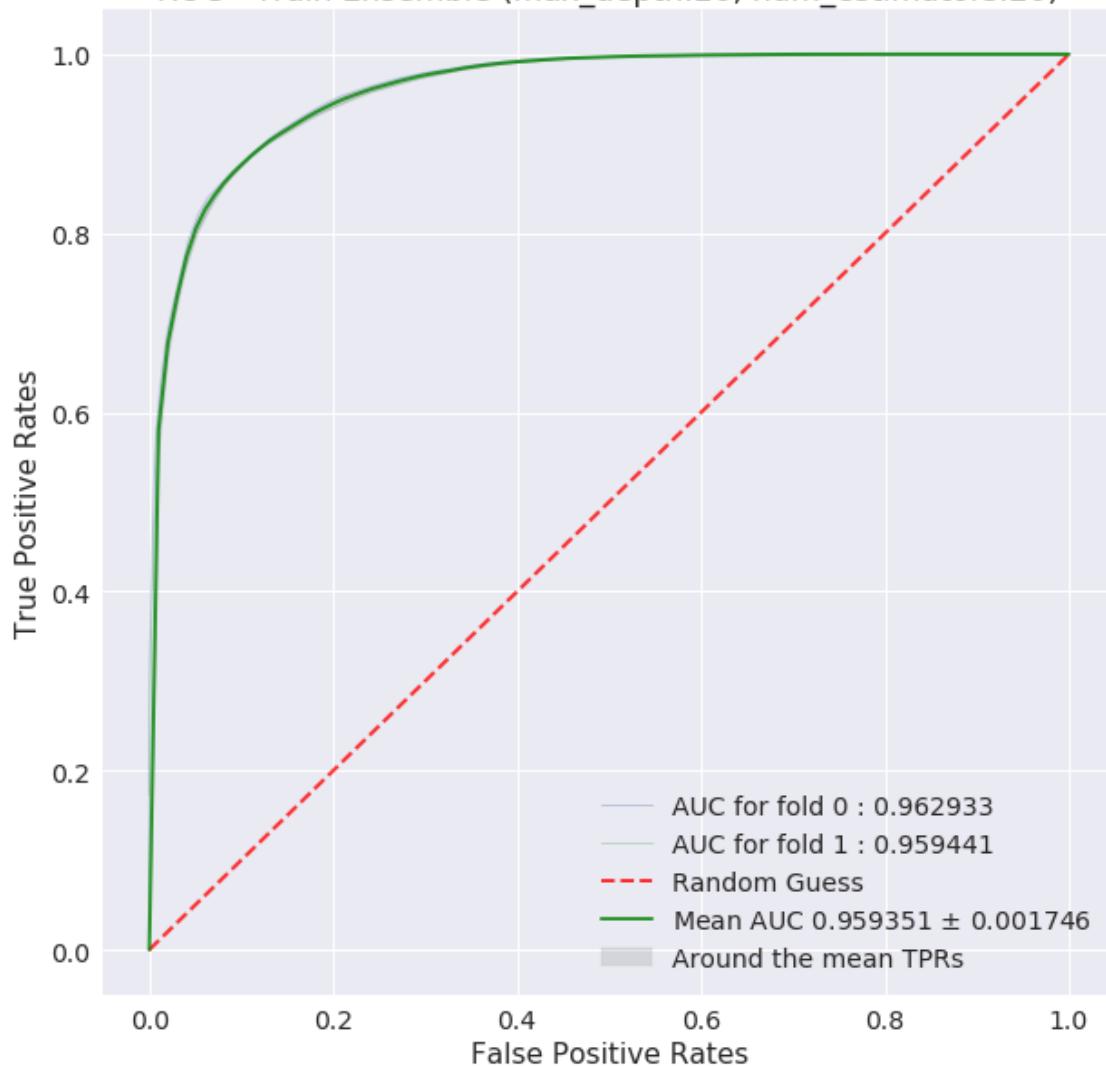
ROC - Train Ensemble (max\_depth:10, num\_estimators:500)

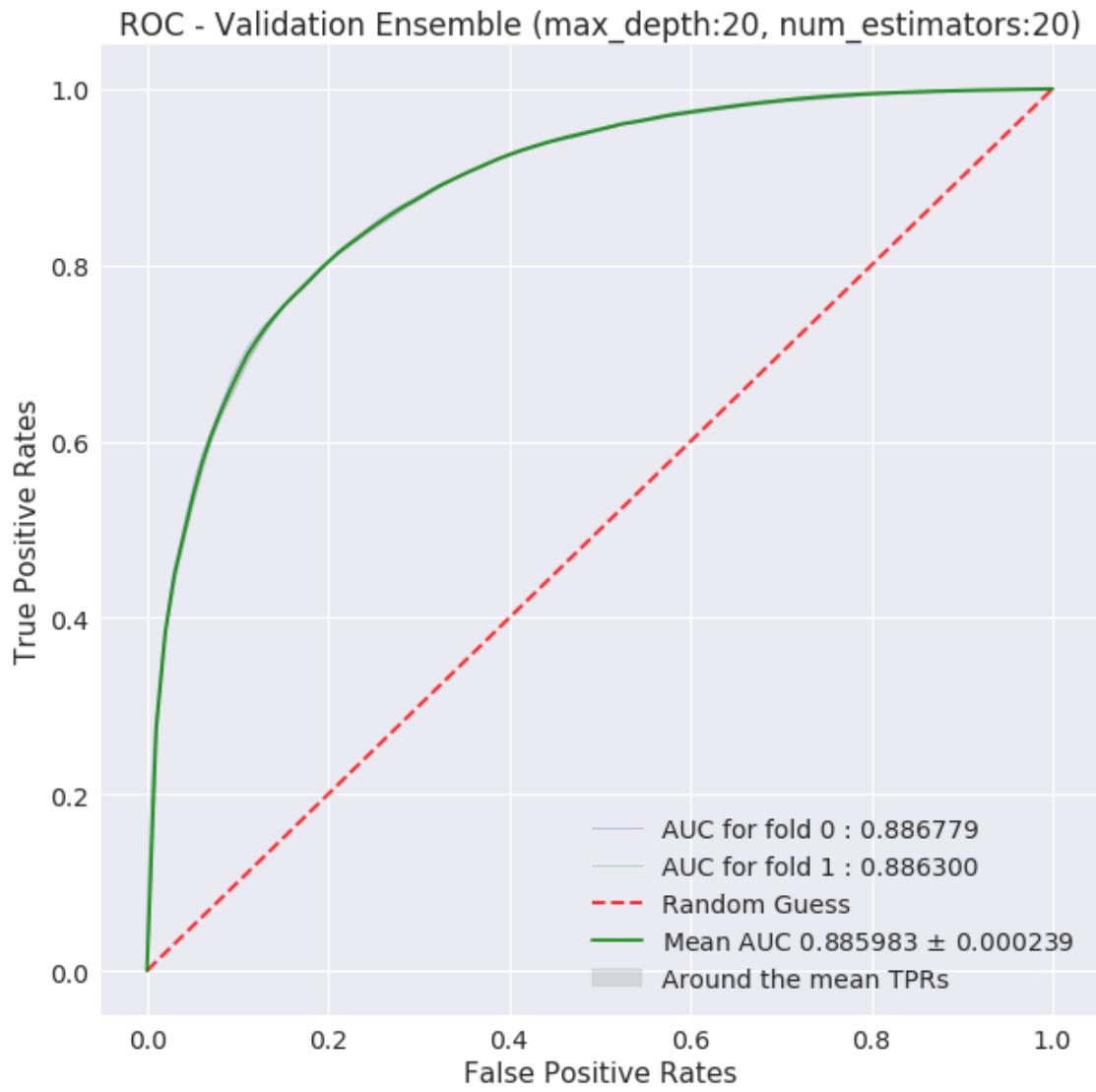


ROC - Validation Ensemble (max\_depth:10, num\_estimators:500)

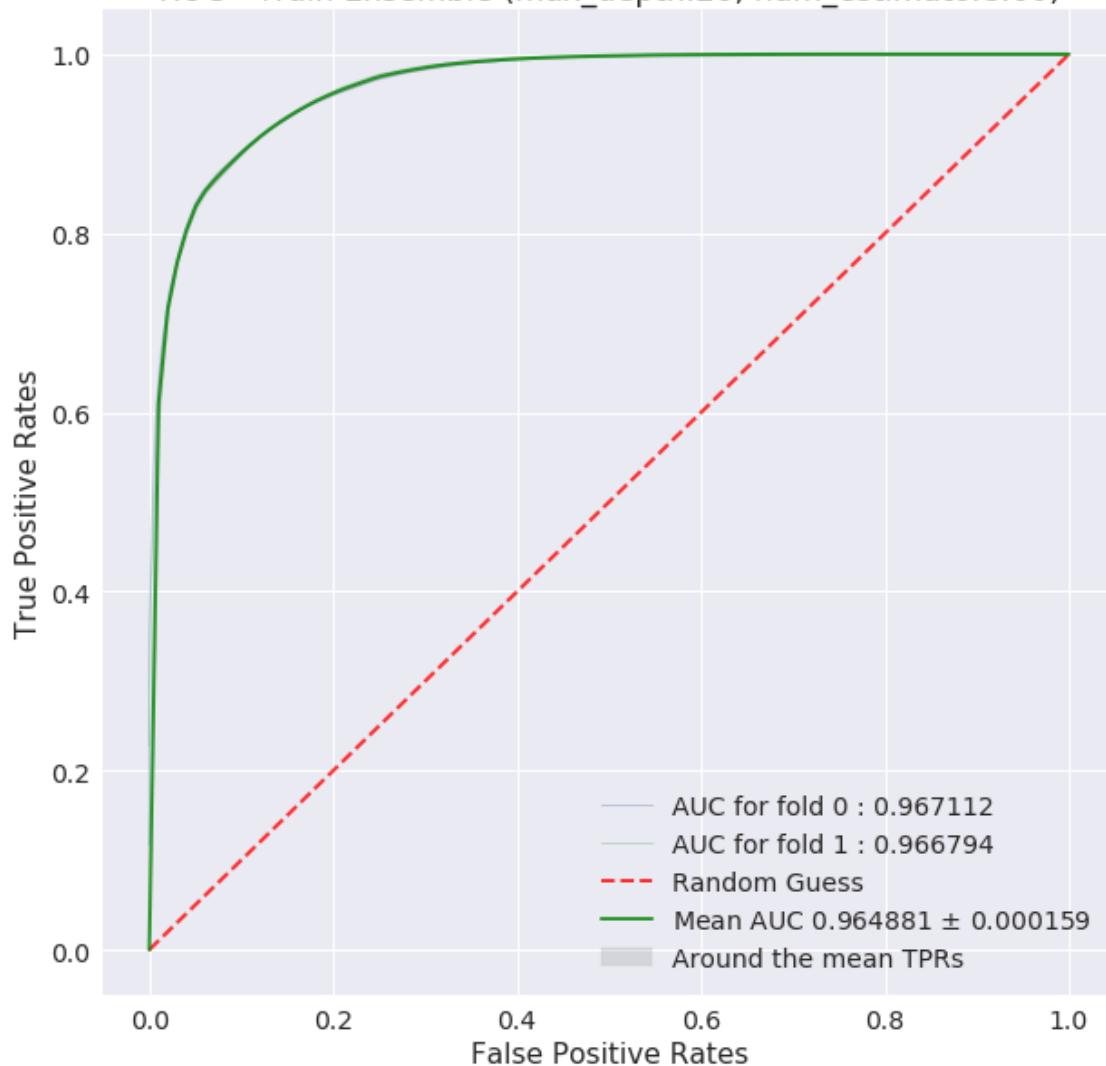


ROC - Train Ensemble (max\_depth:20, num\_estimators:20)

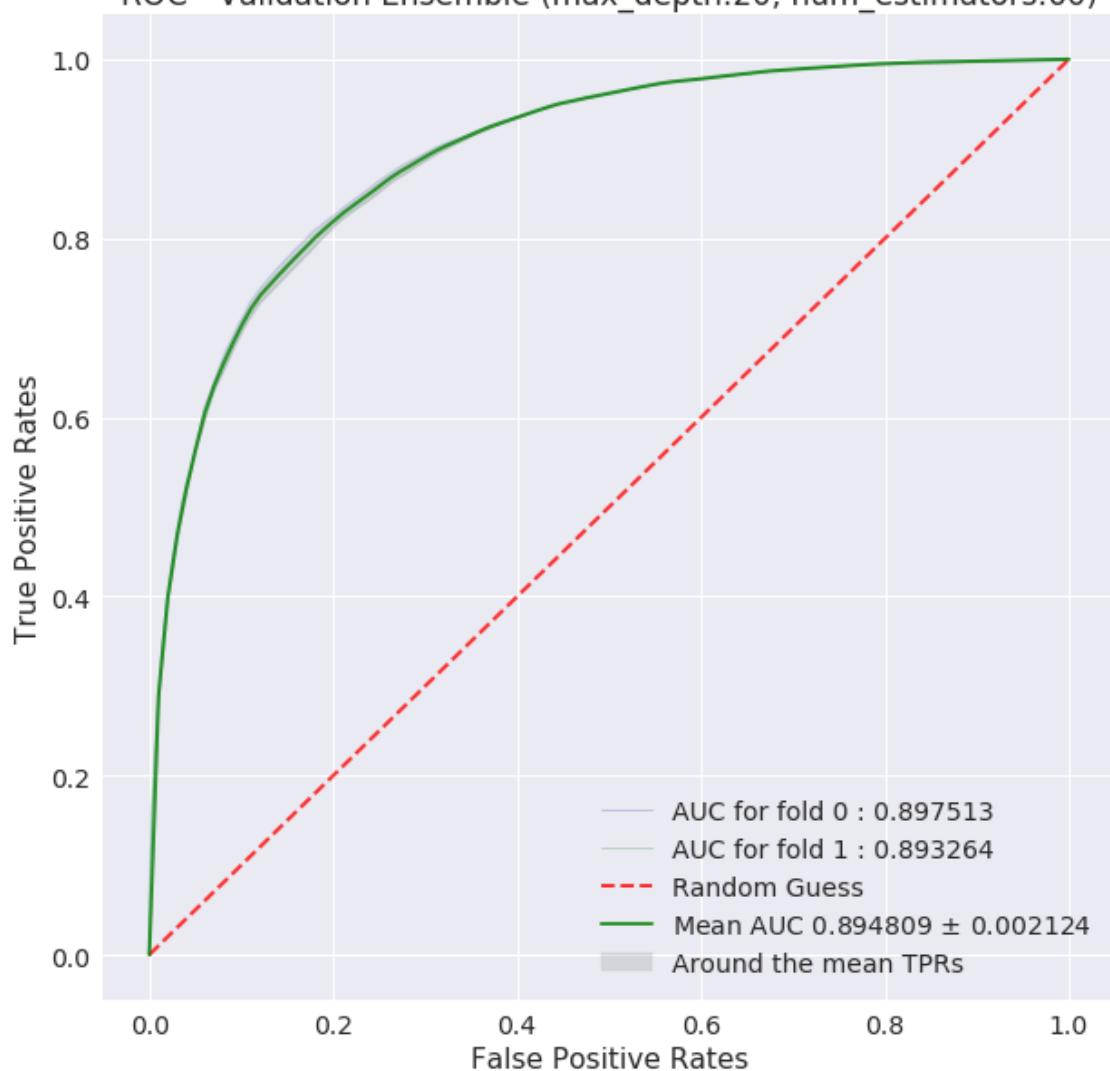




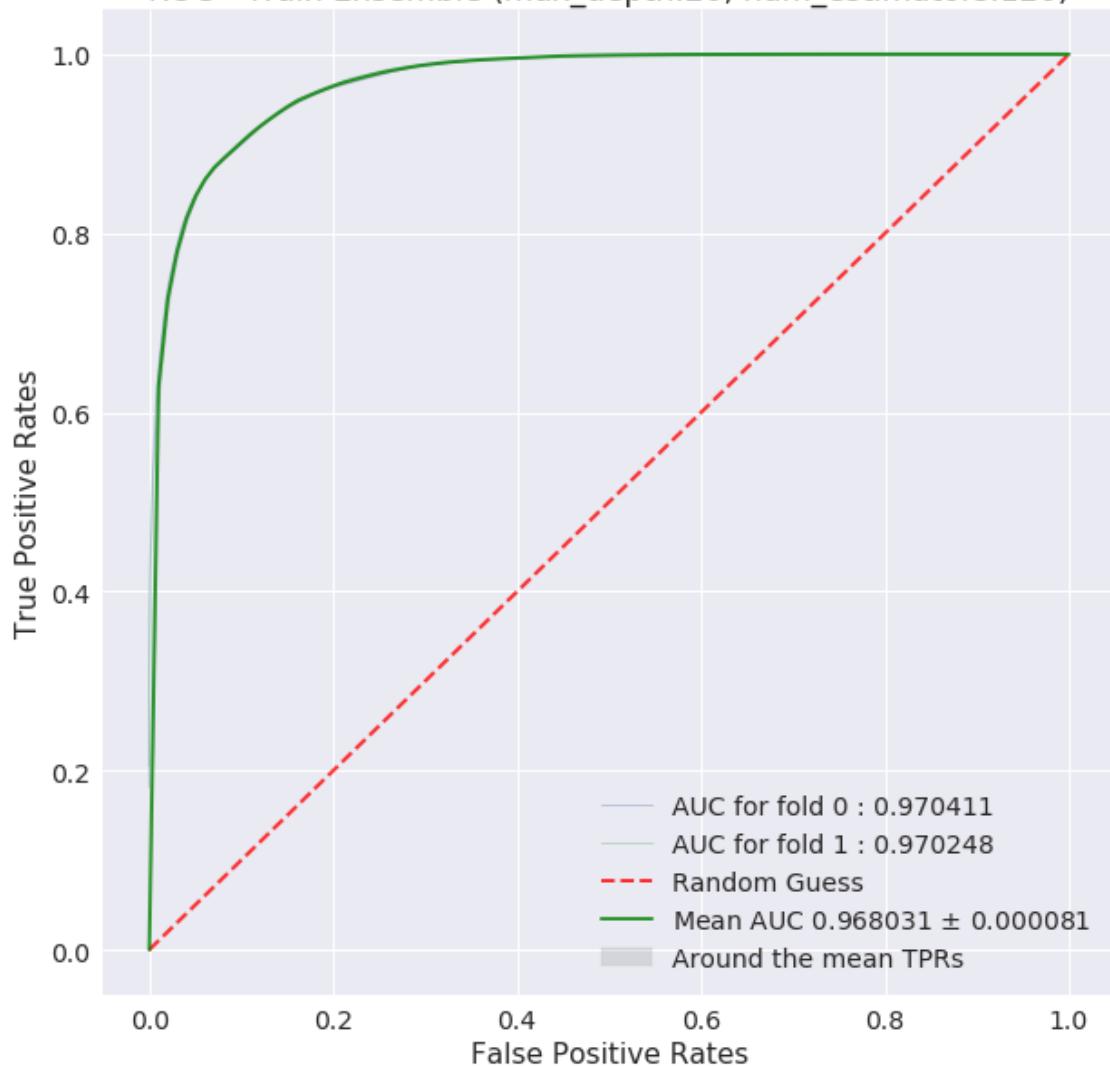
ROC - Train Ensemble (max\_depth:20, num\_estimators:60)



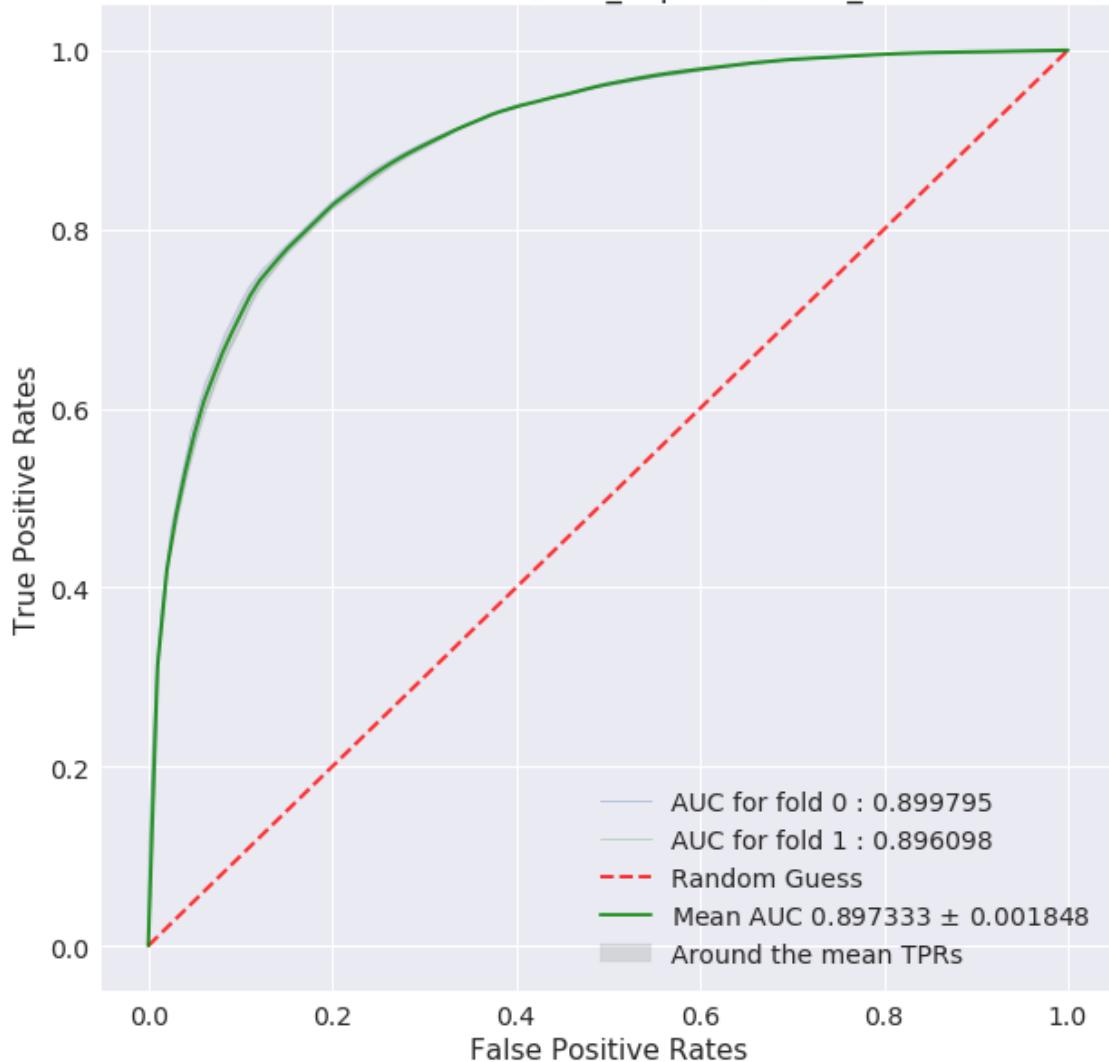
ROC - Validation Ensemble (max\_depth:20, num\_estimators:60)



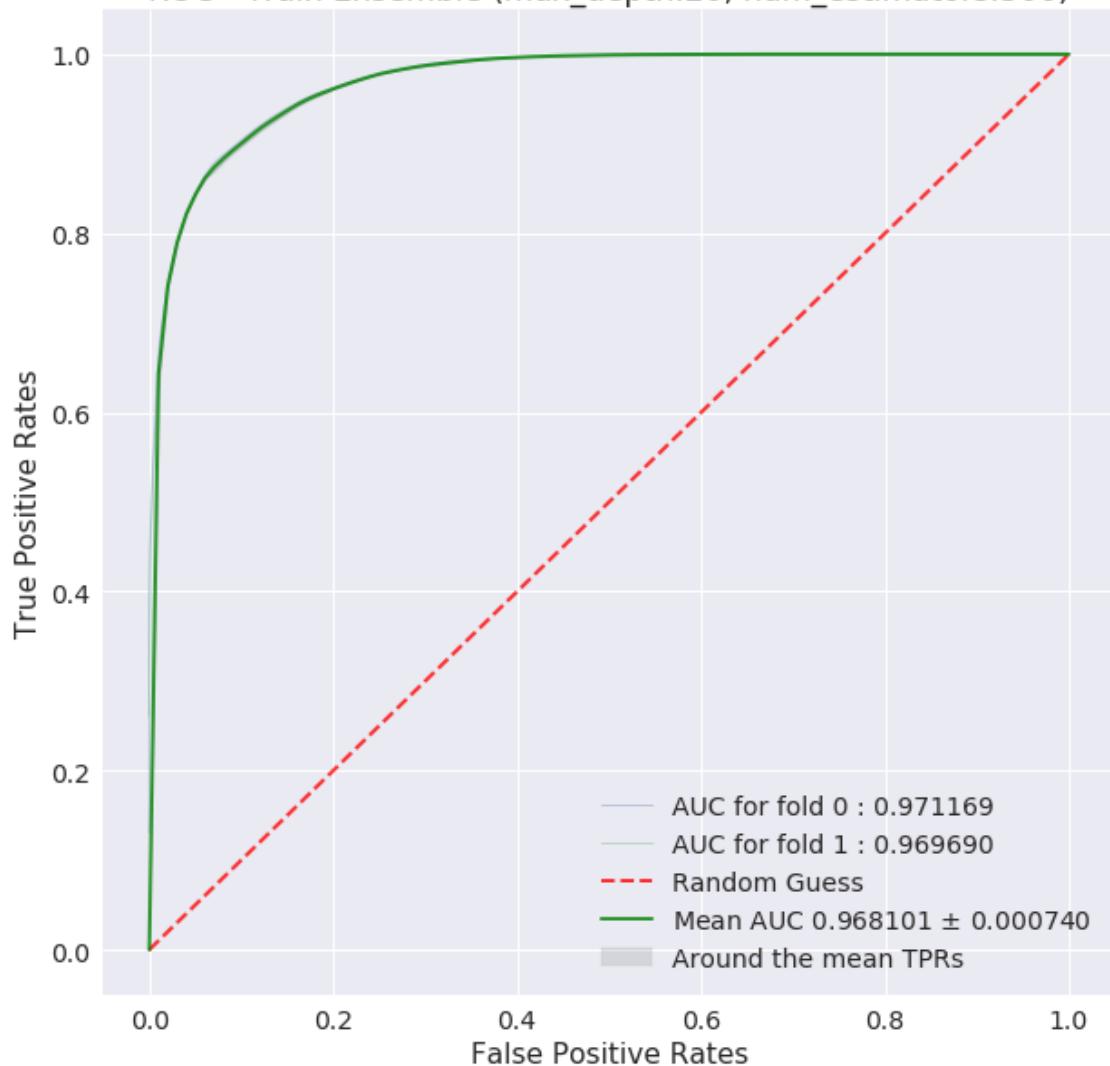
ROC - Train Ensemble (max\_depth:20, num\_estimators:120)



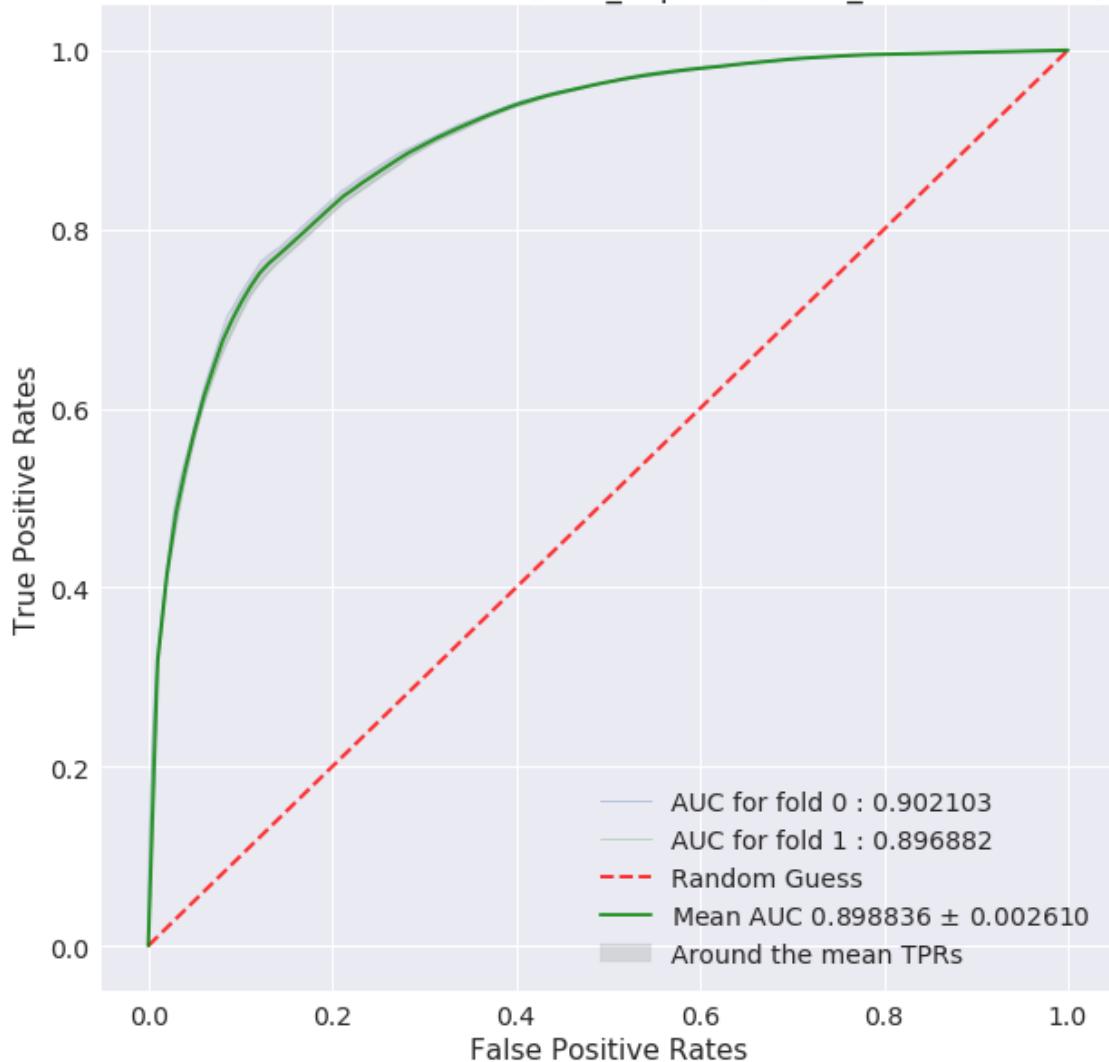
ROC - Validation Ensemble (max\_depth:20, num\_estimators:120)



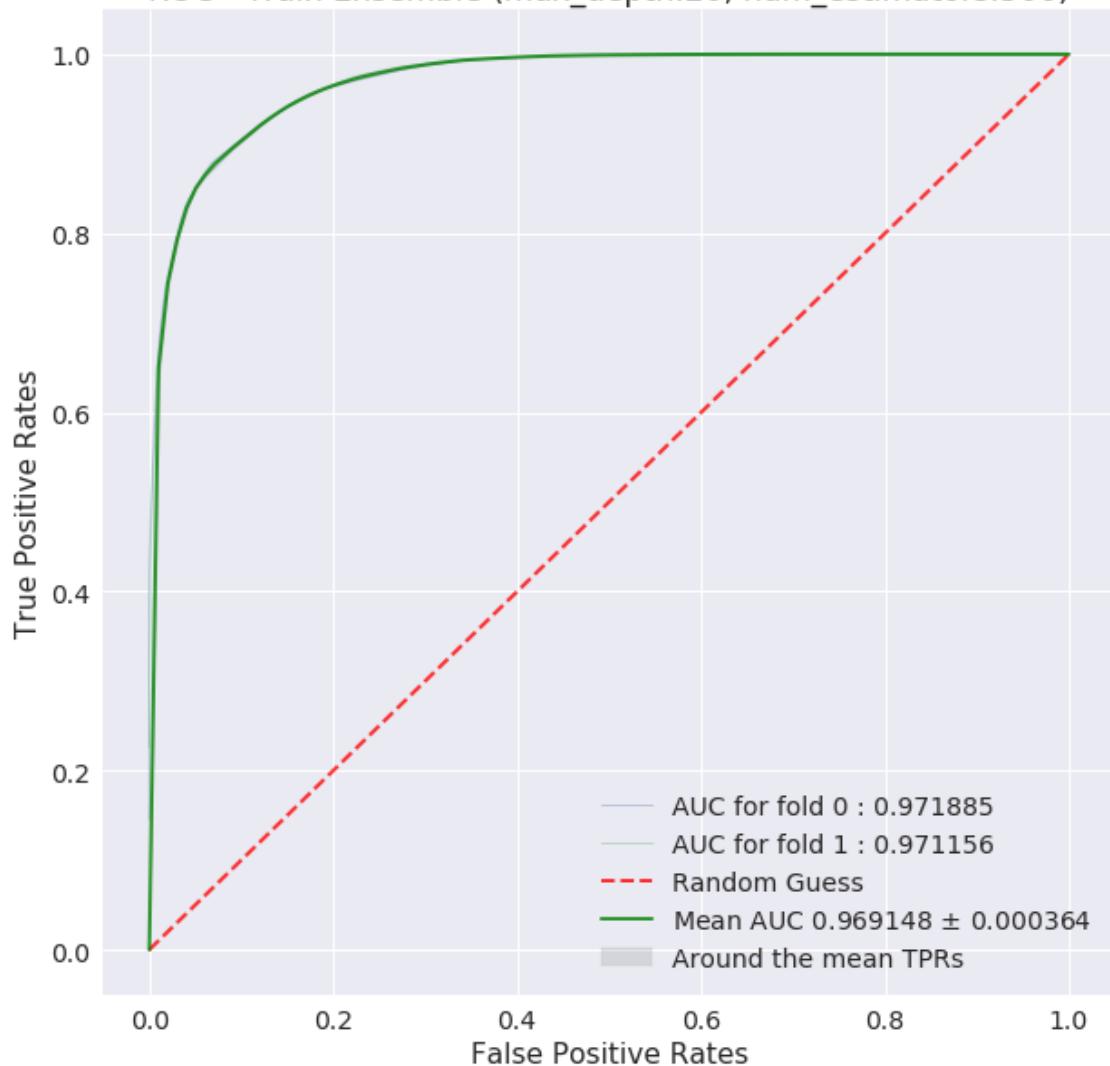
ROC - Train Ensemble (max\_depth:20, num\_estimators:300)



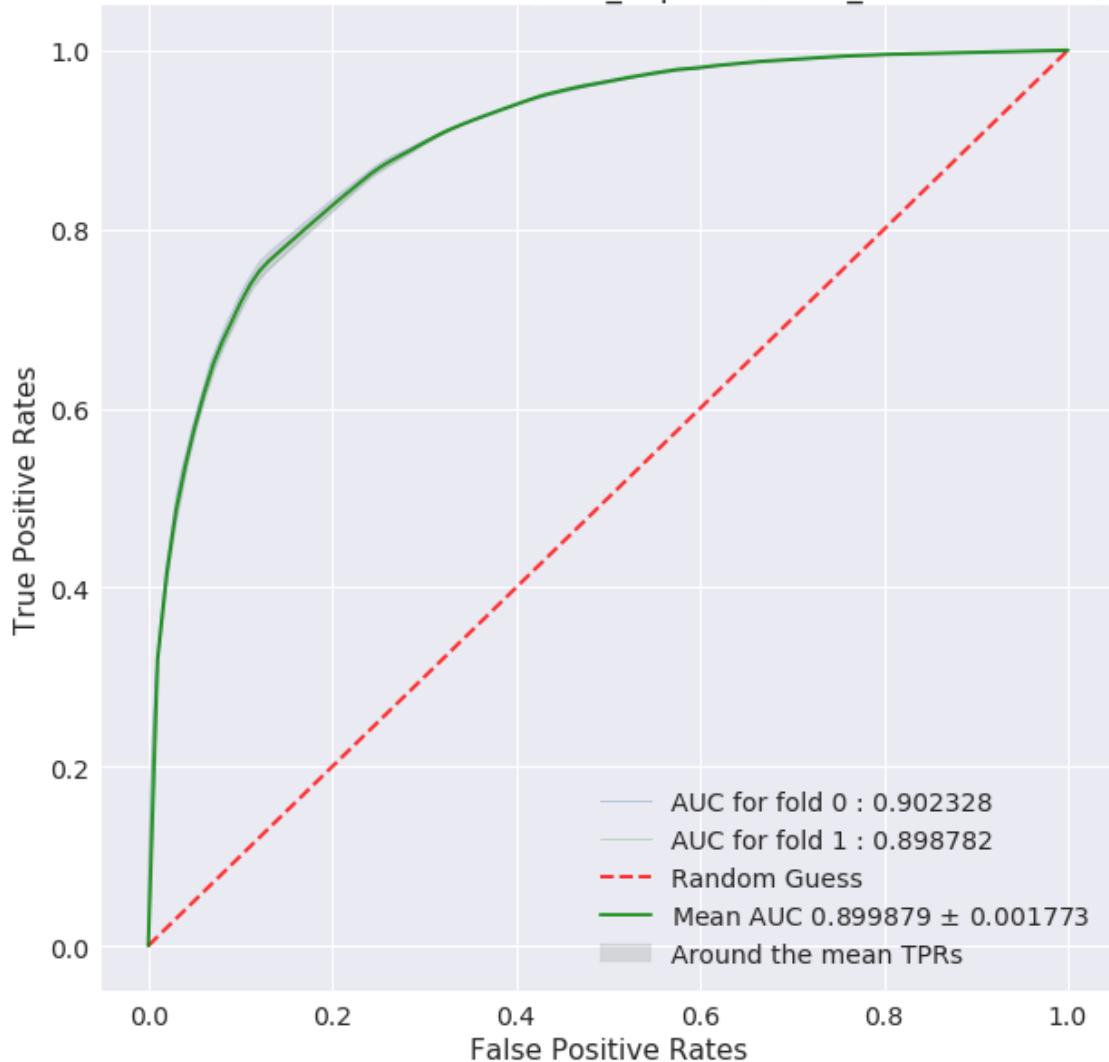
ROC - Validation Ensemble (max\_depth:20, num\_estimators:300)



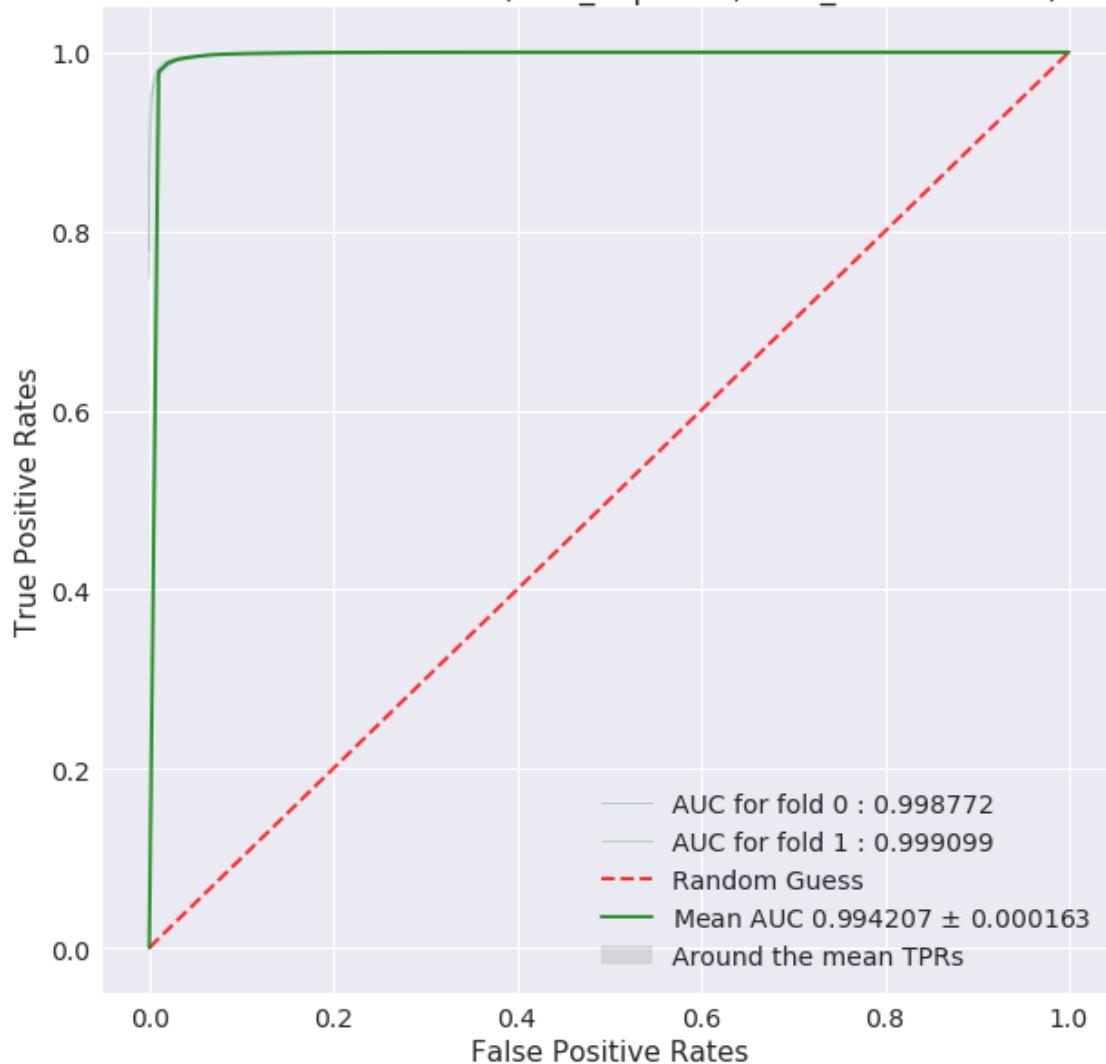
ROC - Train Ensemble (max\_depth:20, num\_estimators:500)

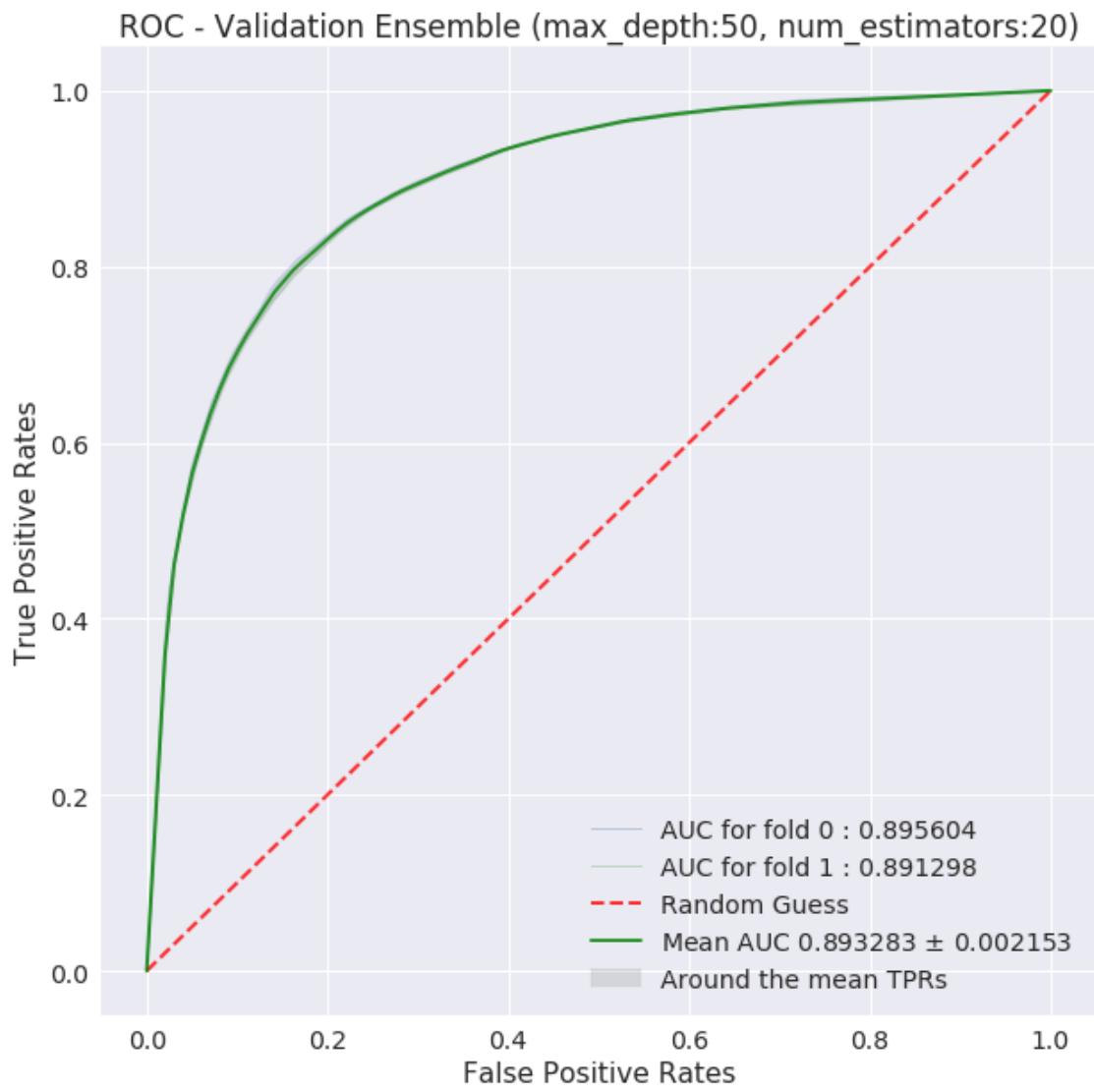


ROC - Validation Ensemble (max\_depth:20, num\_estimators:500)

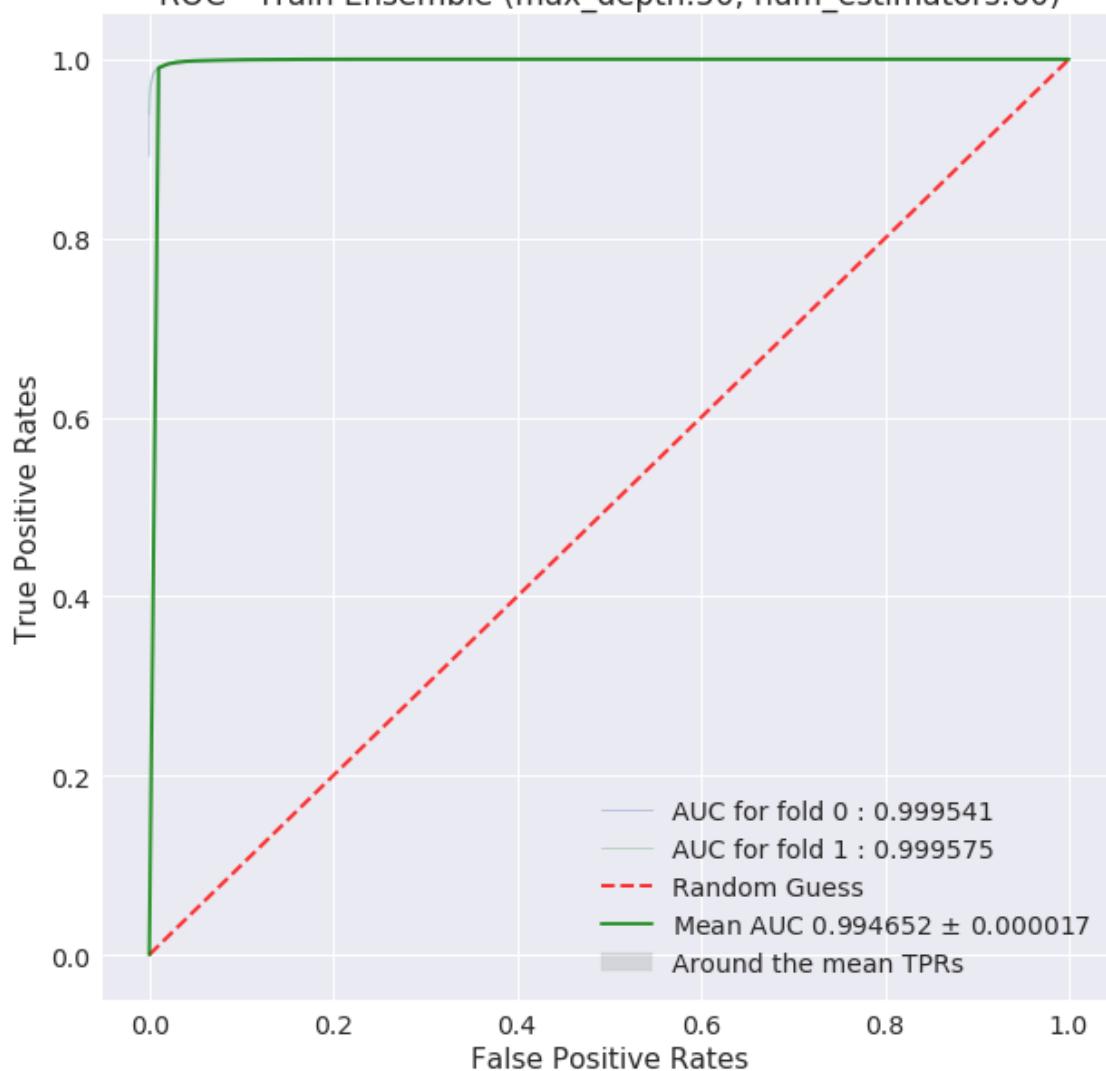


ROC - Train Ensemble (max\_depth:50, num\_estimators:20)

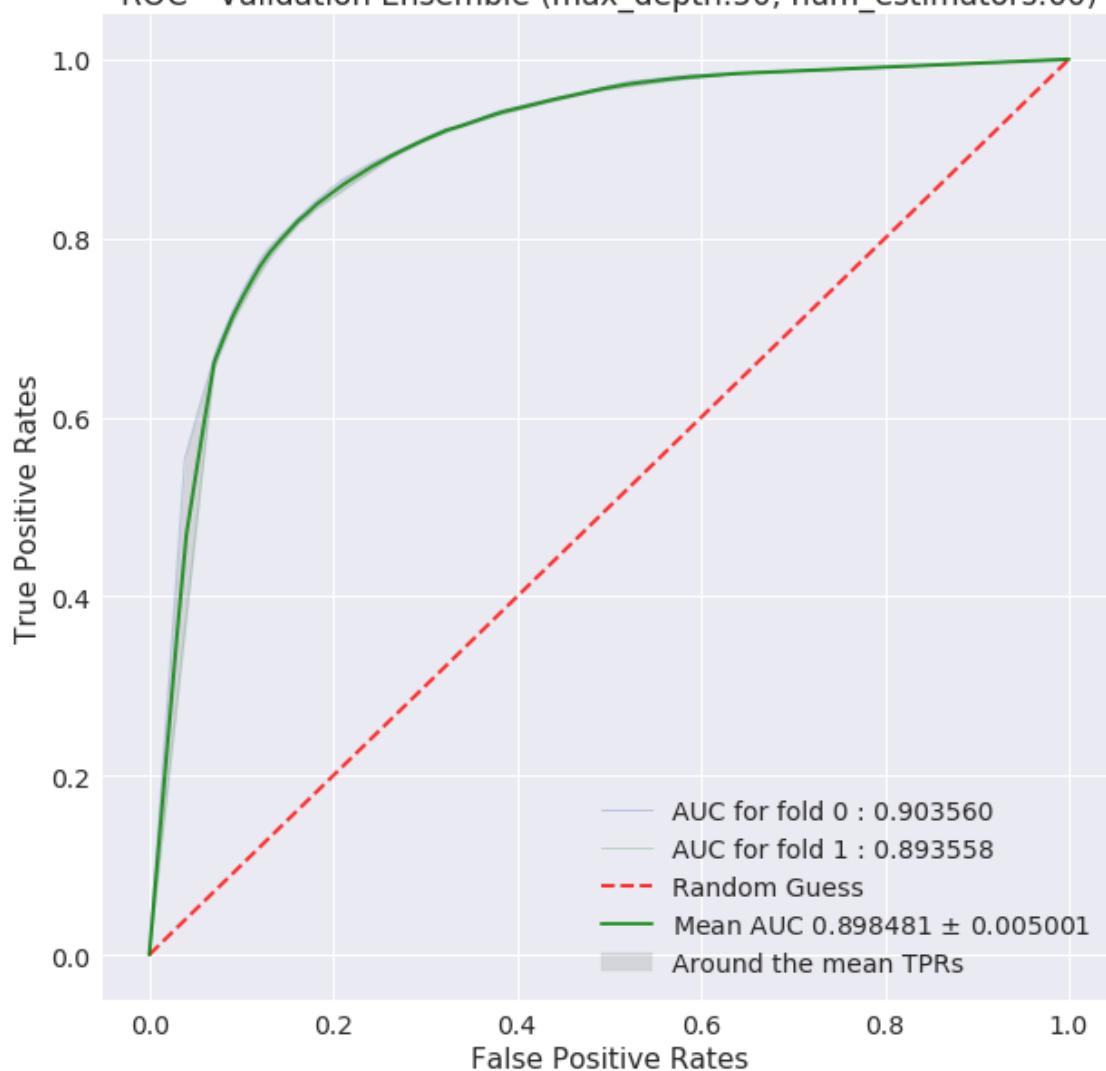




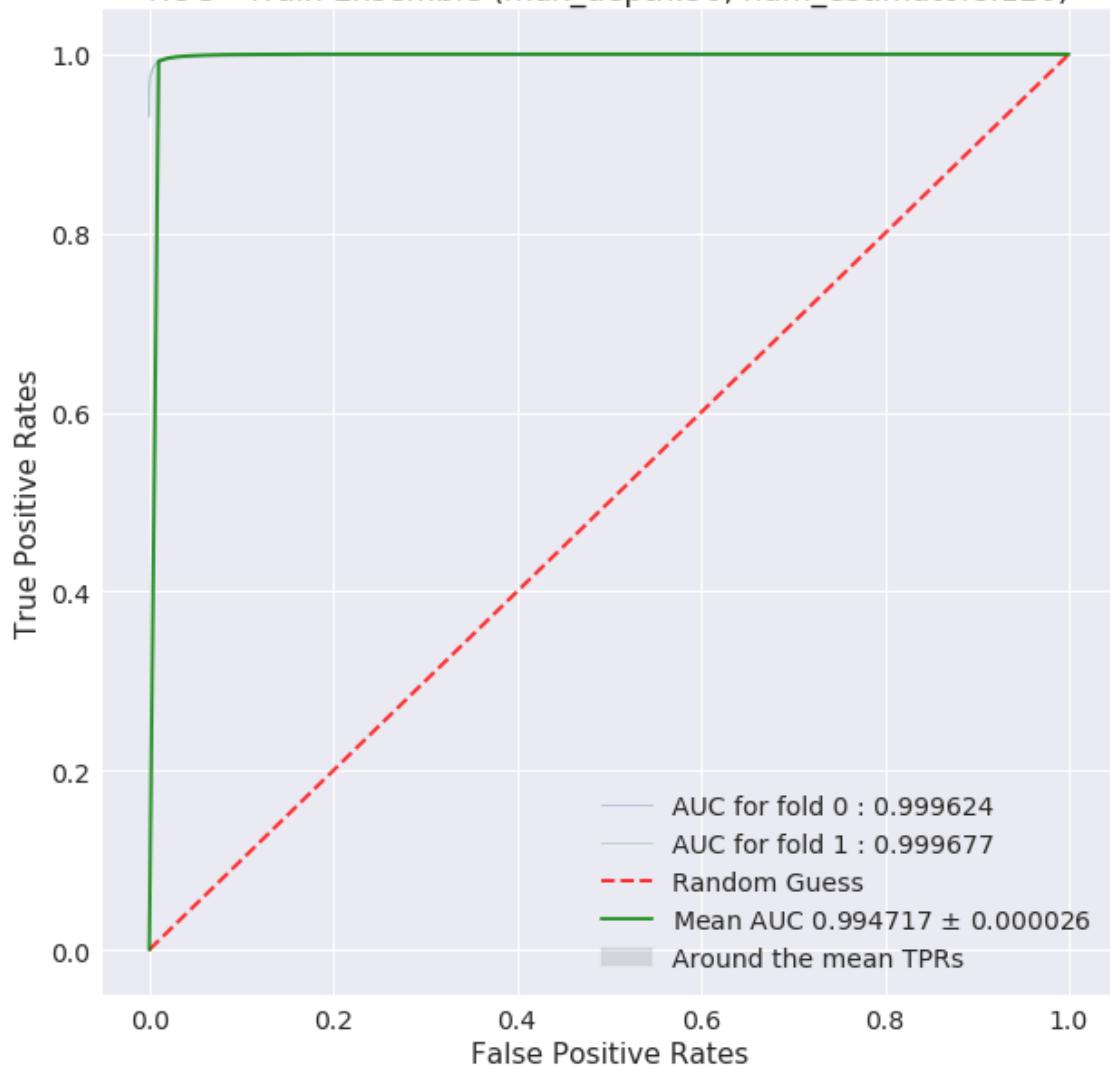
ROC - Train Ensemble (max\_depth:50, num\_estimators:60)



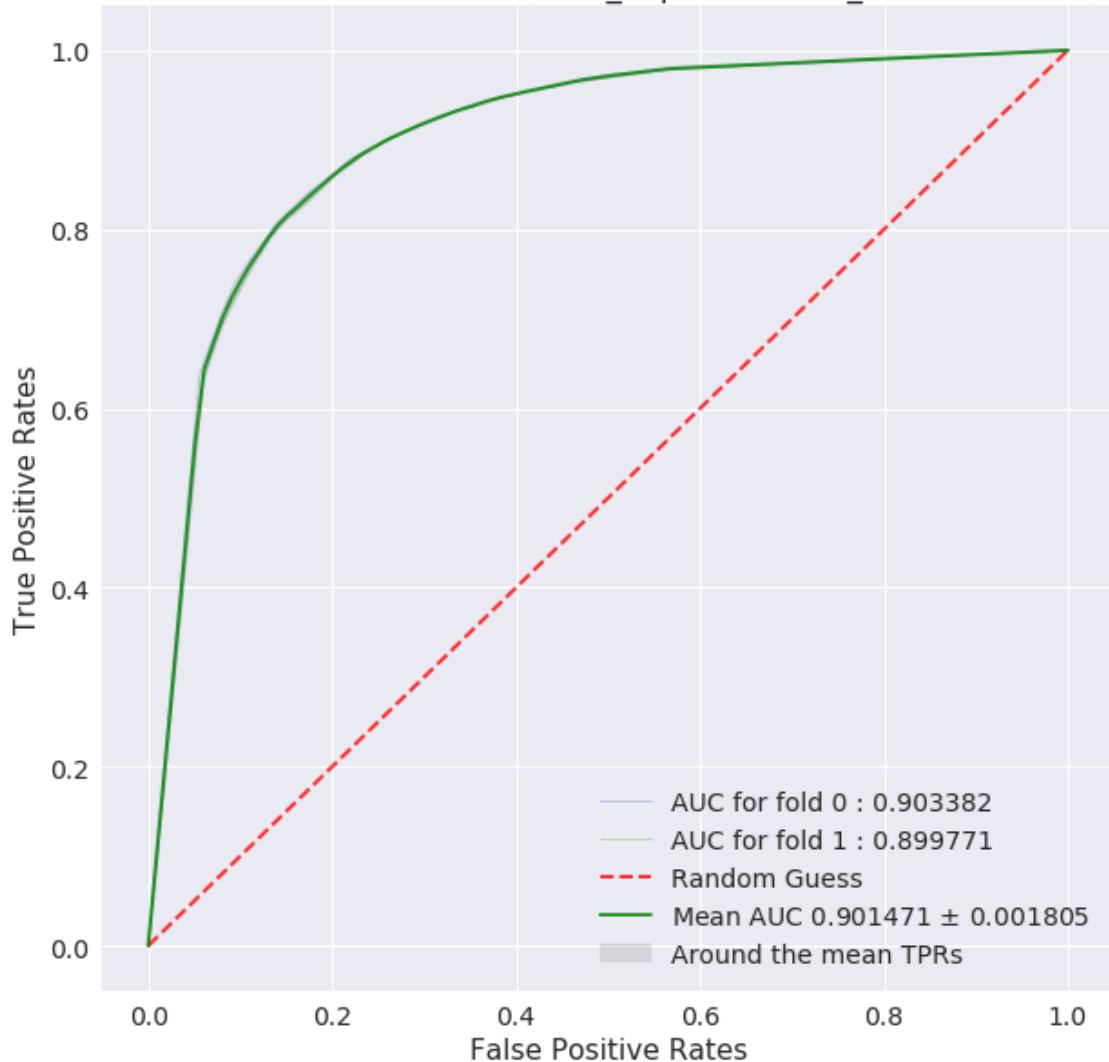
ROC - Validation Ensemble (max\_depth:50, num\_estimators:60)



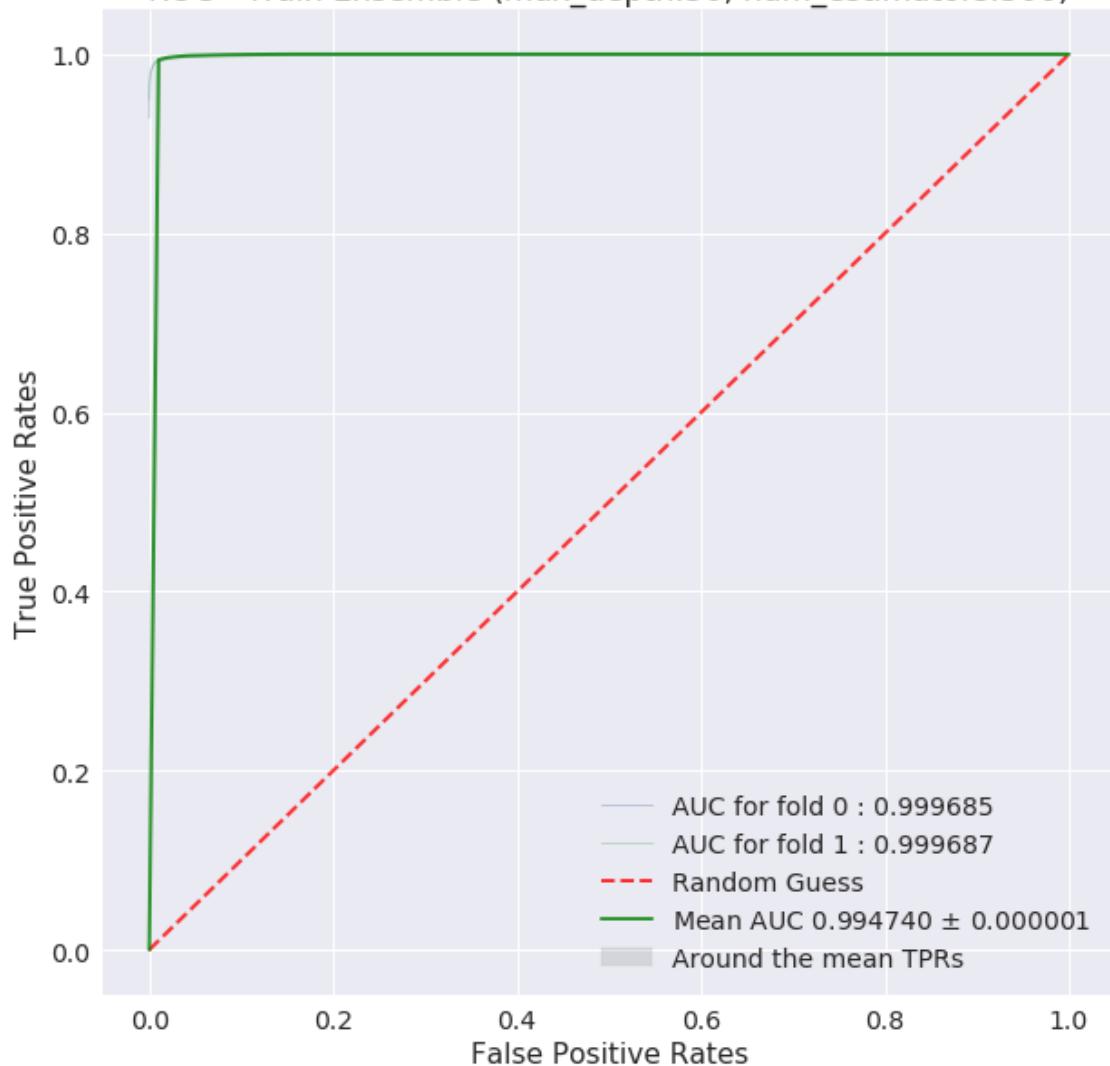
ROC - Train Ensemble (max\_depth:50, num\_estimators:120)



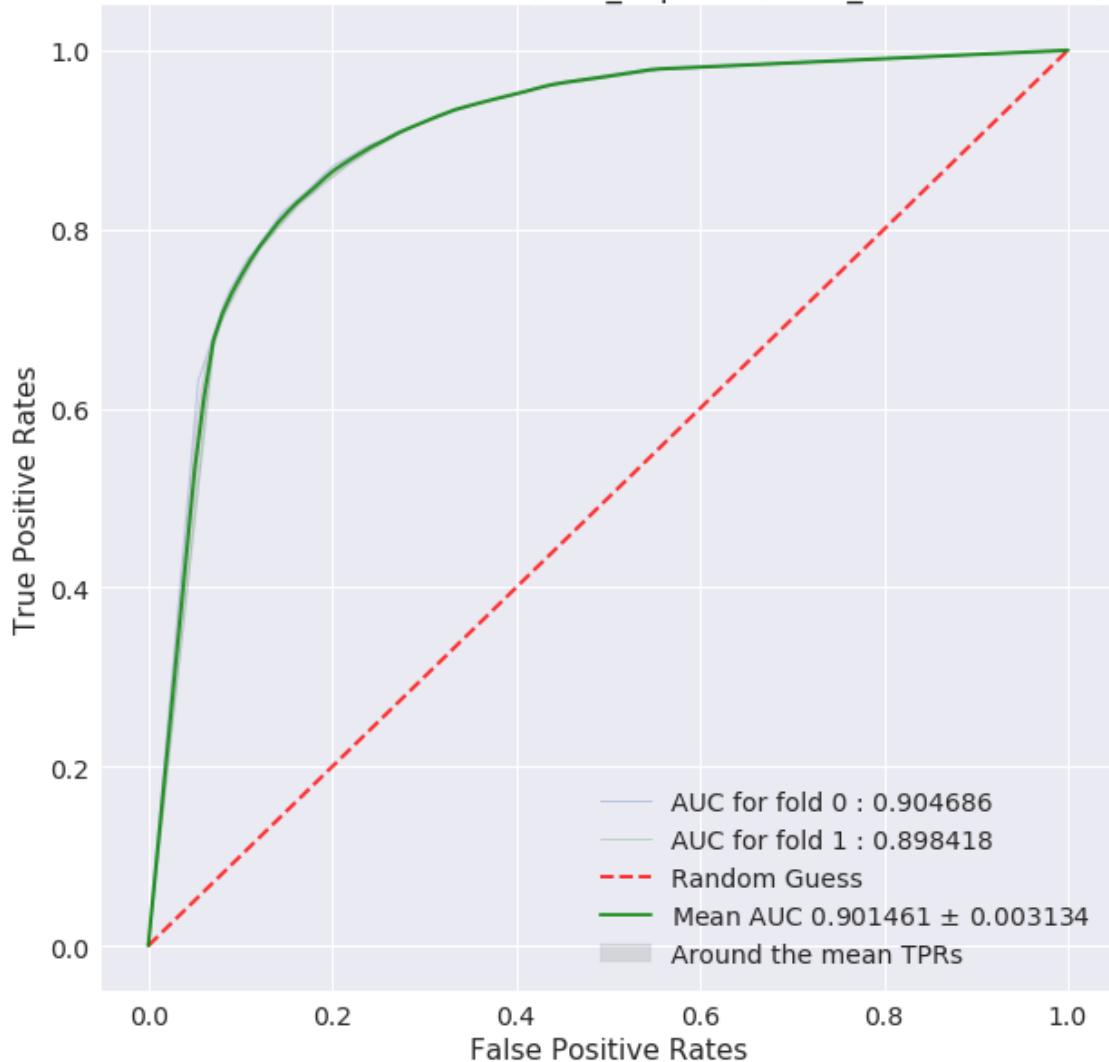
ROC - Validation Ensemble (max\_depth:50, num\_estimators:120)



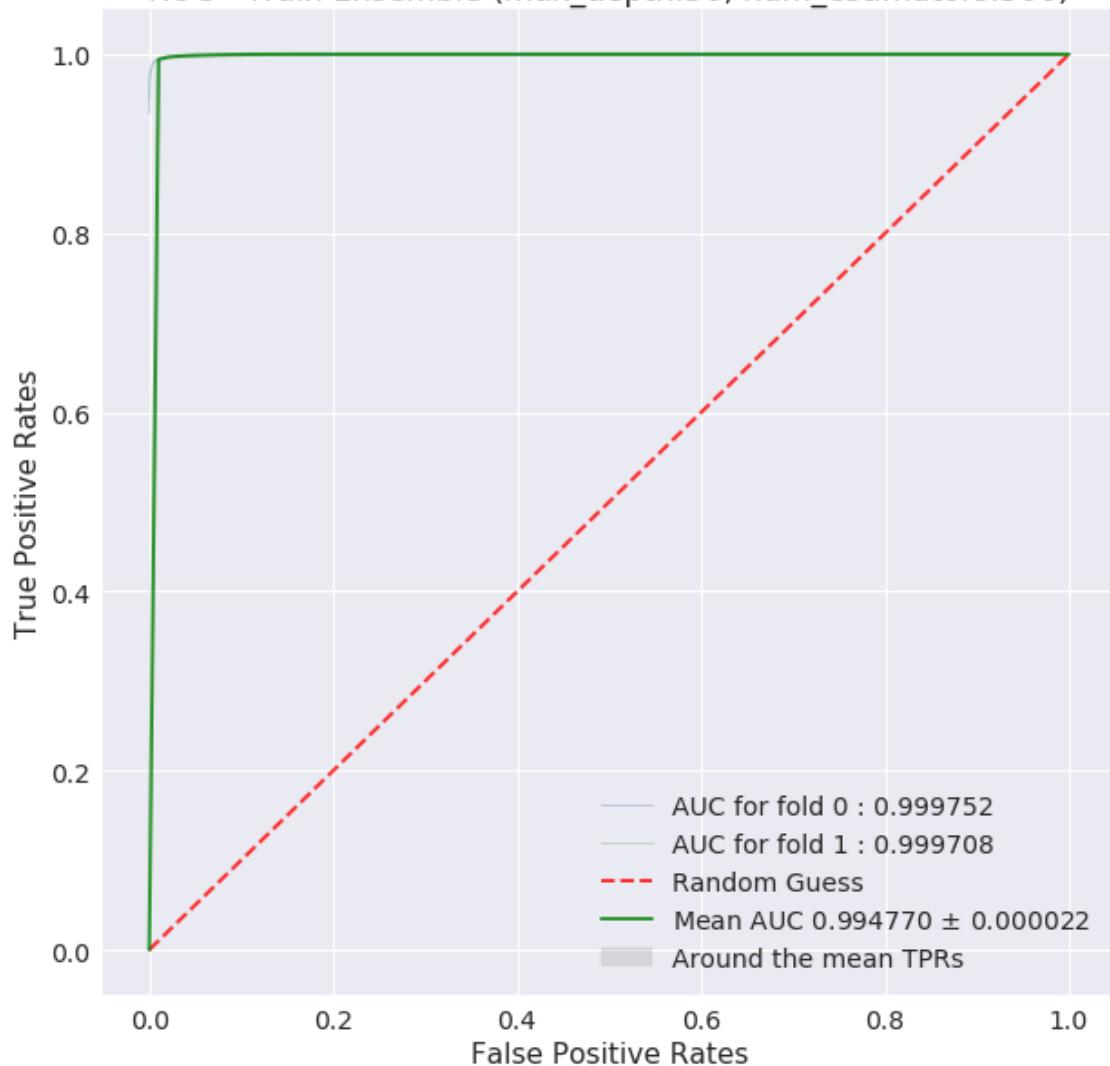
ROC - Train Ensemble (max\_depth:50, num\_estimators:300)



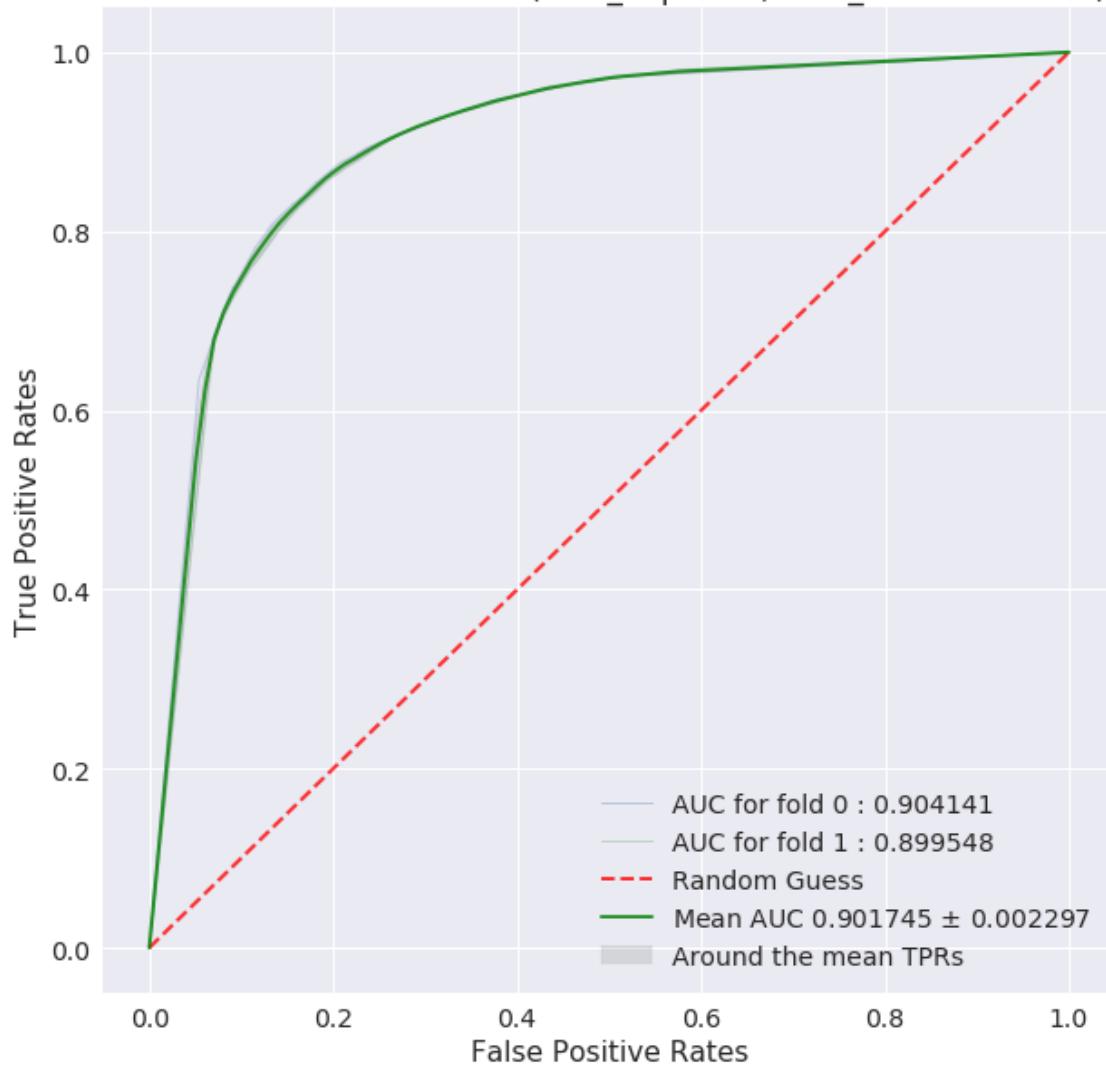
ROC - Validation Ensemble (max\_depth:50, num\_estimators:300)



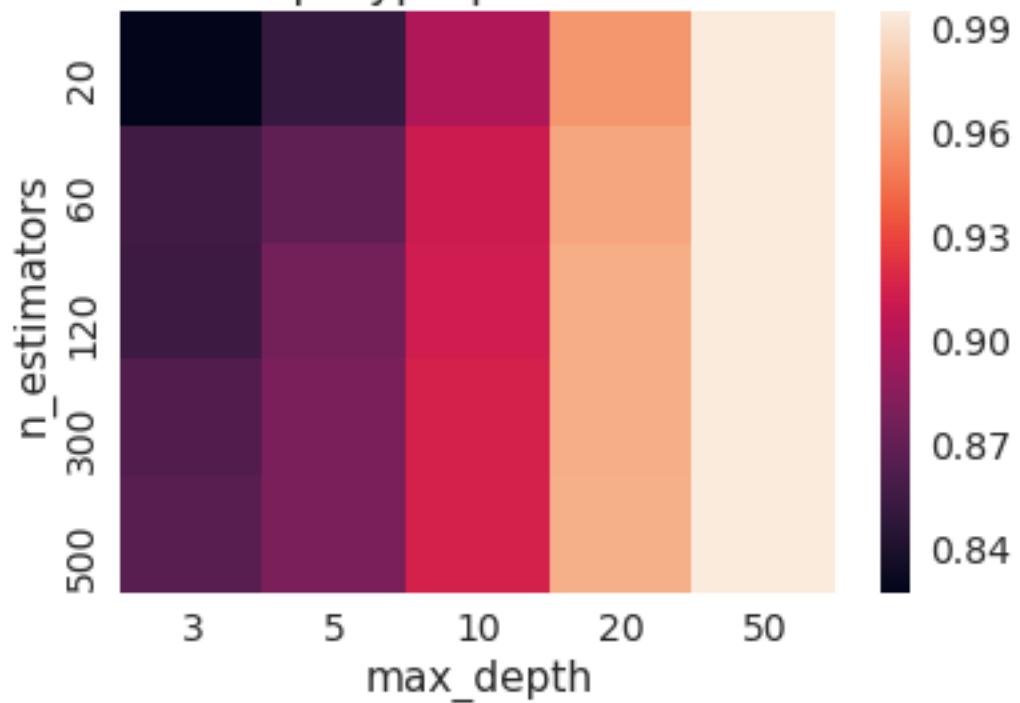
ROC - Train Ensemble (max\_depth:50, num\_estimators:500)



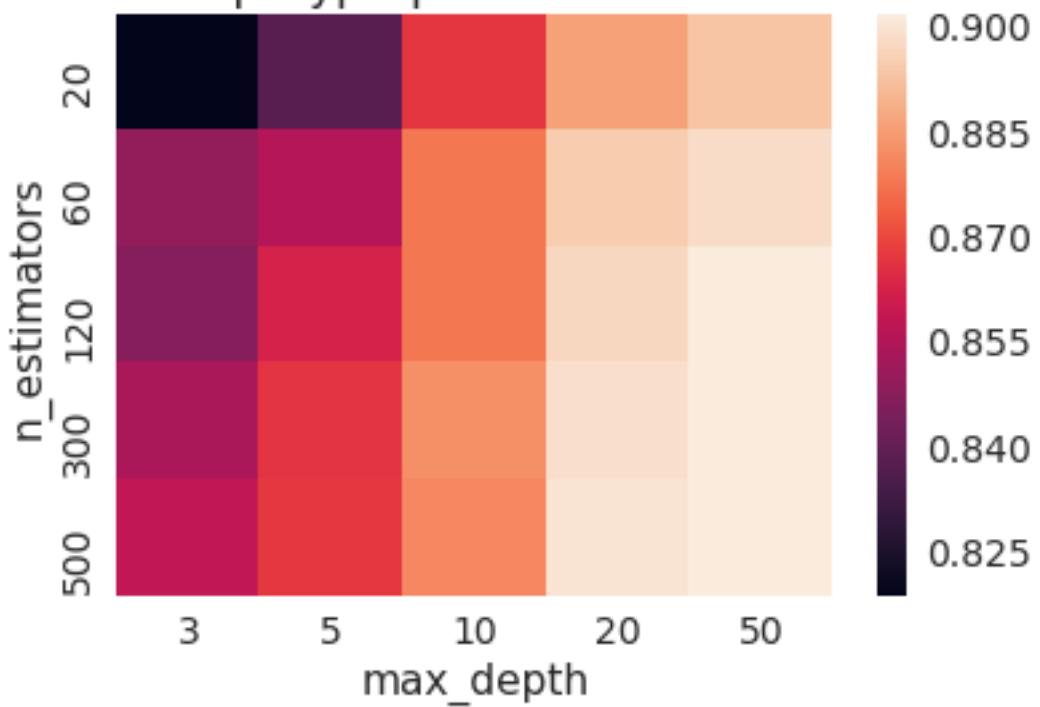
ROC - Validation Ensemble (max\_depth:50, num\_estimators:500)



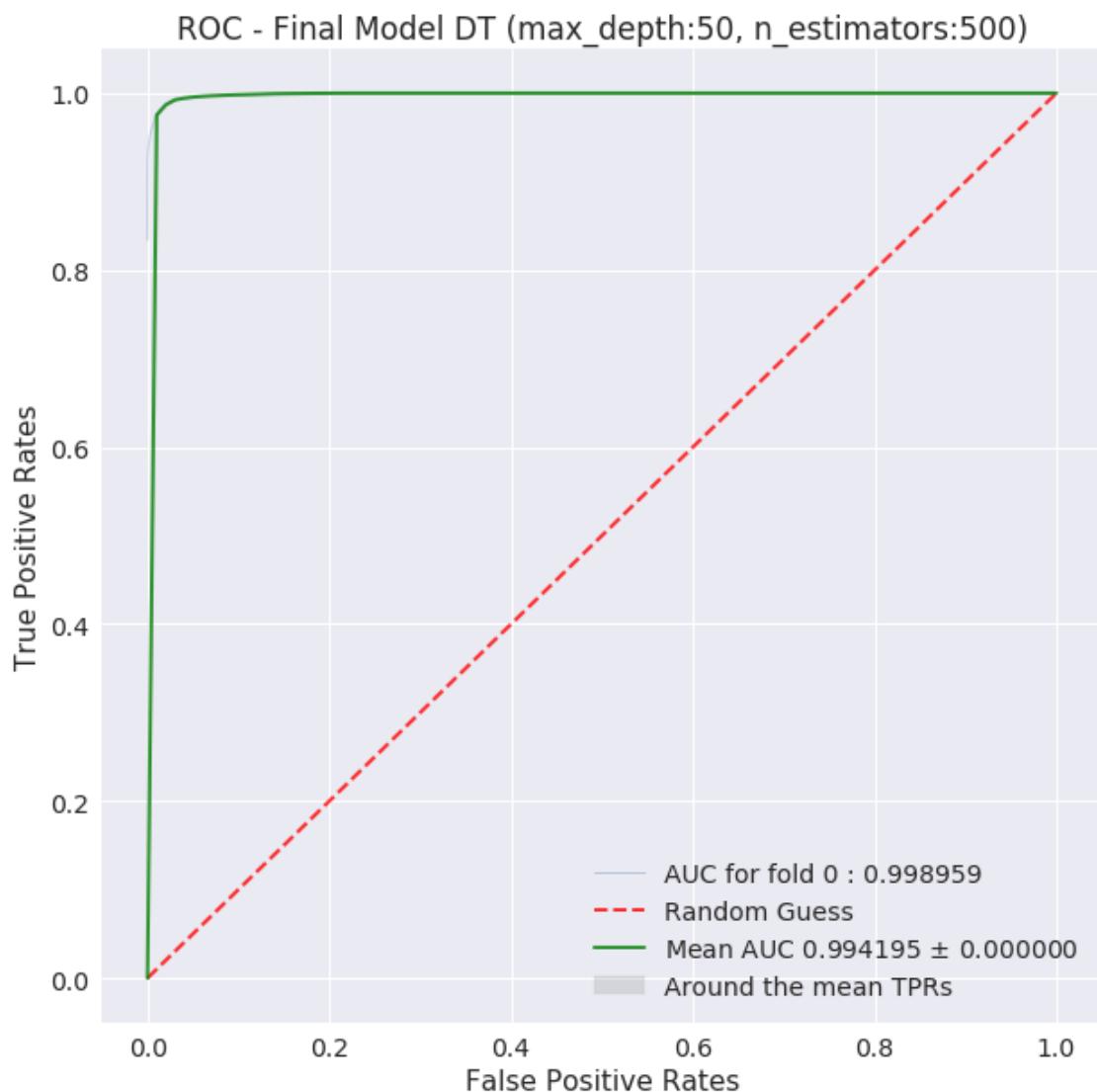
Heatmap Hyperparams for Train

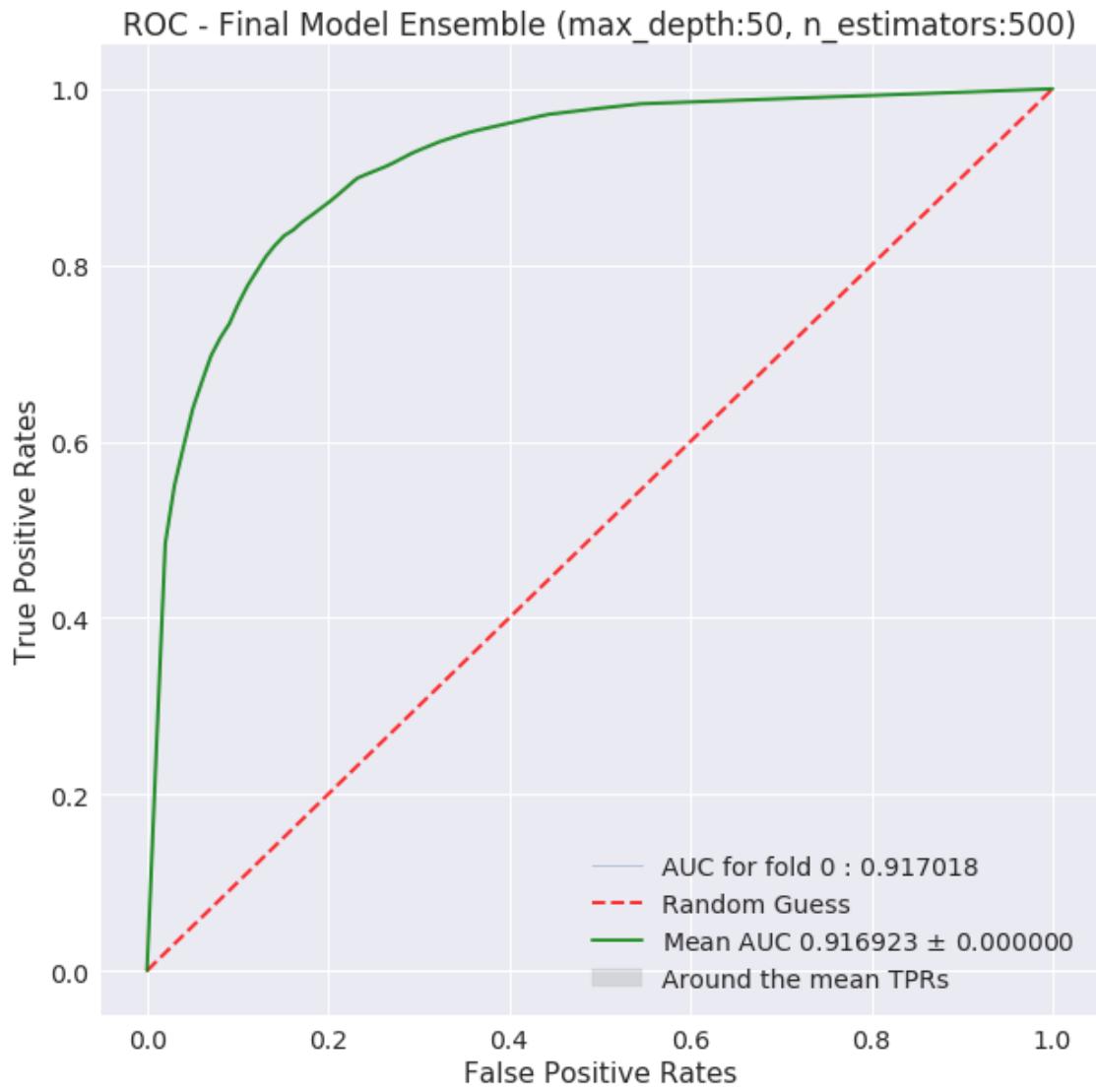


Heatmap Hyperparams for Validation

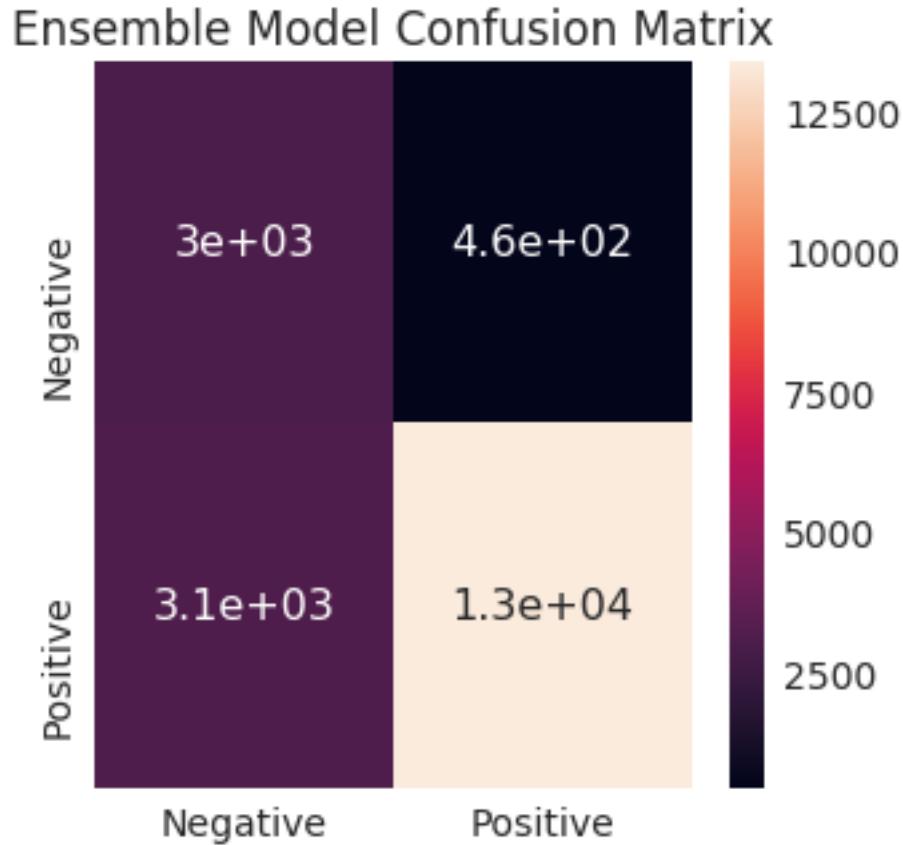


Best hyperparam value: (50, 500)





Test auc score 0.916922935940239



	Negative	Positive
Precision	0.492097	0.966818
Recall	0.867816	0.811320
Fscore	0.628054	0.882270
Support	3480.000000	16520.000000

#### 4.3.2 [A.4] Wordcloud of top 20 important features from SET 2

```
In [15]: # get feature and its importance as tuple
feature_imp_info = list(zip(feature_name_list, model[0].feature_importances_))

# filter only those features which have a value greater than zero
feature_imp_dict = dict(list(filter(lambda x: x[1] > 0.0, feature_imp_info)))

# create word cloud object for displaying the output
wc = WordCloud(background_color='white', width=800, height=800)
wc_output = wc.generate_from_frequencies(feature_imp_dict)

In [16]: plt.figure(figsize=(8,8))
plt.imshow(wc_output)
```

```
plt.axis('off')
plt.tight_layout(pad=0.0)
plt.title('RF Feature Importances')
plt.show()
```



#### 4.4 Observation

best hyper param identified is max\_depth = 50, and n\_estimators=500

Review length is identified as one of the important feature

Positive words such as 'great', 'best' & negative words such as 'disappoint', 'bad' are recognized as important features by random forest

#### 4.4.1 [A.5] Applying Random Forests on AVG W2V, SET 3

```
In [17]: # form two lists
    depth_list = [3, 5, 10, 20, 50] # depends on size of dataset
    n_estimators_list = [20, 60, 120, 300, 500] # depends on size of dataset

    # create a configuartion dictionary
    config_dict = {
        'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVW/AVG_W2V',
        'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVW/AVG_W2V',
        'train_size' : 50000,
        'test_size' : 20000,
        'hyperparam_list' : list(product(depth_list, n_estimators_list)),
        'implementation': 'rf' # 'xgb' or 'rf'
    }

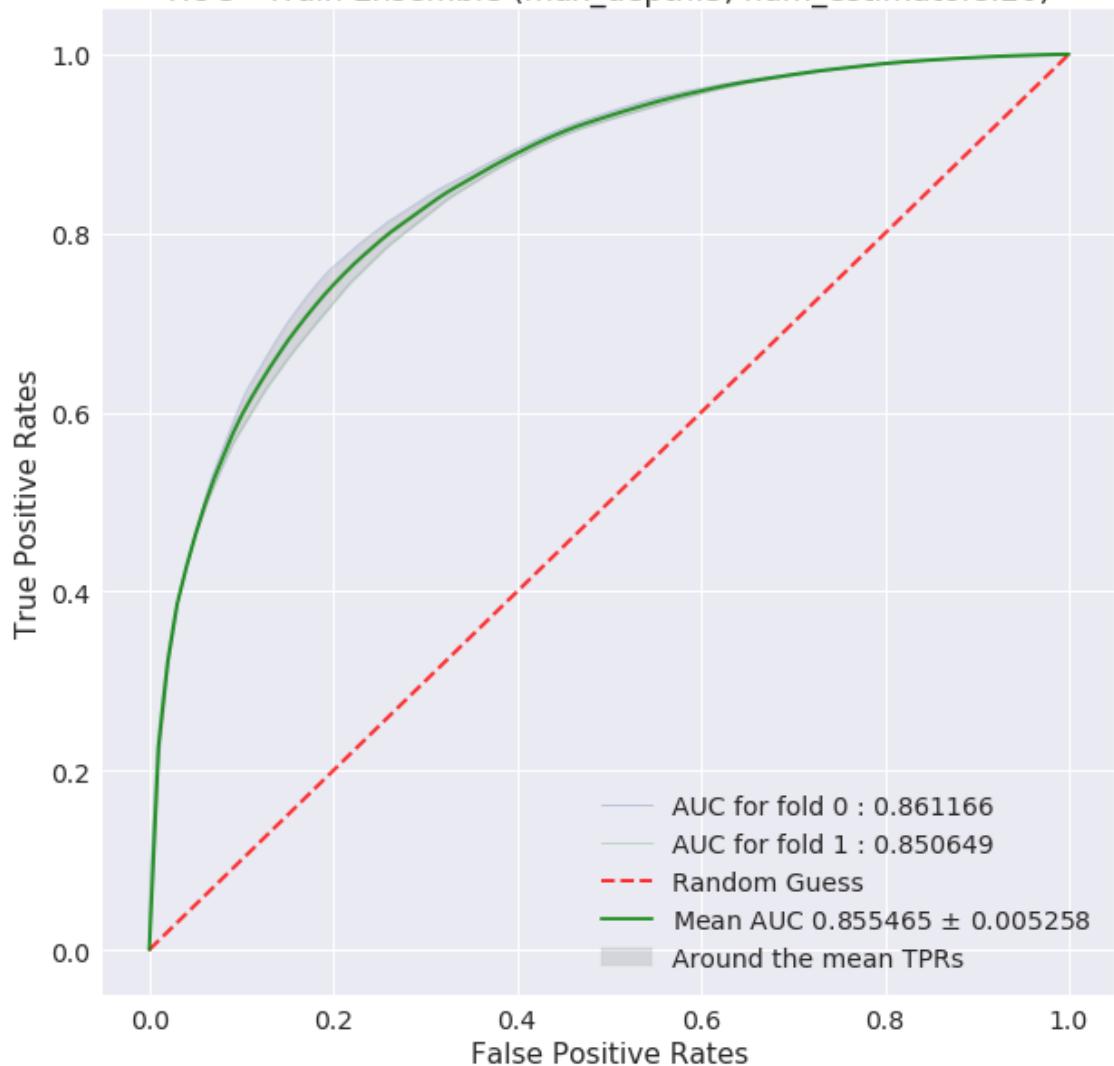
In [18]: # read the train, test data and preprocess it
    train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                                                           scaling=True,
                                                                           dim_reductio

    # train and validate the model
    model = train_and_validate_model(config_dict, train_features, train_labels)

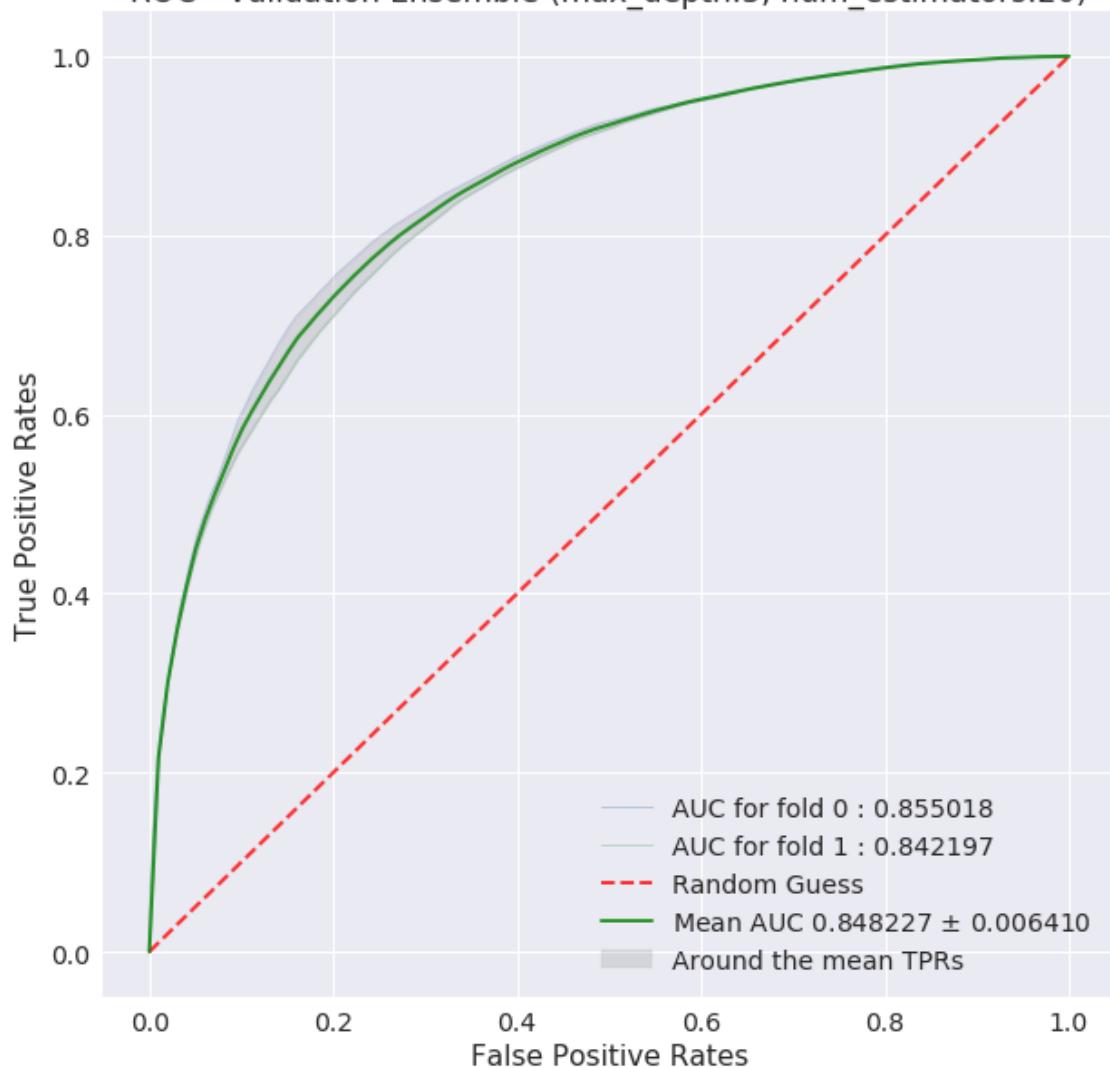
    # test and evaluate the model
    ptabe_entry_a3 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

Train df shape (50000, 52)
Class label distribution in train df:
0    25029
1    24971
Name: Label, dtype: int64
Test df shape (20000, 52)
Class label distribution in test df:
1    16520
0    3480
Name: Label, dtype: int64
Shape of -> train features :50000,50, test features: 20000,50
Shape of -> train labels :50000, test labels: 20000
=====
```

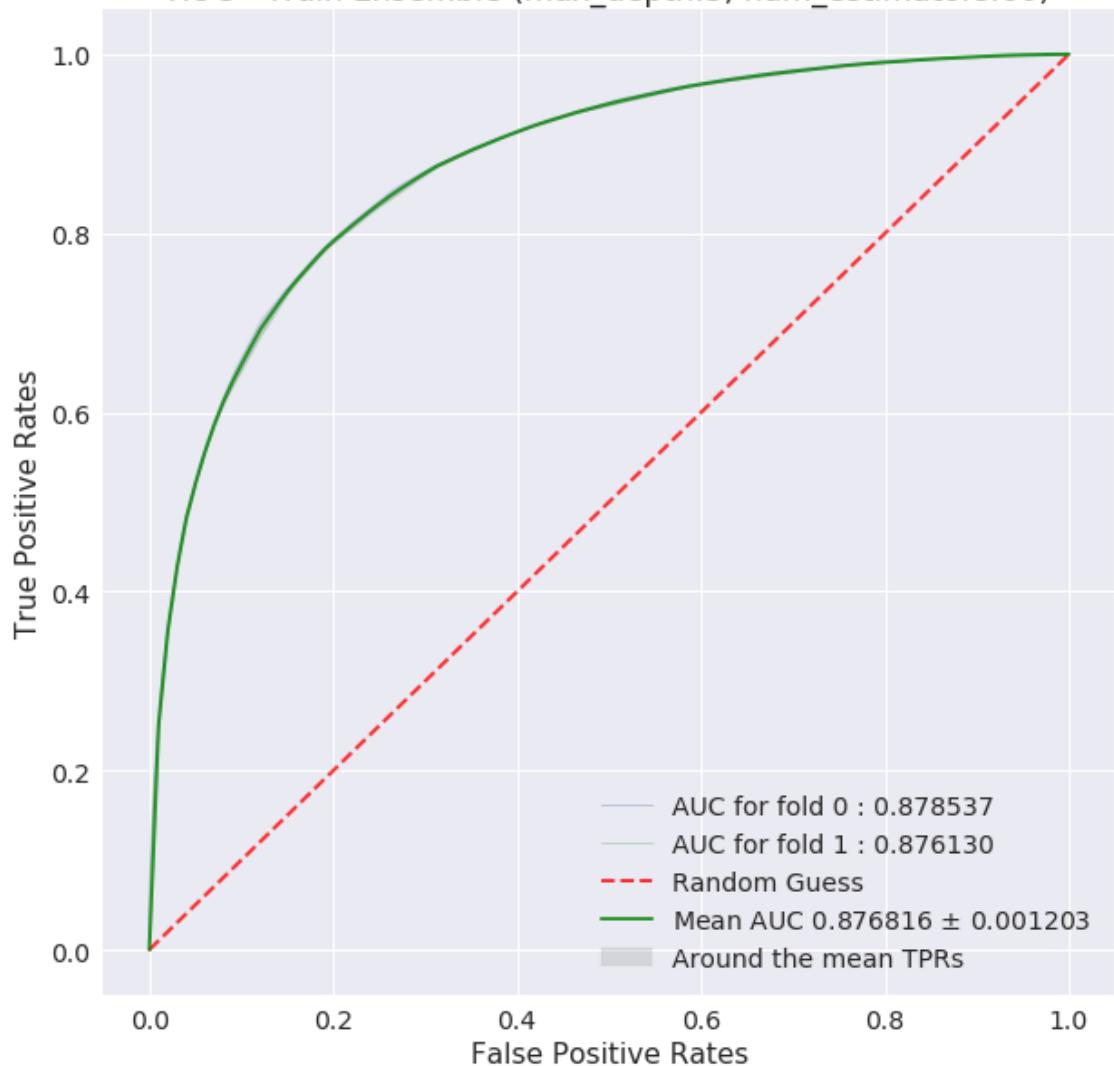
ROC - Train Ensemble (max\_depth:3, num\_estimators:20)



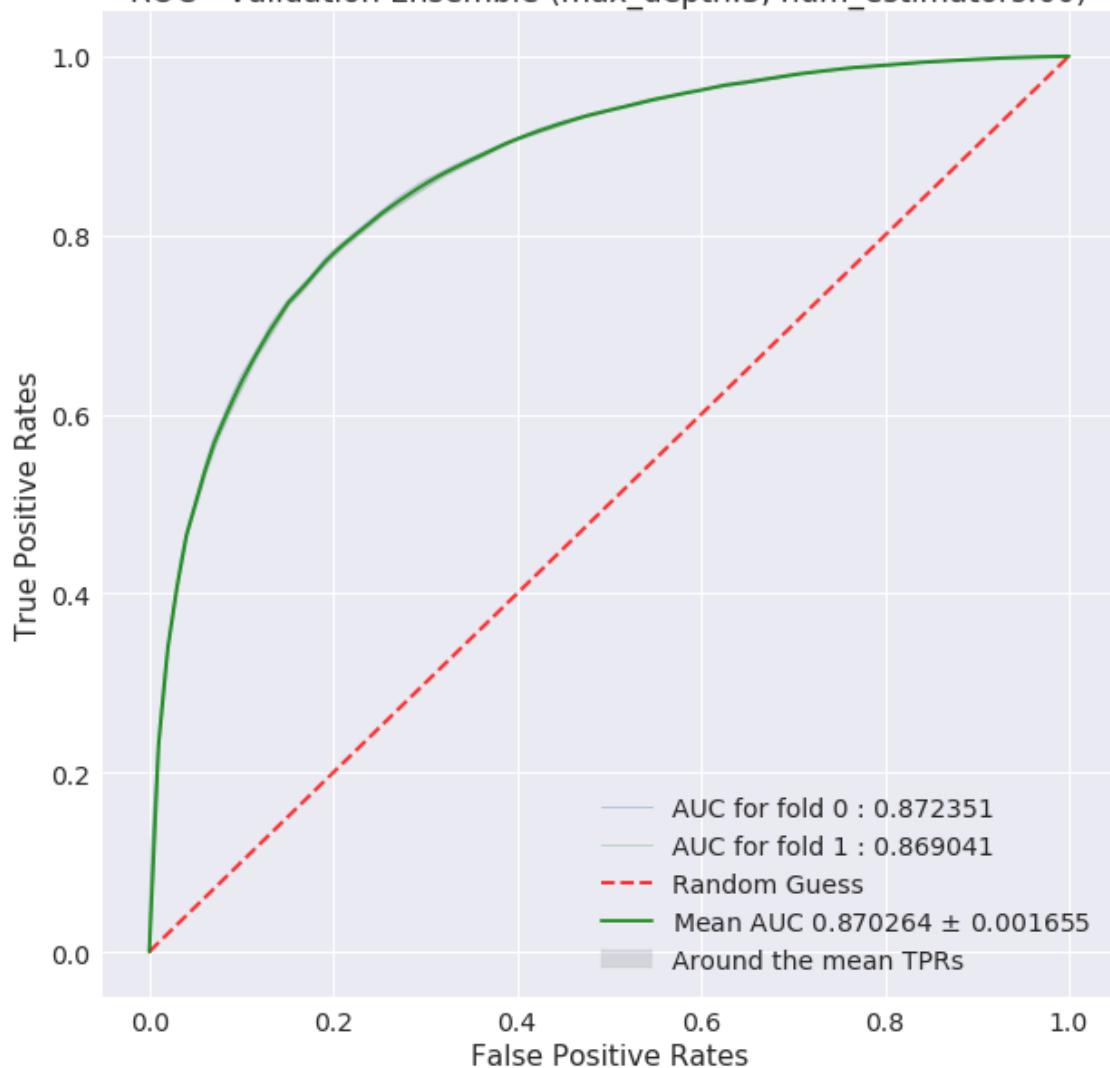
ROC - Validation Ensemble (max\_depth:3, num\_estimators:20)



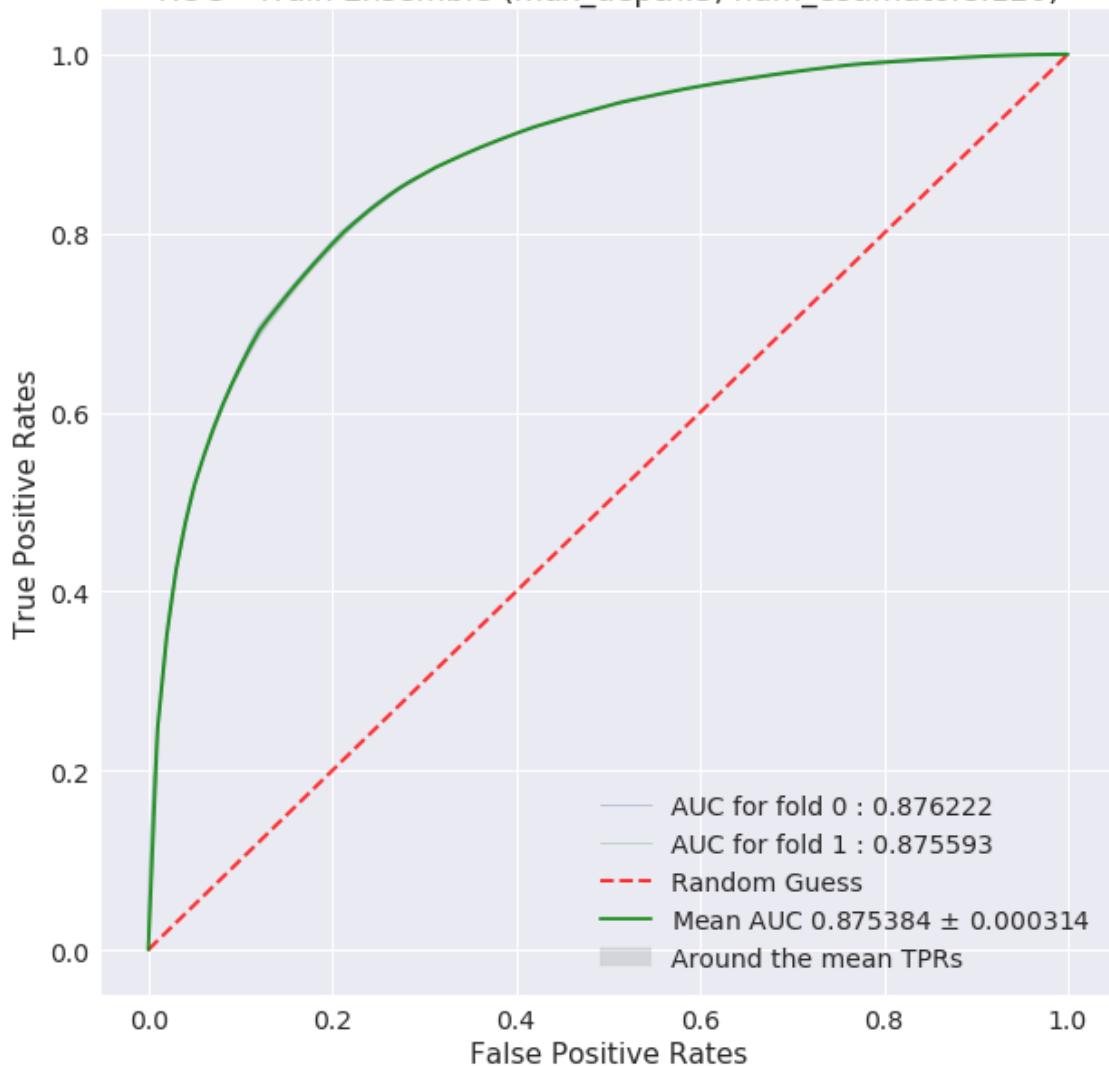
ROC - Train Ensemble (max\_depth:3, num\_estimators:60)



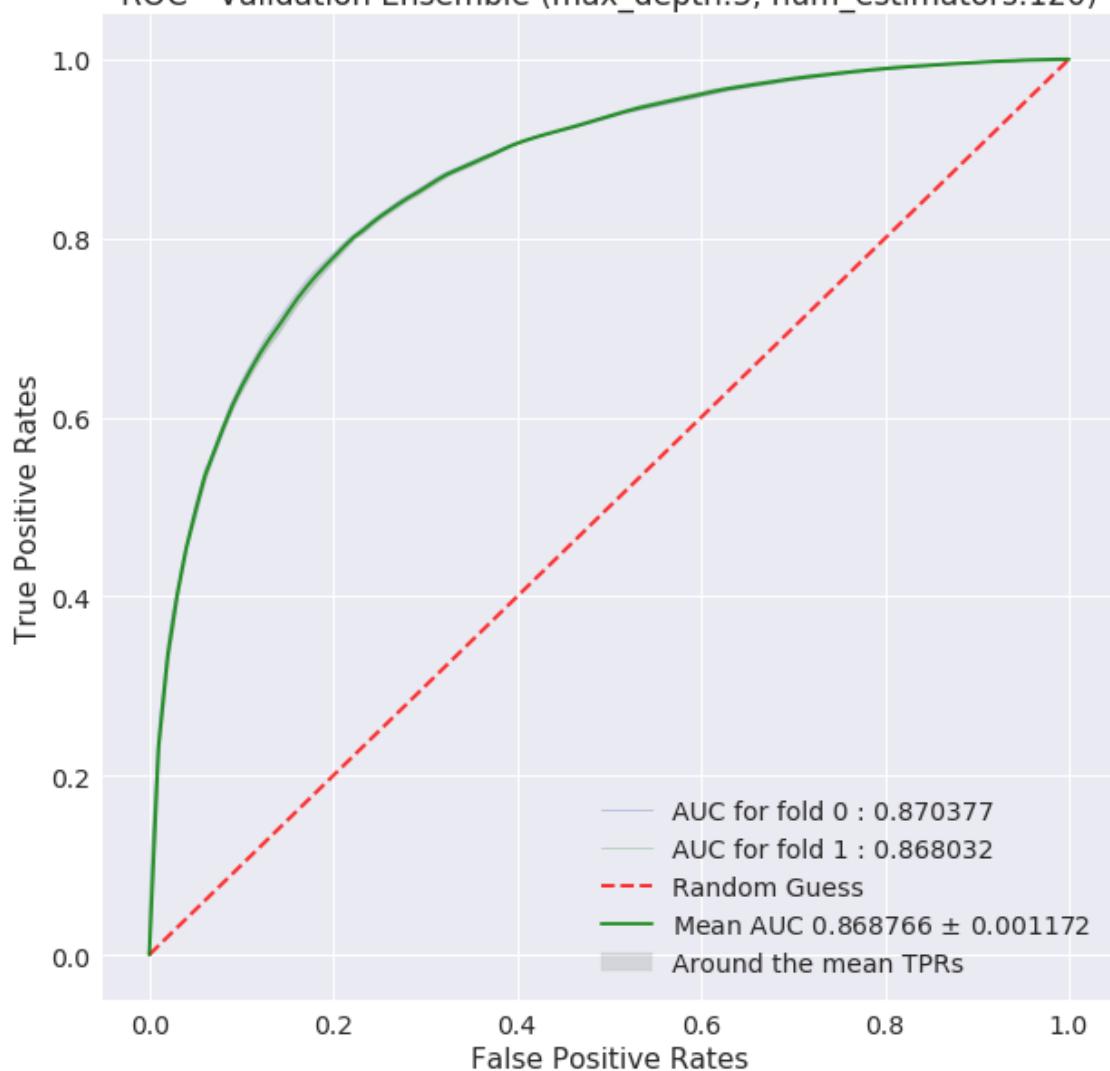
ROC - Validation Ensemble (max\_depth:3, num\_estimators:60)



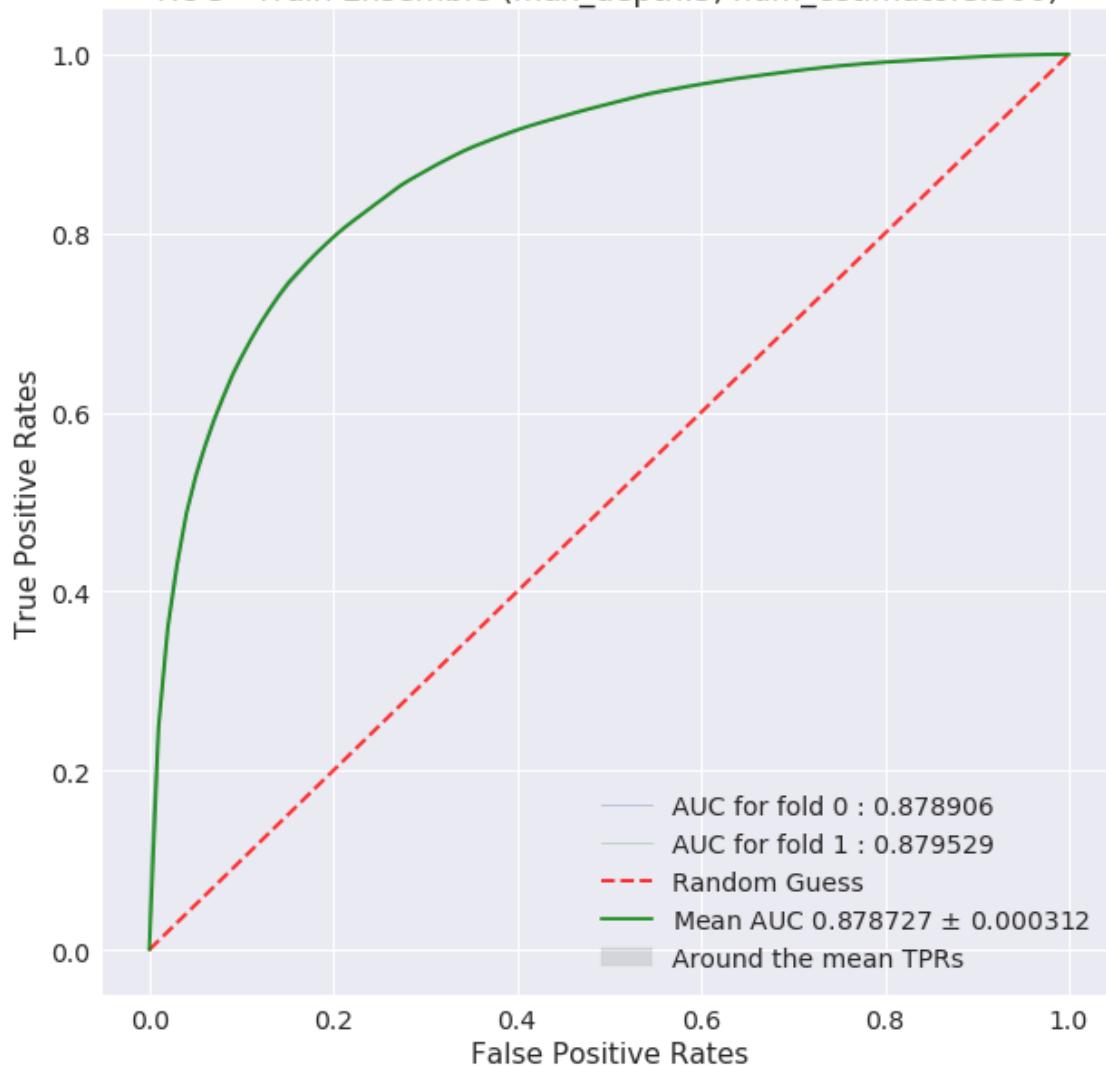
ROC - Train Ensemble (max\_depth:3, num\_estimators:120)



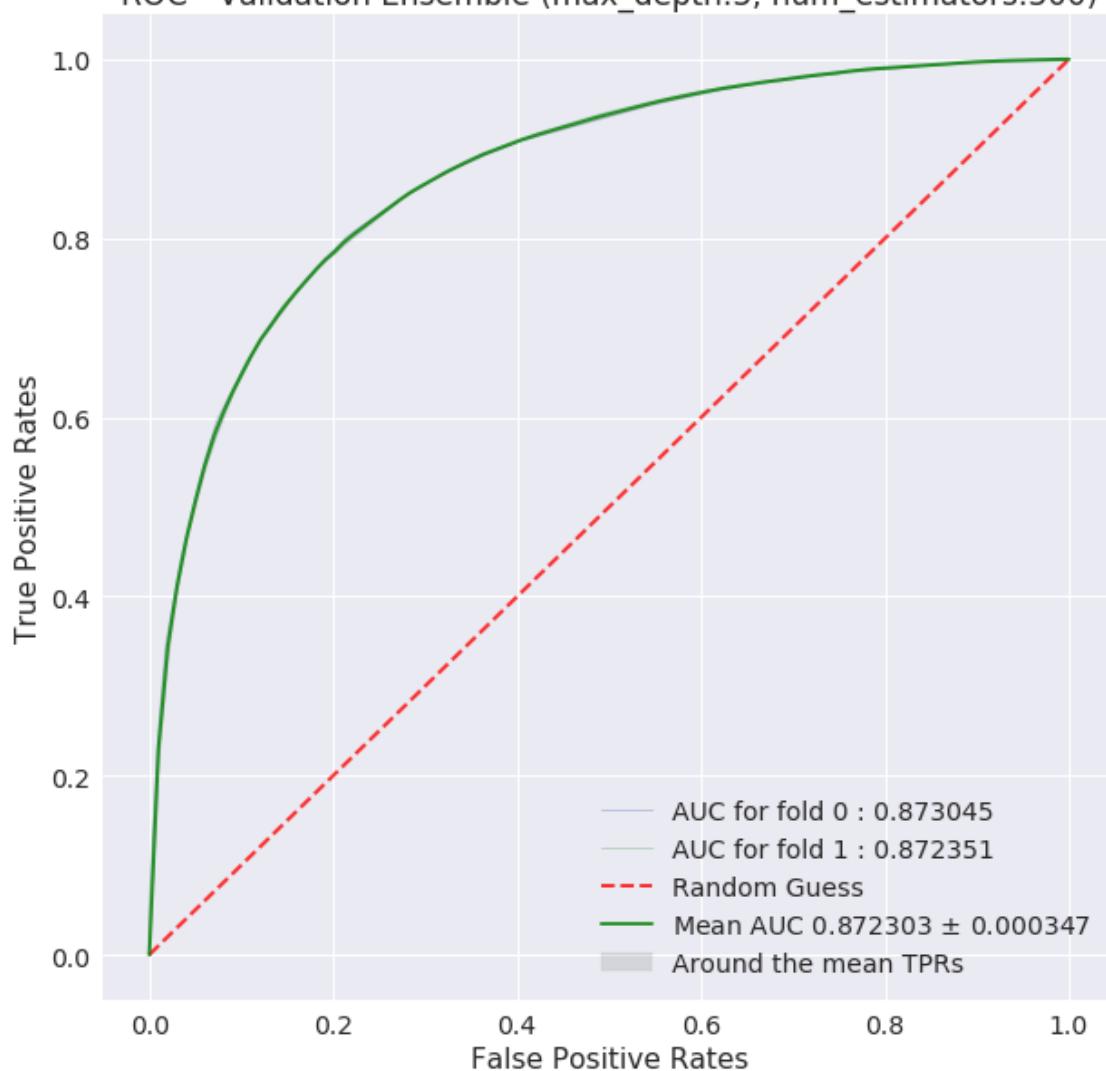
ROC - Validation Ensemble (max\_depth:3, num\_estimators:120)



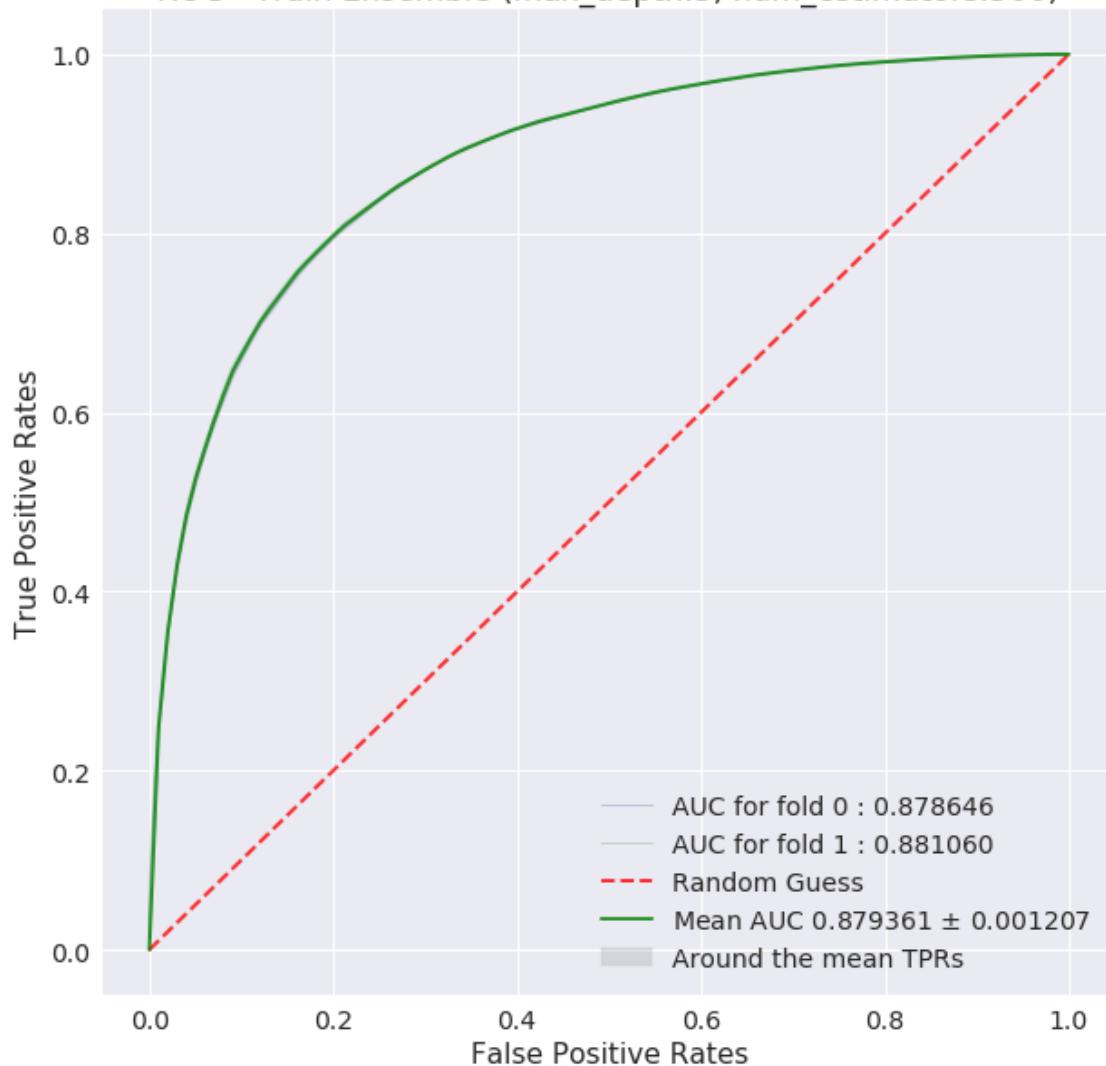
ROC - Train Ensemble (max\_depth:3, num\_estimators:300)



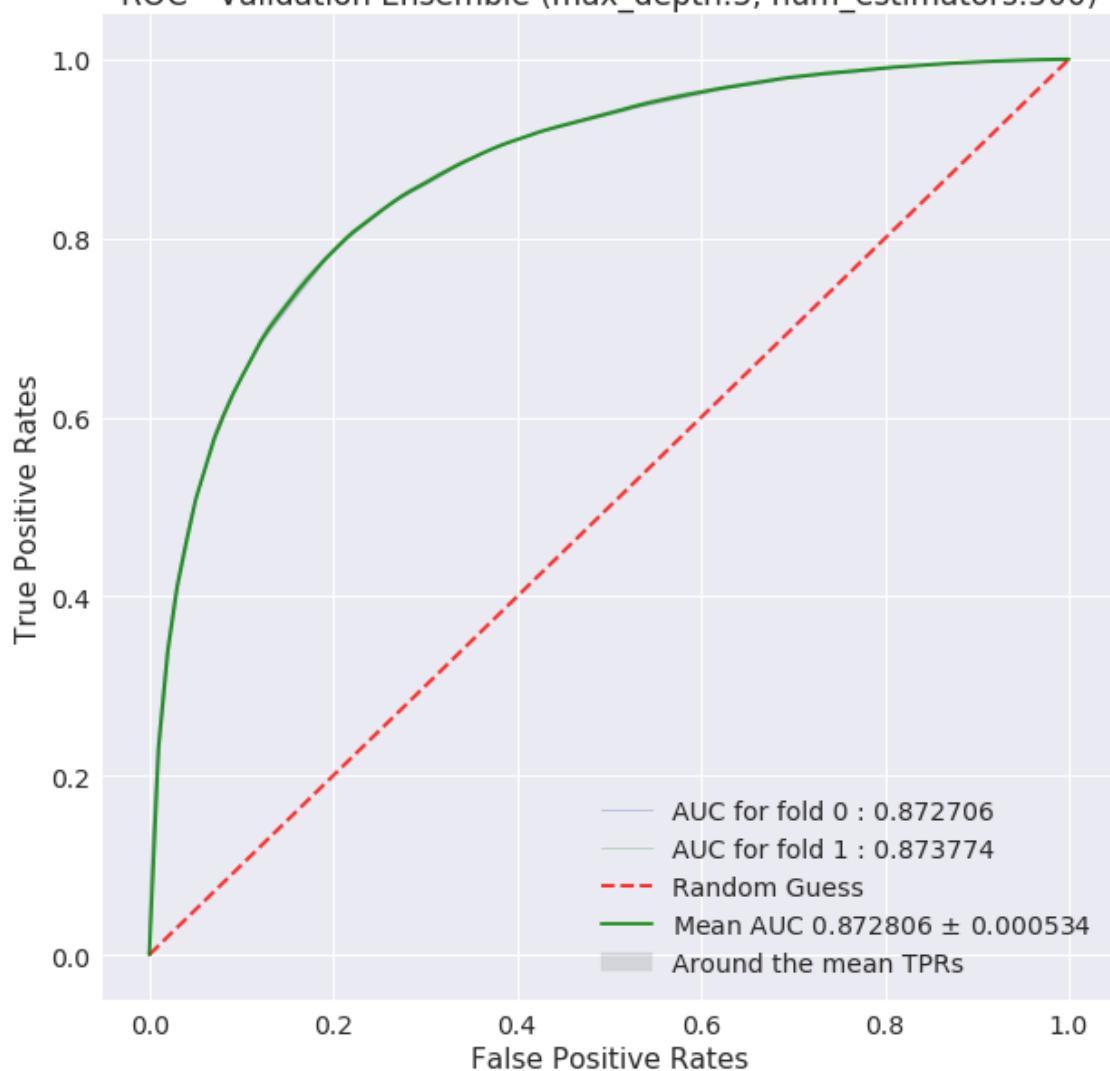
ROC - Validation Ensemble (max\_depth:3, num\_estimators:300)



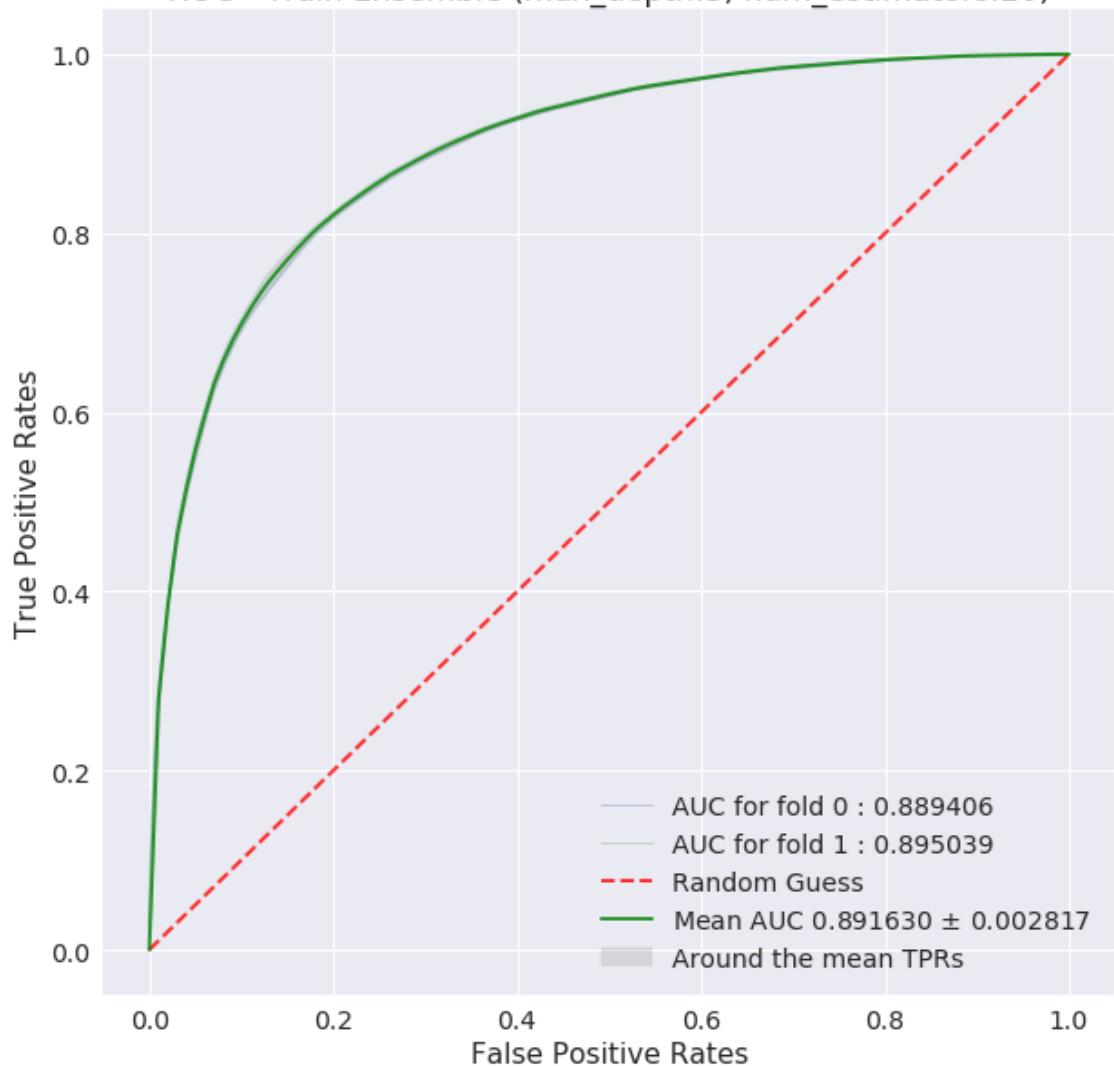
ROC - Train Ensemble (max\_depth:3, num\_estimators:500)



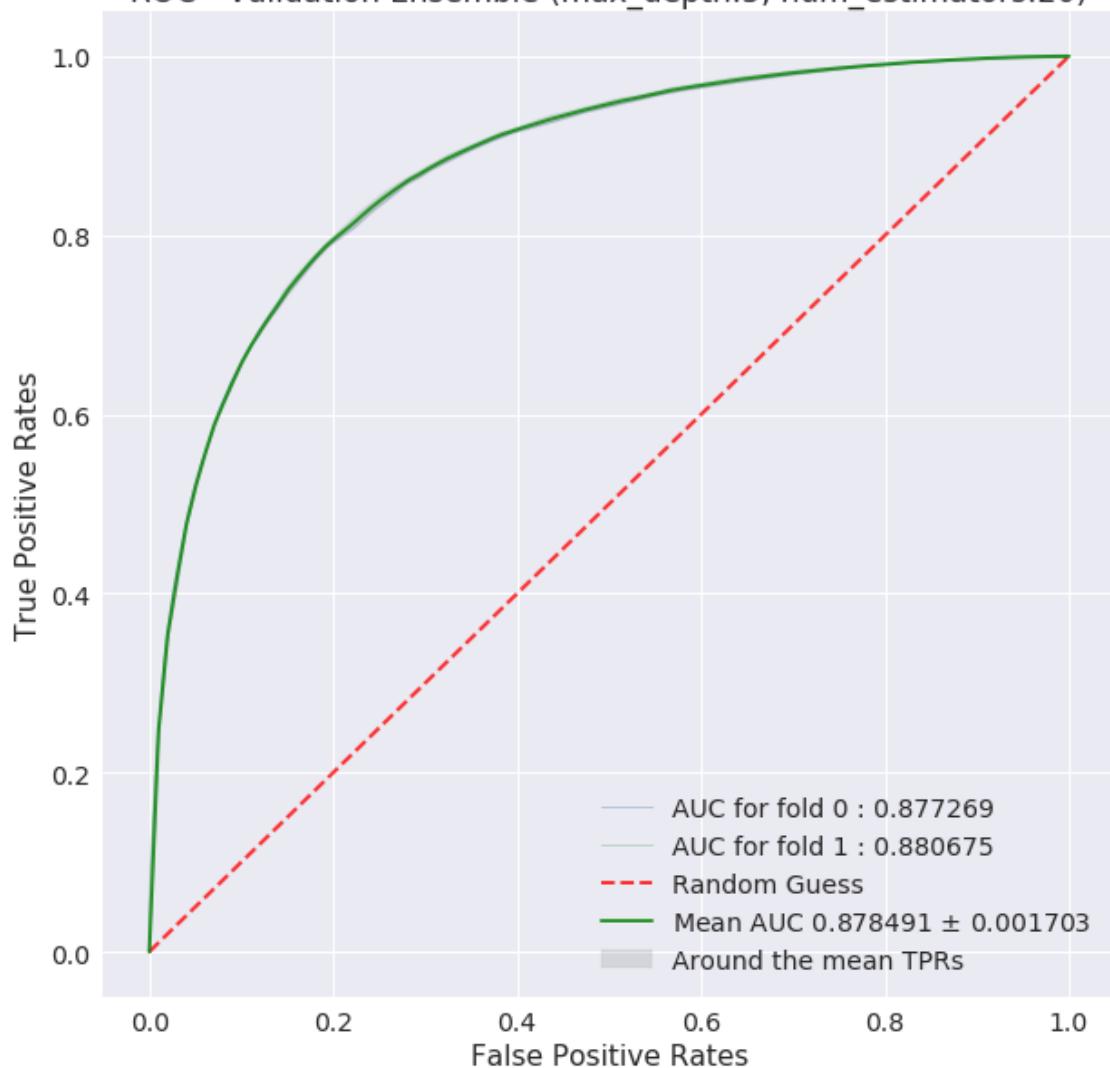
ROC - Validation Ensemble (max\_depth:3, num\_estimators:500)



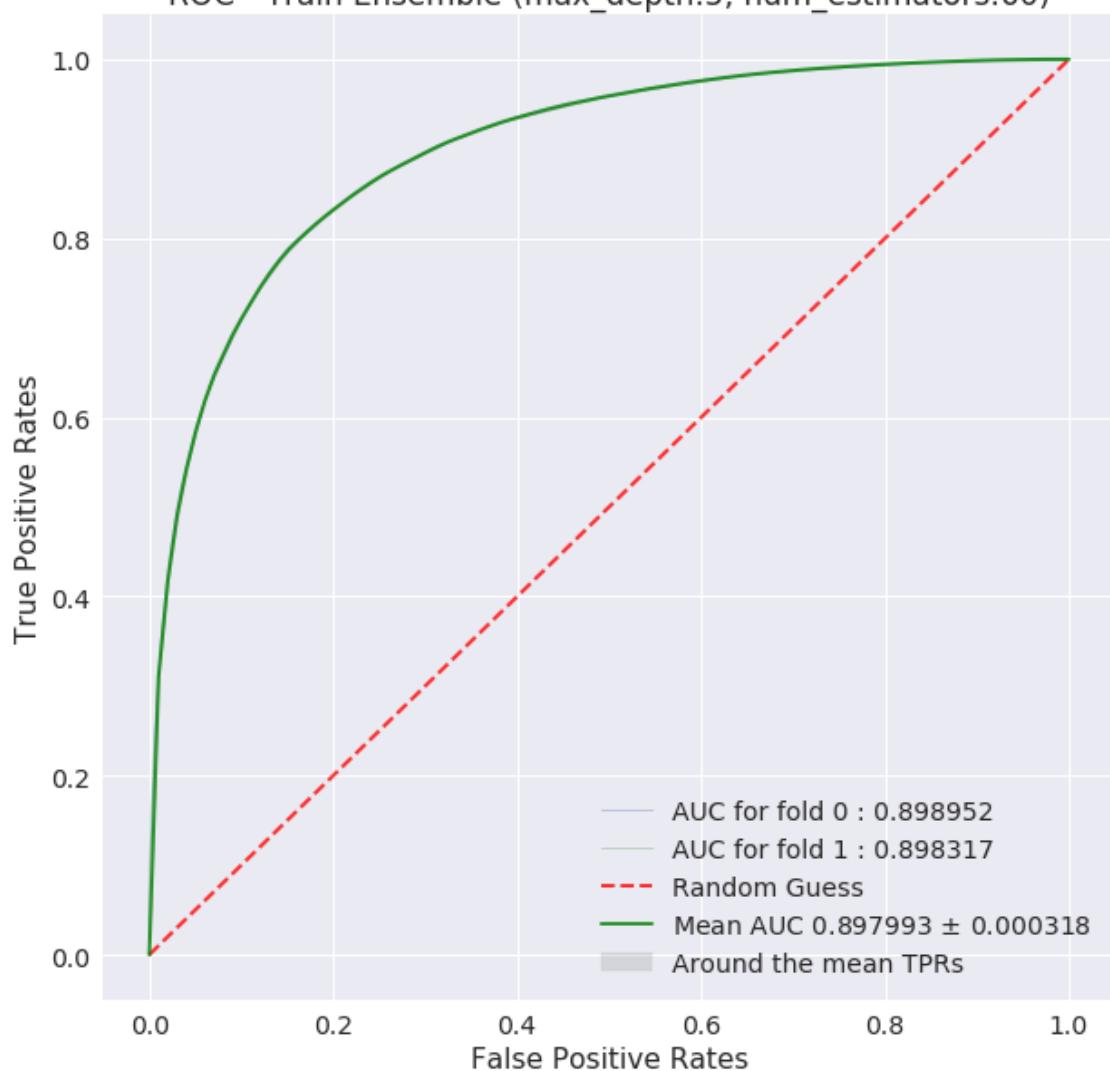
ROC - Train Ensemble (max\_depth:5, num\_estimators:20)



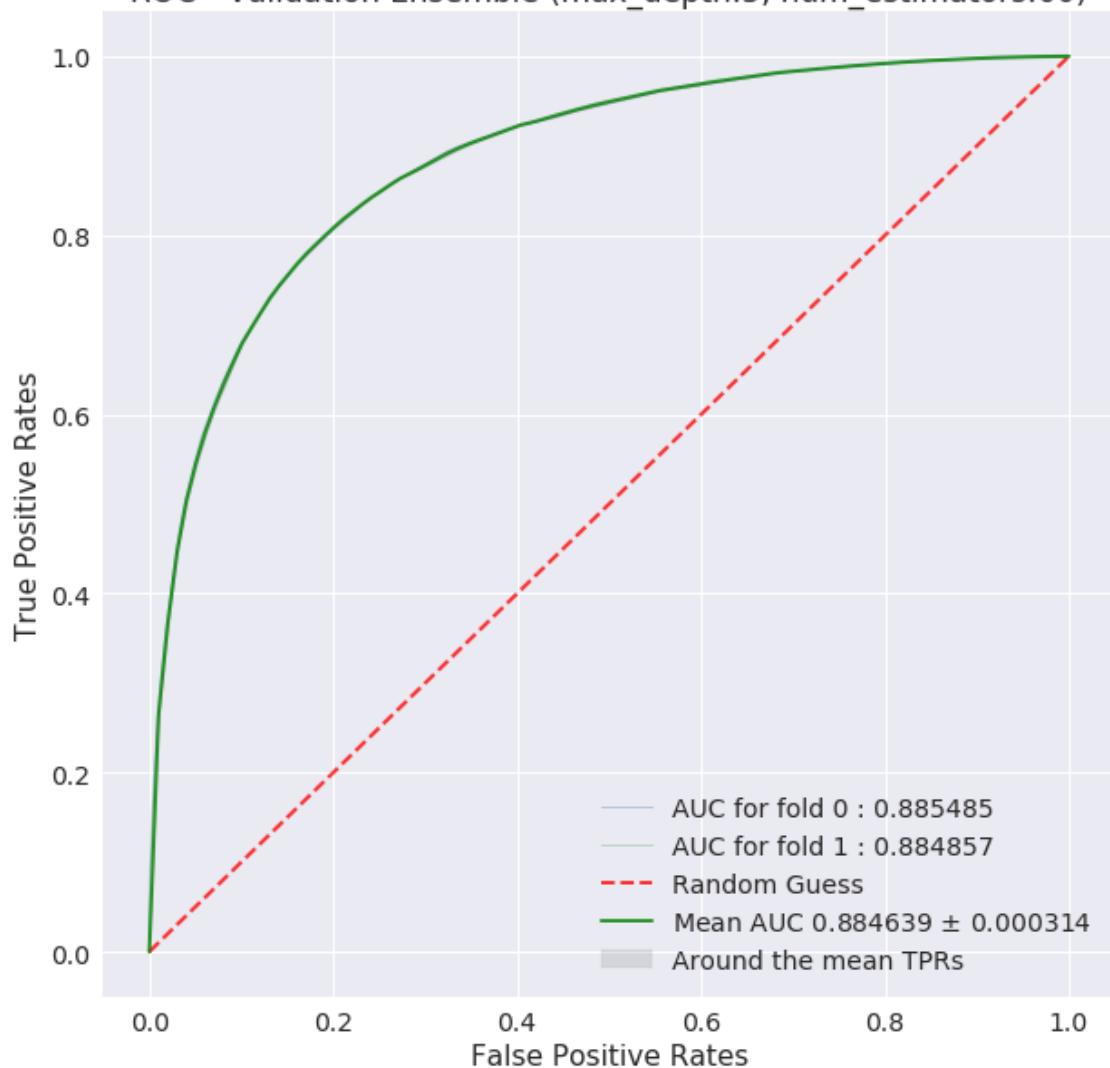
ROC - Validation Ensemble (max\_depth:5, num\_estimators:20)



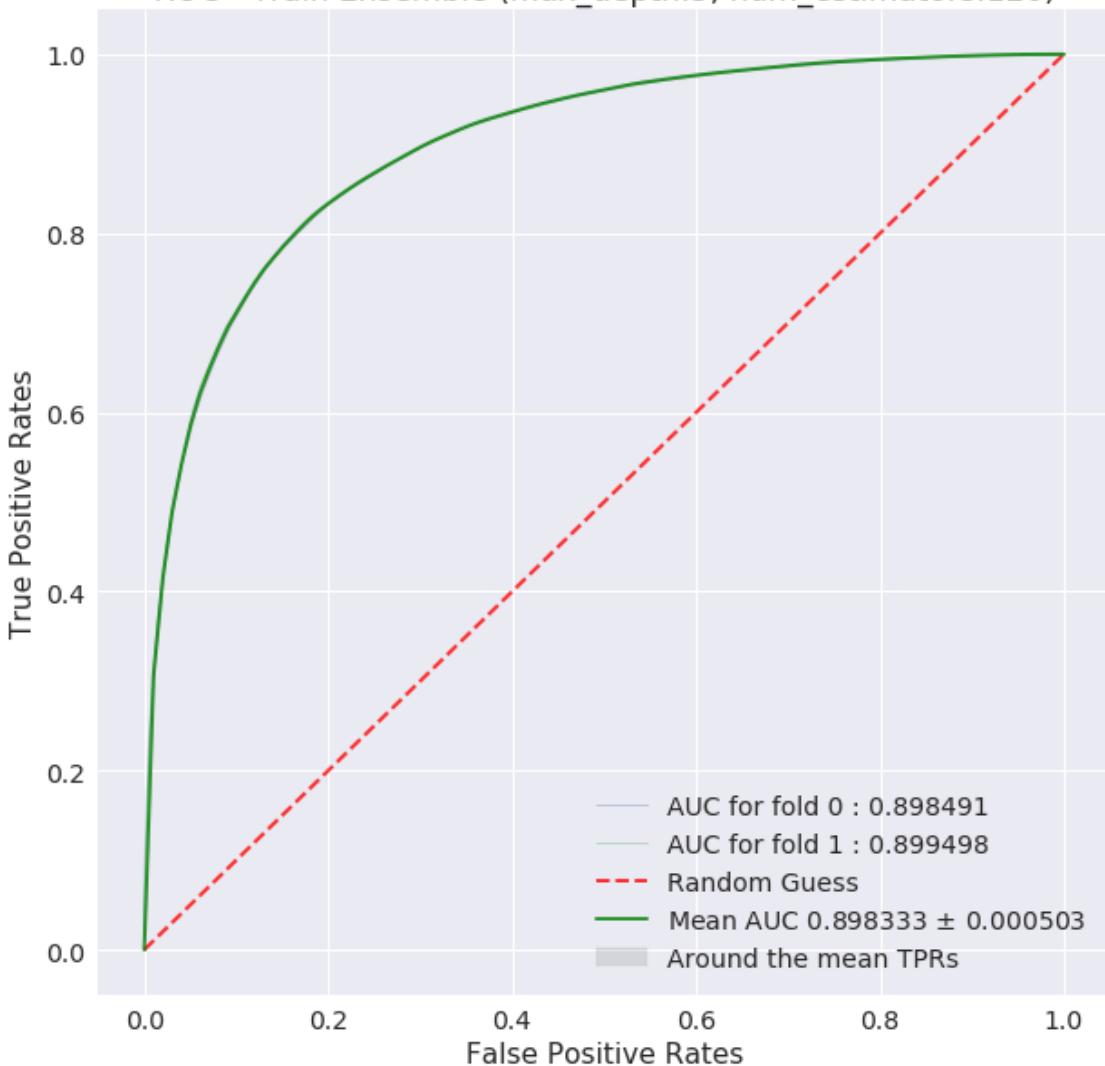
ROC - Train Ensemble (max\_depth:5, num\_estimators:60)



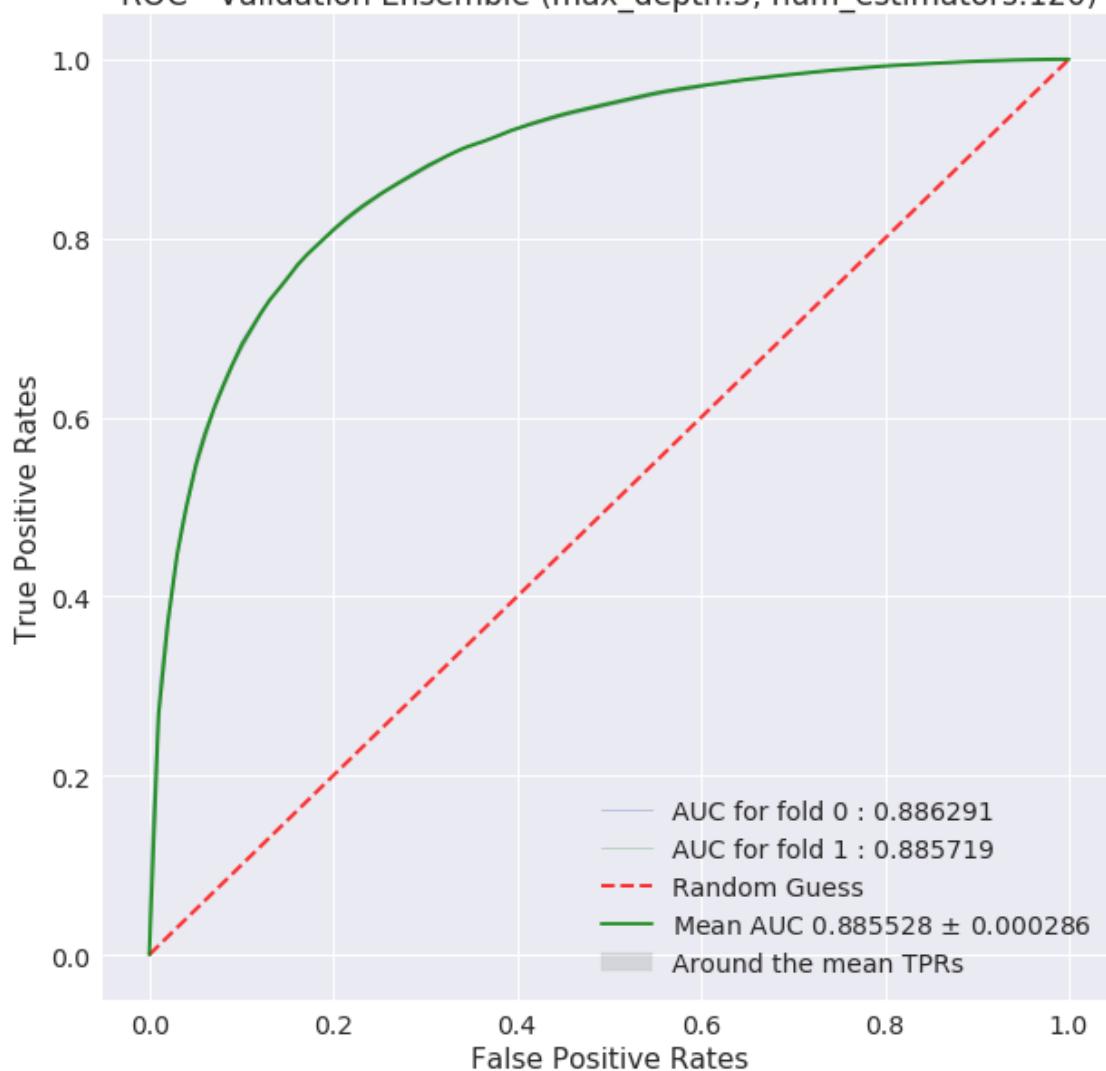
ROC - Validation Ensemble (max\_depth:5, num\_estimators:60)



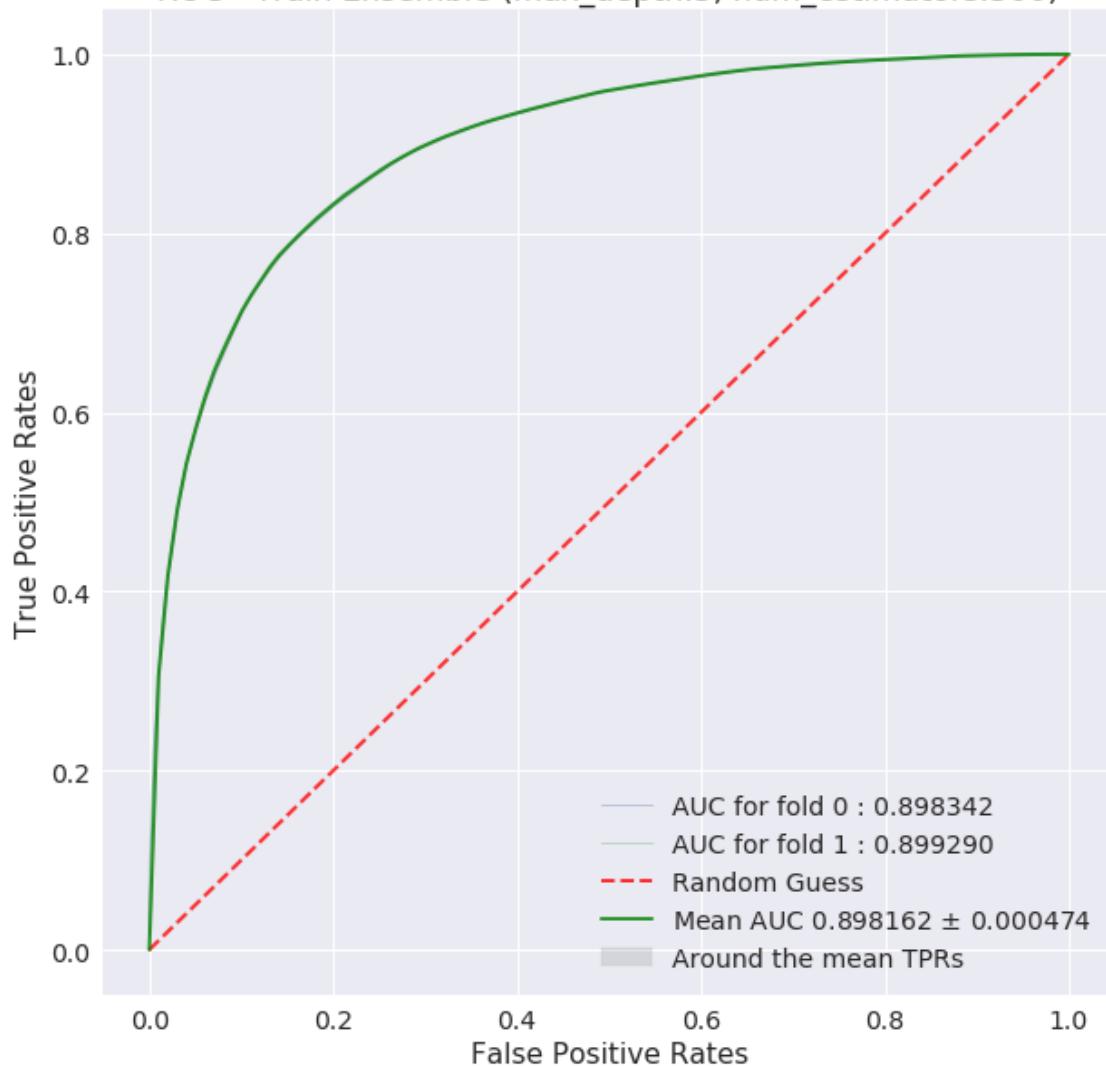
ROC - Train Ensemble (max\_depth:5, num\_estimators:120)



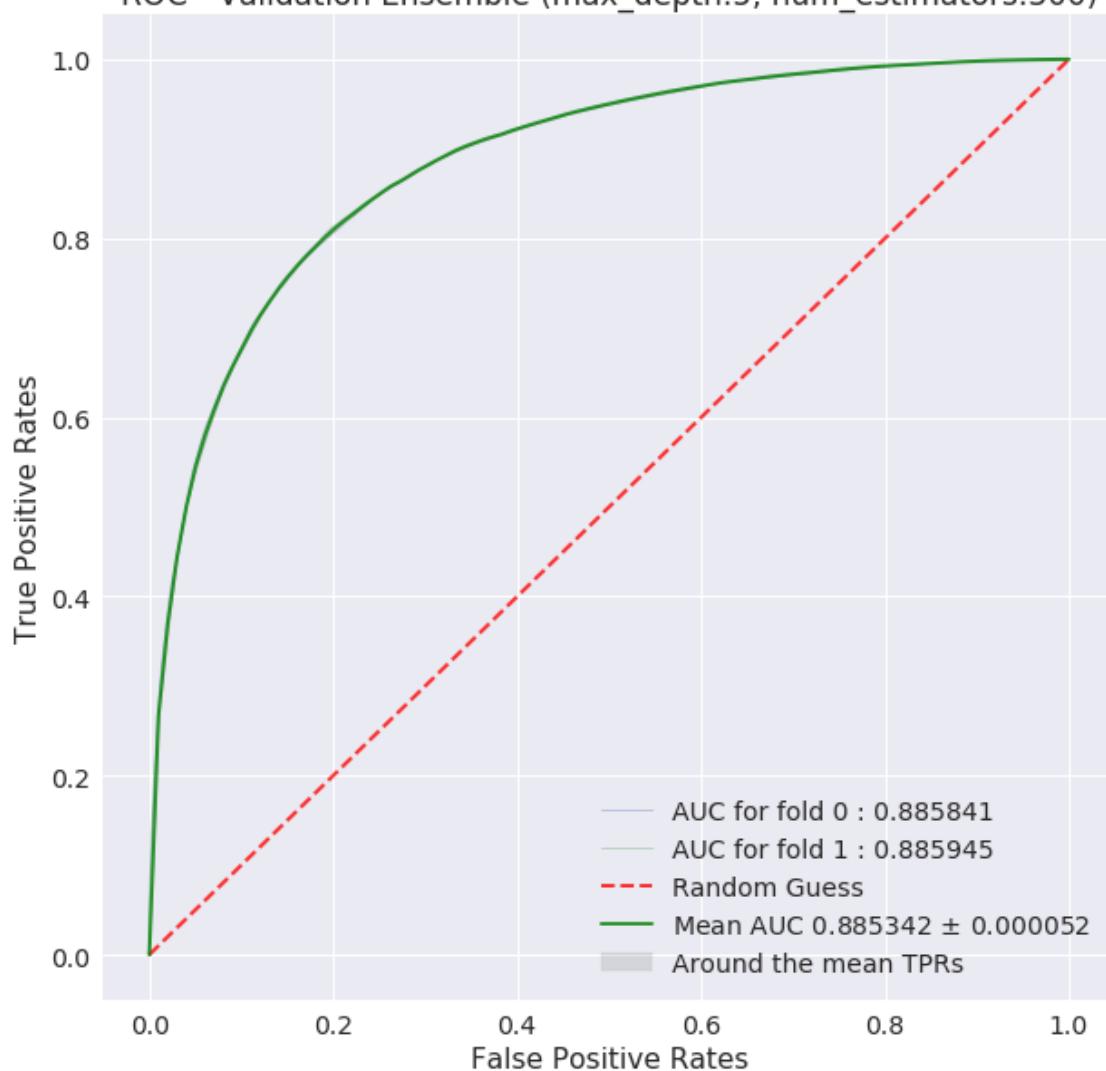
ROC - Validation Ensemble (max\_depth:5, num\_estimators:120)



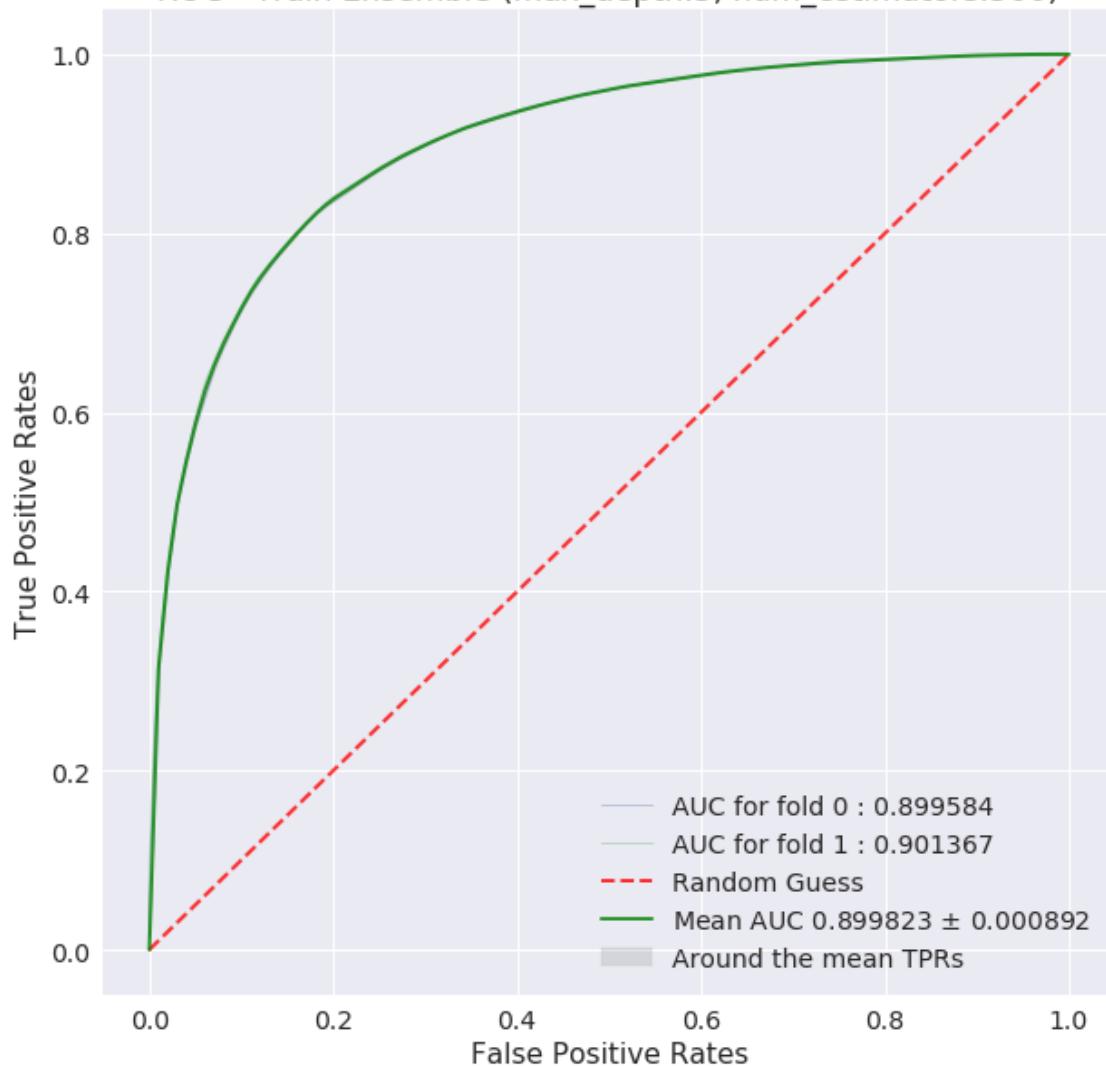
ROC - Train Ensemble (max\_depth:5, num\_estimators:300)



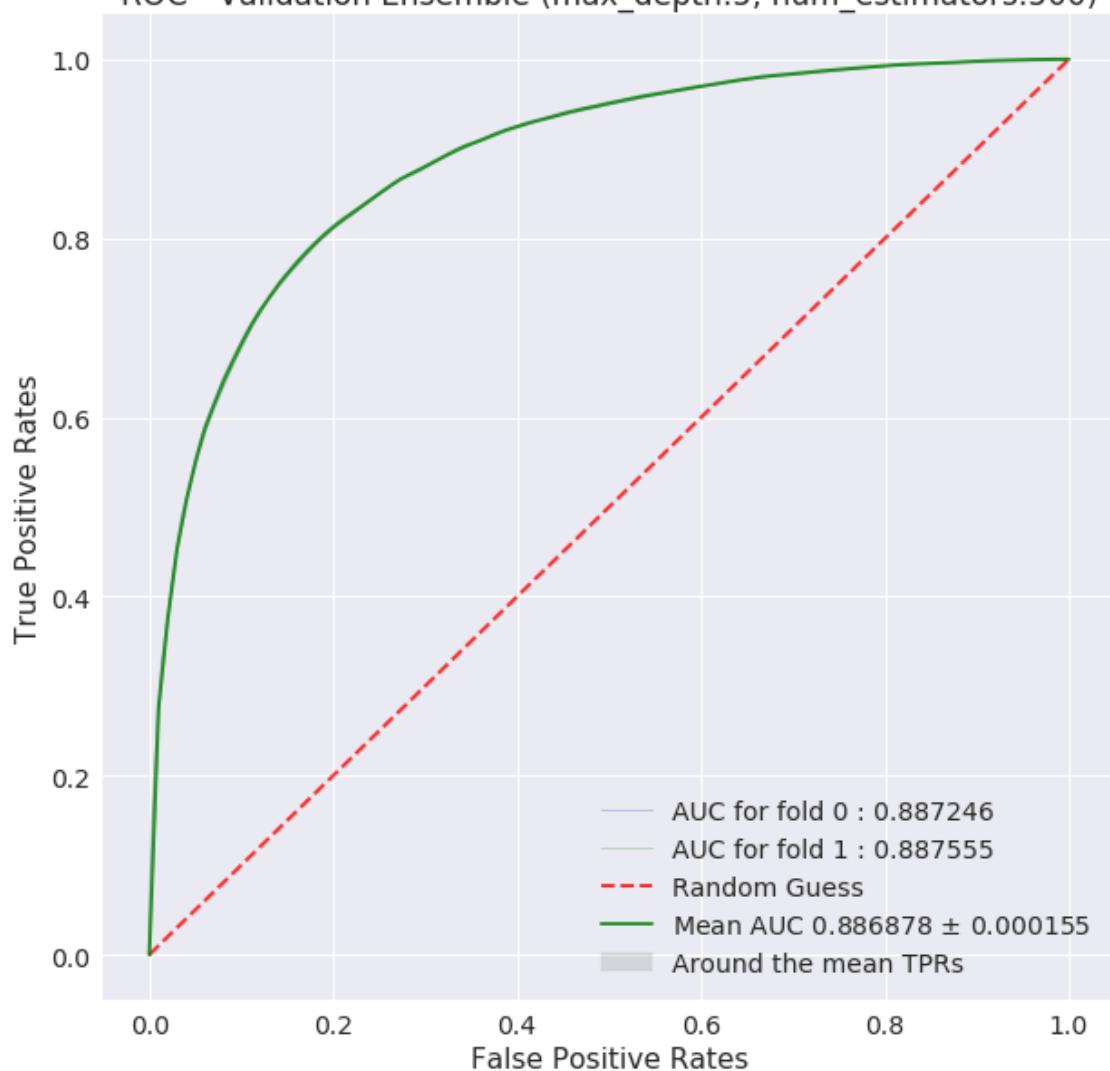
ROC - Validation Ensemble (max\_depth:5, num\_estimators:300)



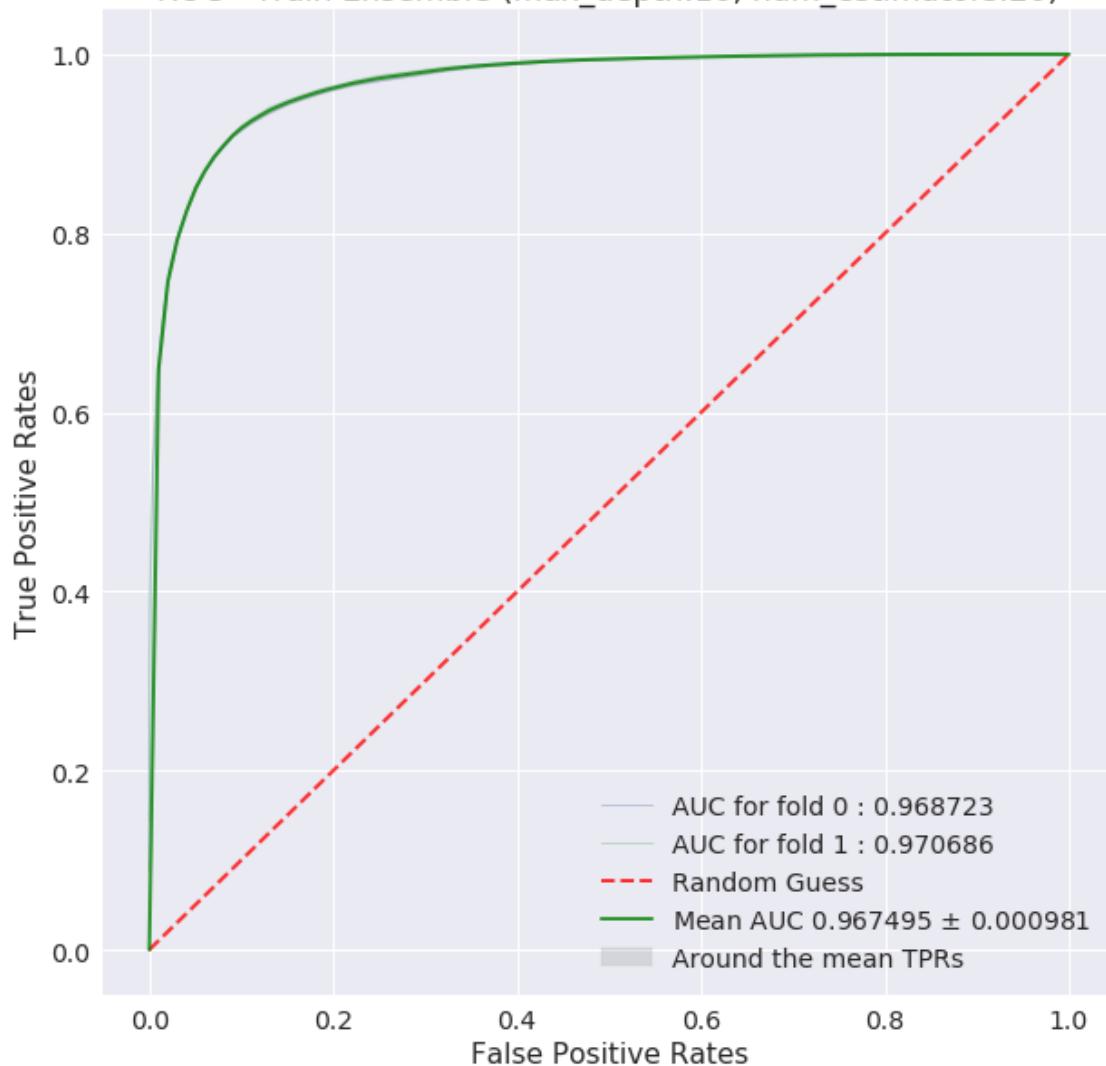
ROC - Train Ensemble (max\_depth:5, num\_estimators:500)

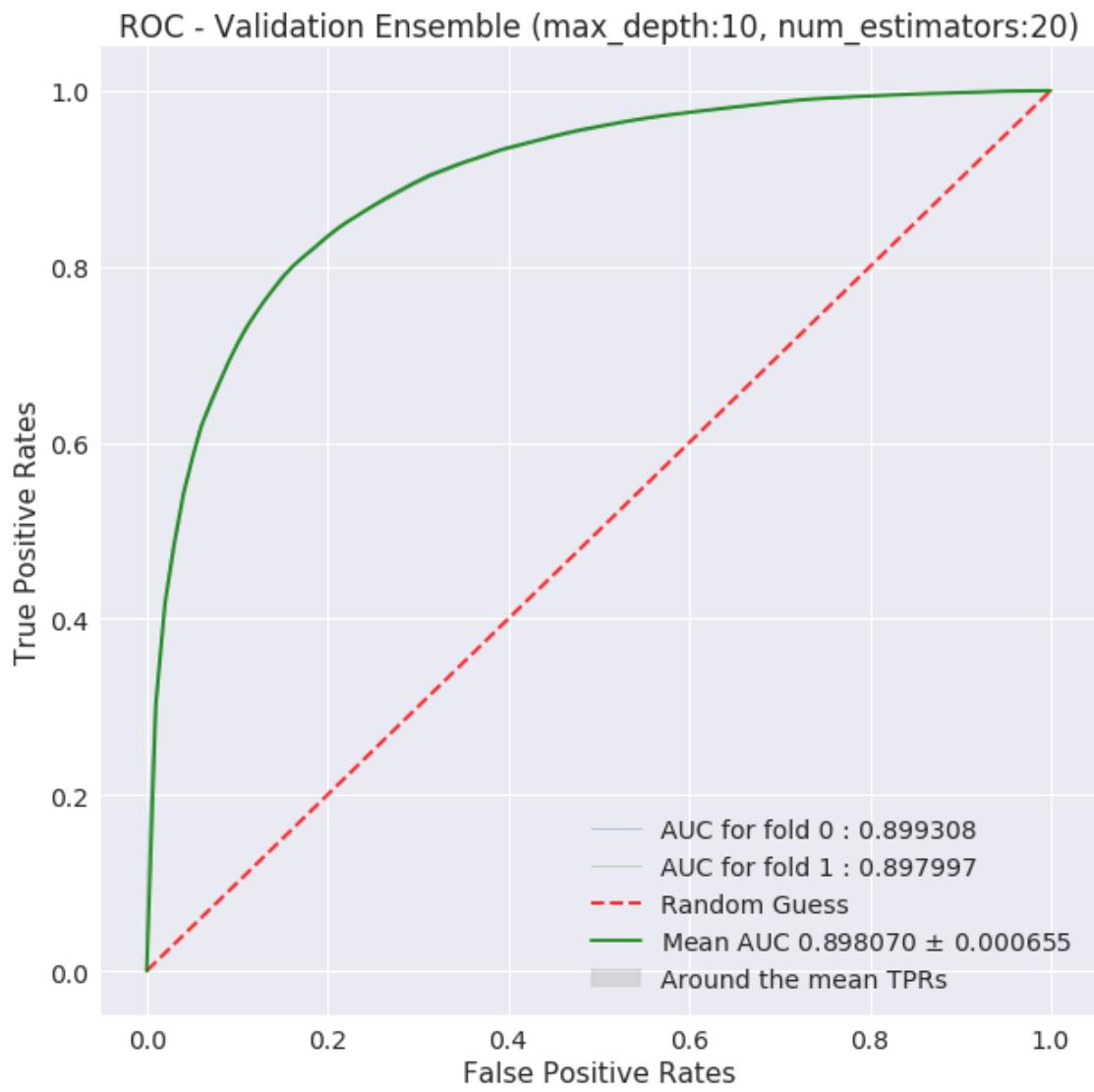


ROC - Validation Ensemble (max\_depth:5, num\_estimators:500)

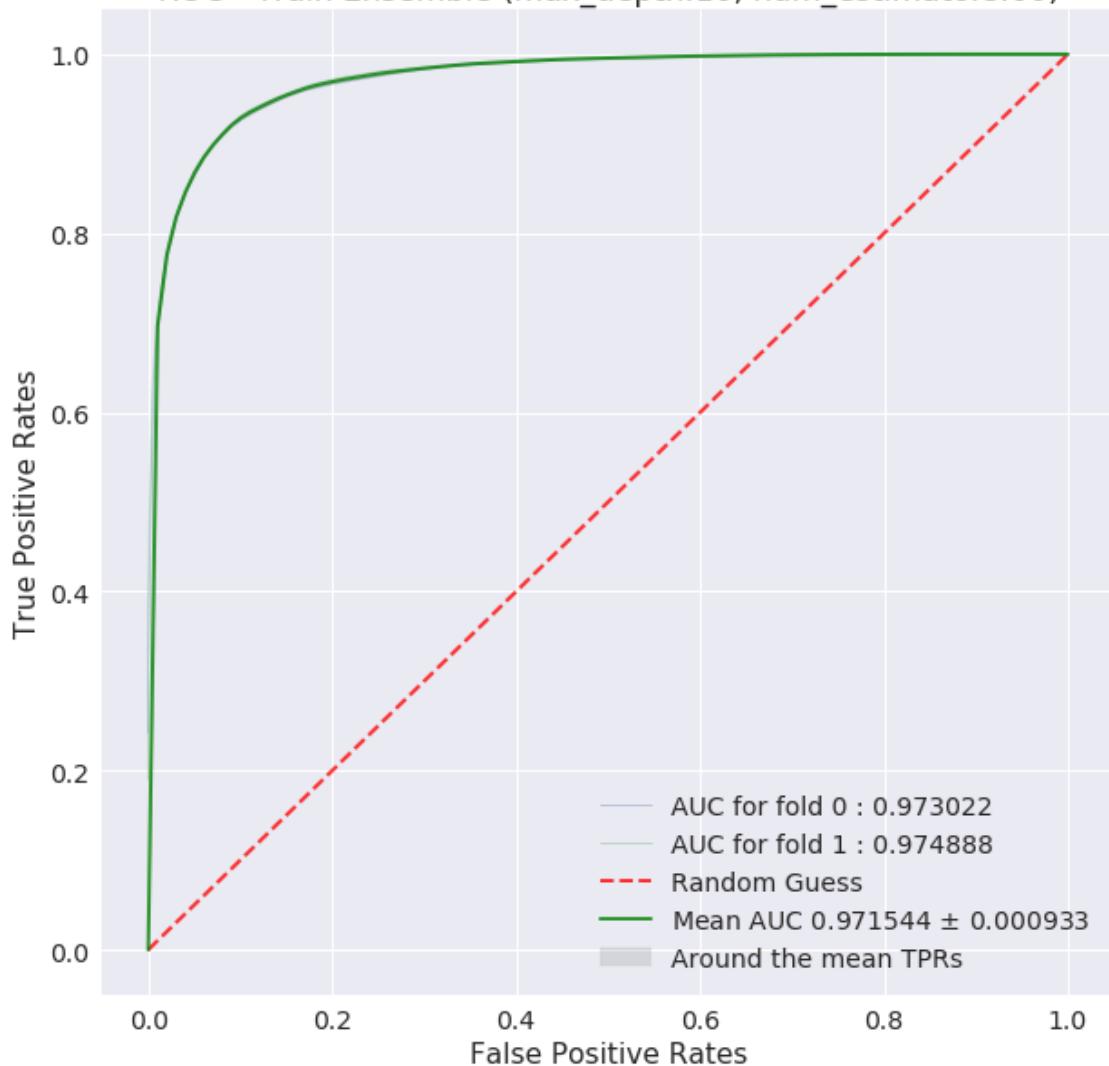


ROC - Train Ensemble (max\_depth:10, num\_estimators:20)

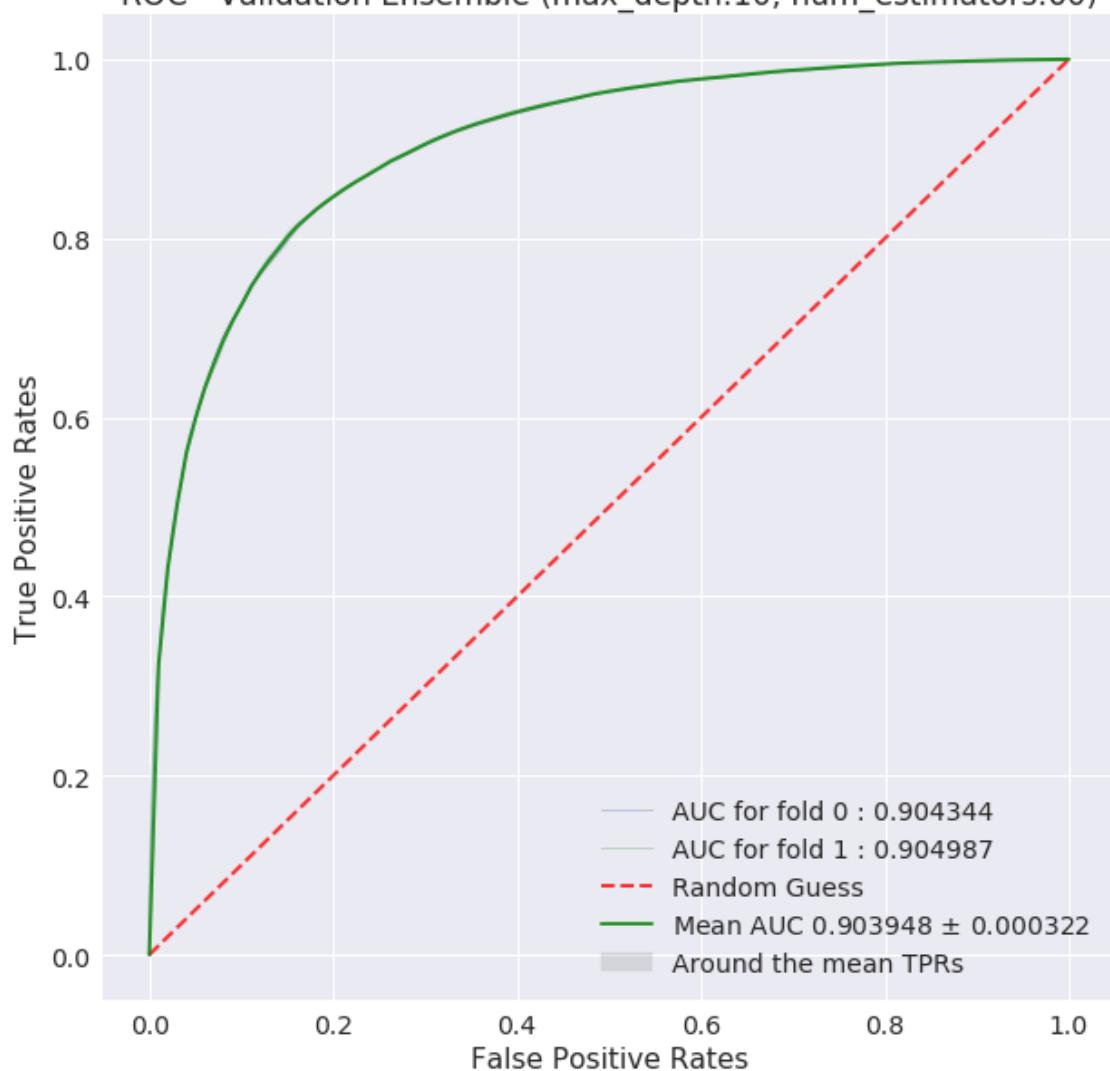




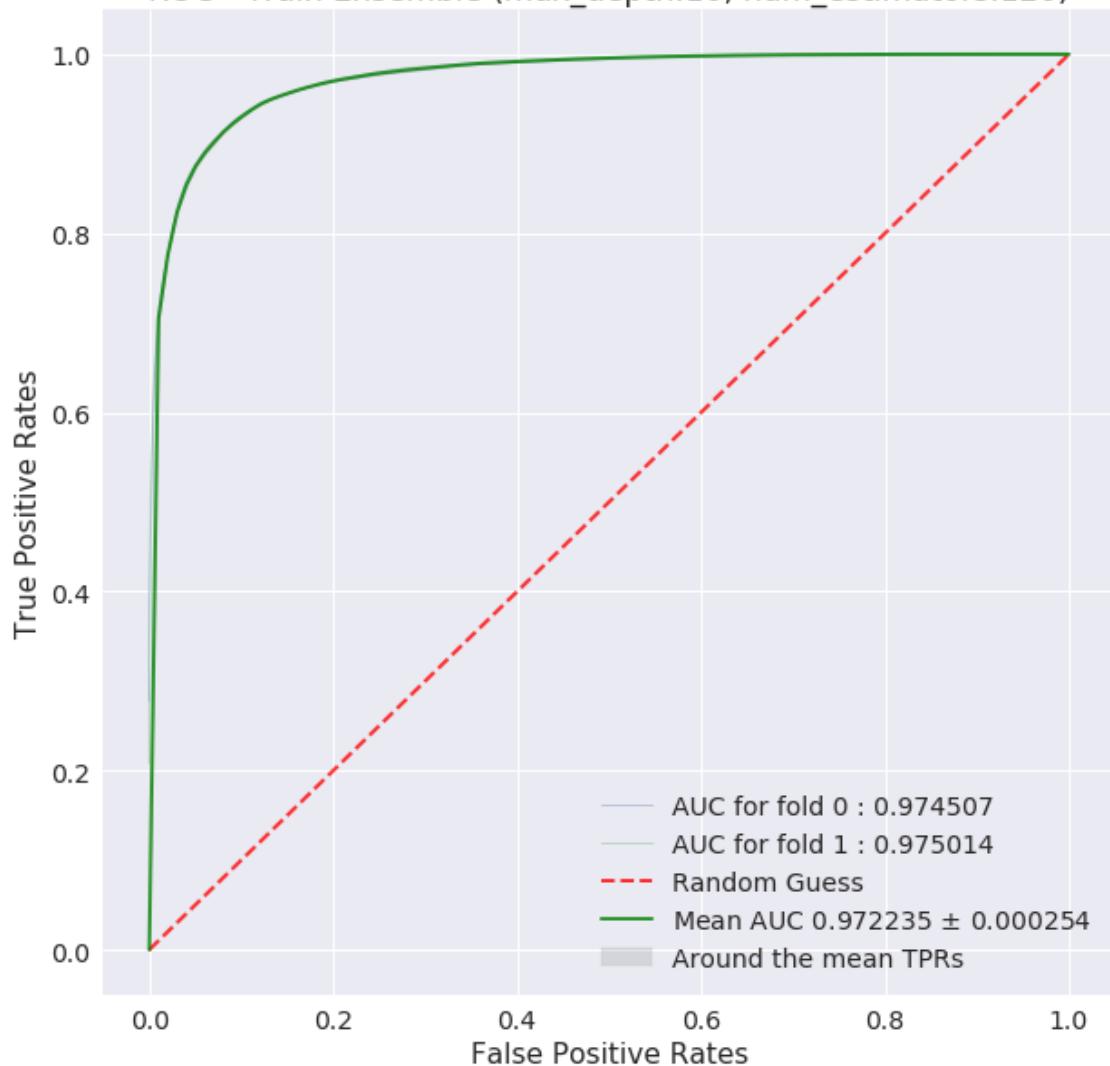
ROC - Train Ensemble (max\_depth:10, num\_estimators:60)



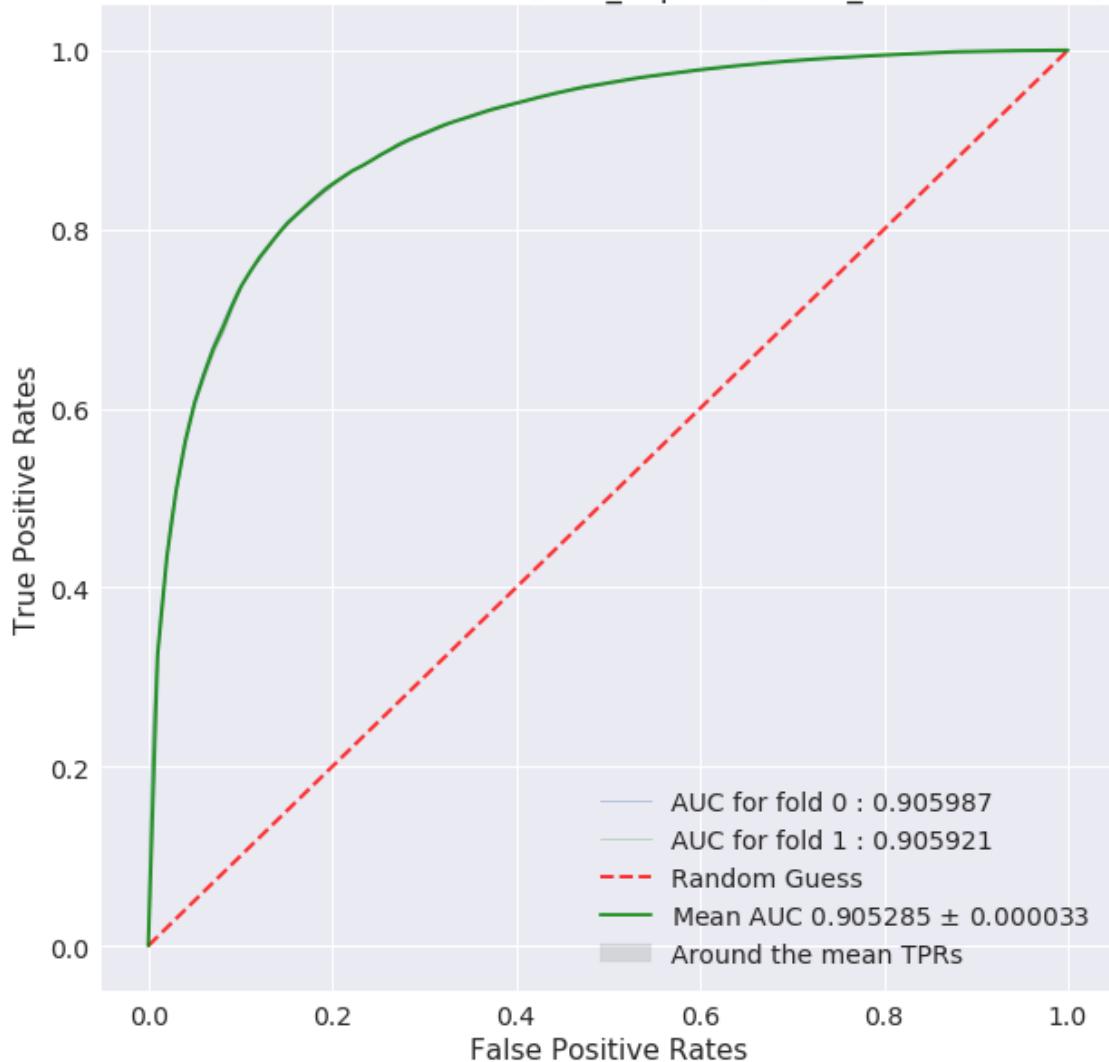
ROC - Validation Ensemble (max\_depth:10, num\_estimators:60)



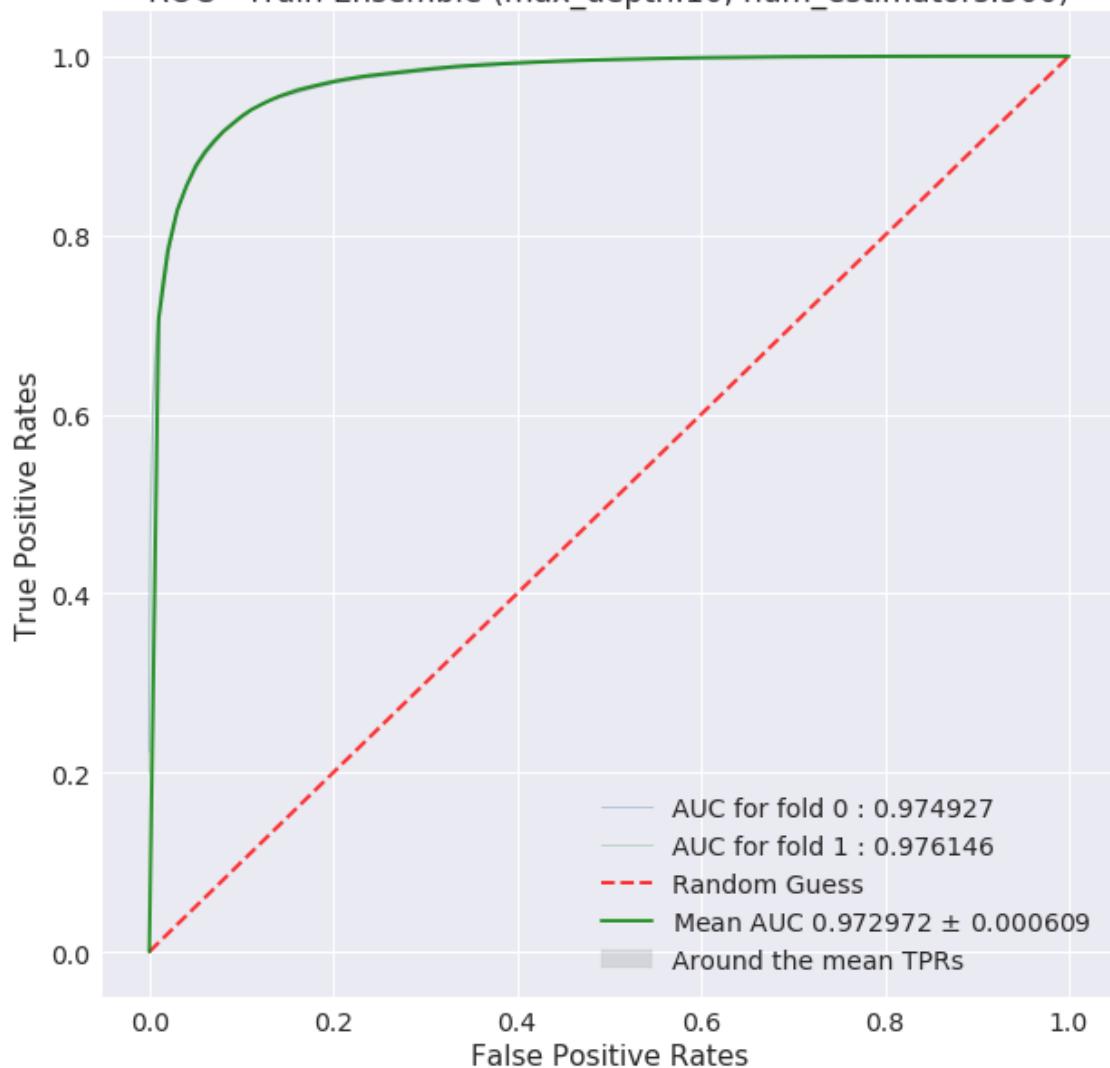
ROC - Train Ensemble (max\_depth:10, num\_estimators:120)



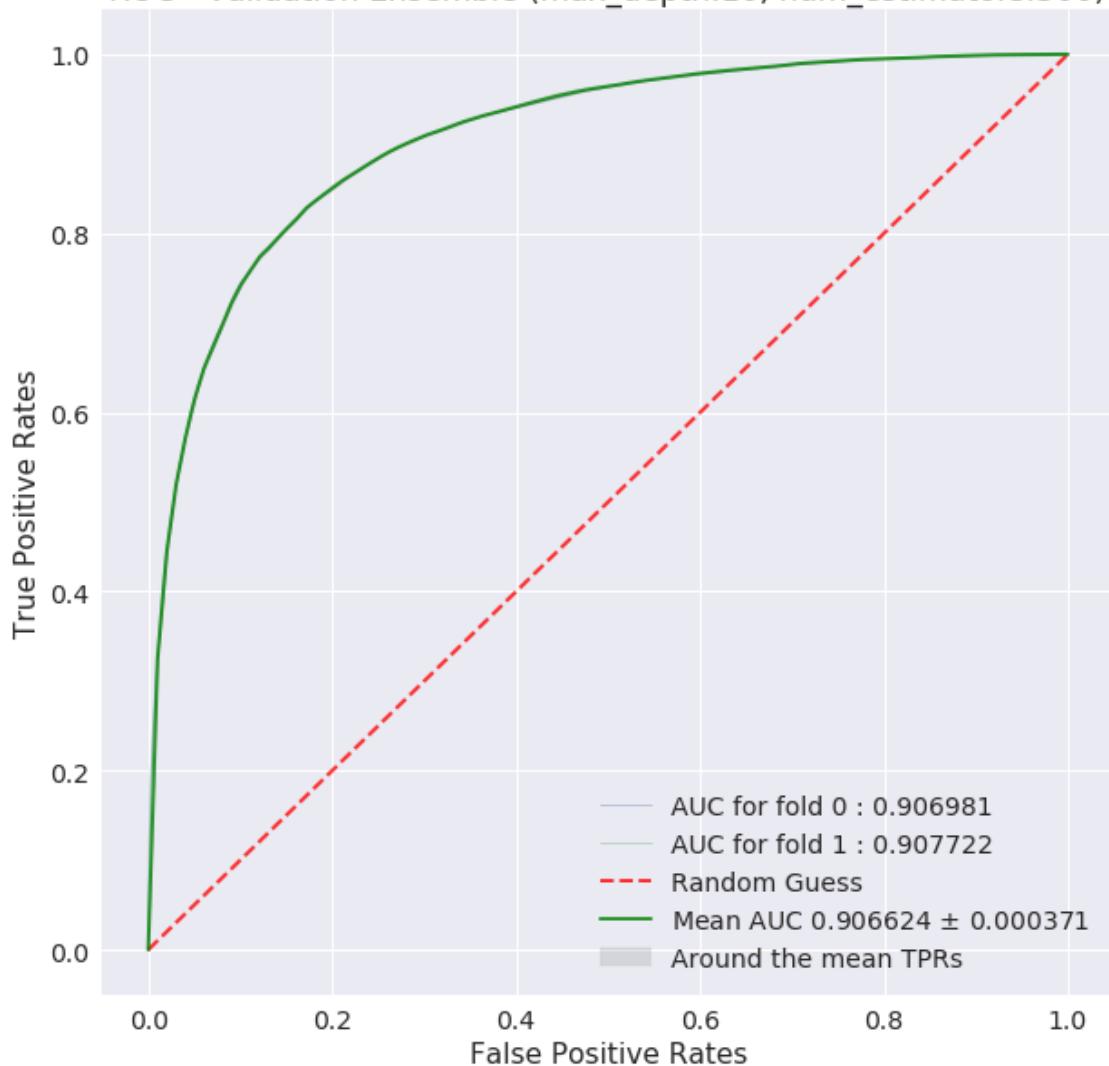
ROC - Validation Ensemble (max\_depth:10, num\_estimators:120)



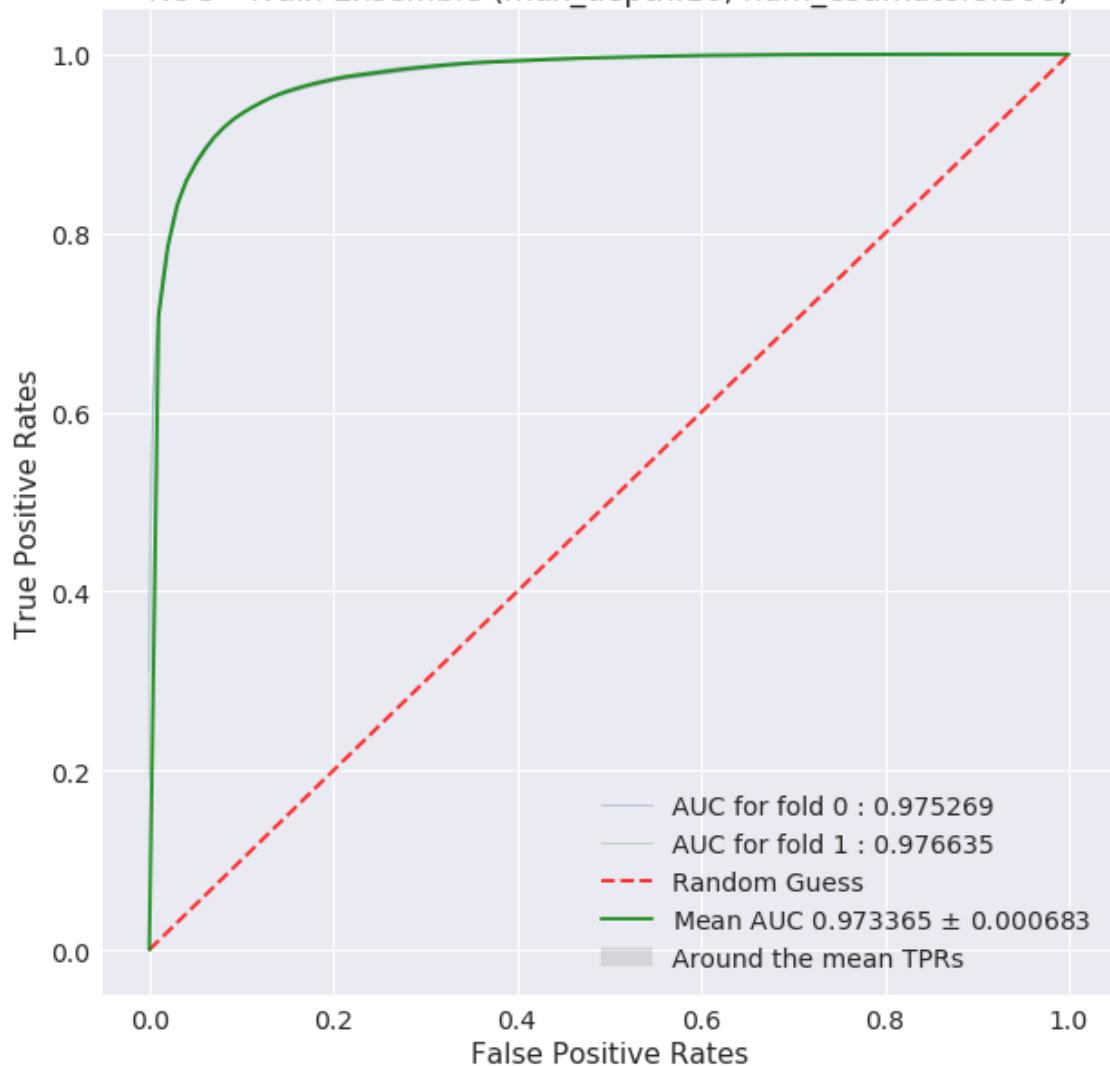
ROC - Train Ensemble (max\_depth:10, num\_estimators:300)



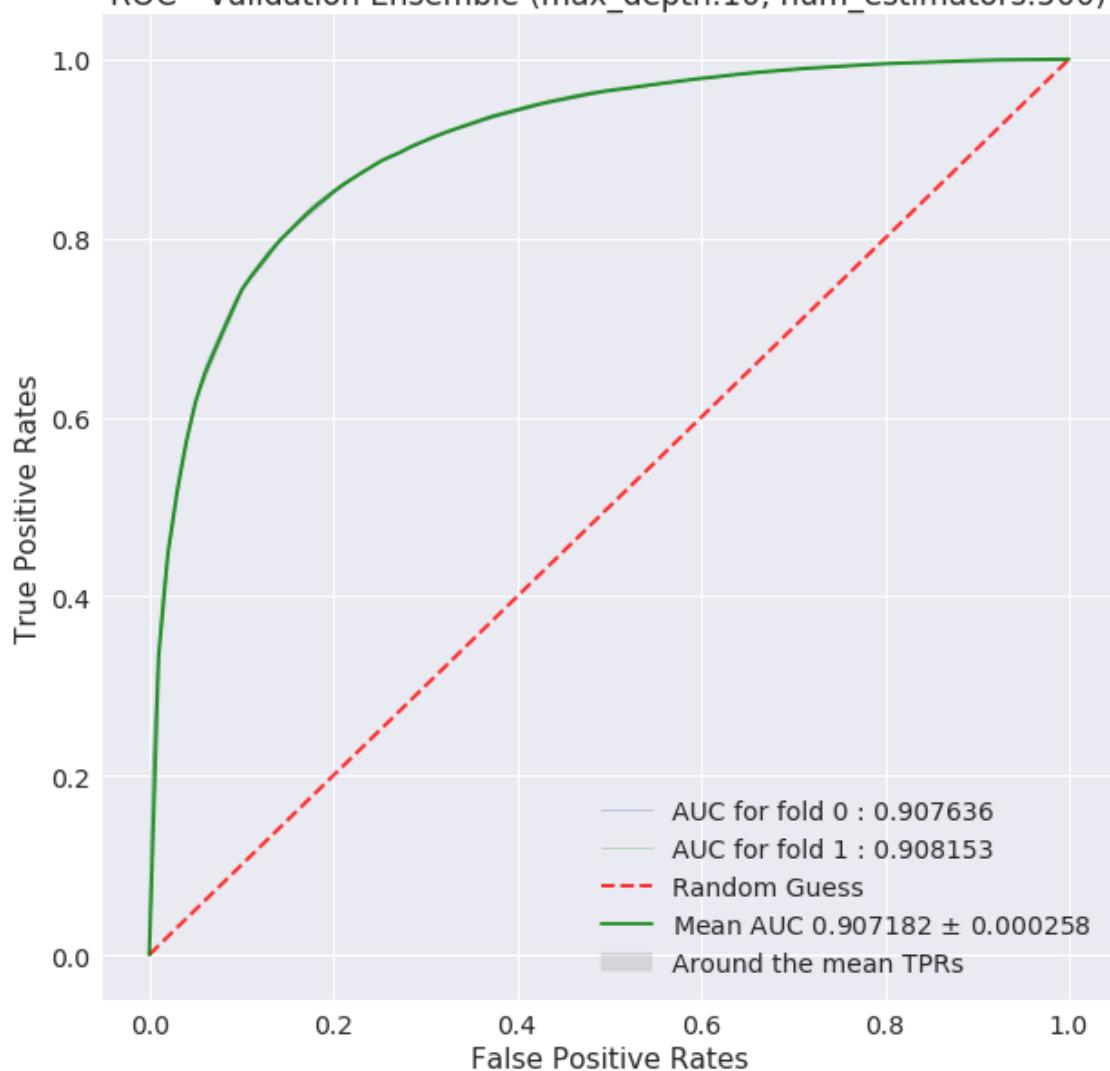
ROC - Validation Ensemble (max\_depth:10, num\_estimators:300)



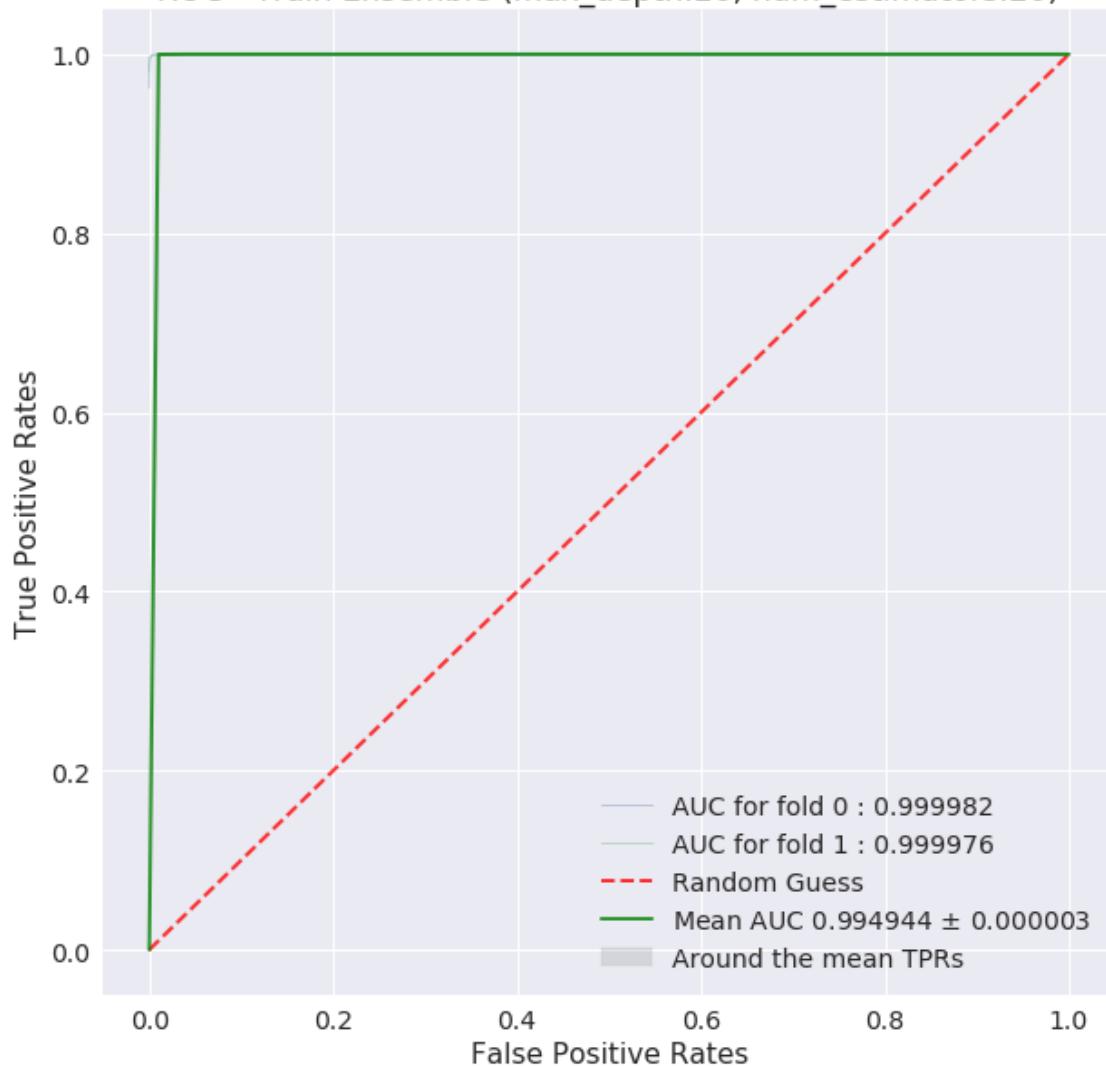
ROC - Train Ensemble (max\_depth:10, num\_estimators:500)



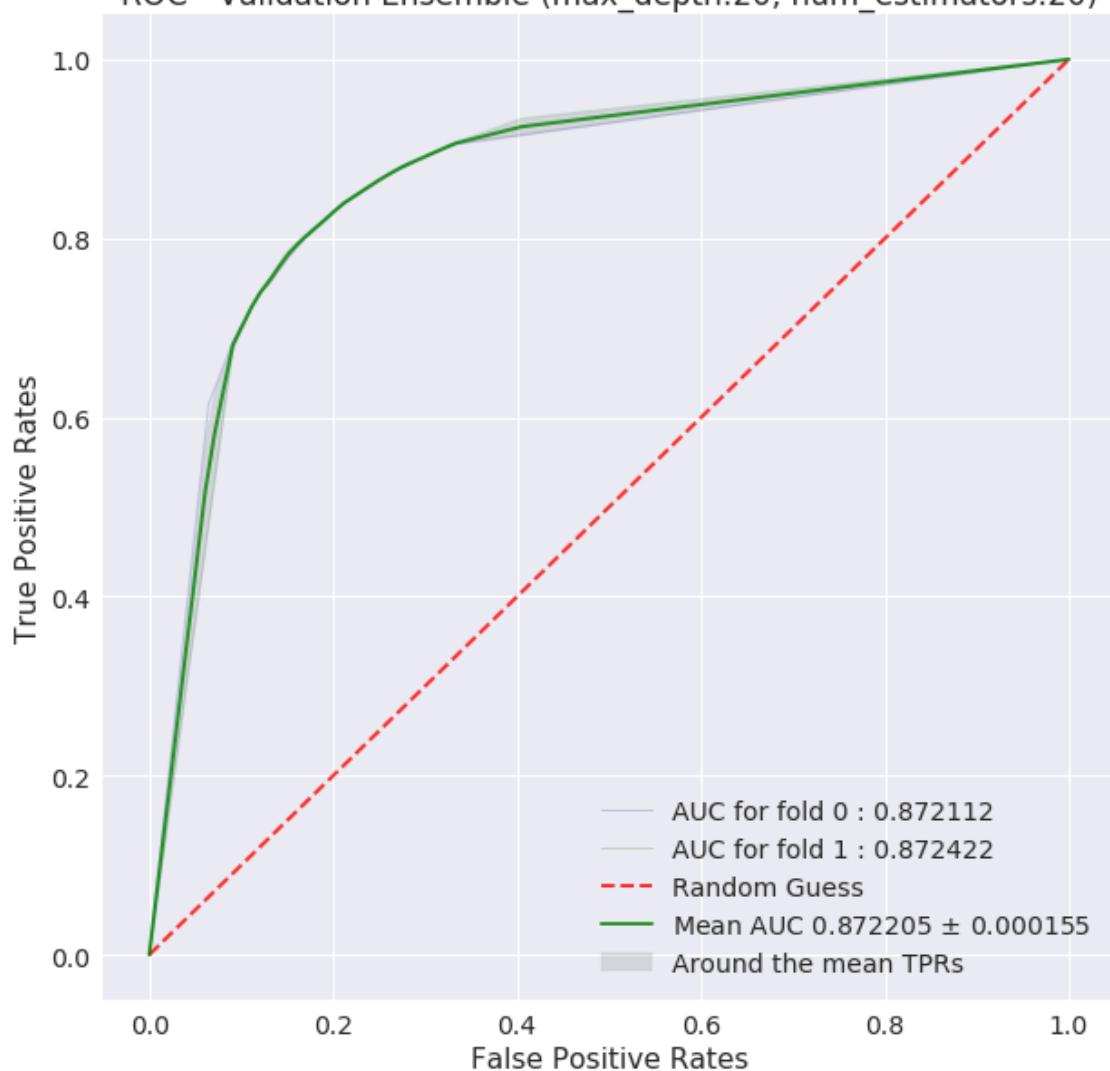
ROC - Validation Ensemble (max\_depth:10, num\_estimators:500)



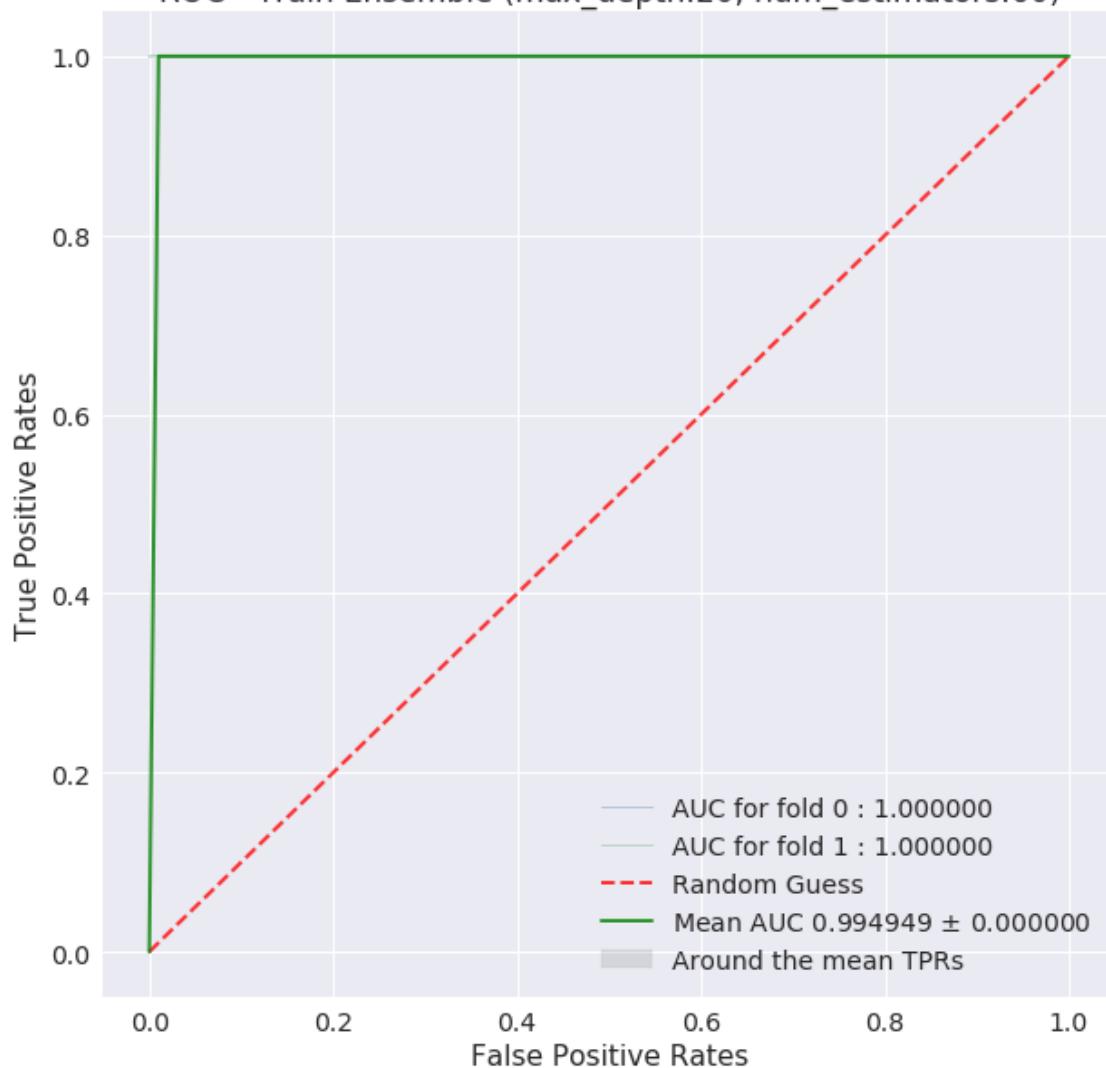
ROC - Train Ensemble (max\_depth:20, num\_estimators:20)



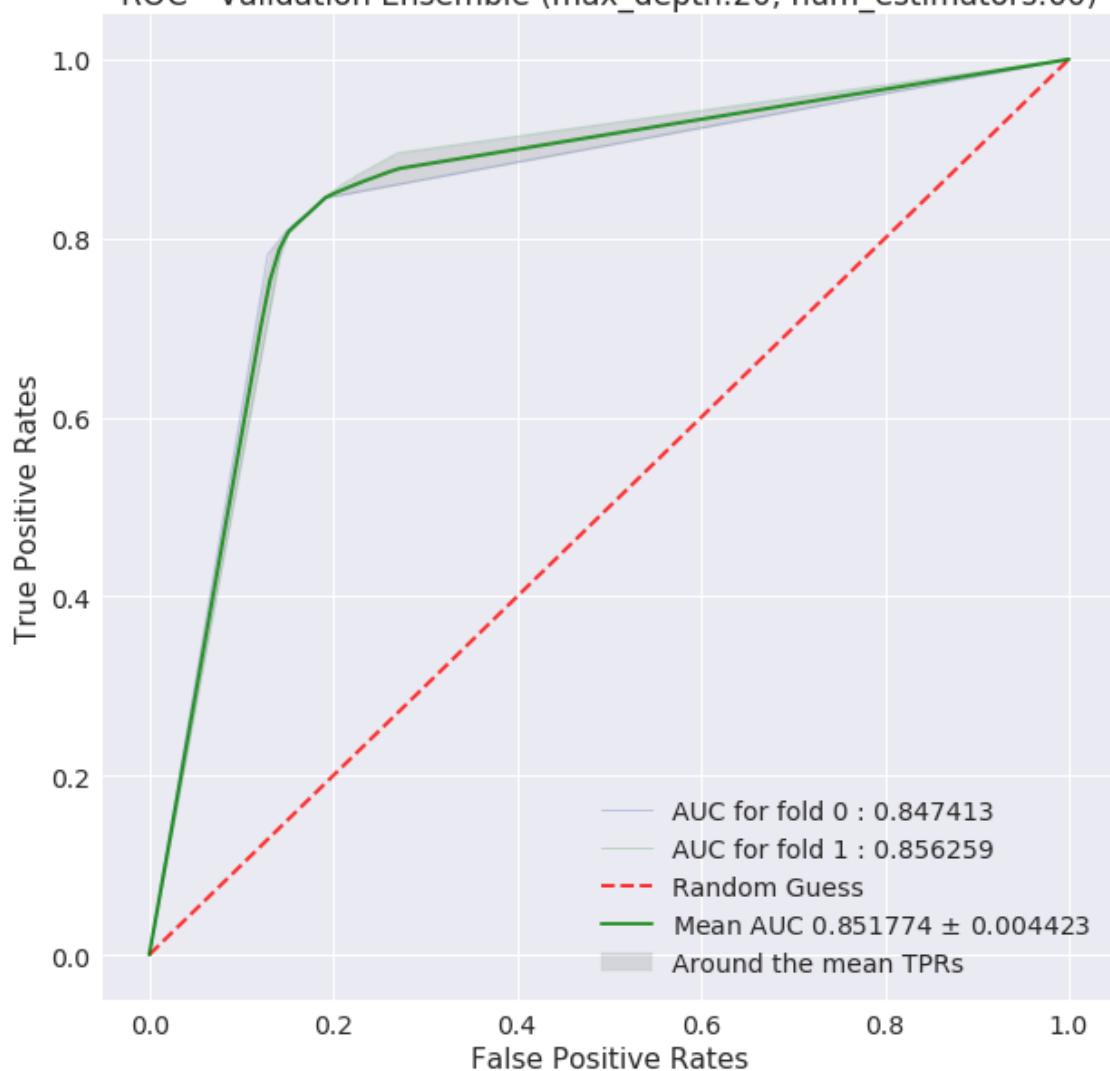
ROC - Validation Ensemble (max\_depth:20, num\_estimators:20)



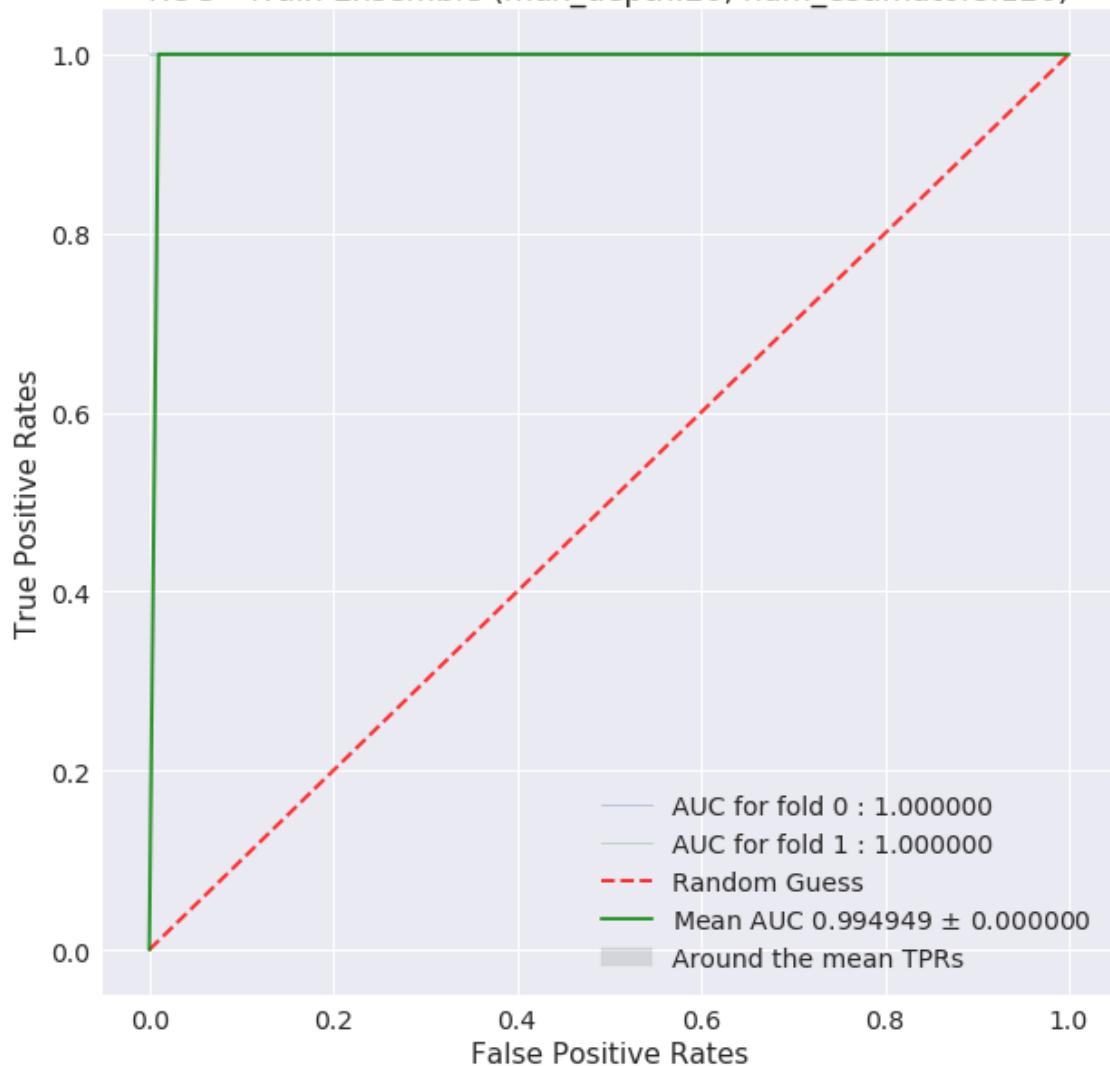
ROC - Train Ensemble (max\_depth:20, num\_estimators:60)



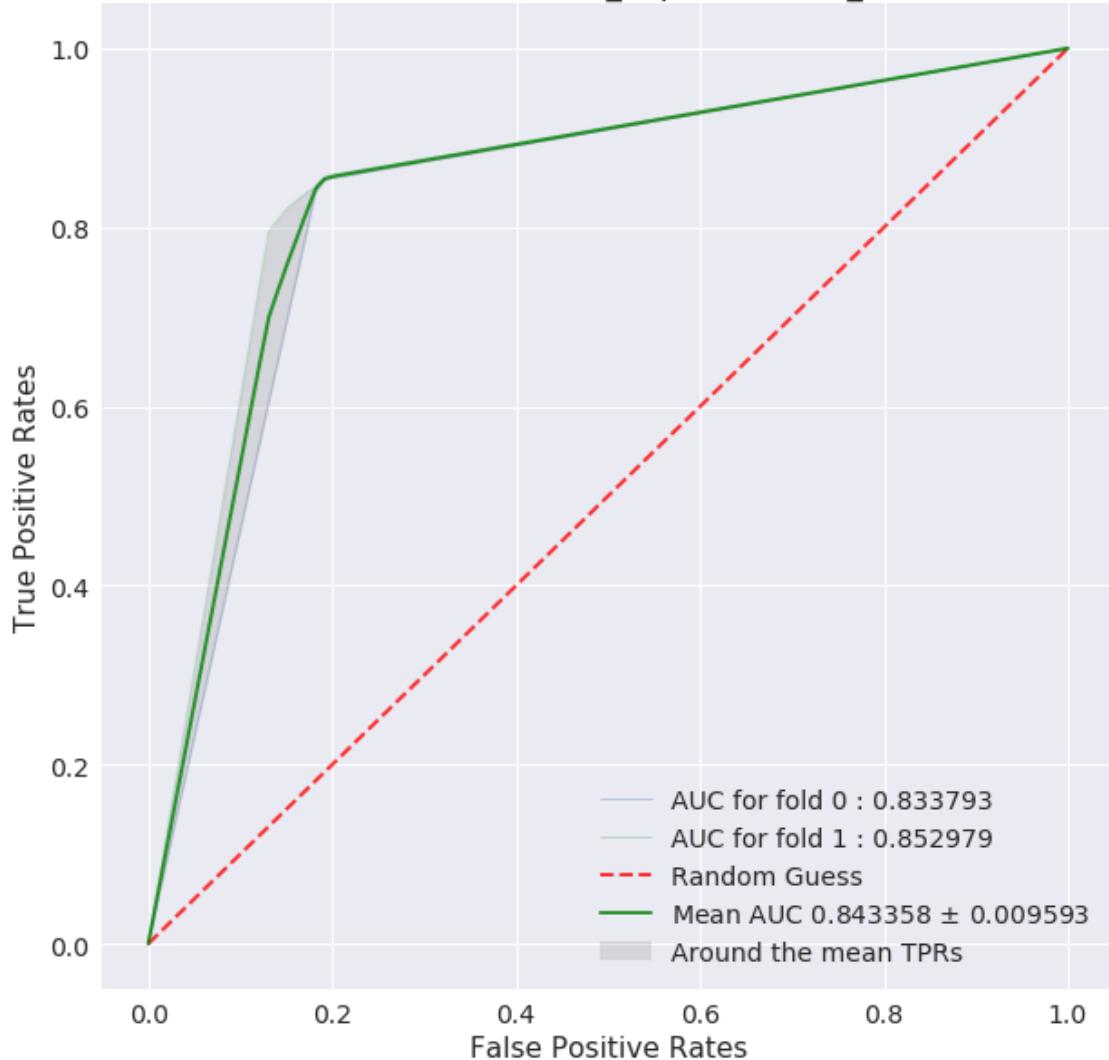
ROC - Validation Ensemble (max\_depth:20, num\_estimators:60)



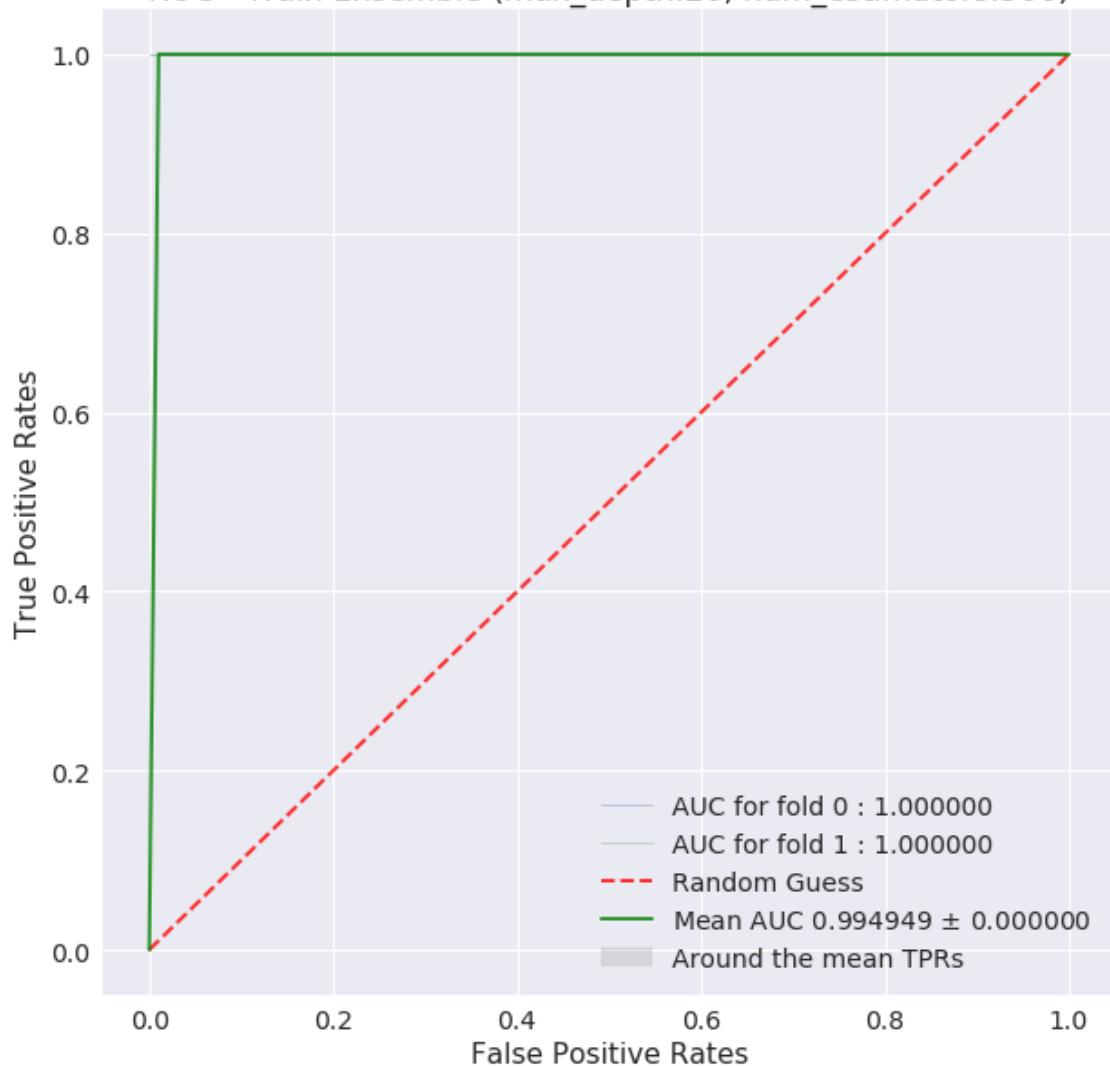
ROC - Train Ensemble (max\_depth:20, num\_estimators:120)



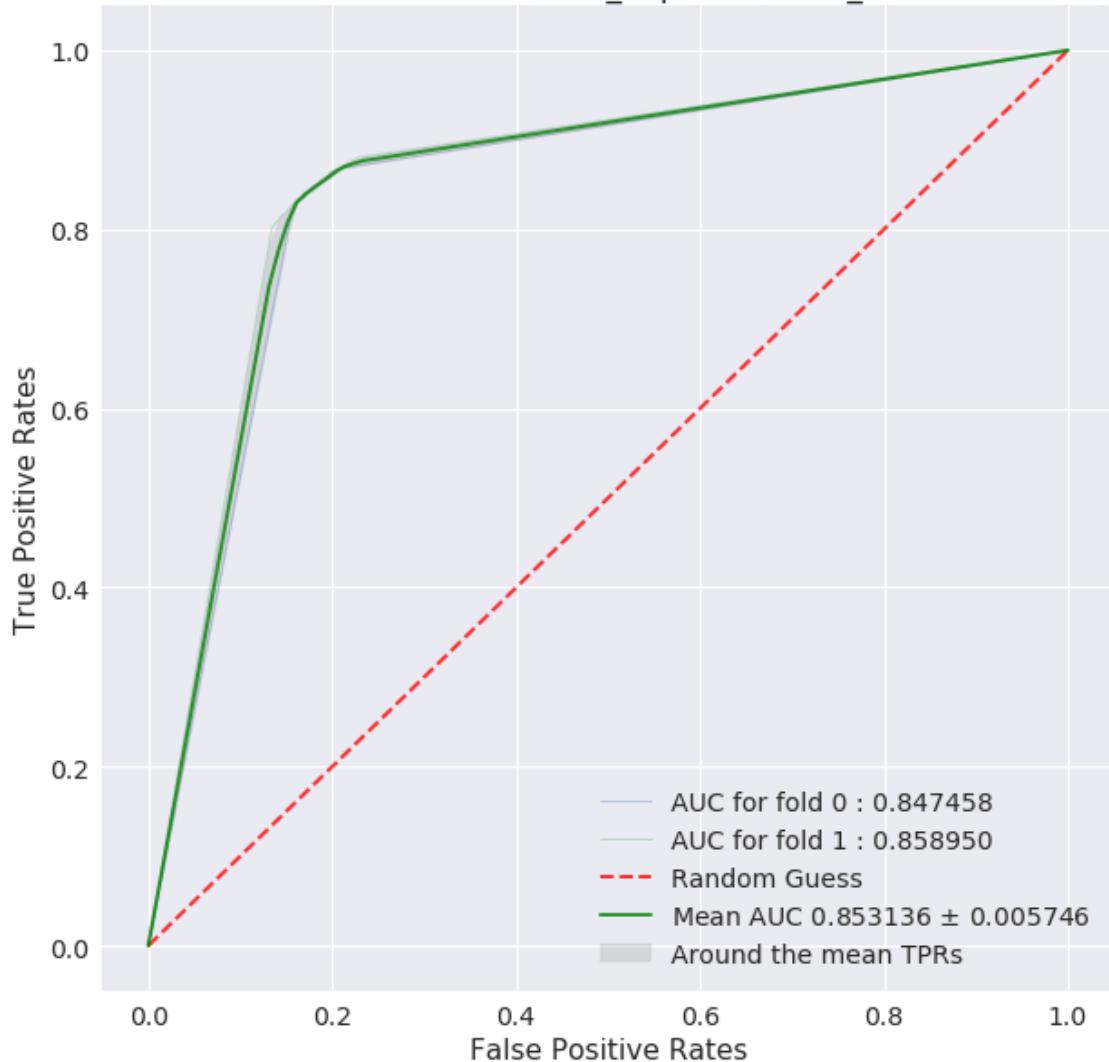
ROC - Validation Ensemble (max\_depth:20, num\_estimators:120)



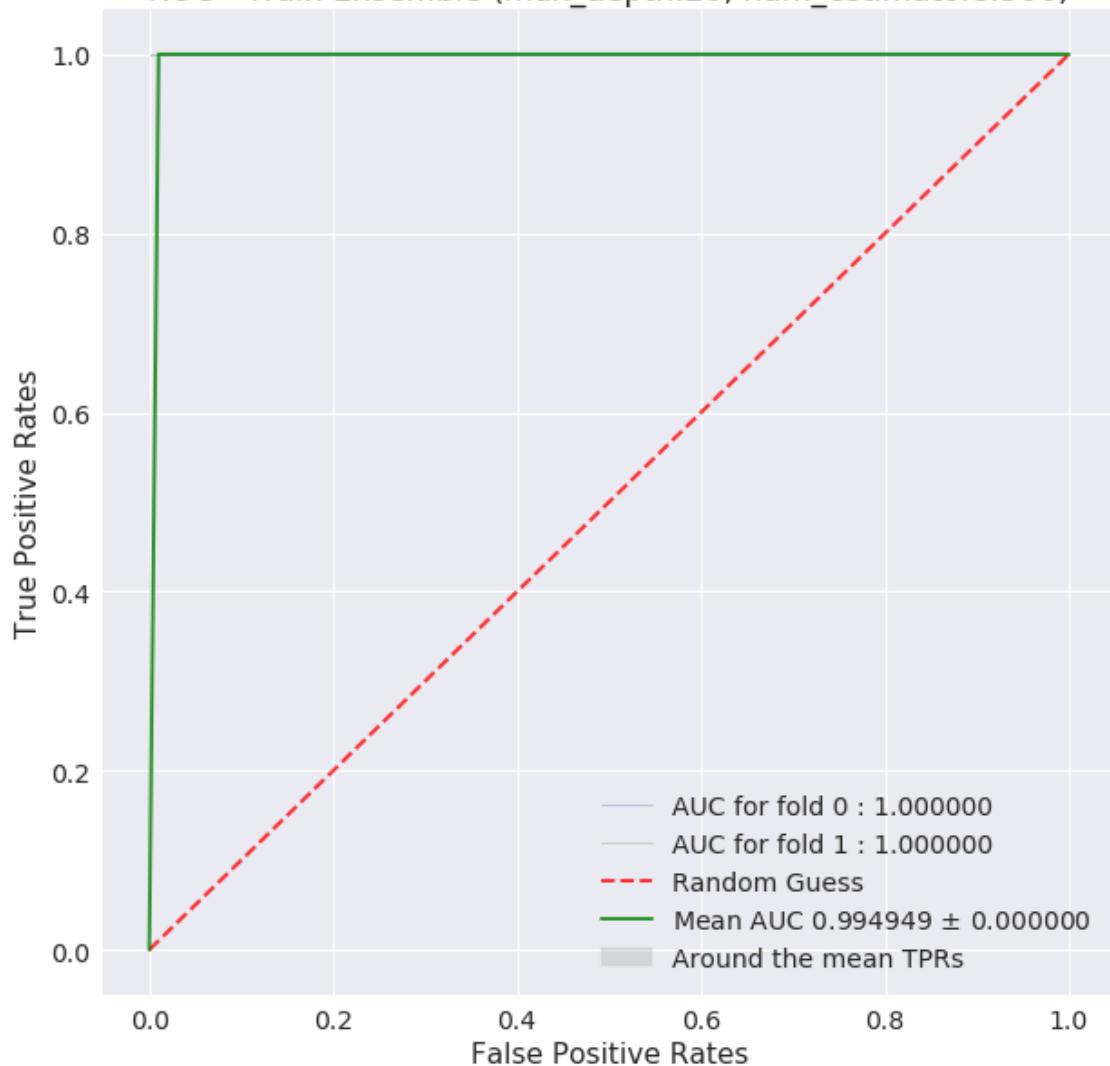
ROC - Train Ensemble (max\_depth:20, num\_estimators:300)



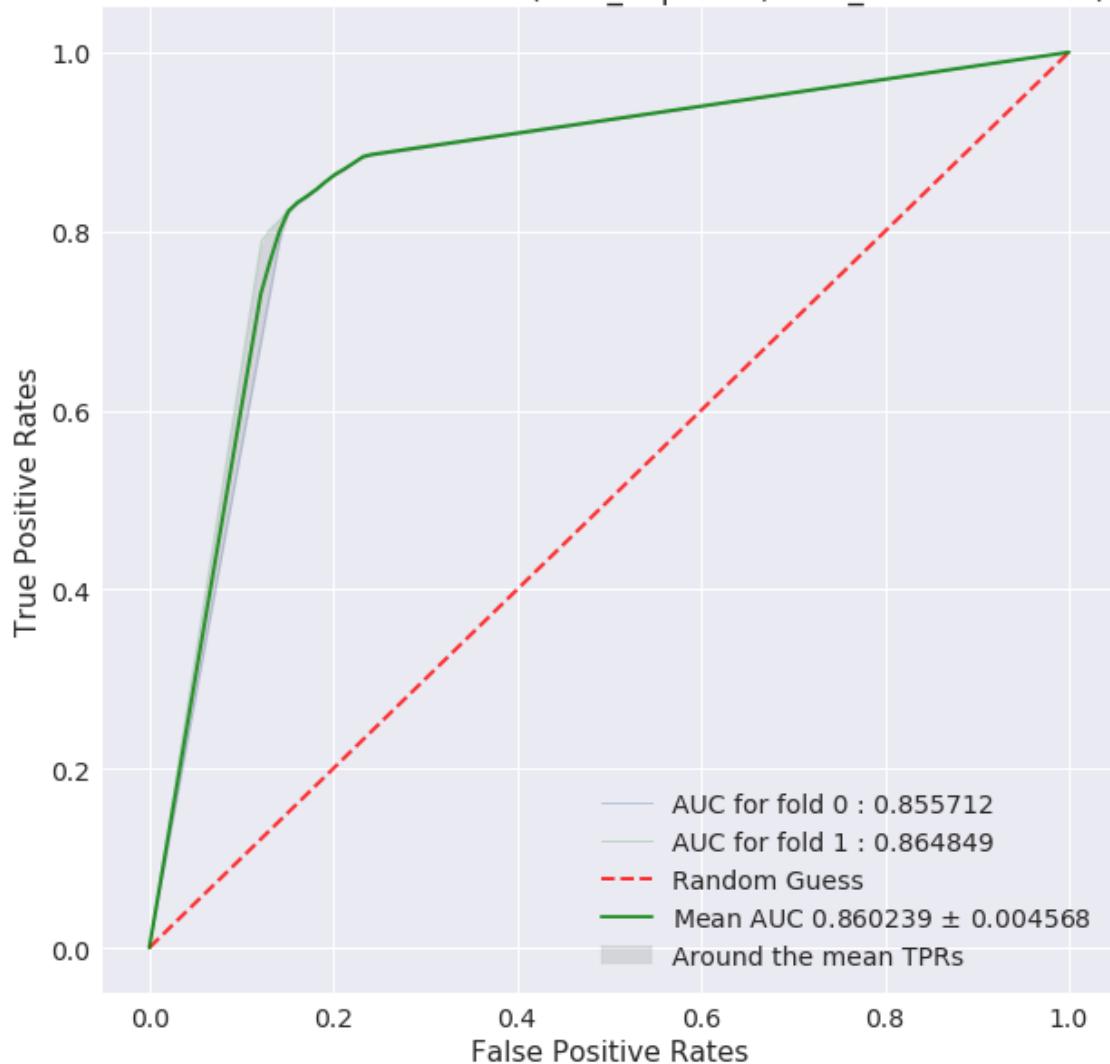
ROC - Validation Ensemble (max\_depth:20, num\_estimators:300)



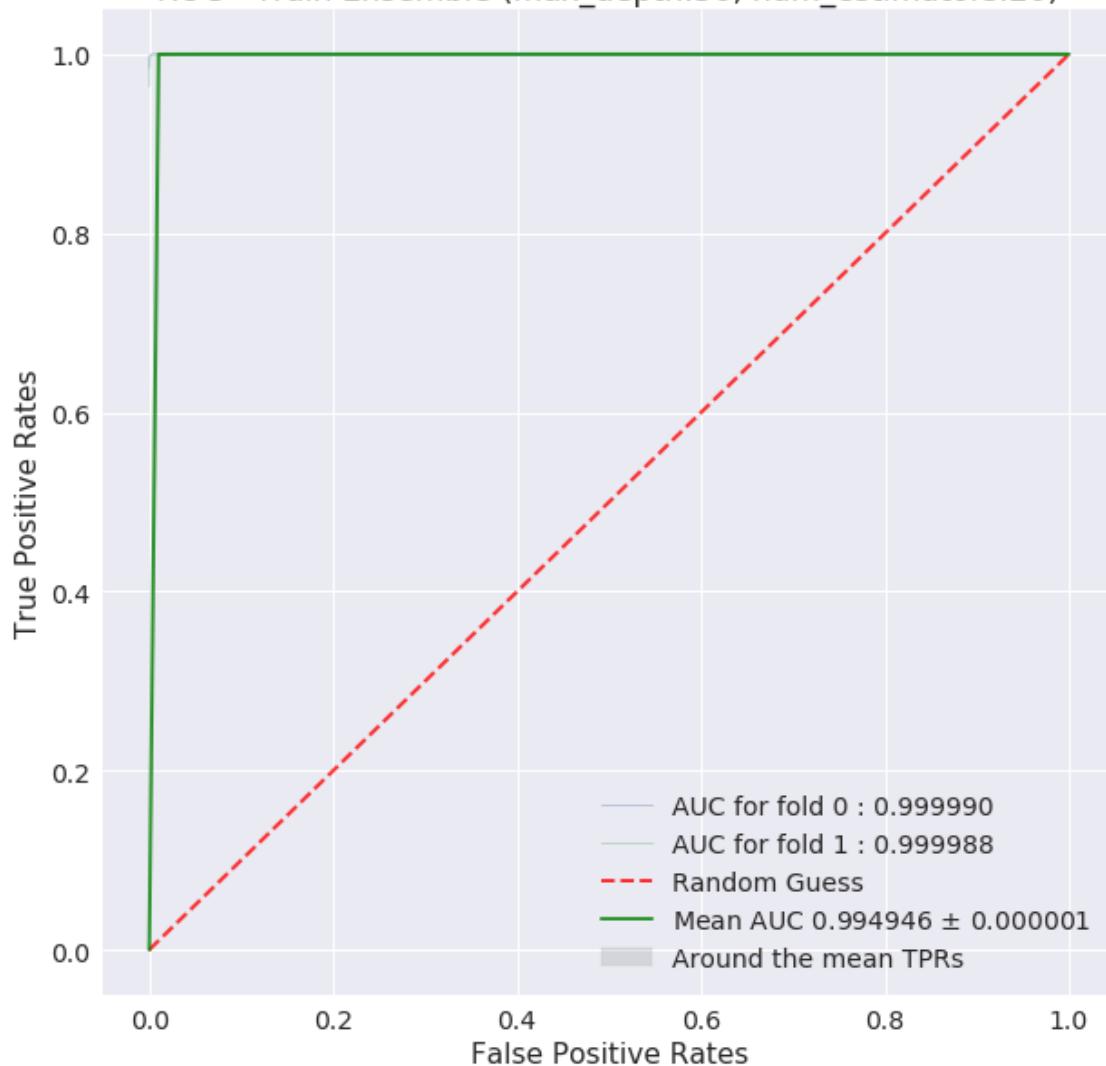
ROC - Train Ensemble (max\_depth:20, num\_estimators:500)



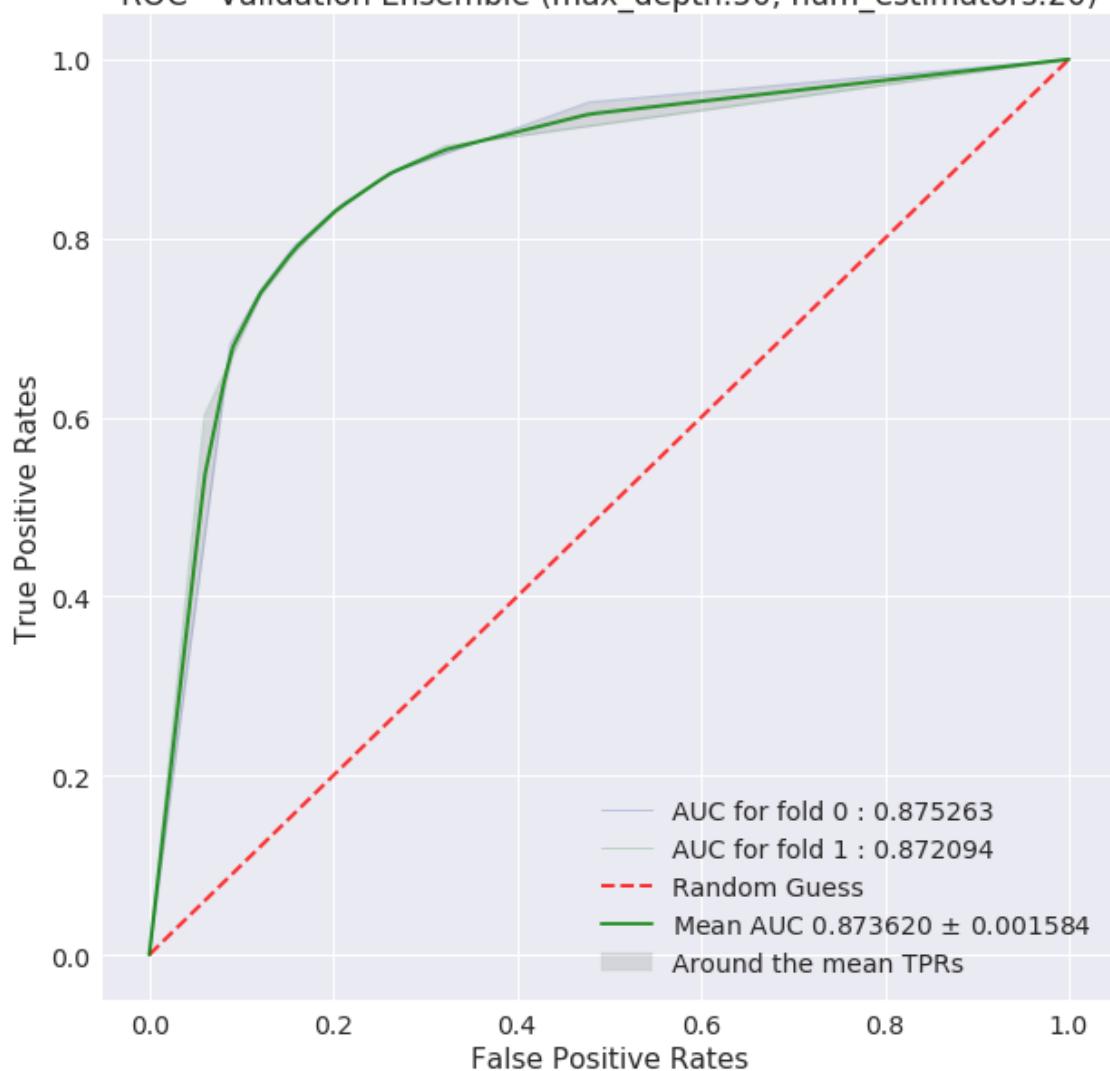
ROC - Validation Ensemble (max\_depth:20, num\_estimators:500)



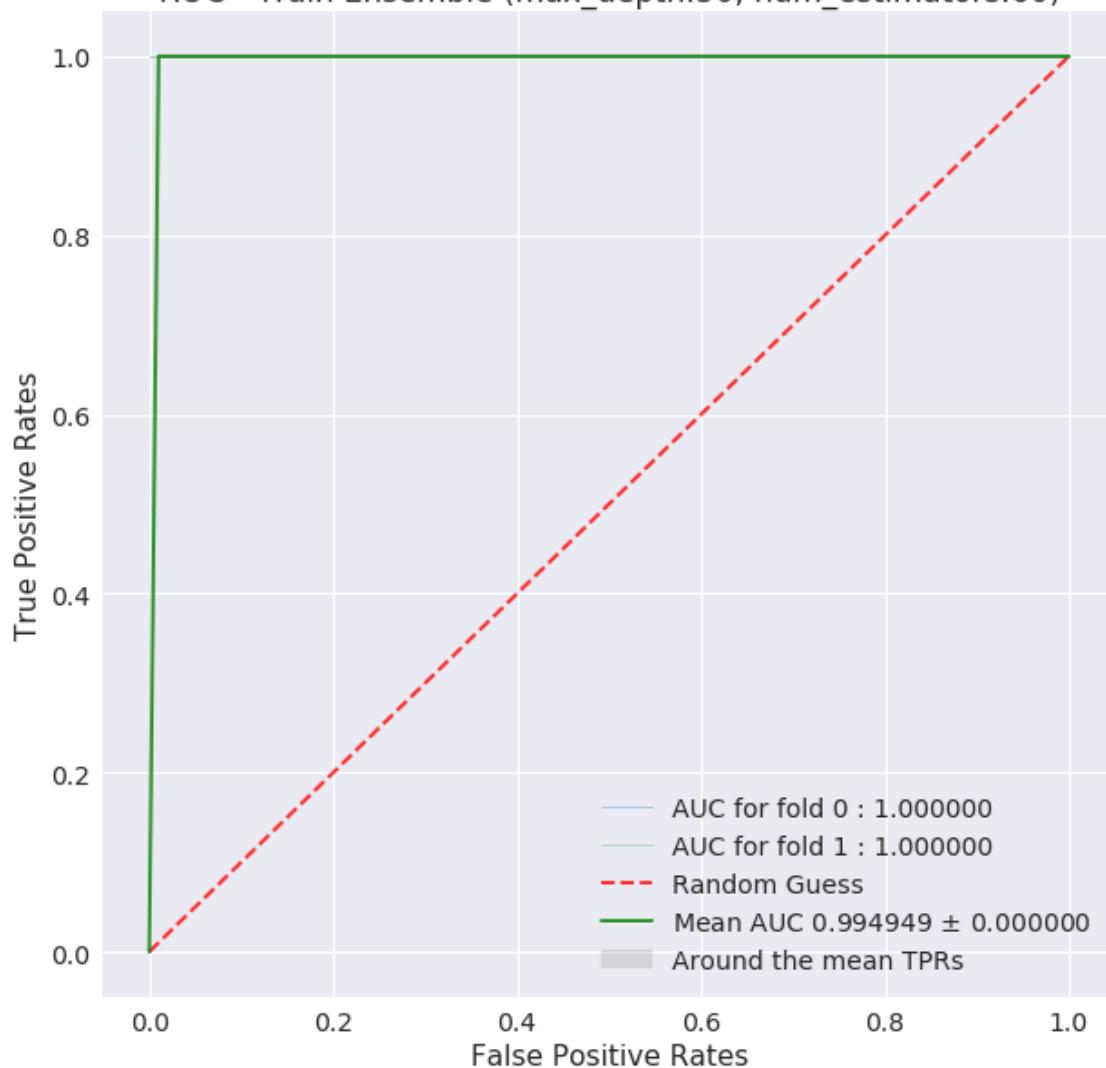
ROC - Train Ensemble (max\_depth:50, num\_estimators:20)



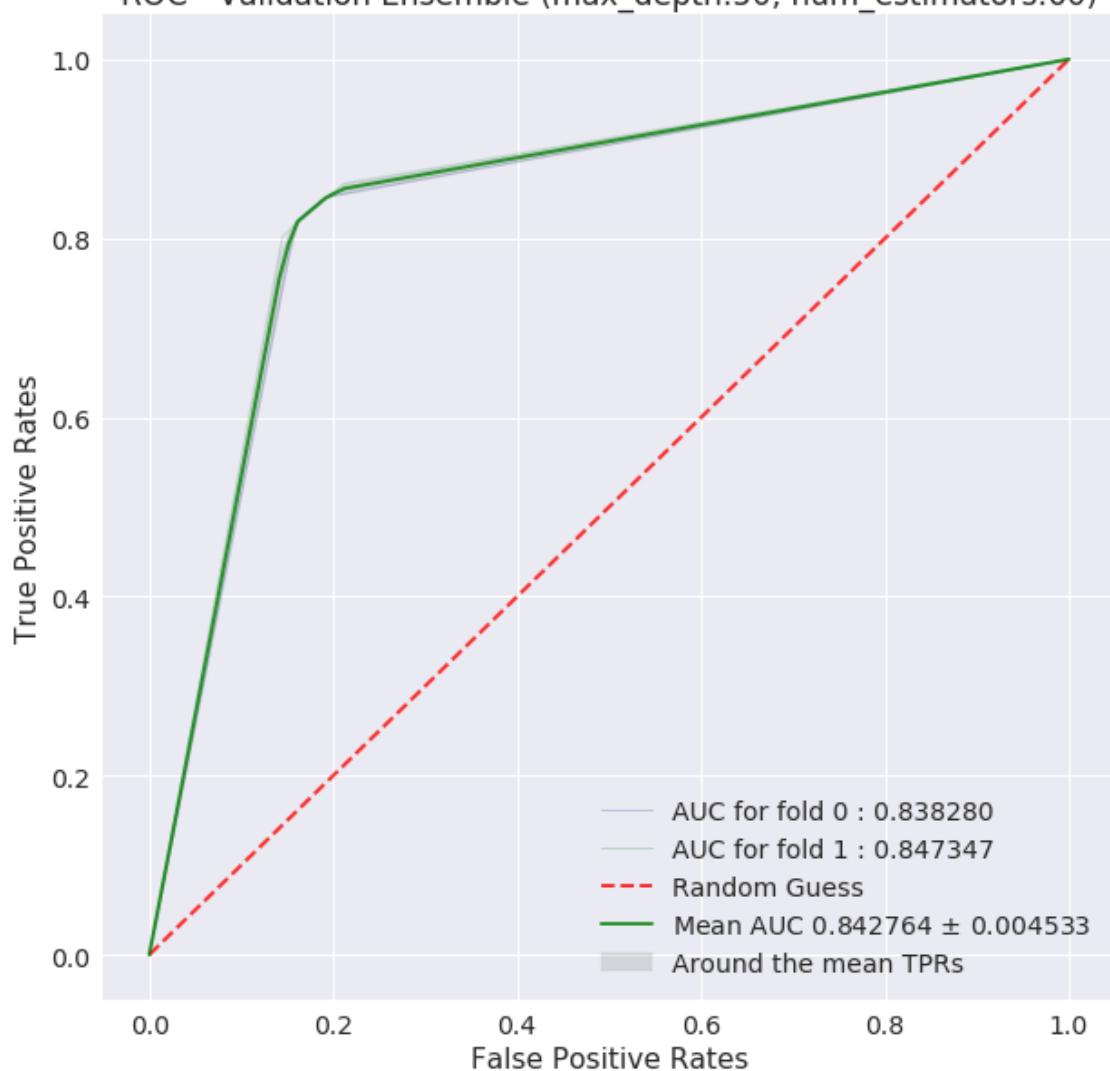
ROC - Validation Ensemble (max\_depth:50, num\_estimators:20)



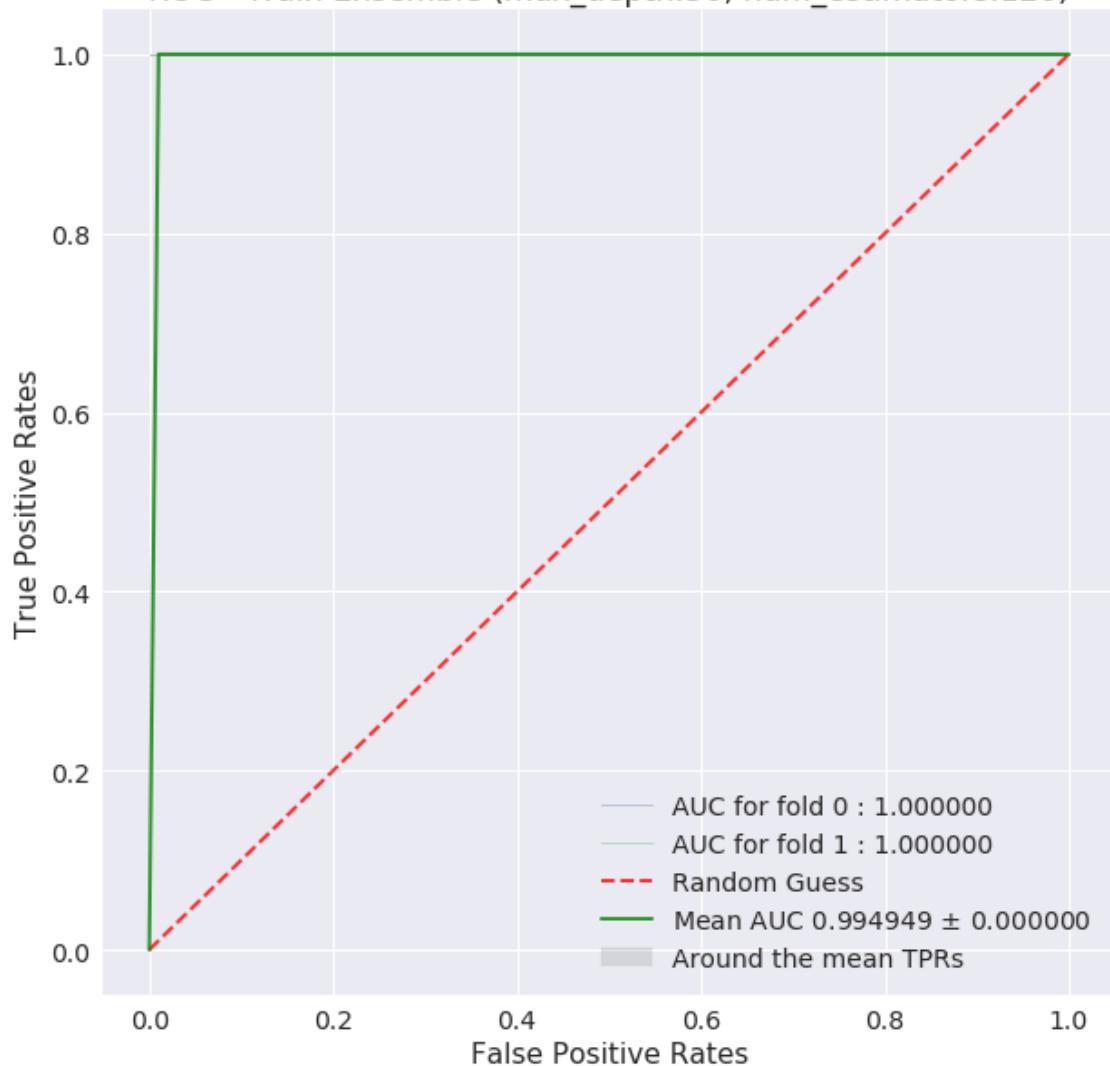
ROC - Train Ensemble (max\_depth:50, num\_estimators:60)



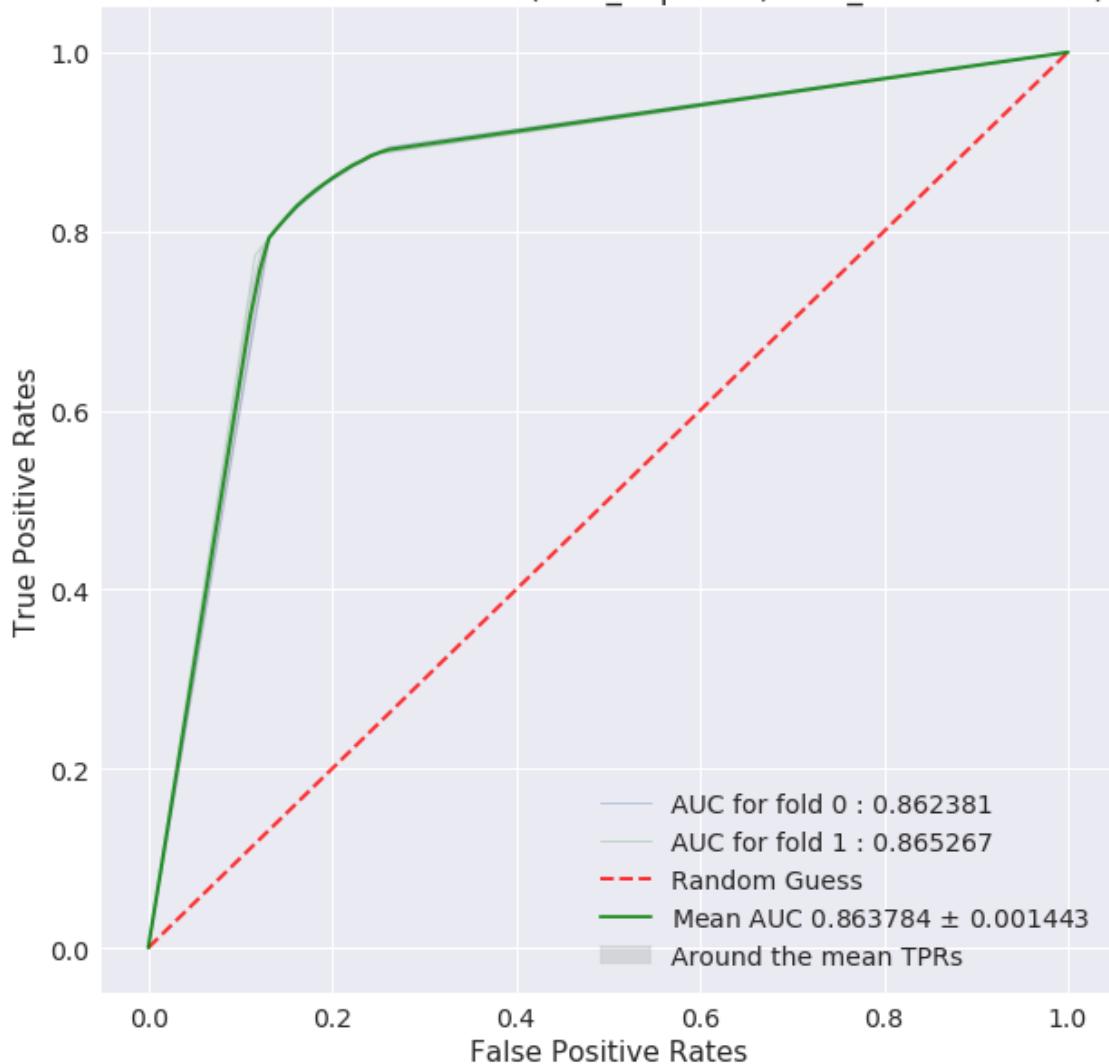
ROC - Validation Ensemble (max\_depth:50, num\_estimators:60)



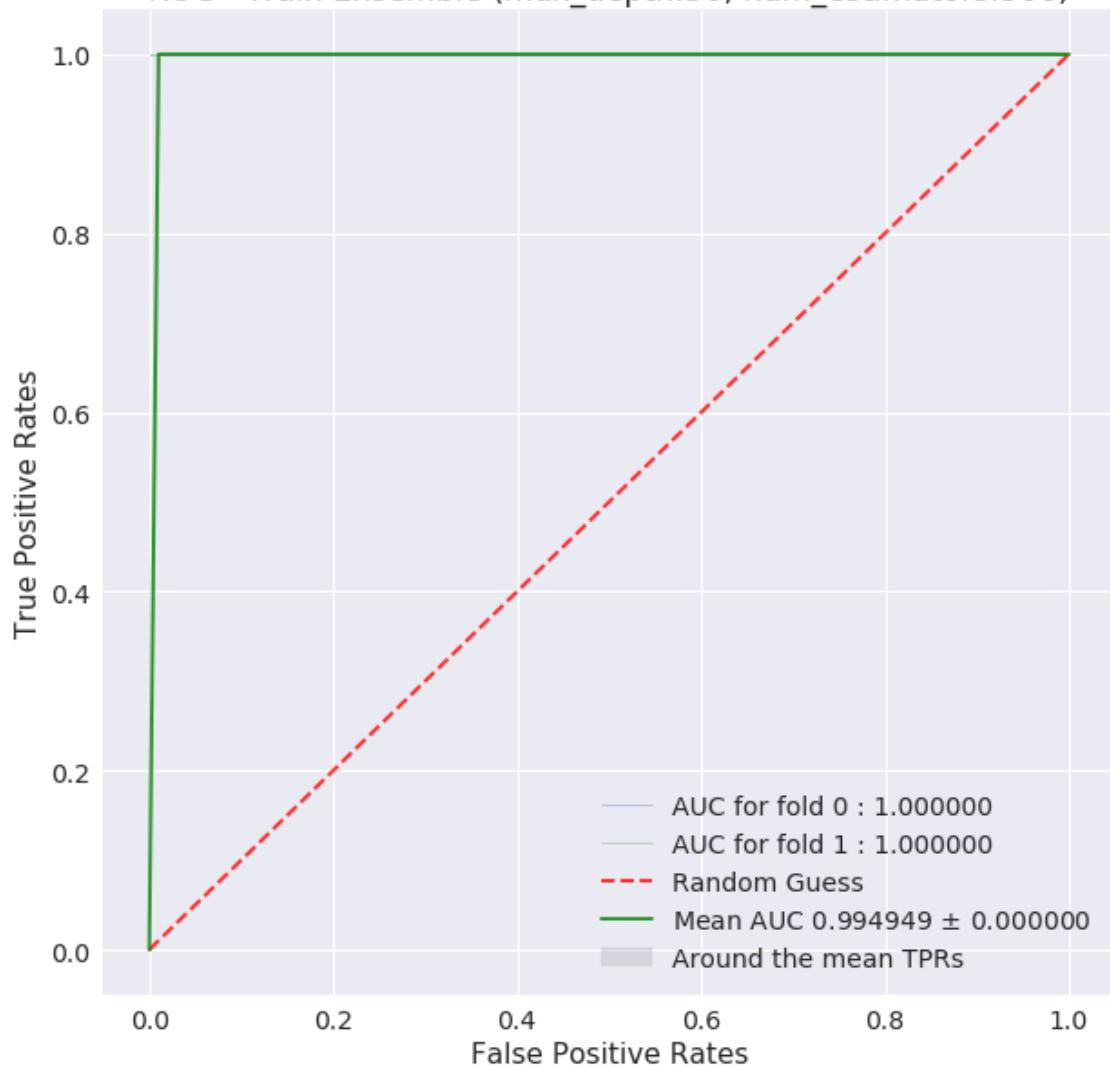
ROC - Train Ensemble (max\_depth:50, num\_estimators:120)



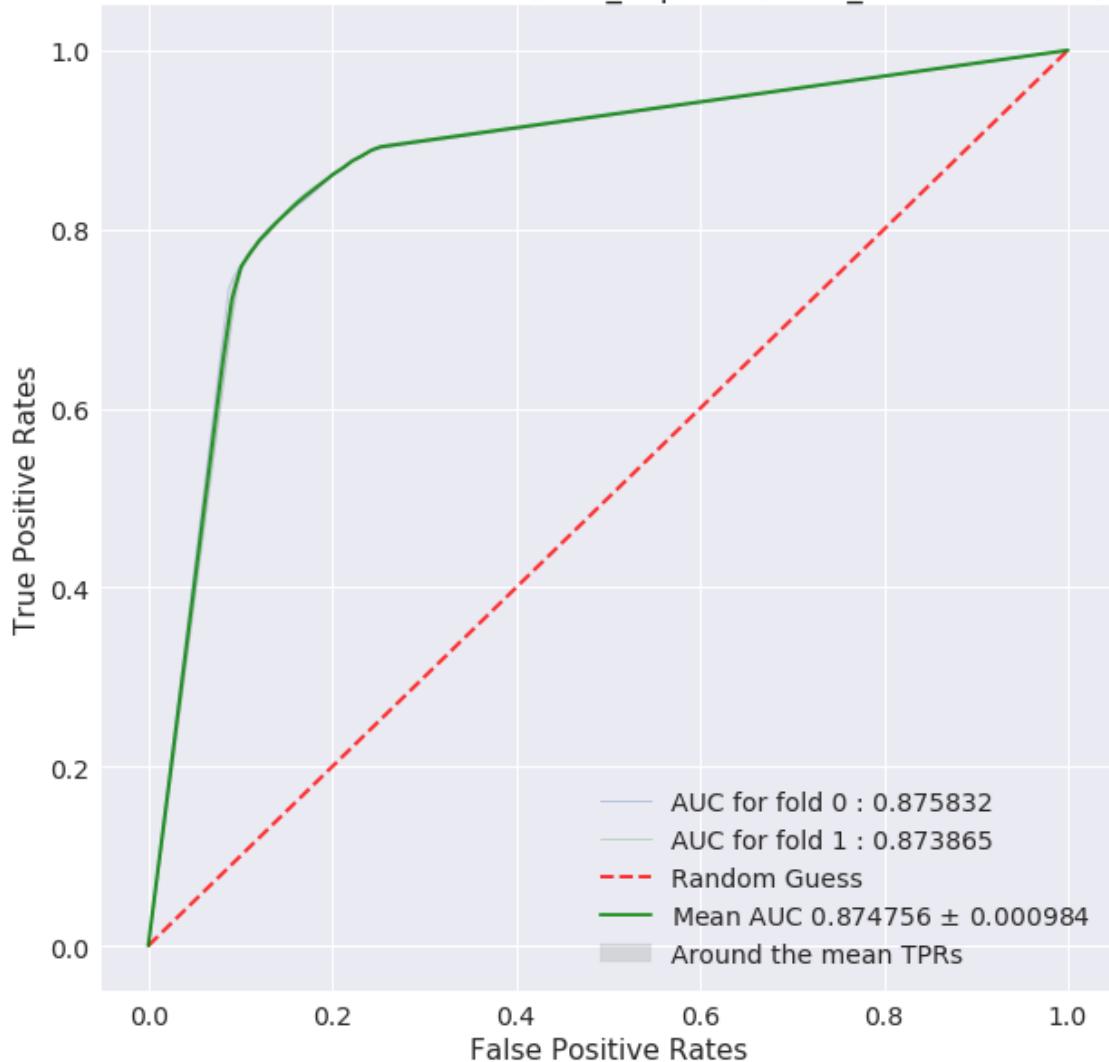
ROC - Validation Ensemble (max\_depth:50, num\_estimators:120)



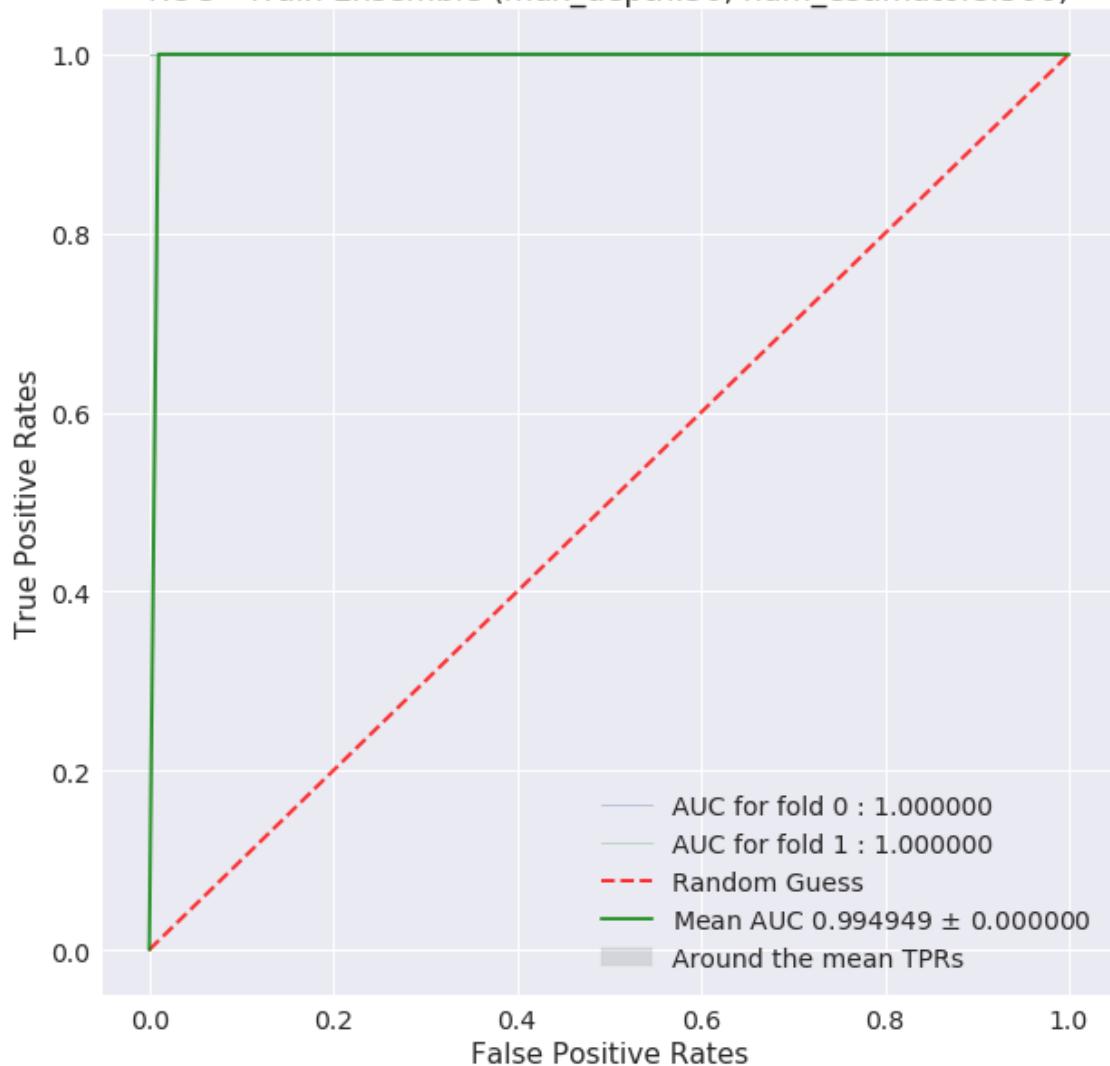
ROC - Train Ensemble (max\_depth:50, num\_estimators:300)



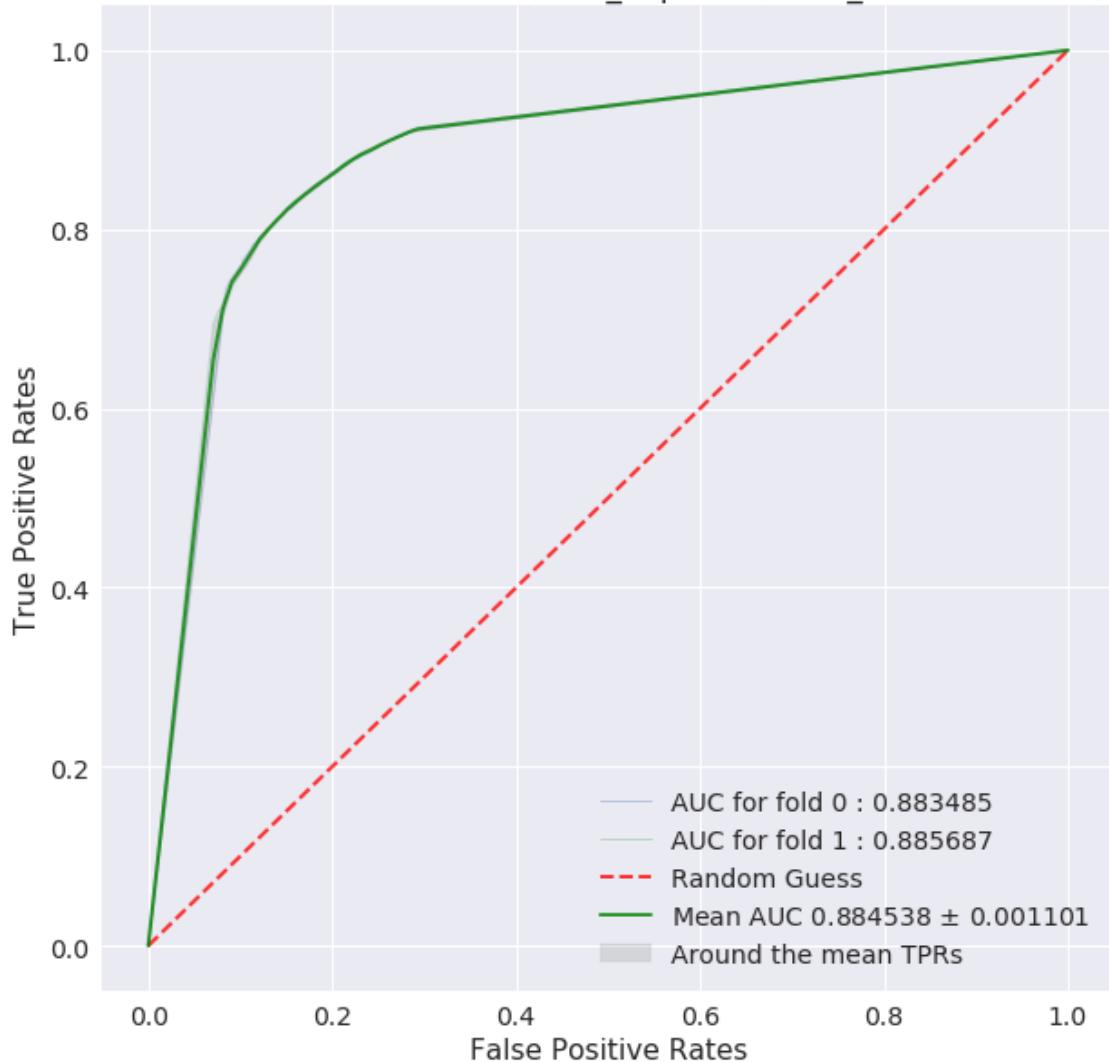
ROC - Validation Ensemble (max\_depth:50, num\_estimators:300)

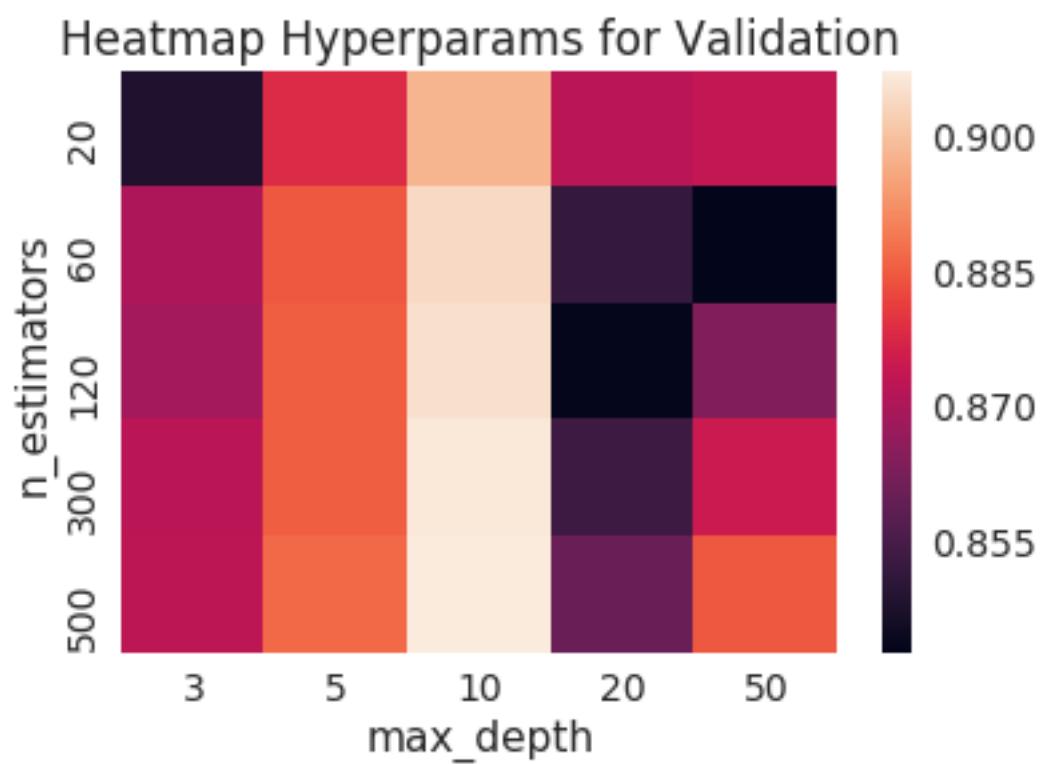
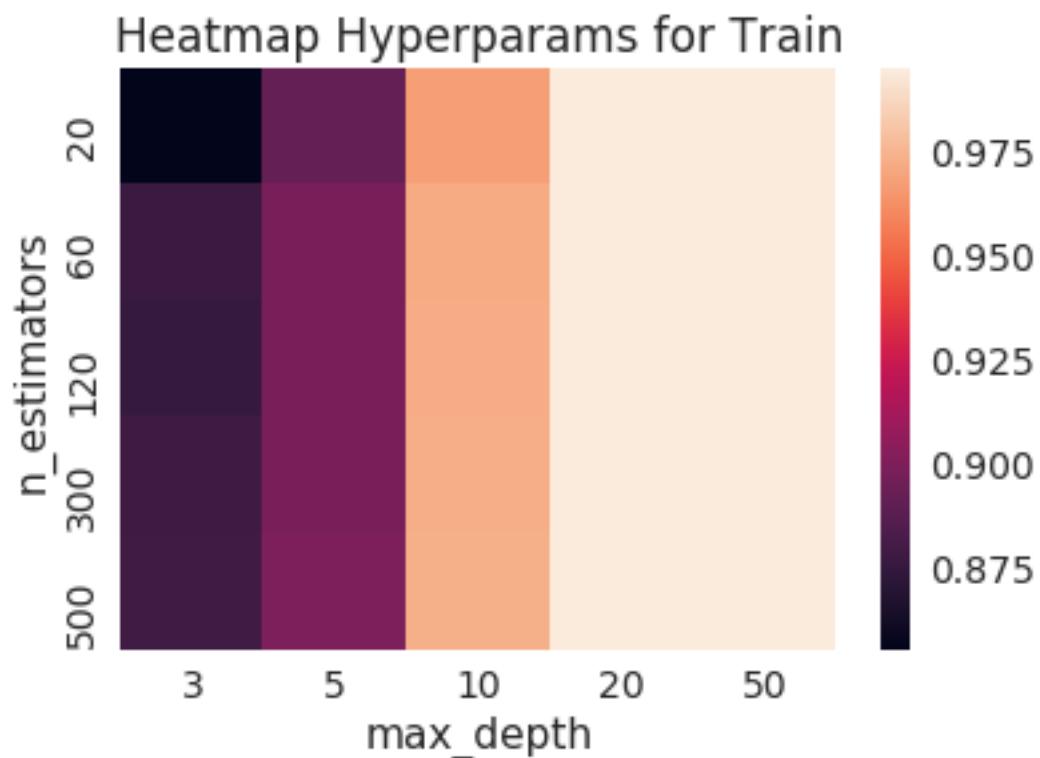


ROC - Train Ensemble (max\_depth:50, num\_estimators:500)

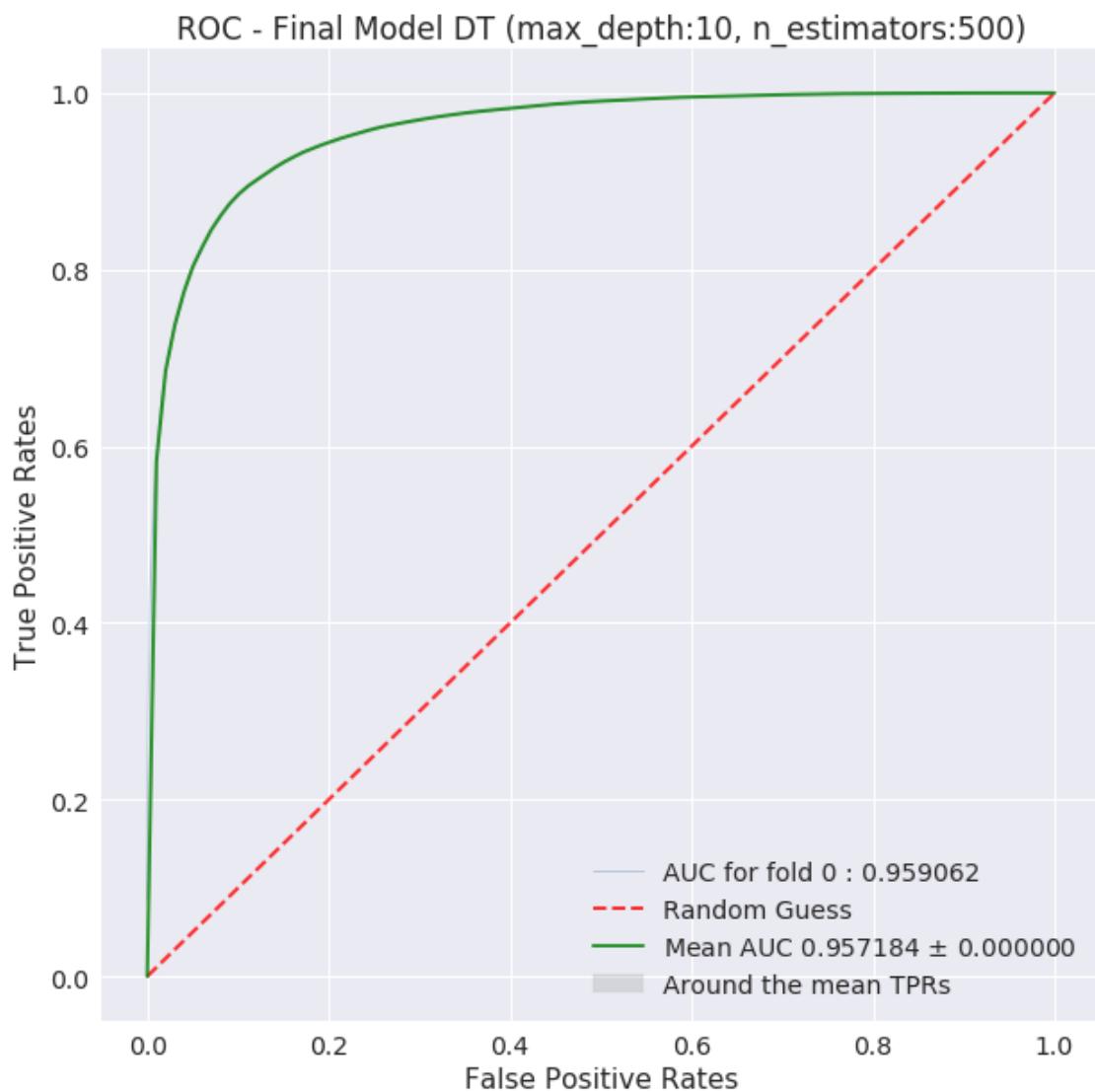


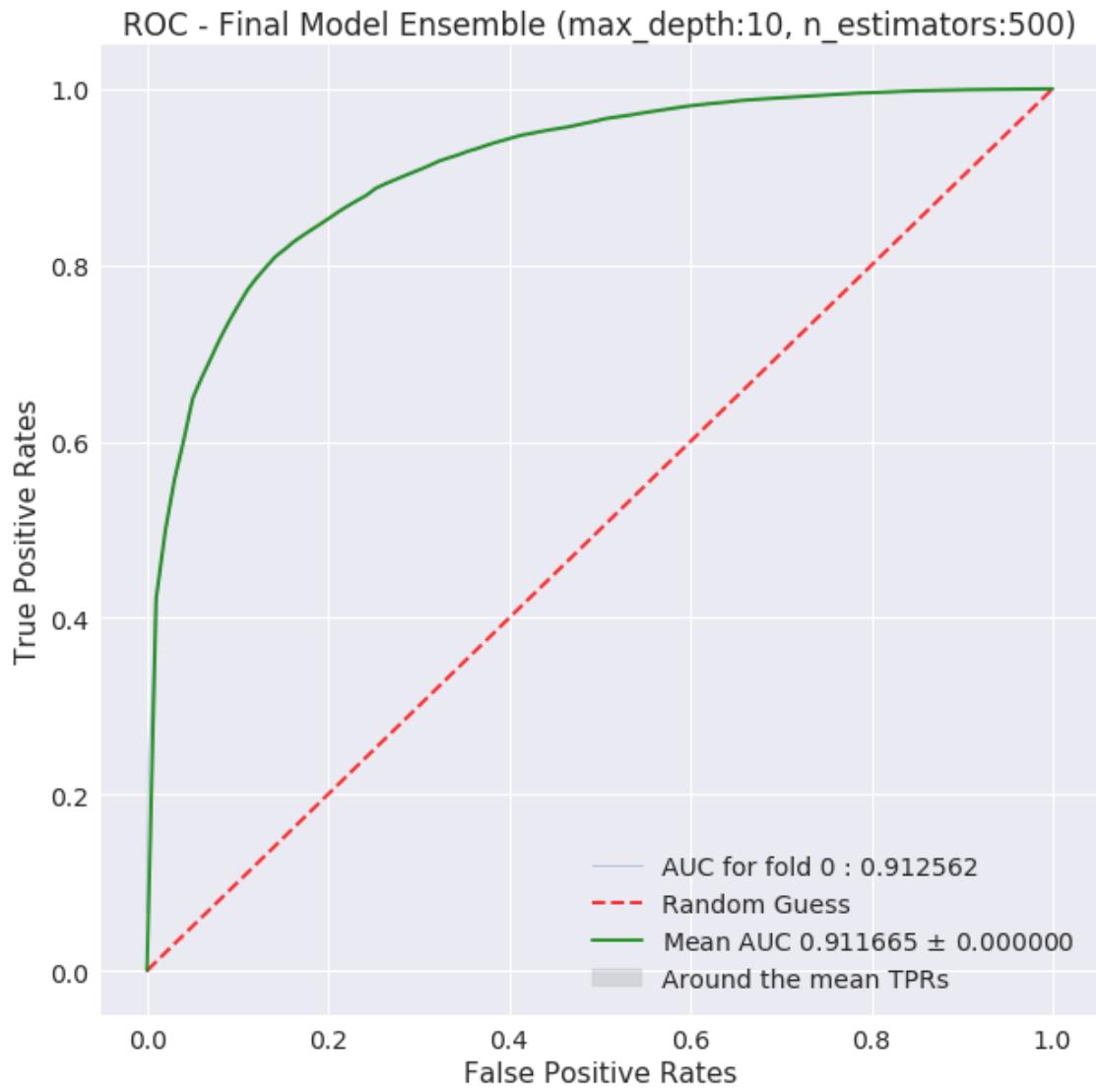
ROC - Validation Ensemble (max\_depth:50, num\_estimators:500)





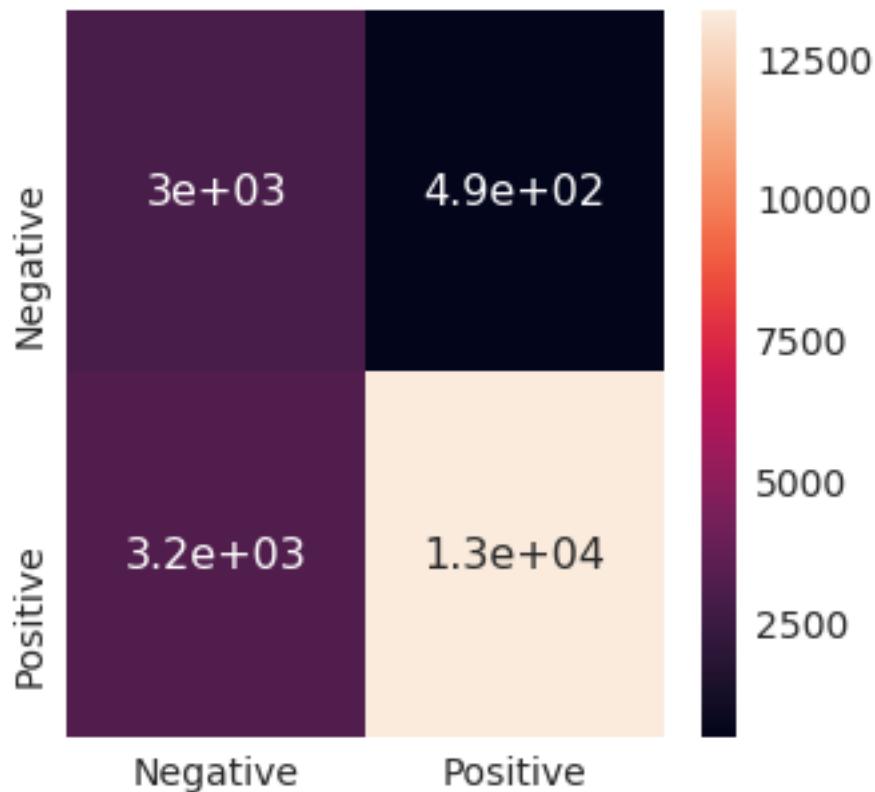
Best hyperparam value: (10, 500)





Test auc score 0.9116651157195222

### Ensemble Model Confusion Matrix



	Negative	Positive
Precision	0.485001	0.964650
Recall	0.859483	0.807748
Fscore	0.620089	0.879254
Support	3480.000000	16520.000000

best hyper param identified is max\_depth = 10, and n\_estimators=500

#### 4.4.2 [A.6] Applying Random Forests on TFIDF W2V, SET 4

```
In [19]: # form two lists
depth_list = [3, 5, 10, 20, 50] # depends on size of dataset
n_estimators_list = [20, 60, 120, 300, 500] # depends on size of dataset

# create a configuration dictionary
config_dict = {
    'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF_W2V',
    'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF_W2V',
    'train_size' : 50000,
```

```
'test_size' : 20000,
'hyperparam_list' : list(product(depth_list, n_estimators_list)),
'implementation': 'rf' # 'xgb' or 'rf'
}
```

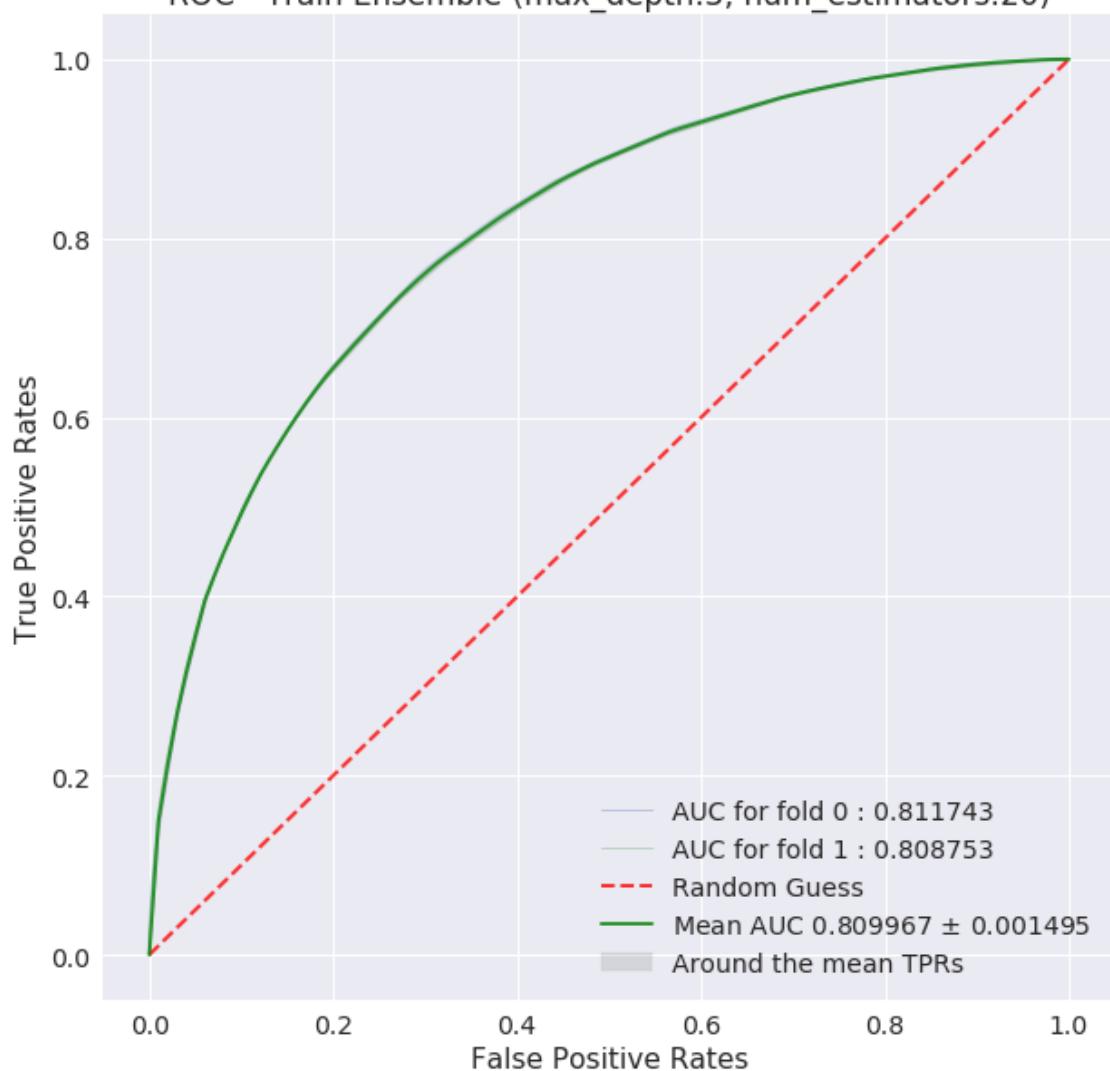
```
In [20]: # read the train, test data and preprocess it
train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                                               scaling=True,
                                                               dim_reduction=True)

# train and validate the model
model = train_and_validate_model(config_dict, train_features, train_labels)

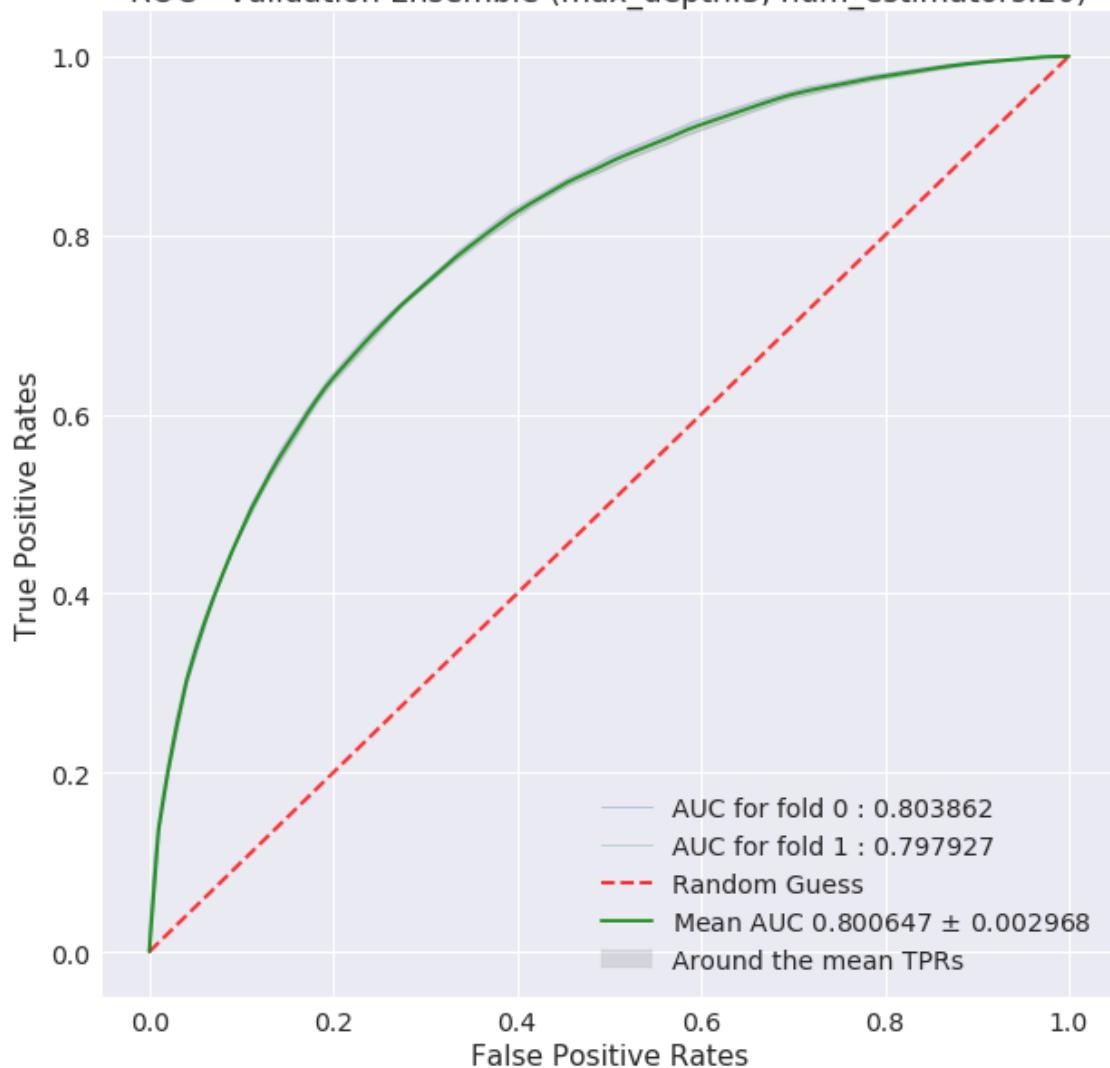
# test and evaluate the model
ptabe_entry_a4 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

Train df shape (50000, 52)
Class label distribution in train df:
0    25029
1    24971
Name: Label, dtype: int64
Test df shape (20000, 52)
Class label distribution in test df:
1    16520
0    3480
Name: Label, dtype: int64
Shape of -> train features :50000,50, test features: 20000,50
Shape of -> train labels :50000, test labels: 20000
=====
```

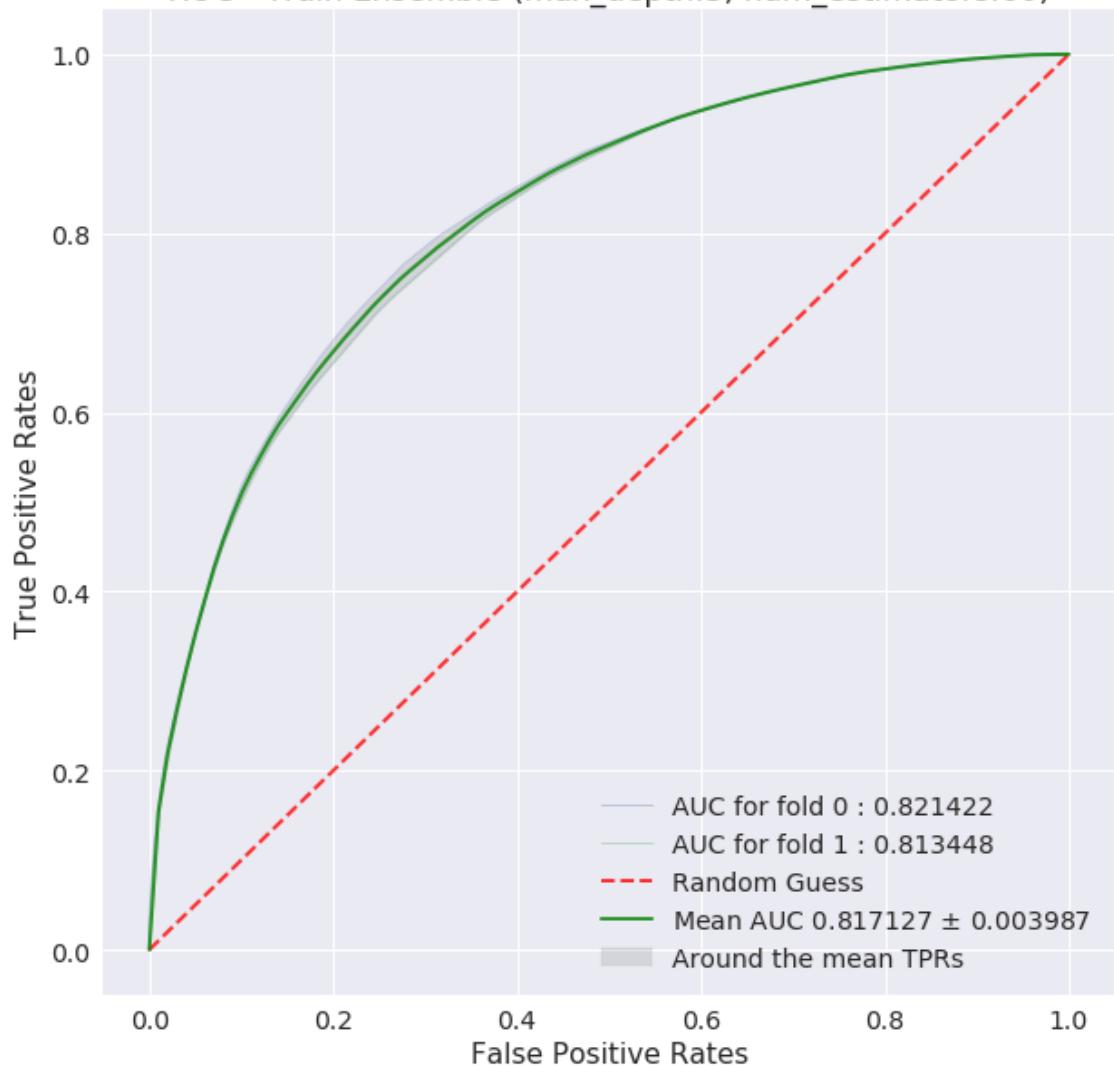
ROC - Train Ensemble (max\_depth:3, num\_estimators:20)



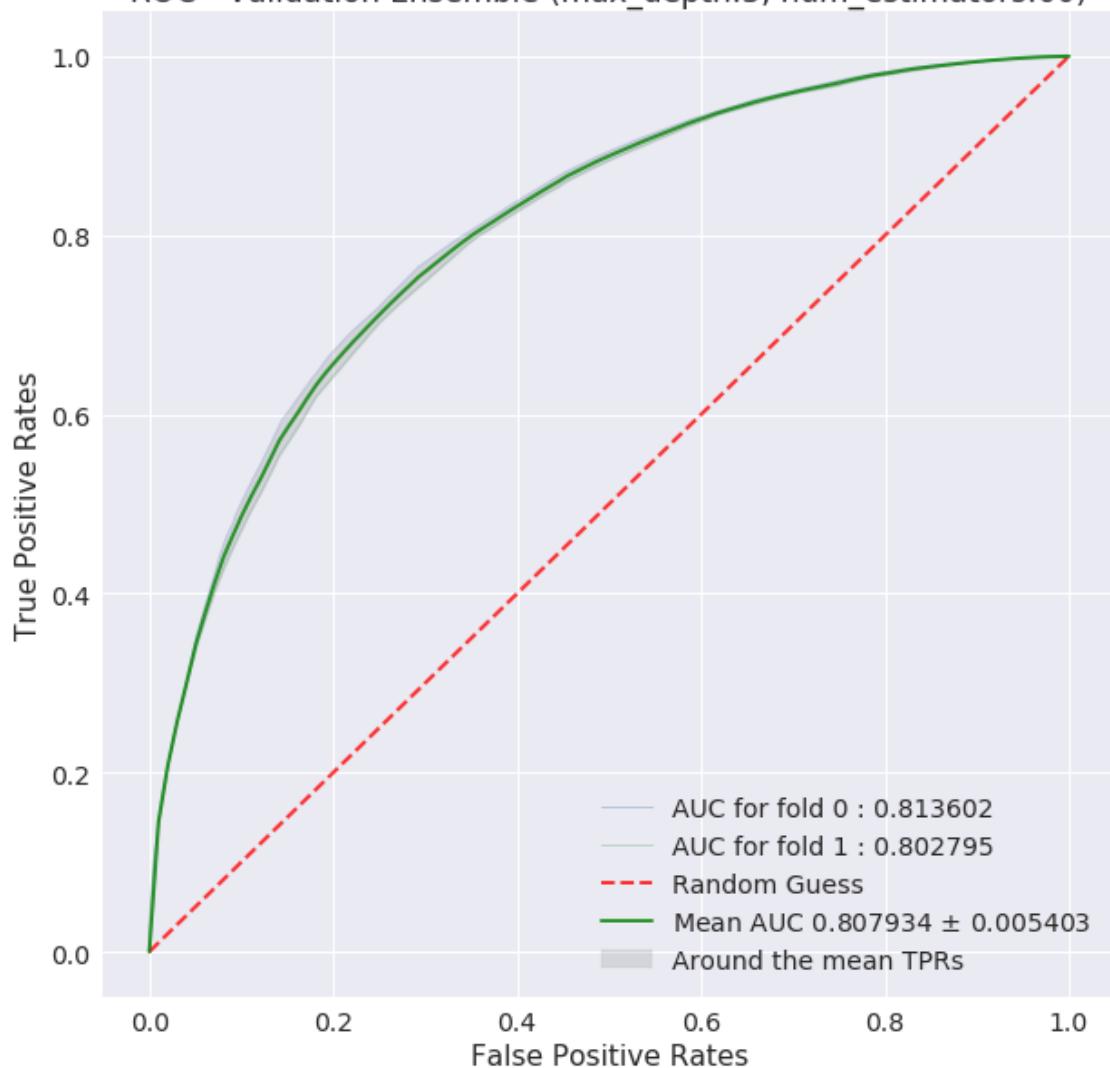
ROC - Validation Ensemble (max\_depth:3, num\_estimators:20)



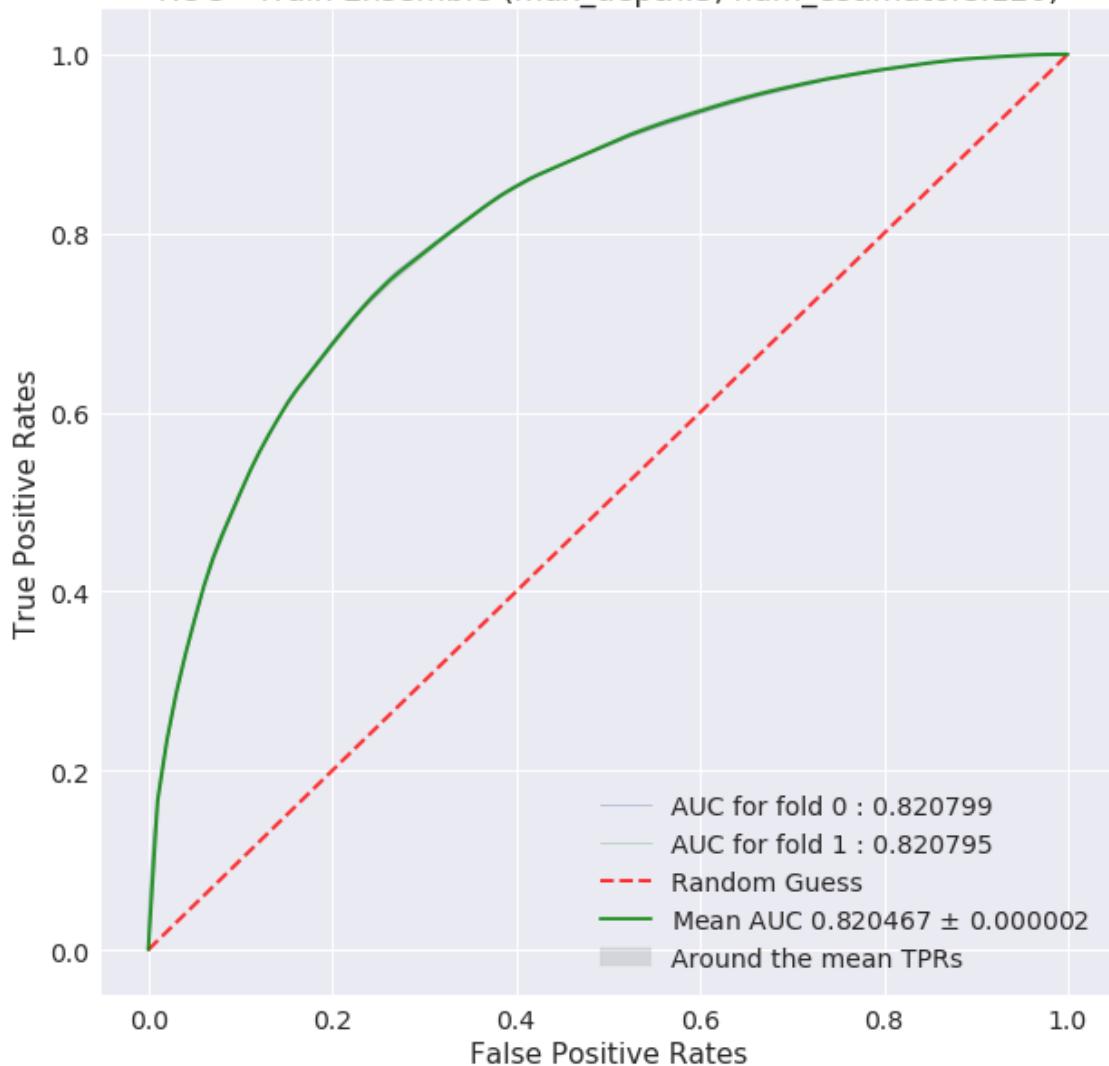
ROC - Train Ensemble (max\_depth:3, num\_estimators:60)



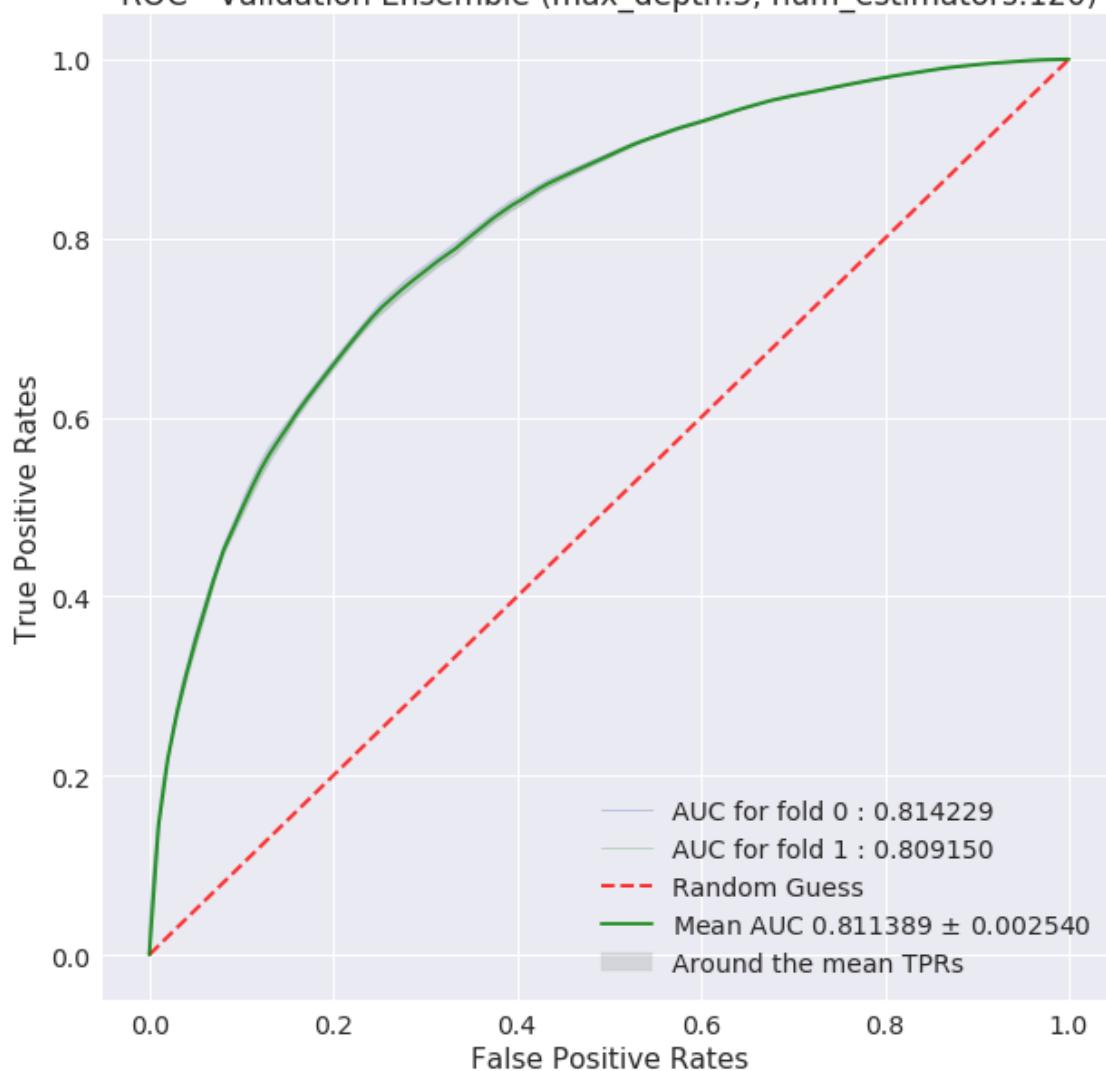
ROC - Validation Ensemble (max\_depth:3, num\_estimators:60)



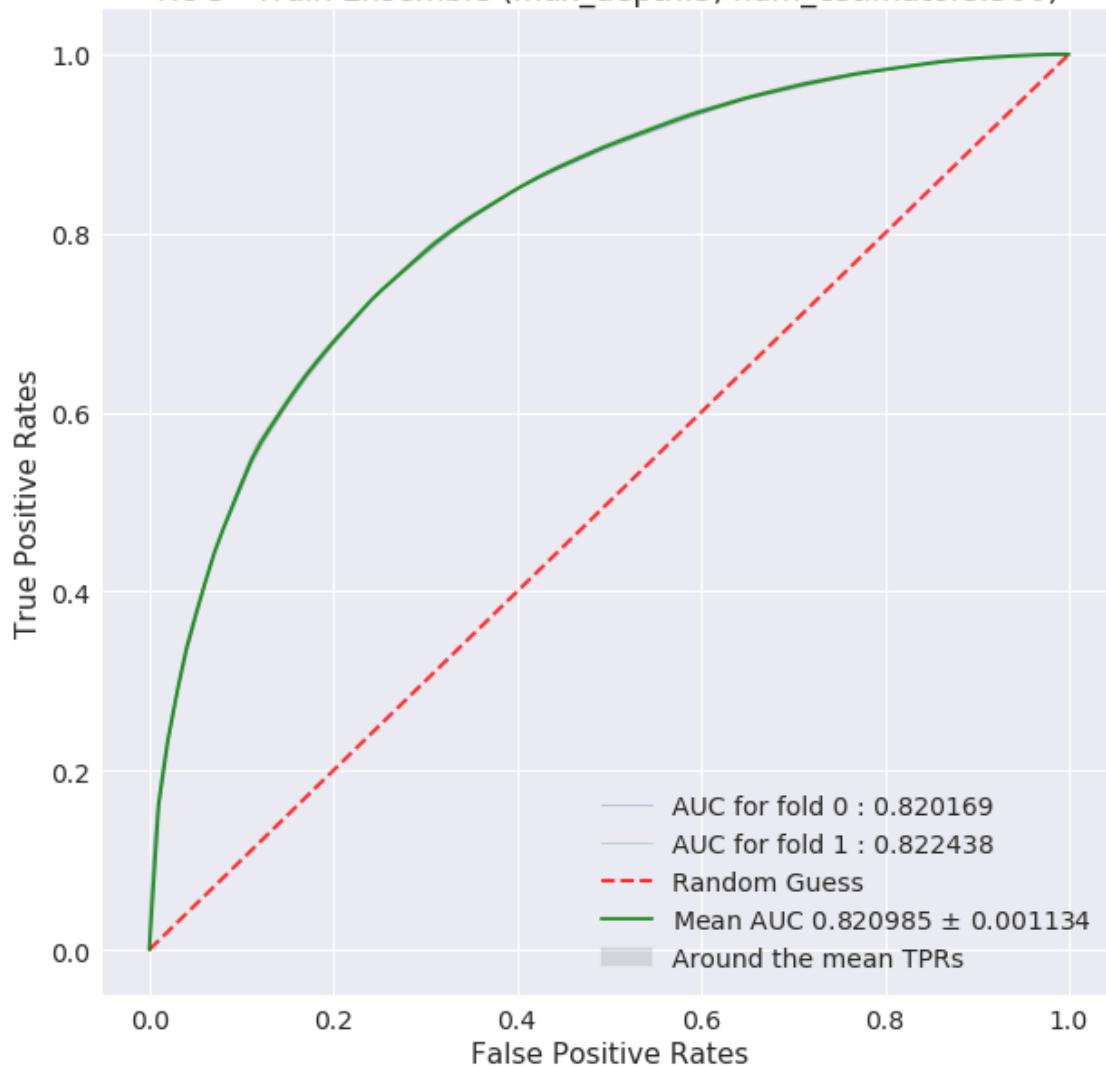
ROC - Train Ensemble (max\_depth:3, num\_estimators:120)



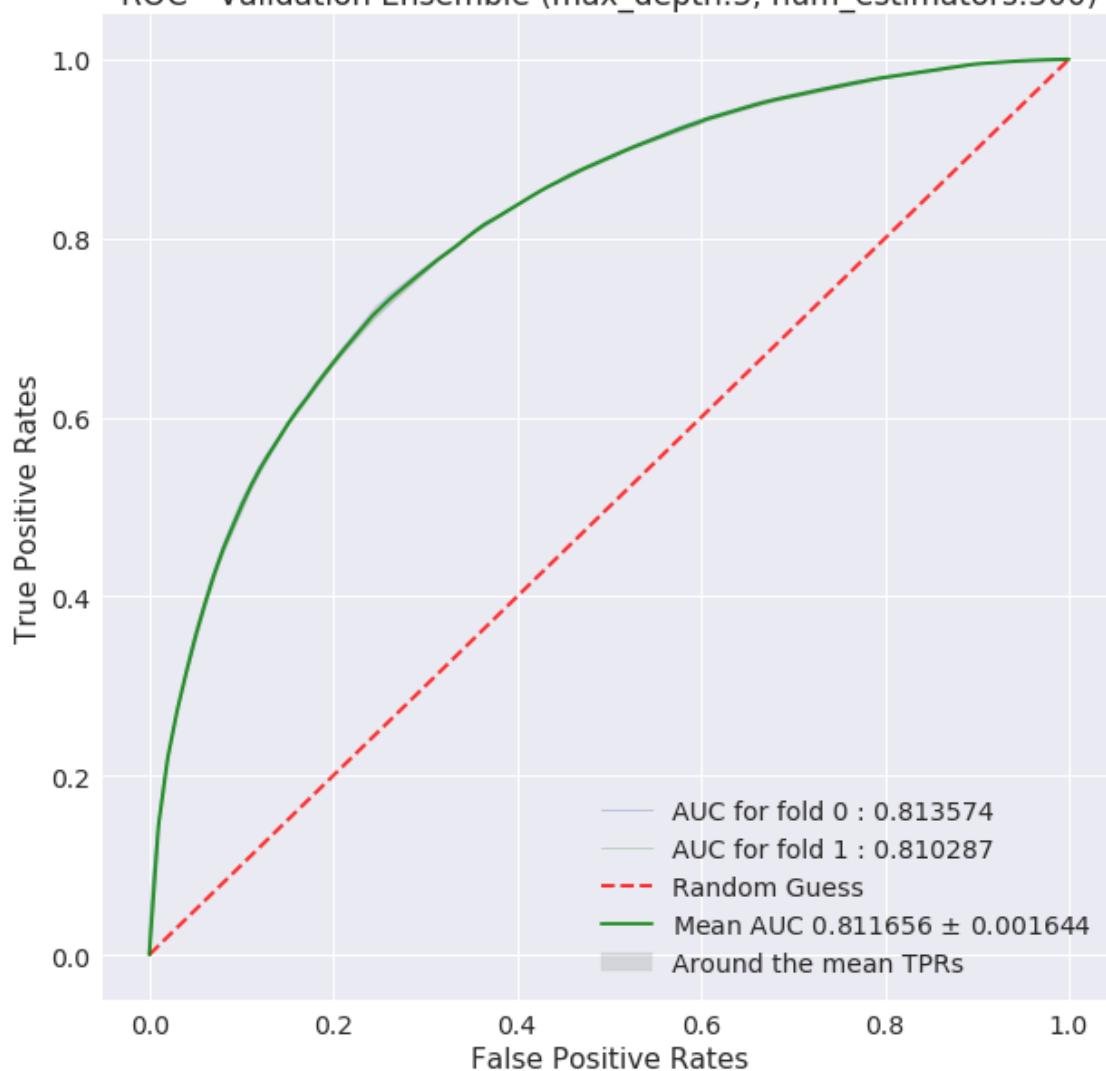
ROC - Validation Ensemble (max\_depth:3, num\_estimators:120)



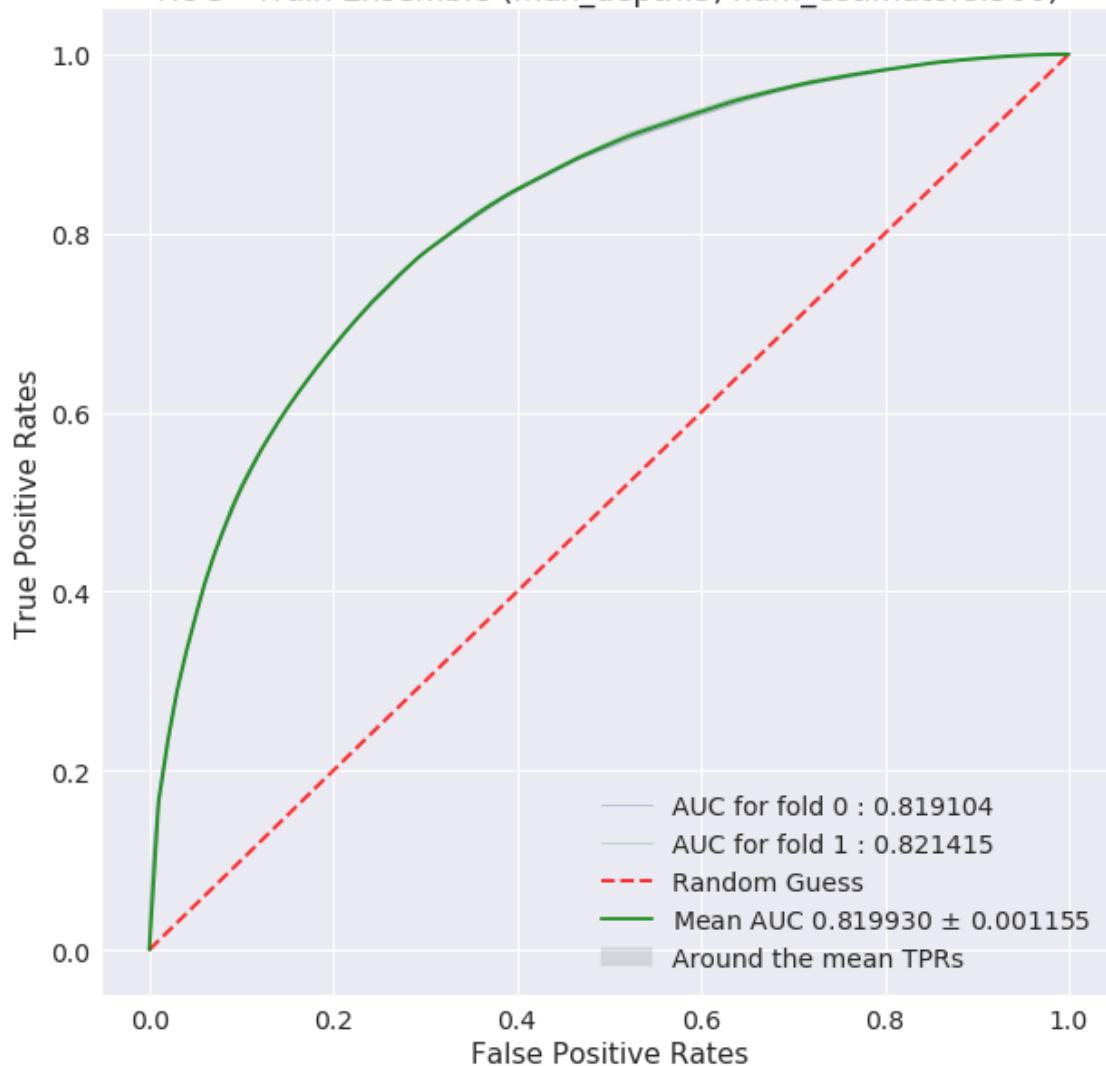
ROC - Train Ensemble (max\_depth:3, num\_estimators:300)



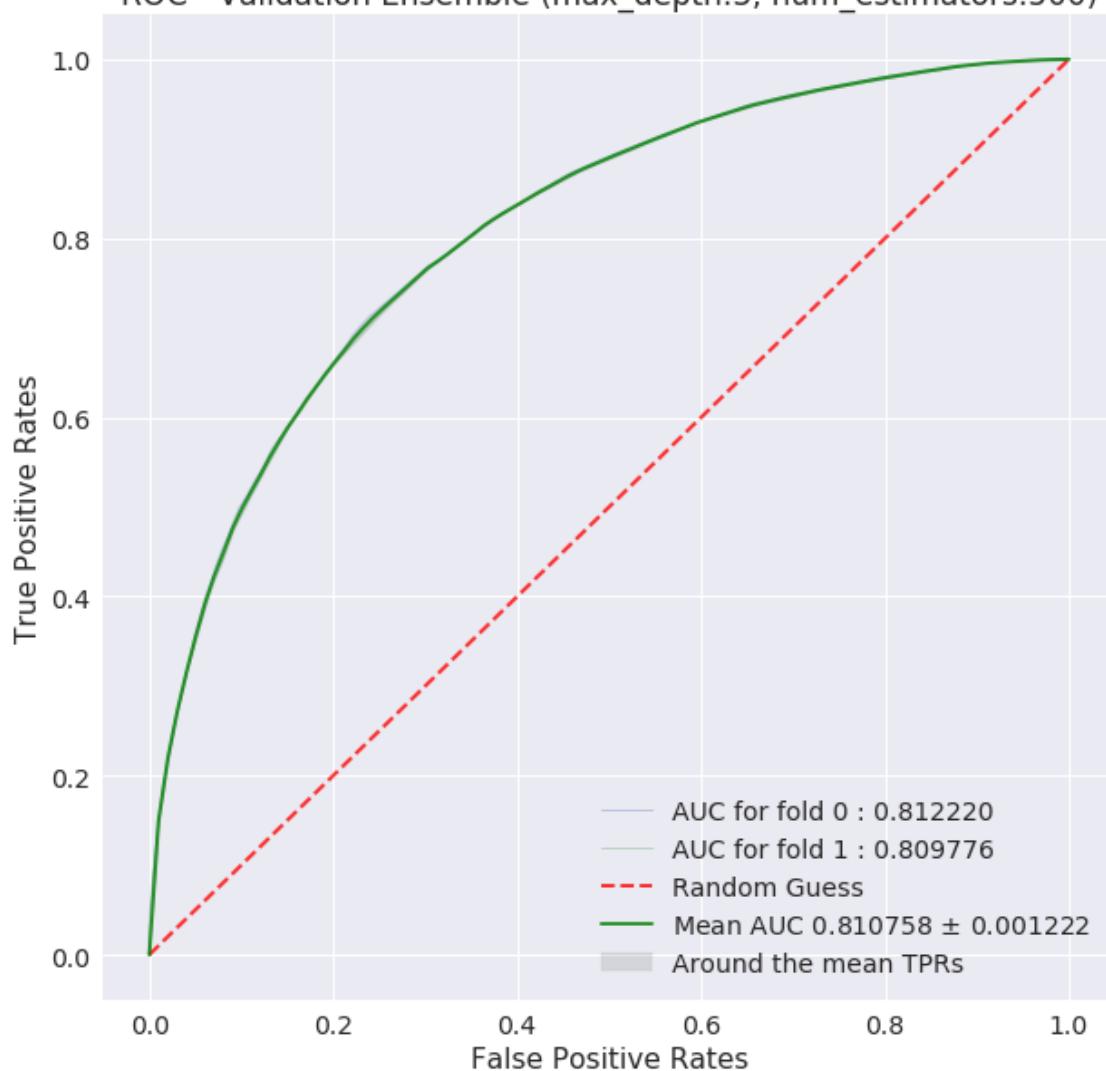
ROC - Validation Ensemble (max\_depth:3, num\_estimators:300)



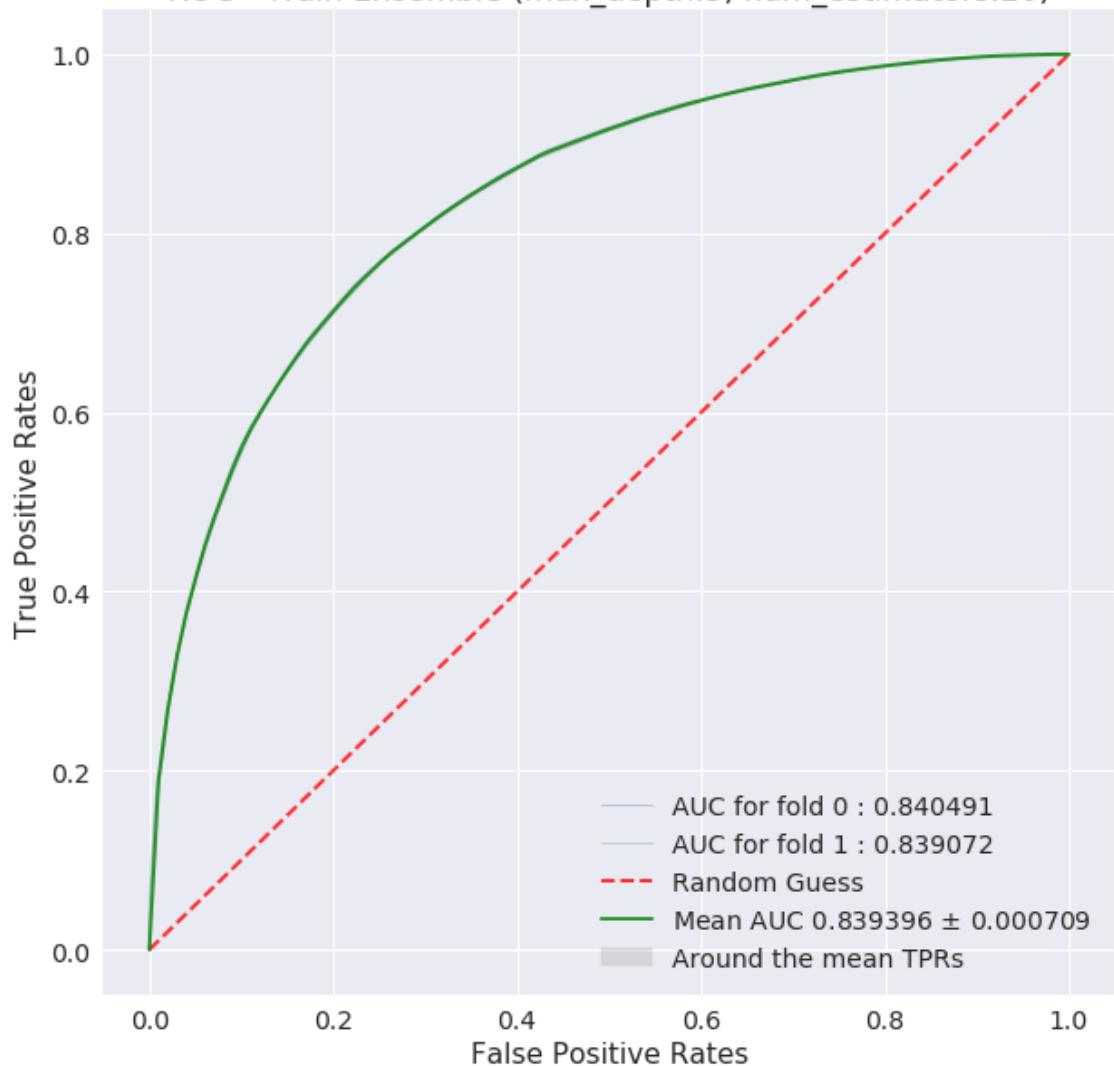
ROC - Train Ensemble (max\_depth:3, num\_estimators:500)



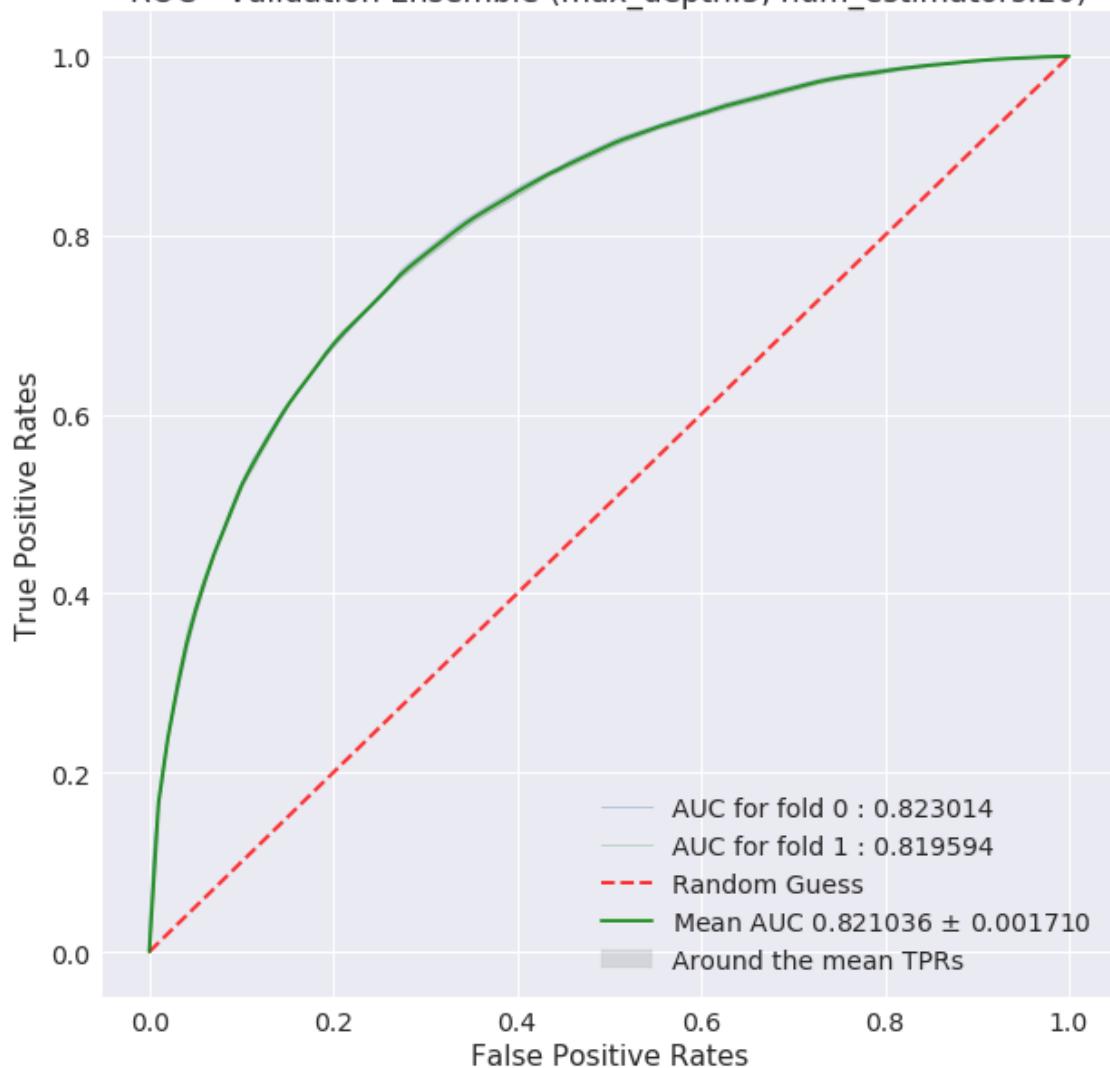
ROC - Validation Ensemble (max\_depth:3, num\_estimators:500)



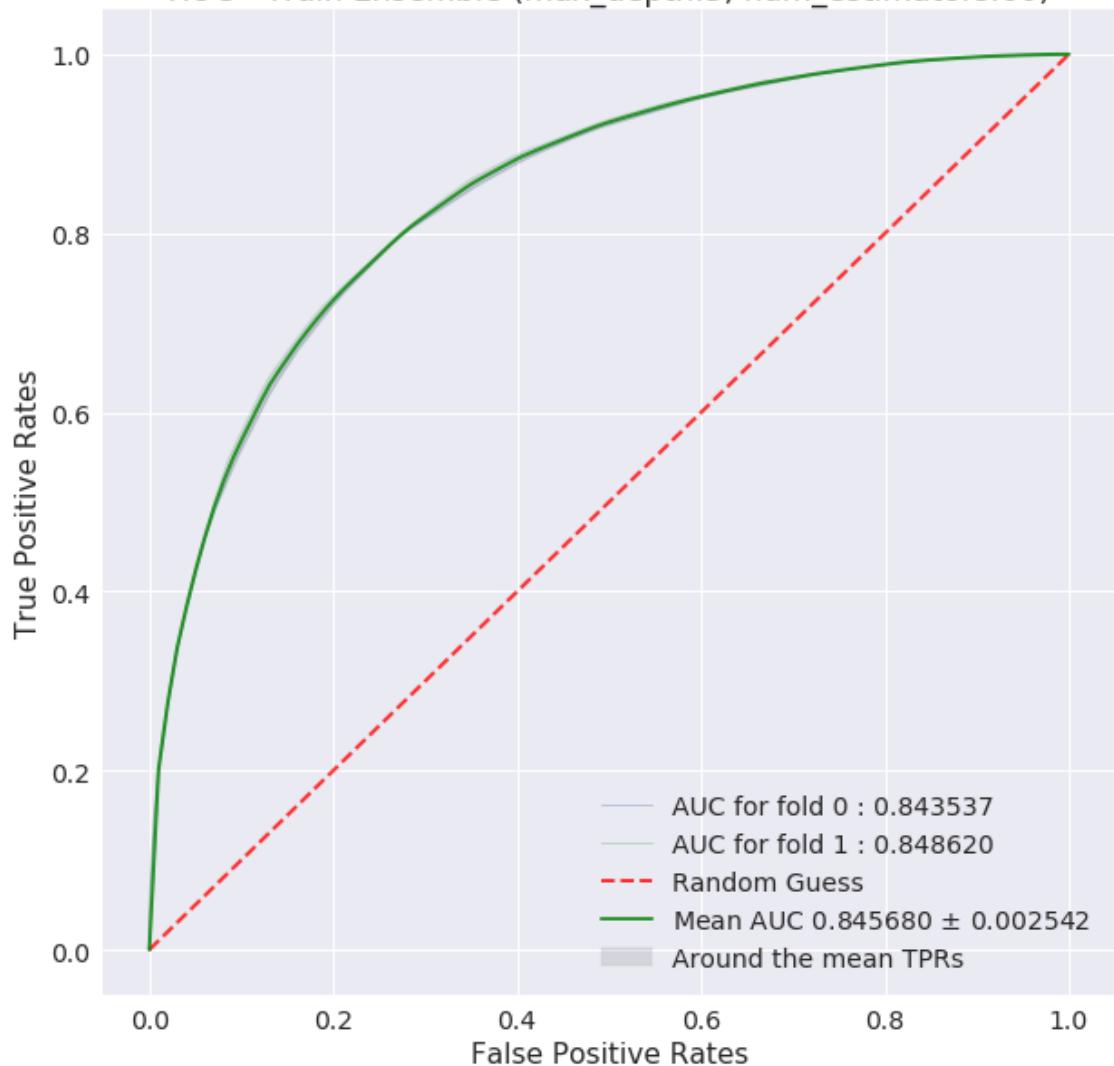
ROC - Train Ensemble (max\_depth:5, num\_estimators:20)



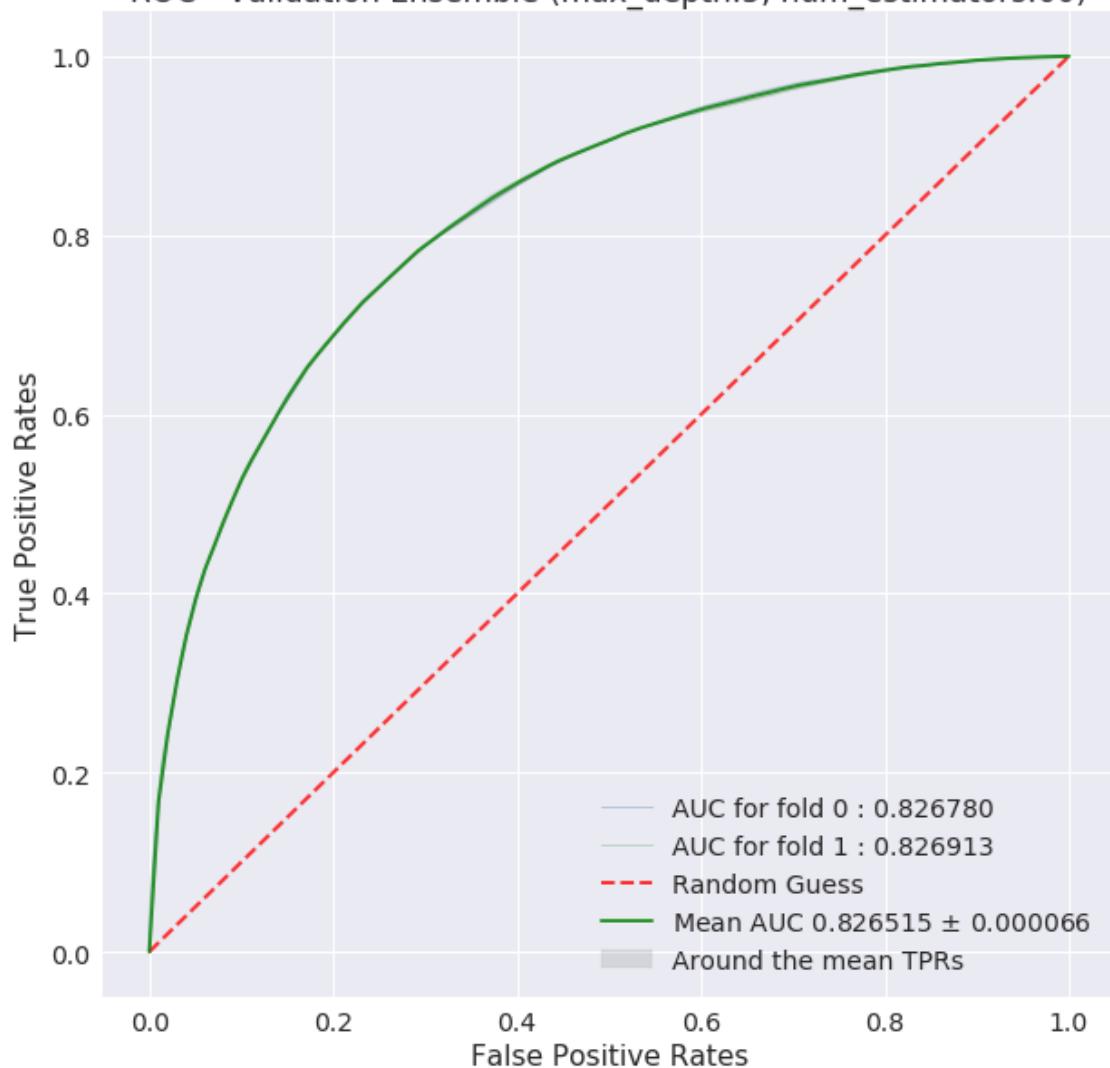
ROC - Validation Ensemble (max\_depth:5, num\_estimators:20)



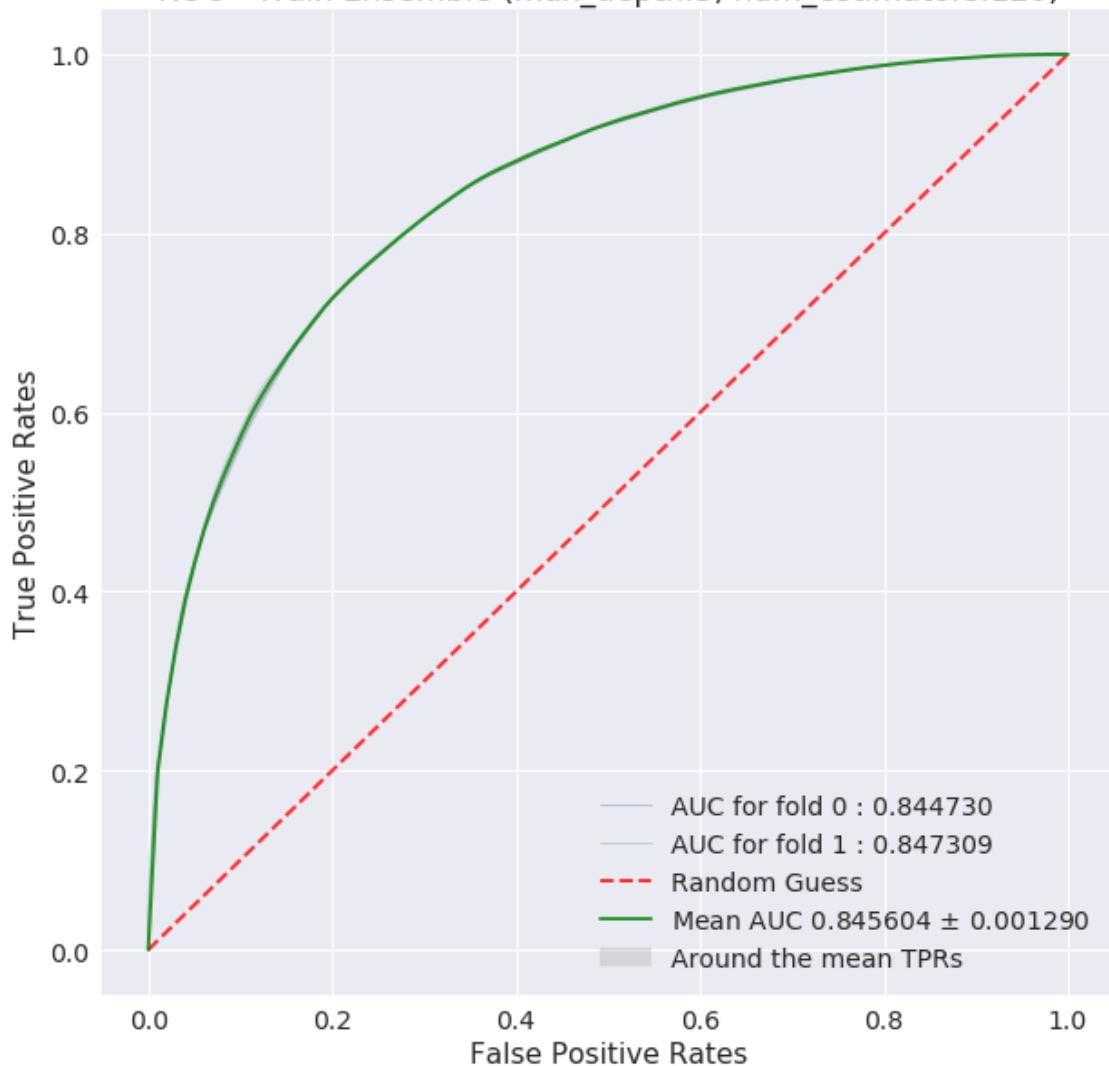
ROC - Train Ensemble (max\_depth:5, num\_estimators:60)



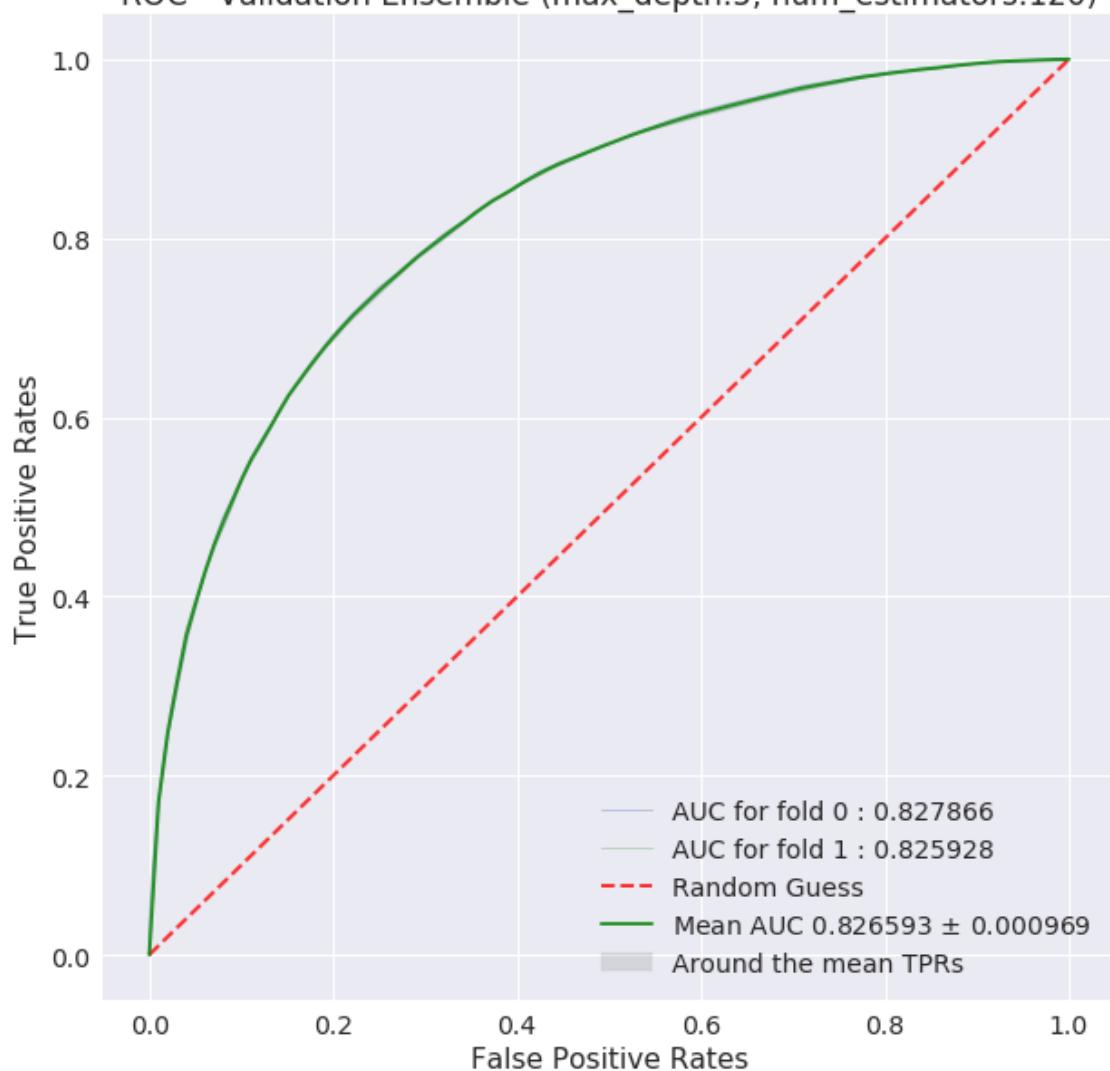
ROC - Validation Ensemble (max\_depth:5, num\_estimators:60)



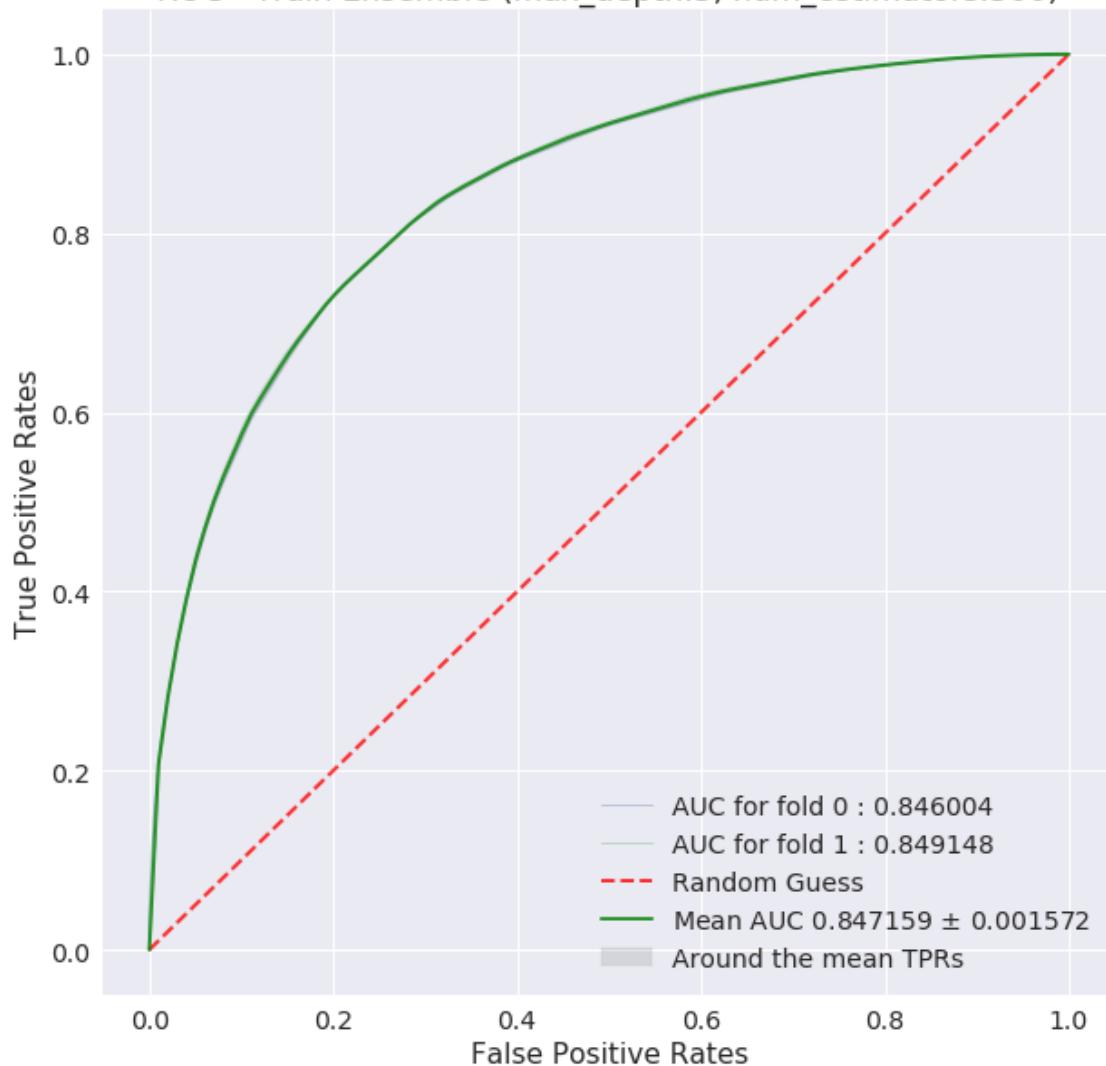
ROC - Train Ensemble (max\_depth:5, num\_estimators:120)



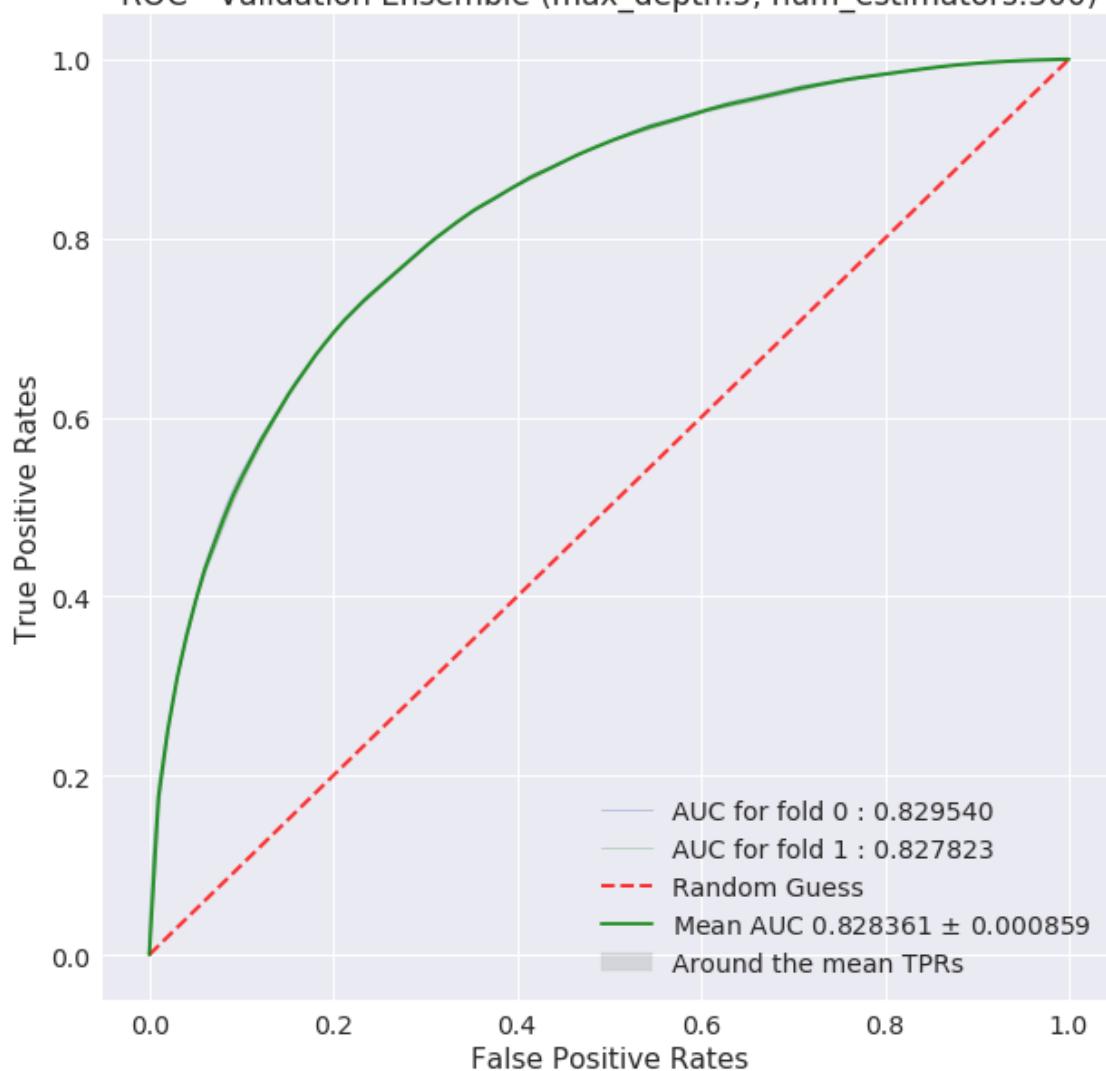
ROC - Validation Ensemble (max\_depth:5, num\_estimators:120)



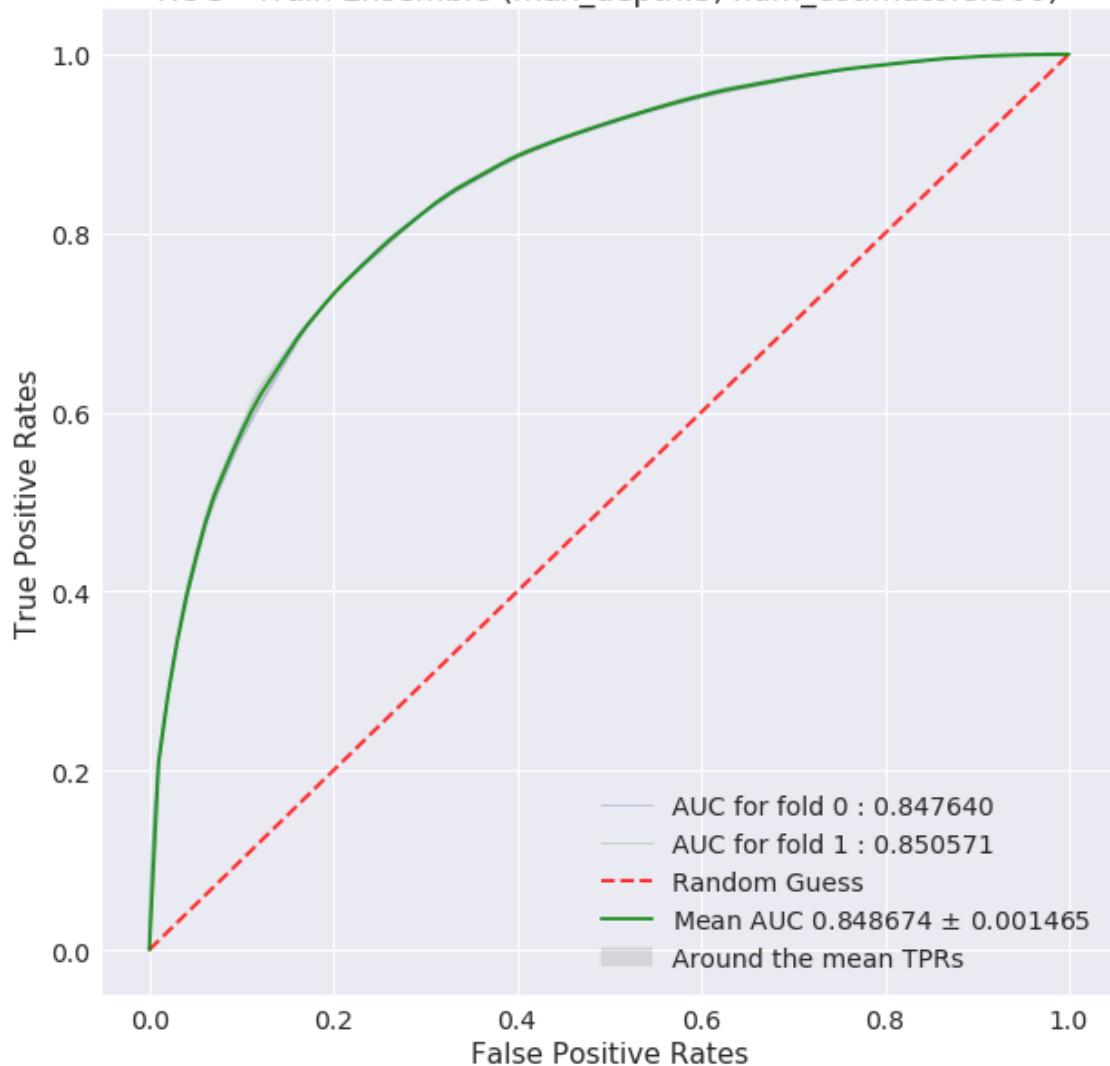
ROC - Train Ensemble (max\_depth:5, num\_estimators:300)



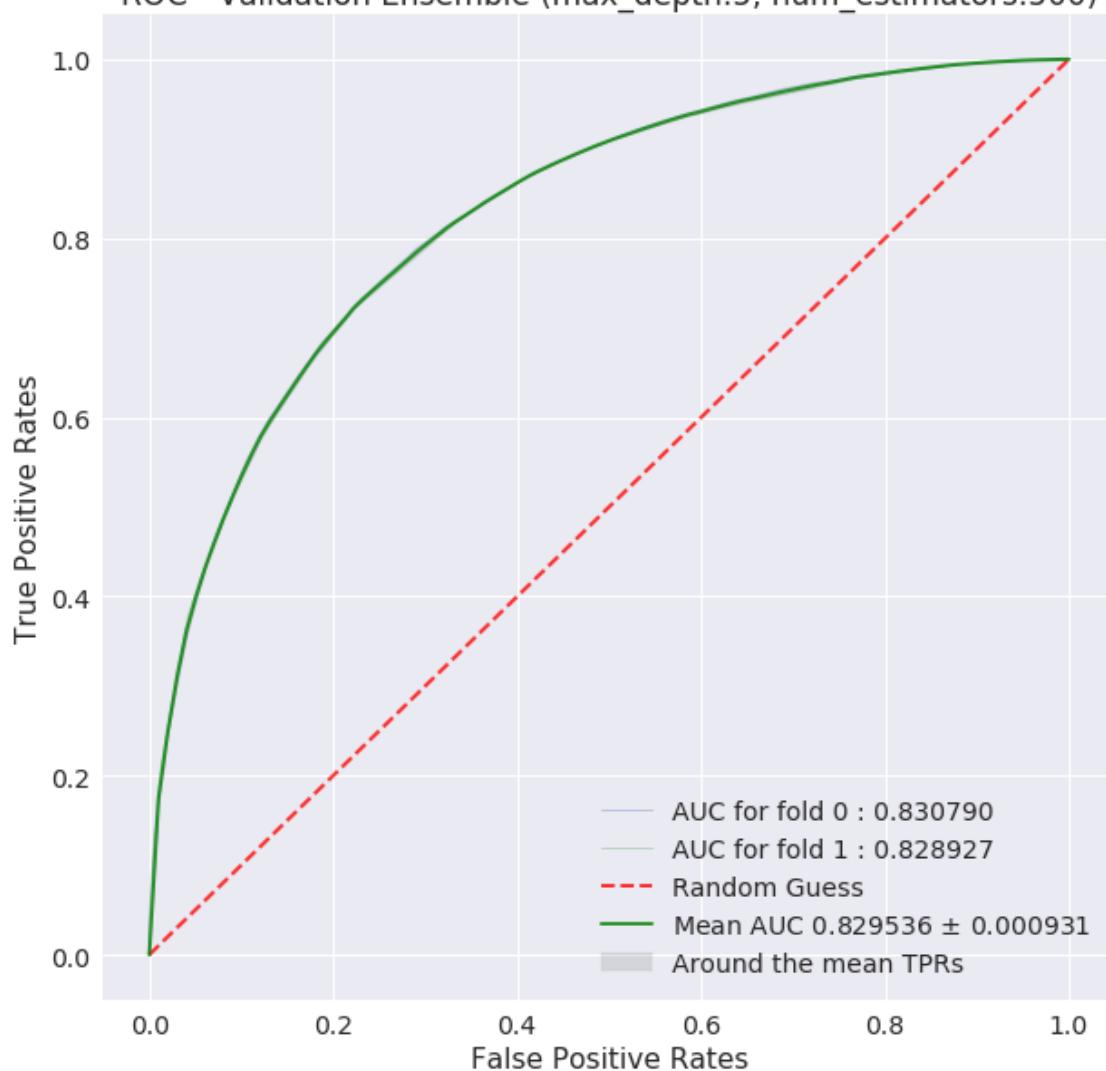
ROC - Validation Ensemble (max\_depth:5, num\_estimators:300)



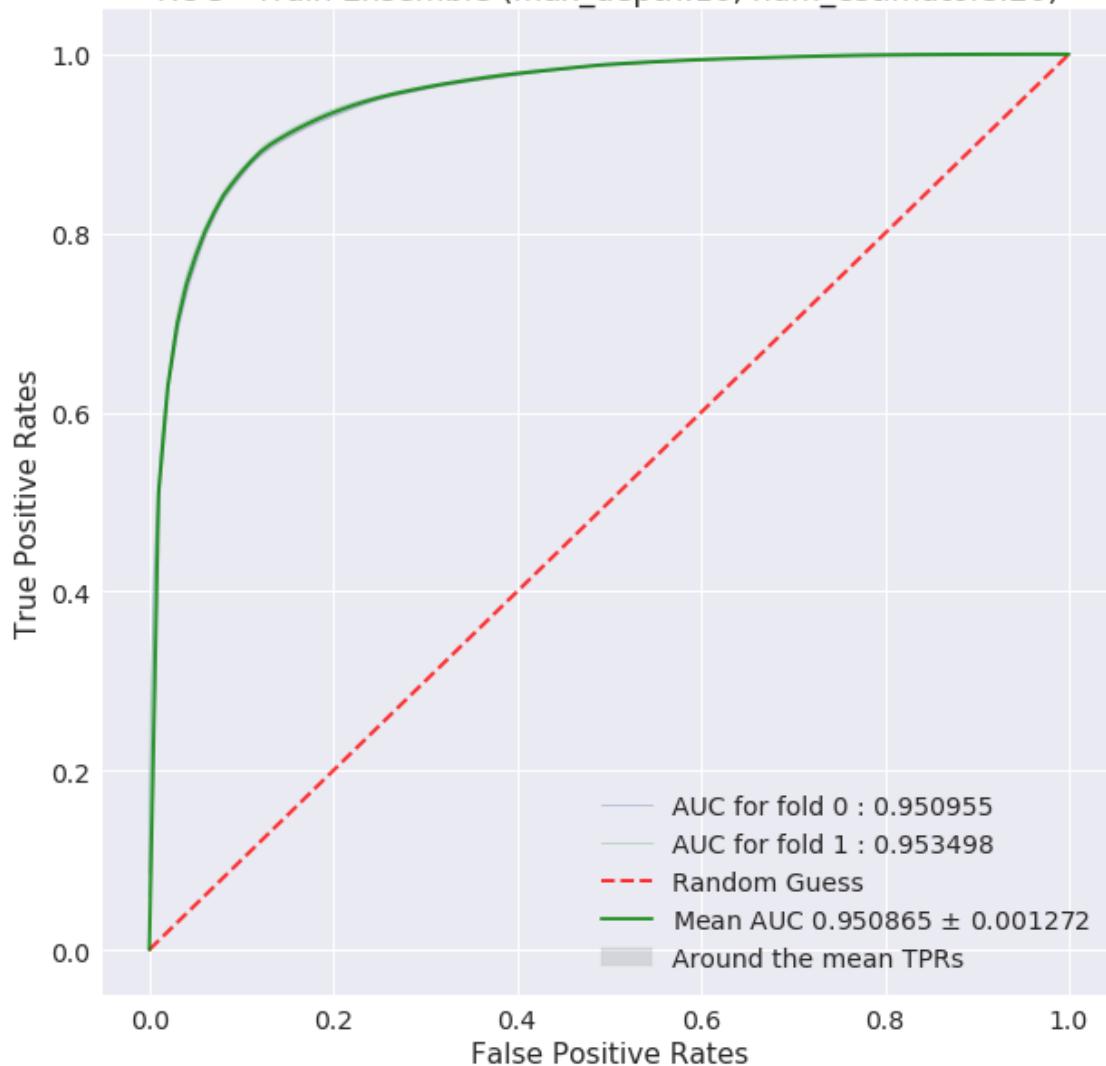
ROC - Train Ensemble (max\_depth:5, num\_estimators:500)



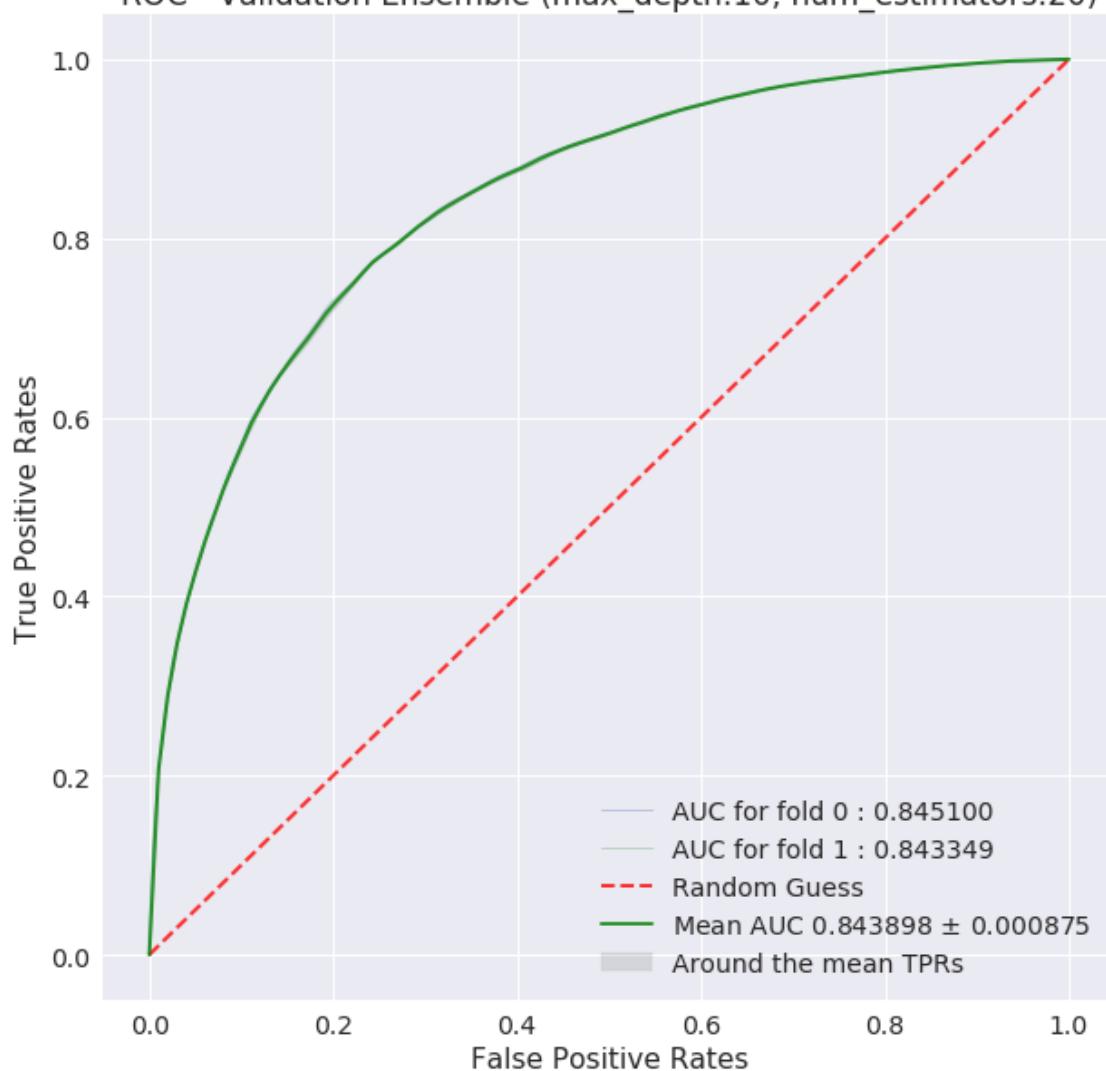
ROC - Validation Ensemble (max\_depth:5, num\_estimators:500)



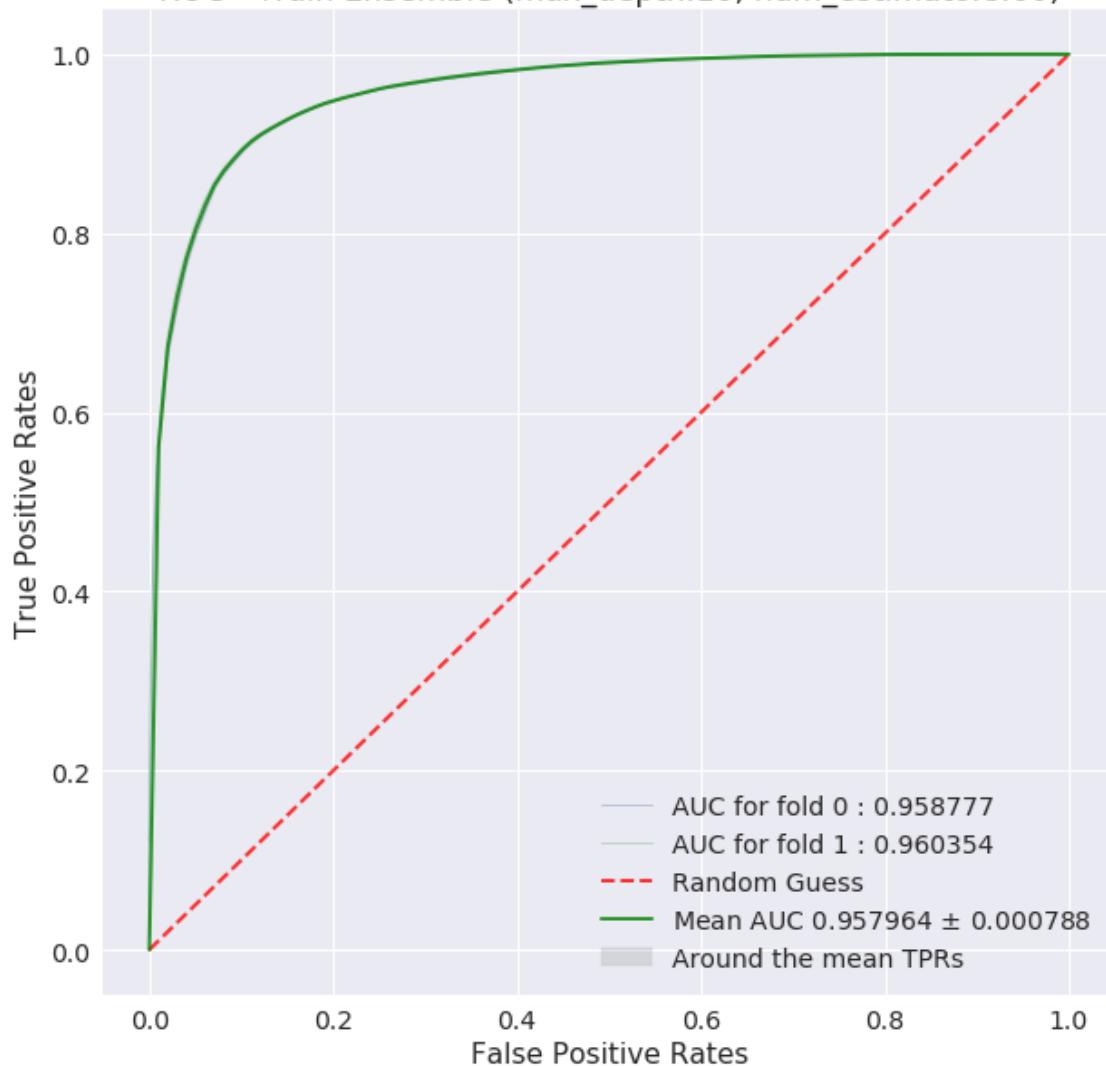
ROC - Train Ensemble (max\_depth:10, num\_estimators:20)



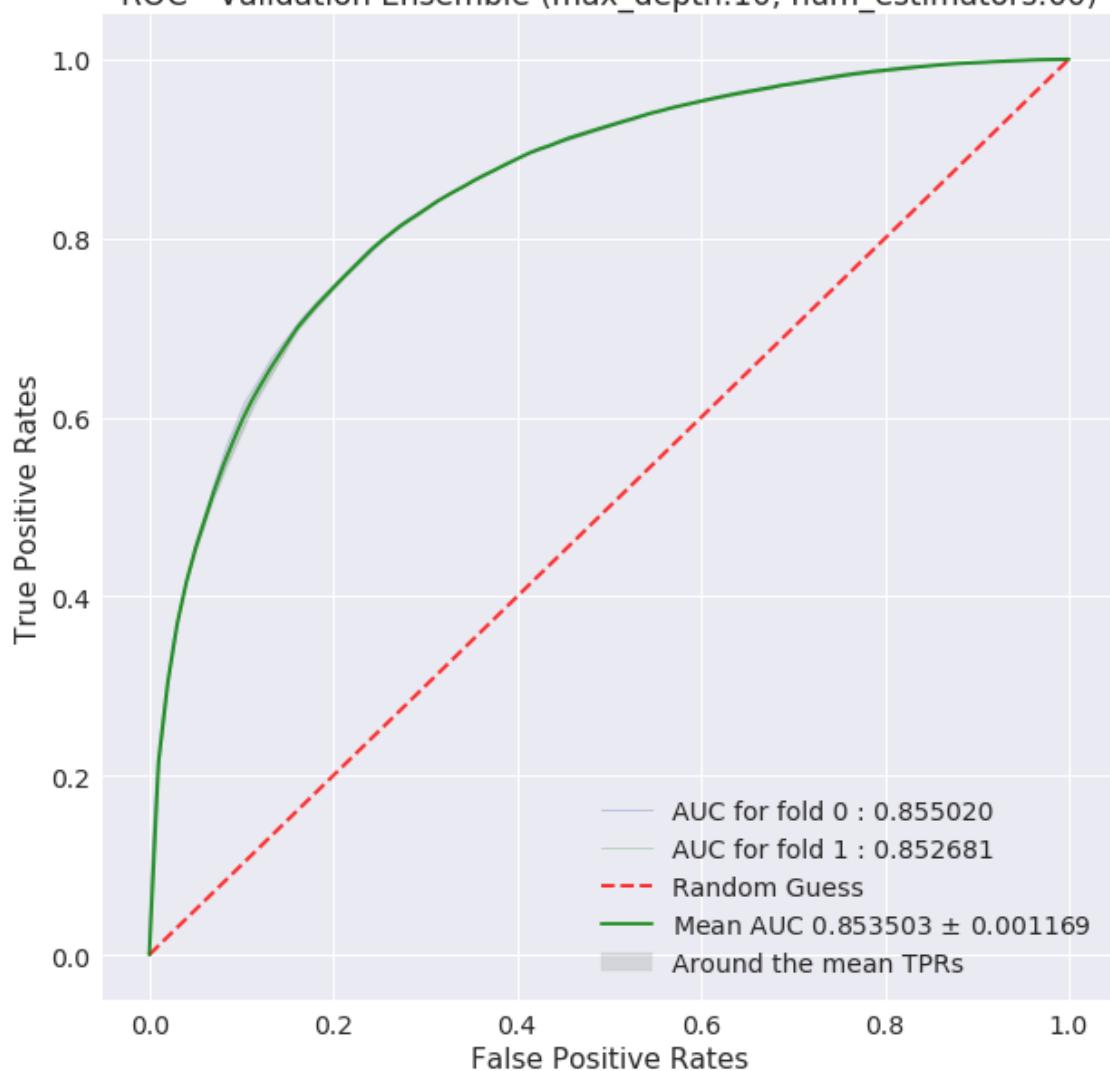
ROC - Validation Ensemble (max\_depth:10, num\_estimators:20)



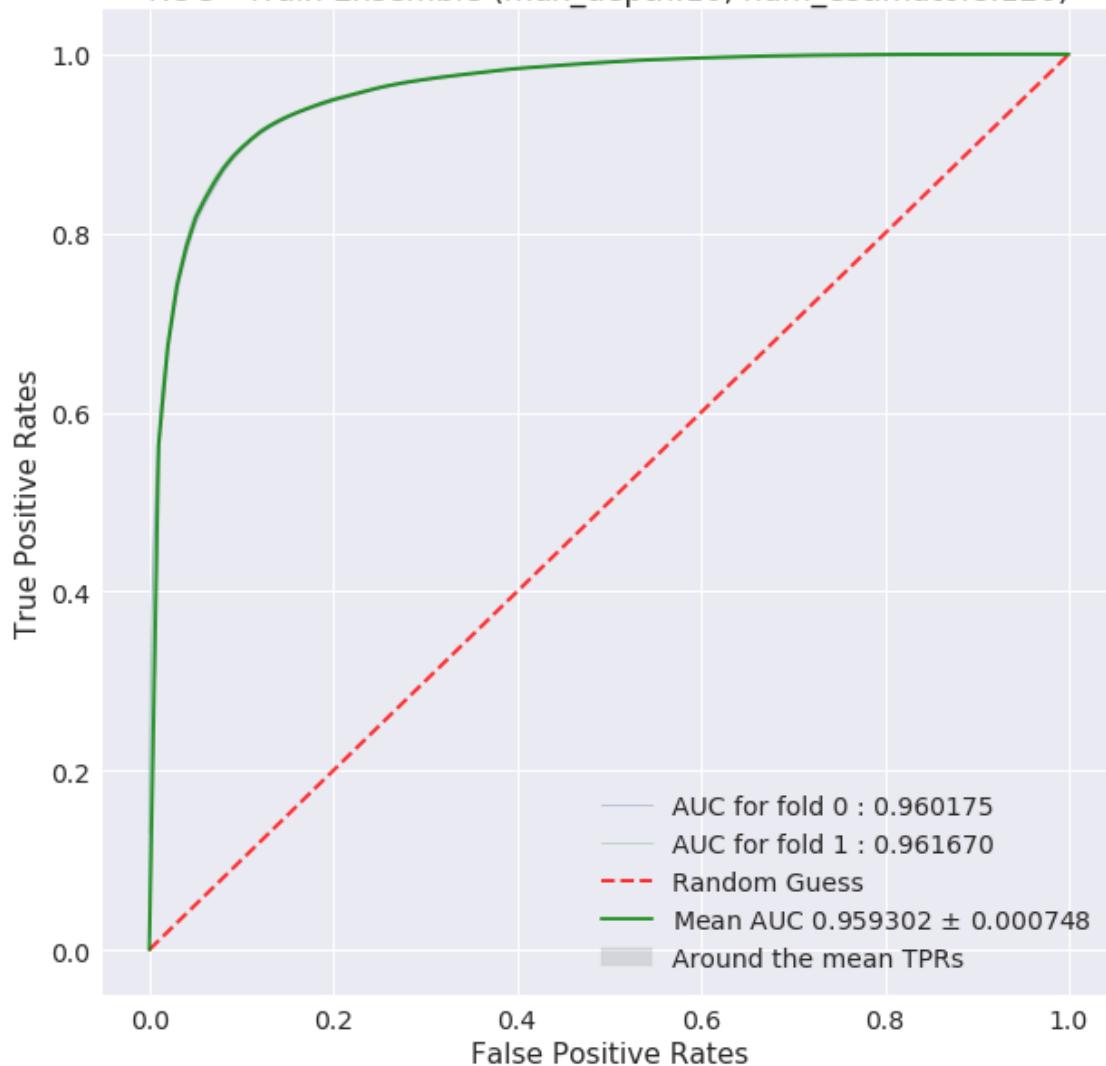
ROC - Train Ensemble (max\_depth:10, num\_estimators:60)



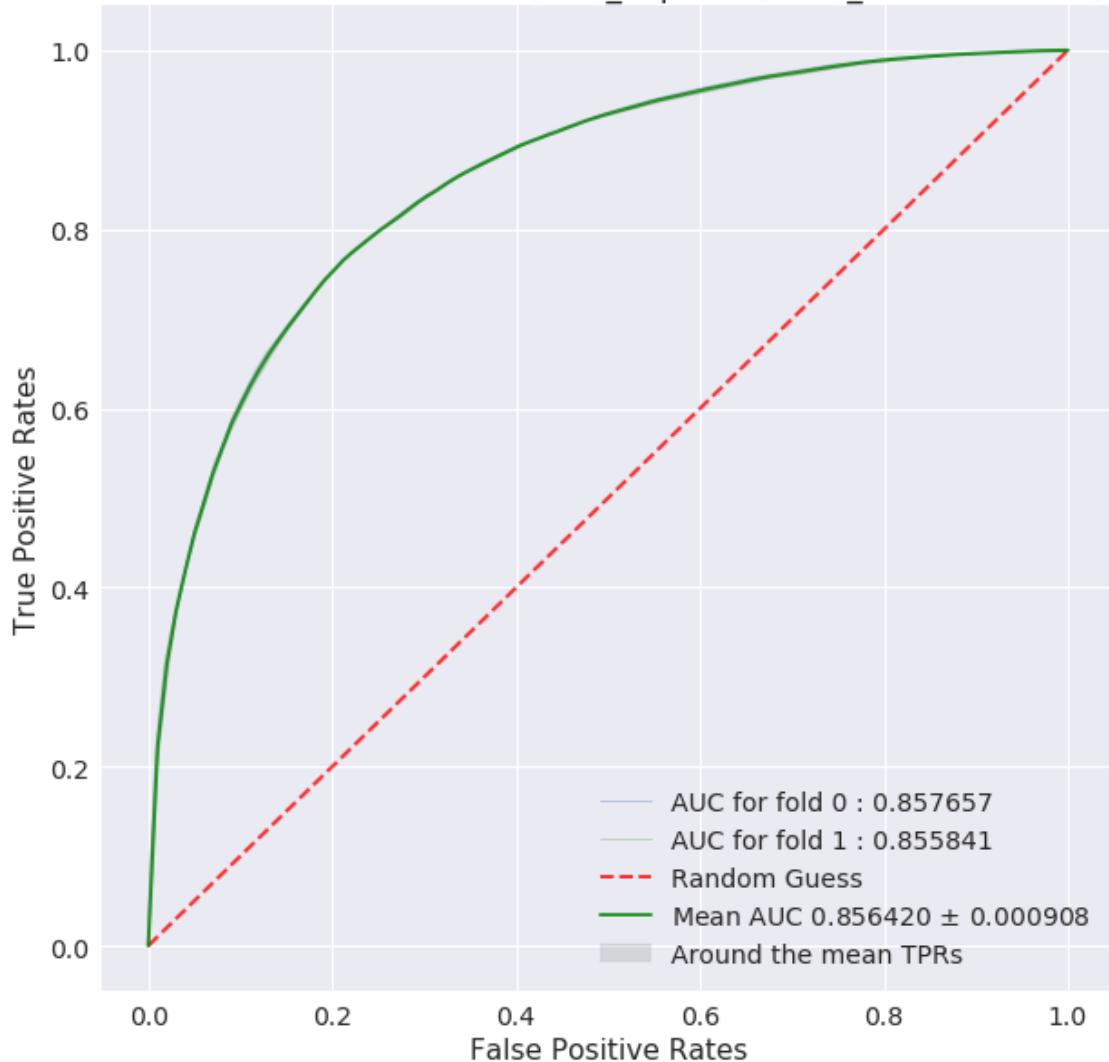
ROC - Validation Ensemble (max\_depth:10, num\_estimators:60)



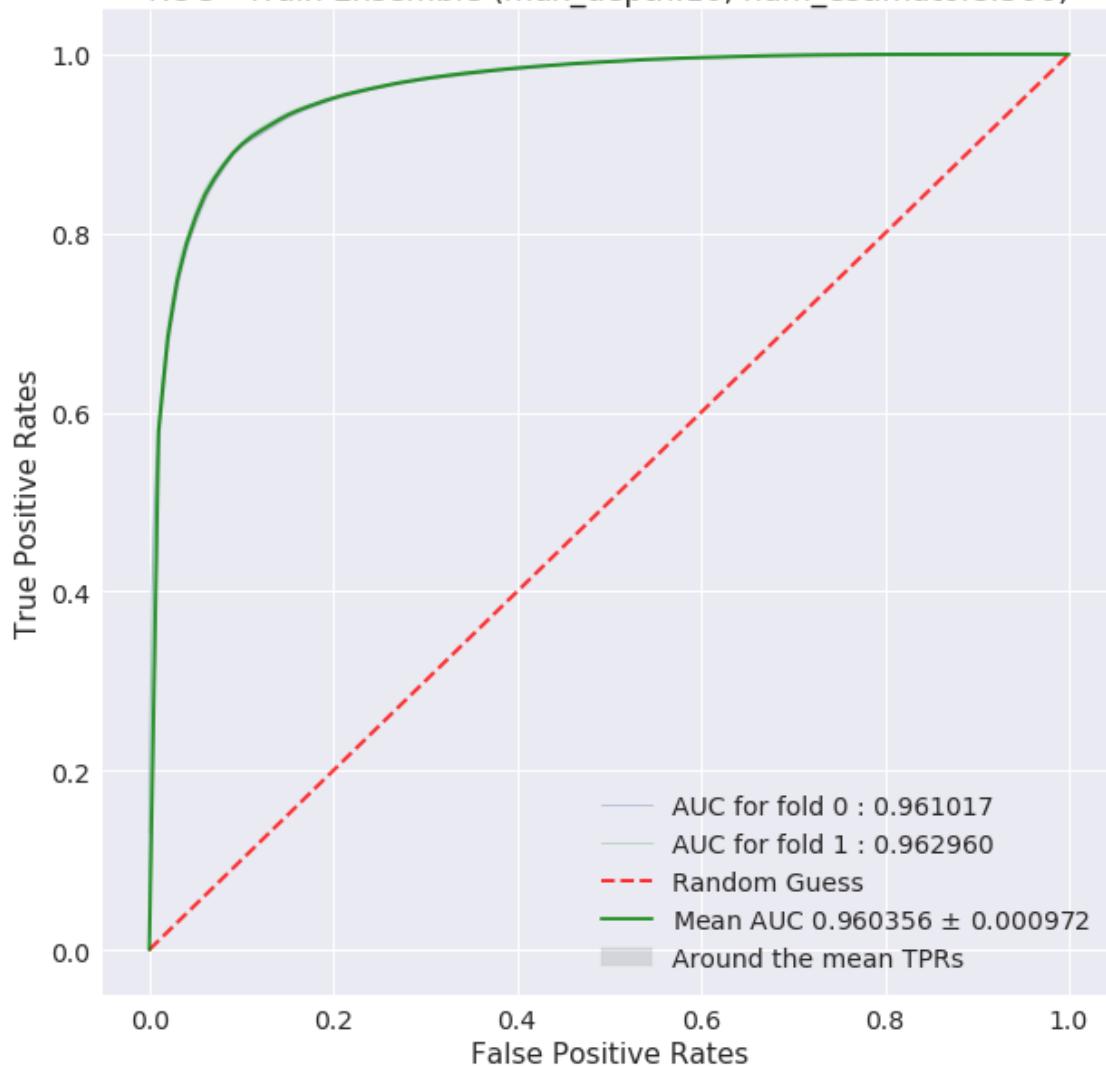
ROC - Train Ensemble (max\_depth:10, num\_estimators:120)



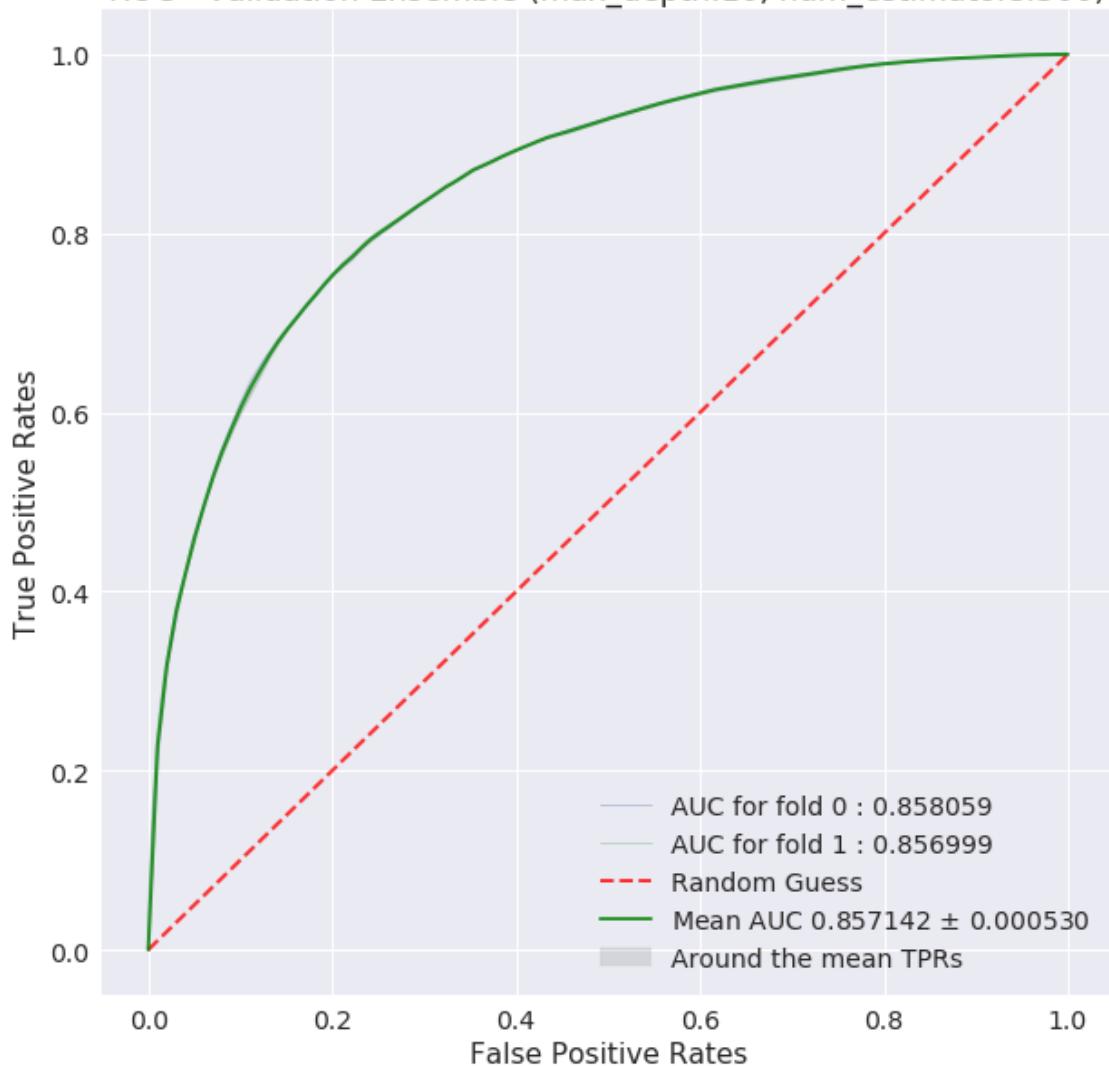
ROC - Validation Ensemble (max\_depth:10, num\_estimators:120)



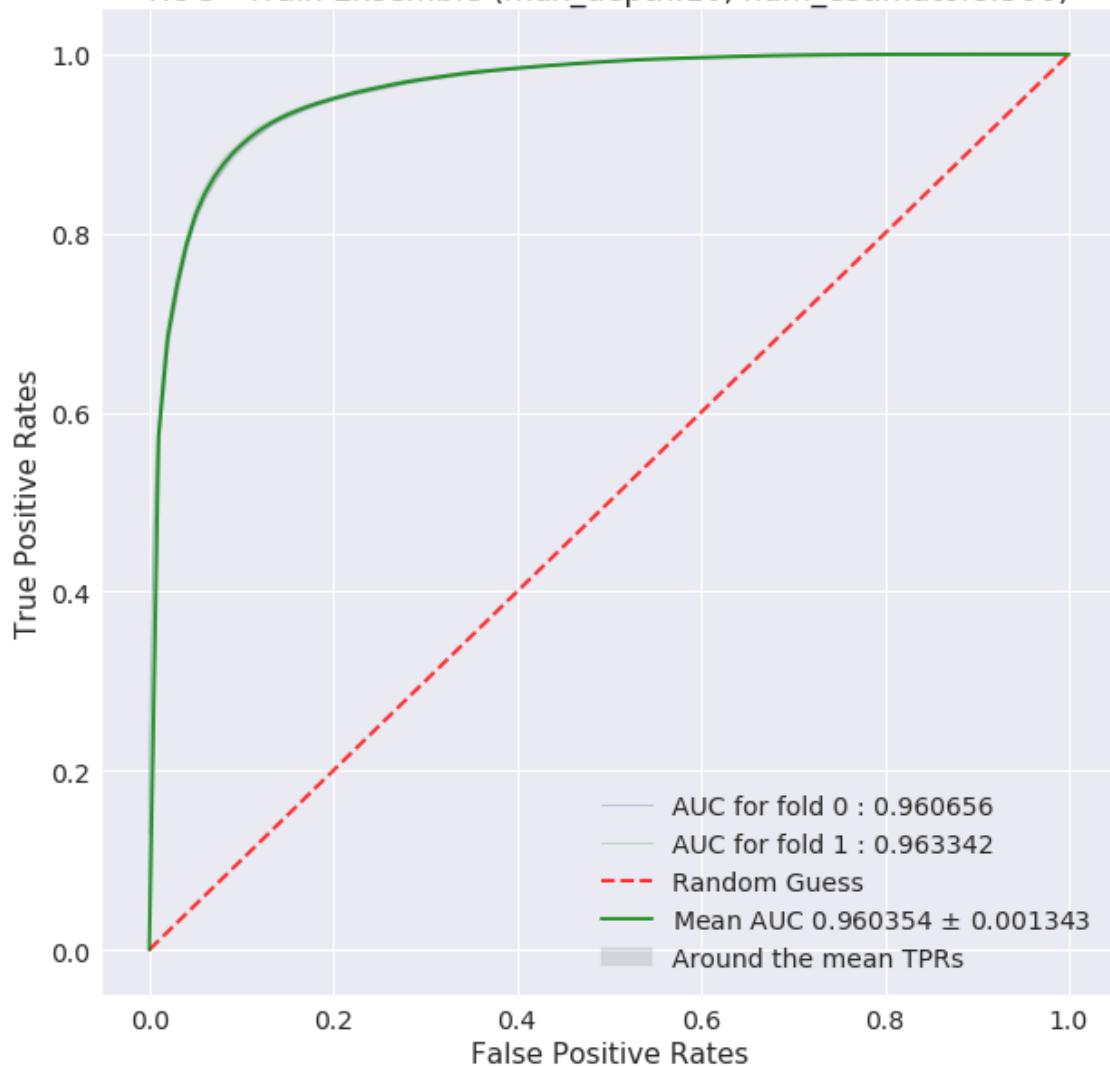
ROC - Train Ensemble (max\_depth:10, num\_estimators:300)



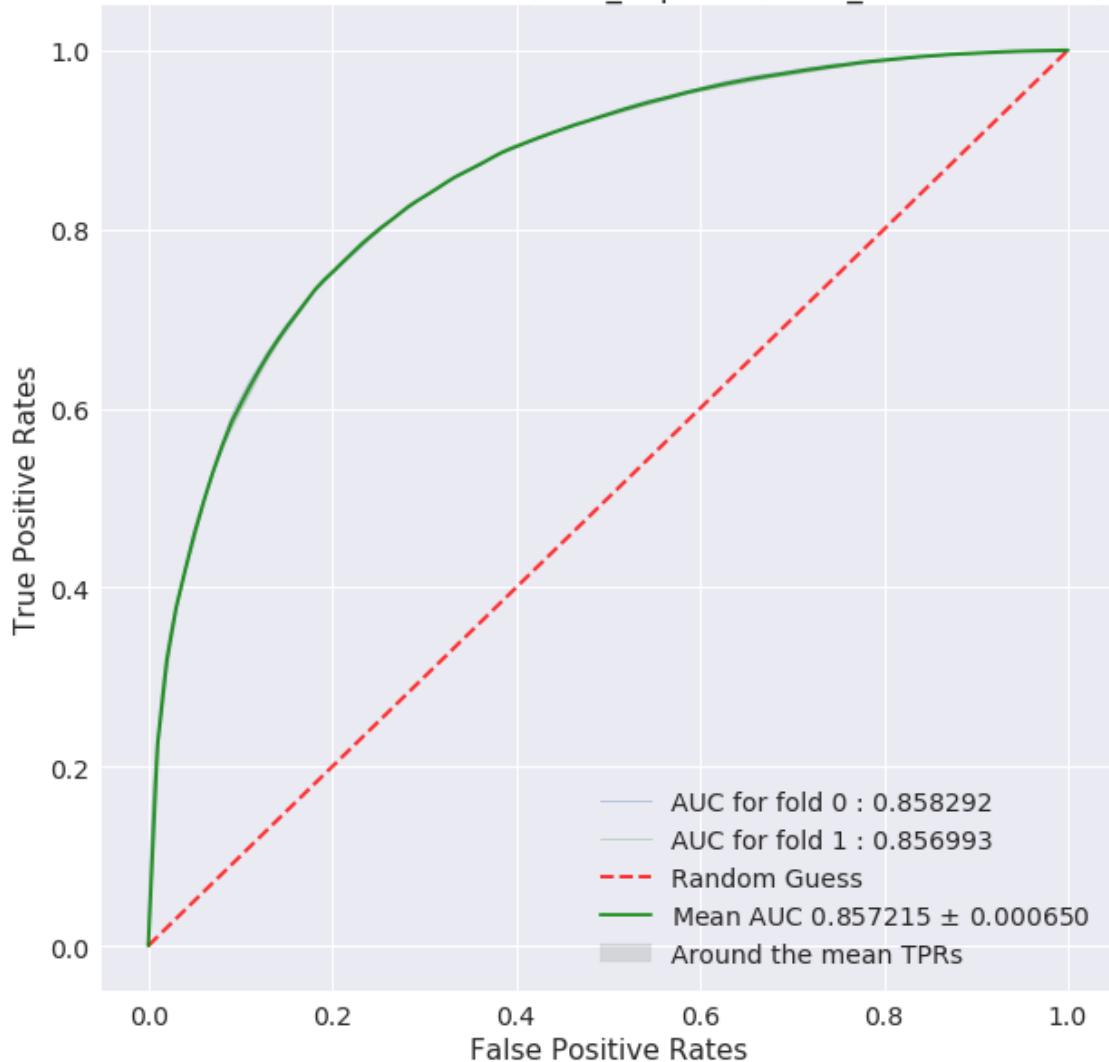
ROC - Validation Ensemble (max\_depth:10, num\_estimators:300)



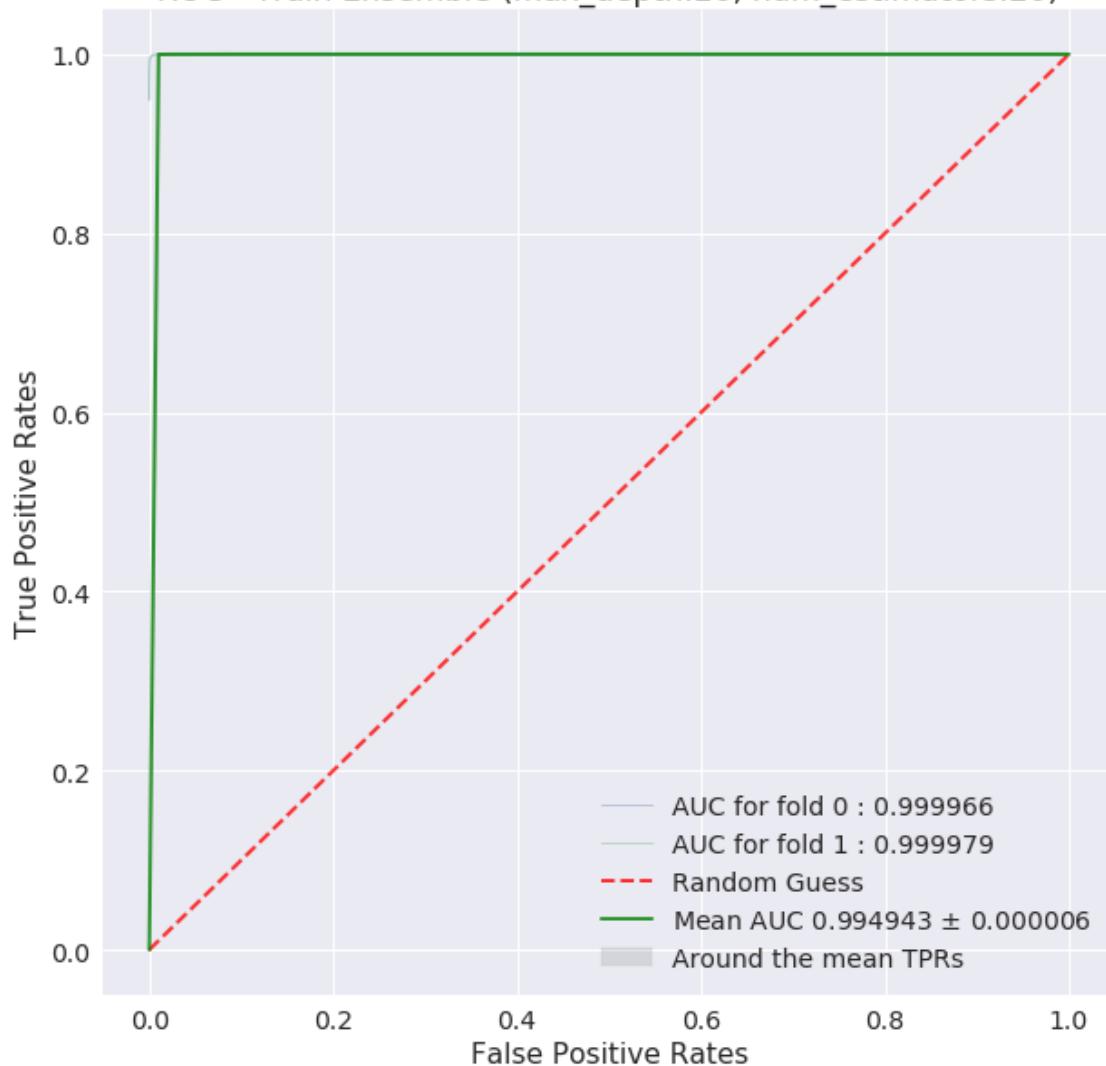
ROC - Train Ensemble (max\_depth:10, num\_estimators:500)



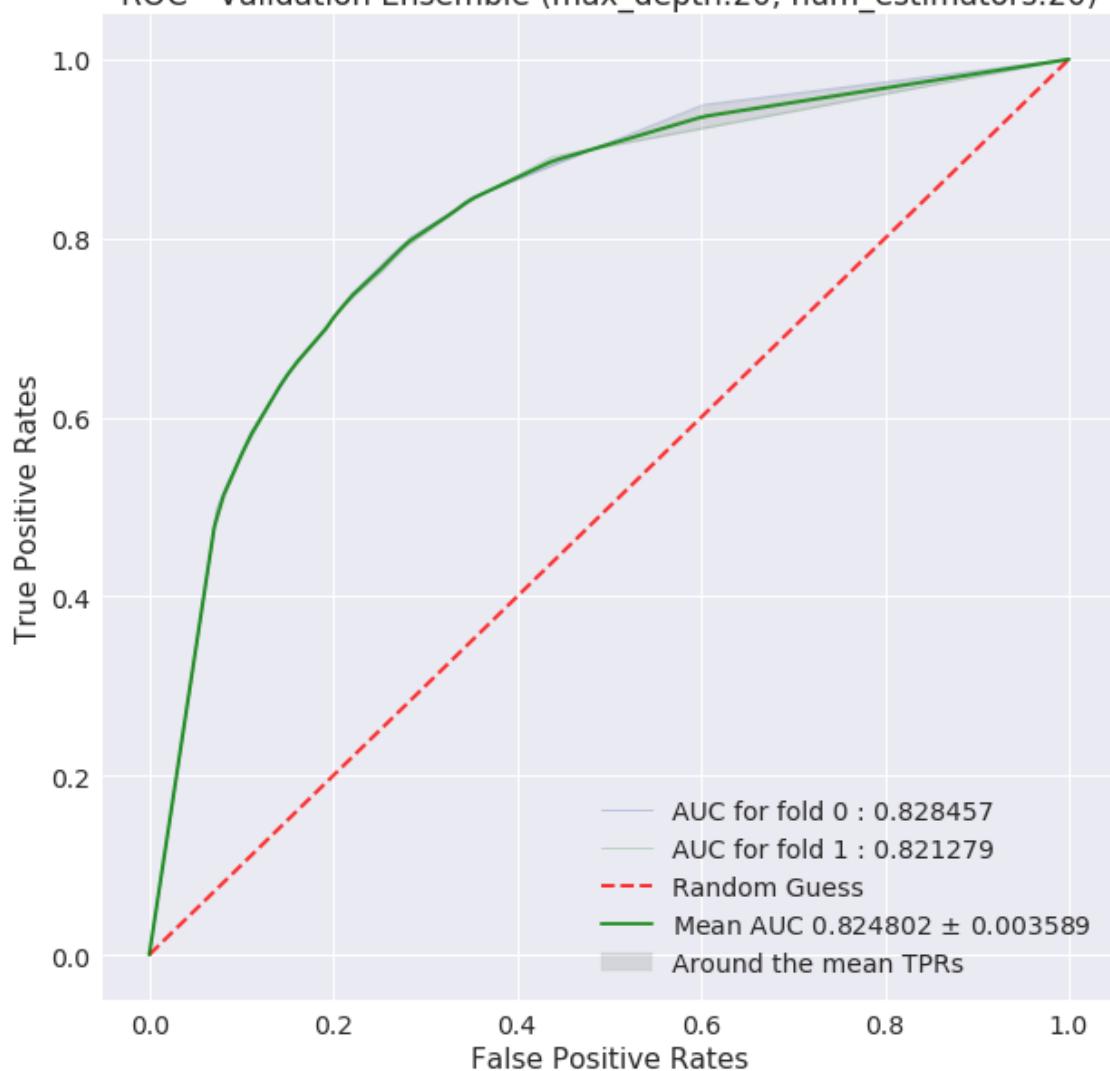
ROC - Validation Ensemble (max\_depth:10, num\_estimators:500)



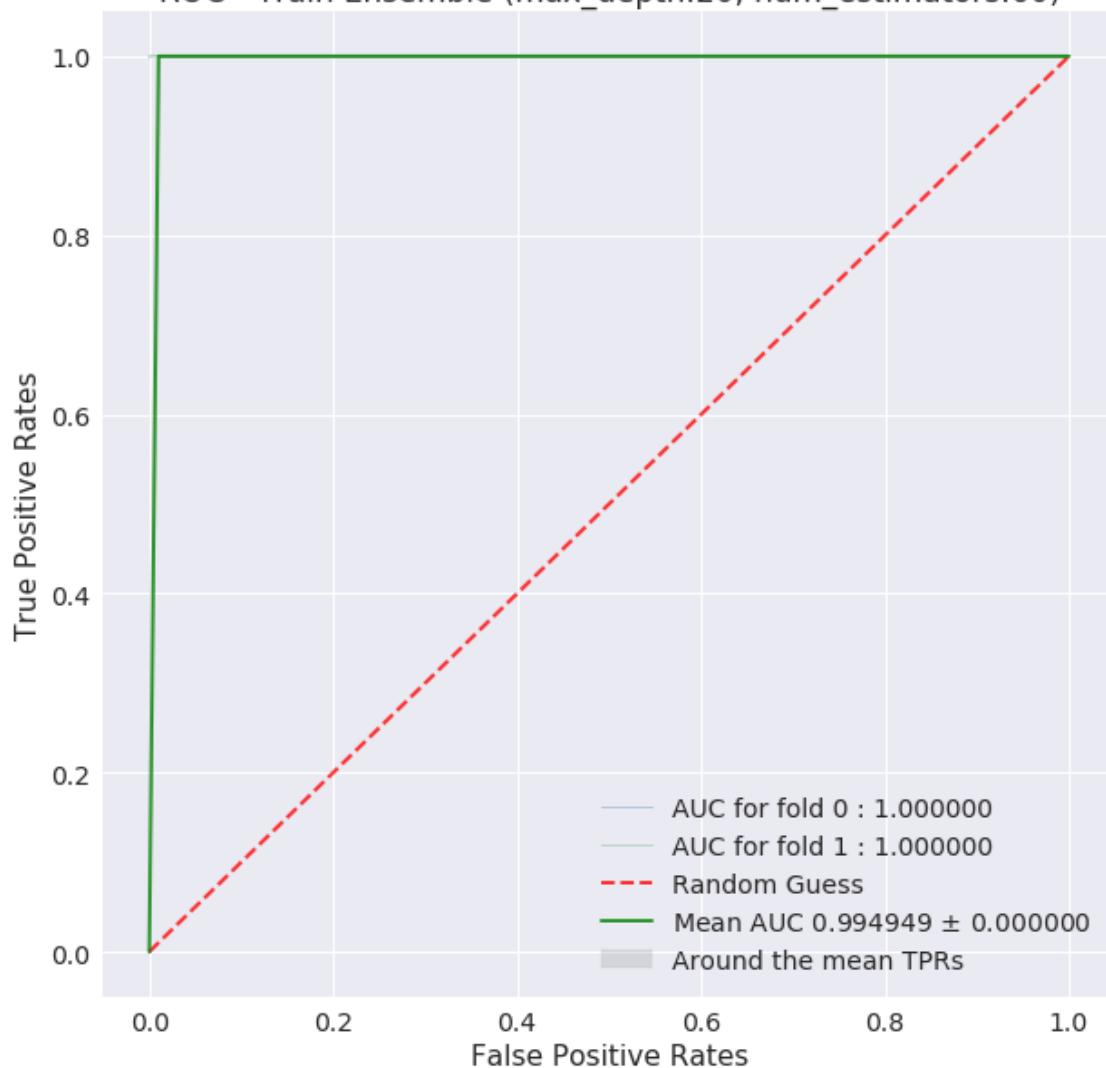
ROC - Train Ensemble (max\_depth:20, num\_estimators:20)



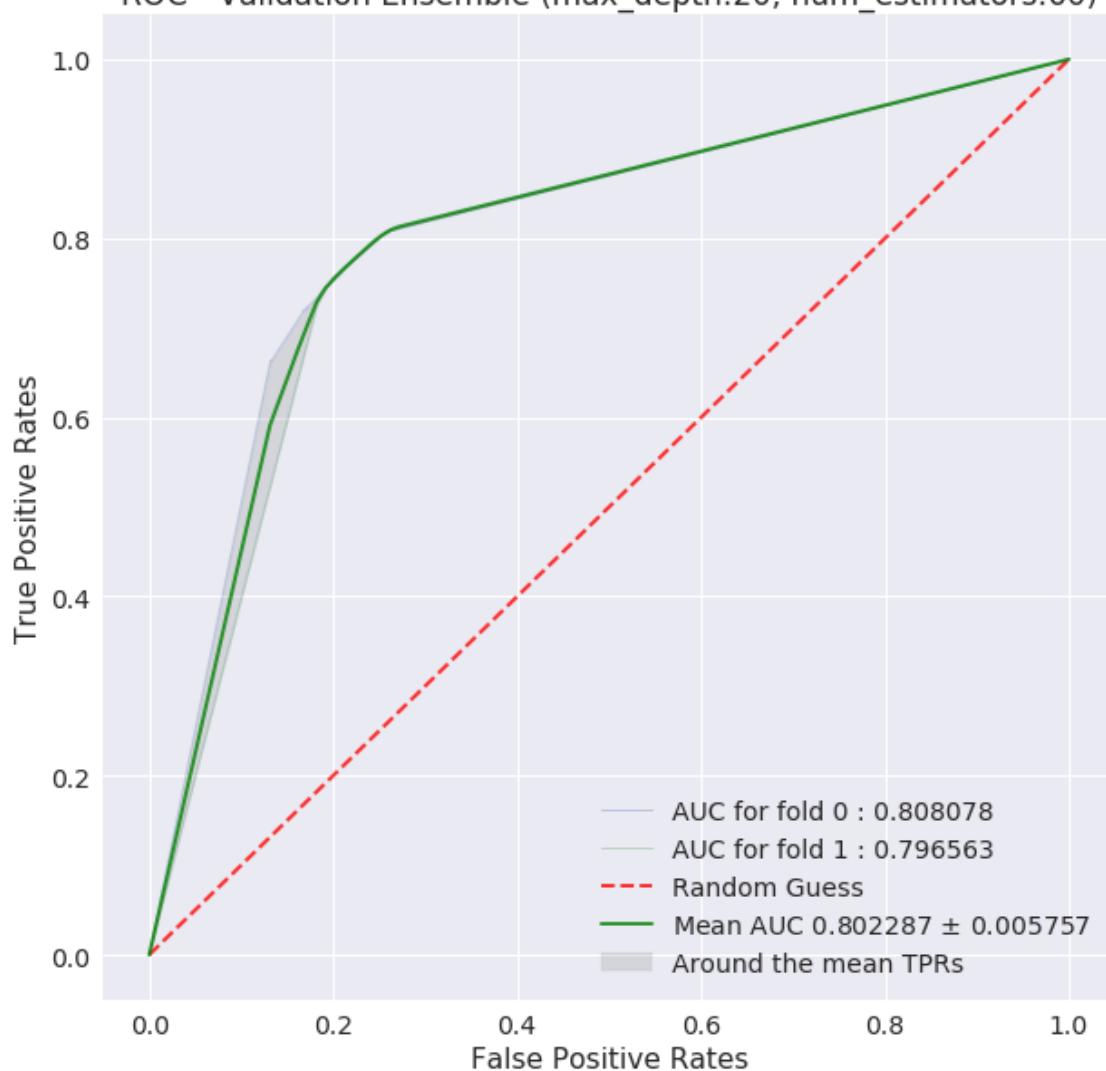
ROC - Validation Ensemble (max\_depth:20, num\_estimators:20)



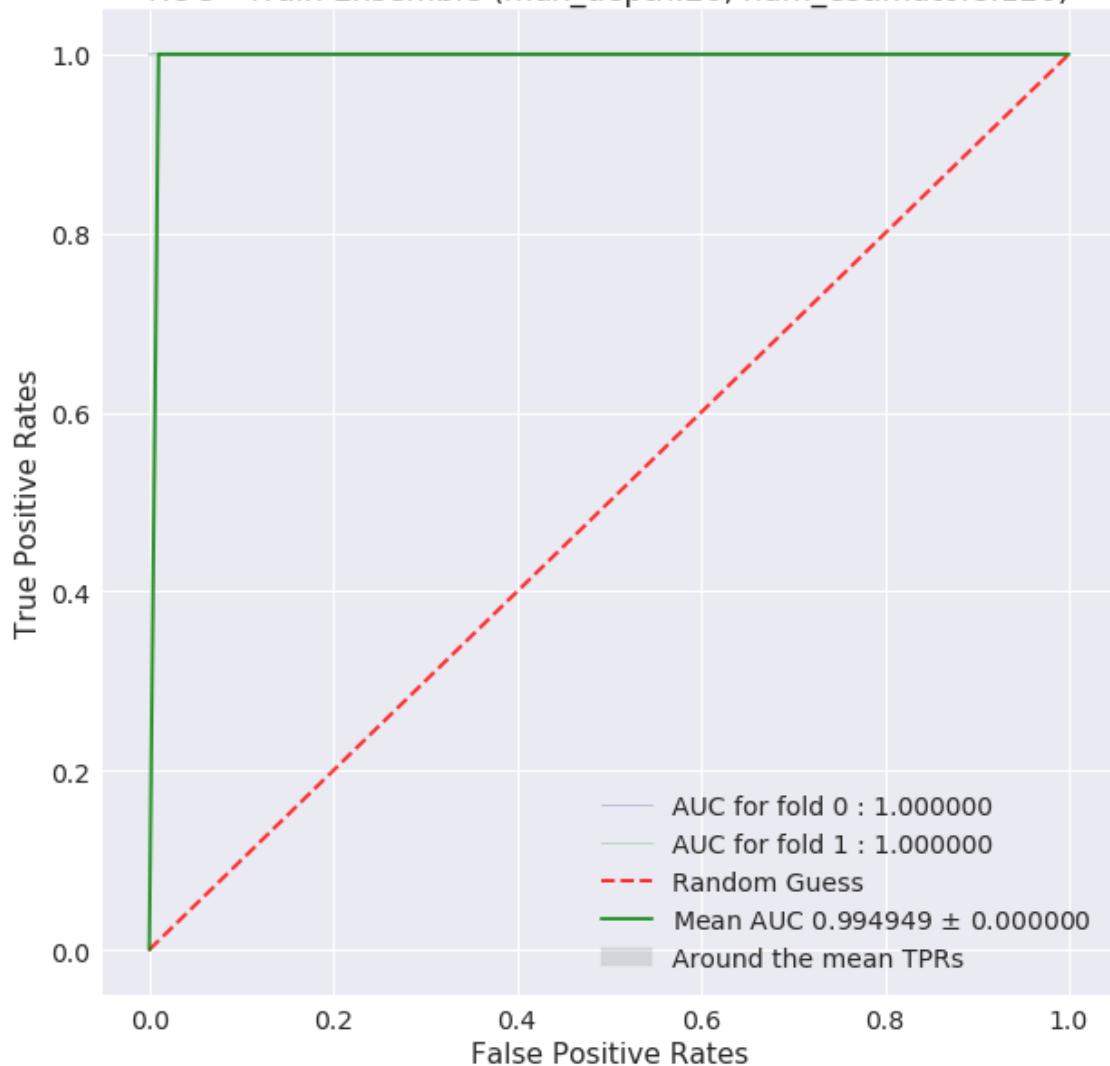
ROC - Train Ensemble (max\_depth:20, num\_estimators:60)



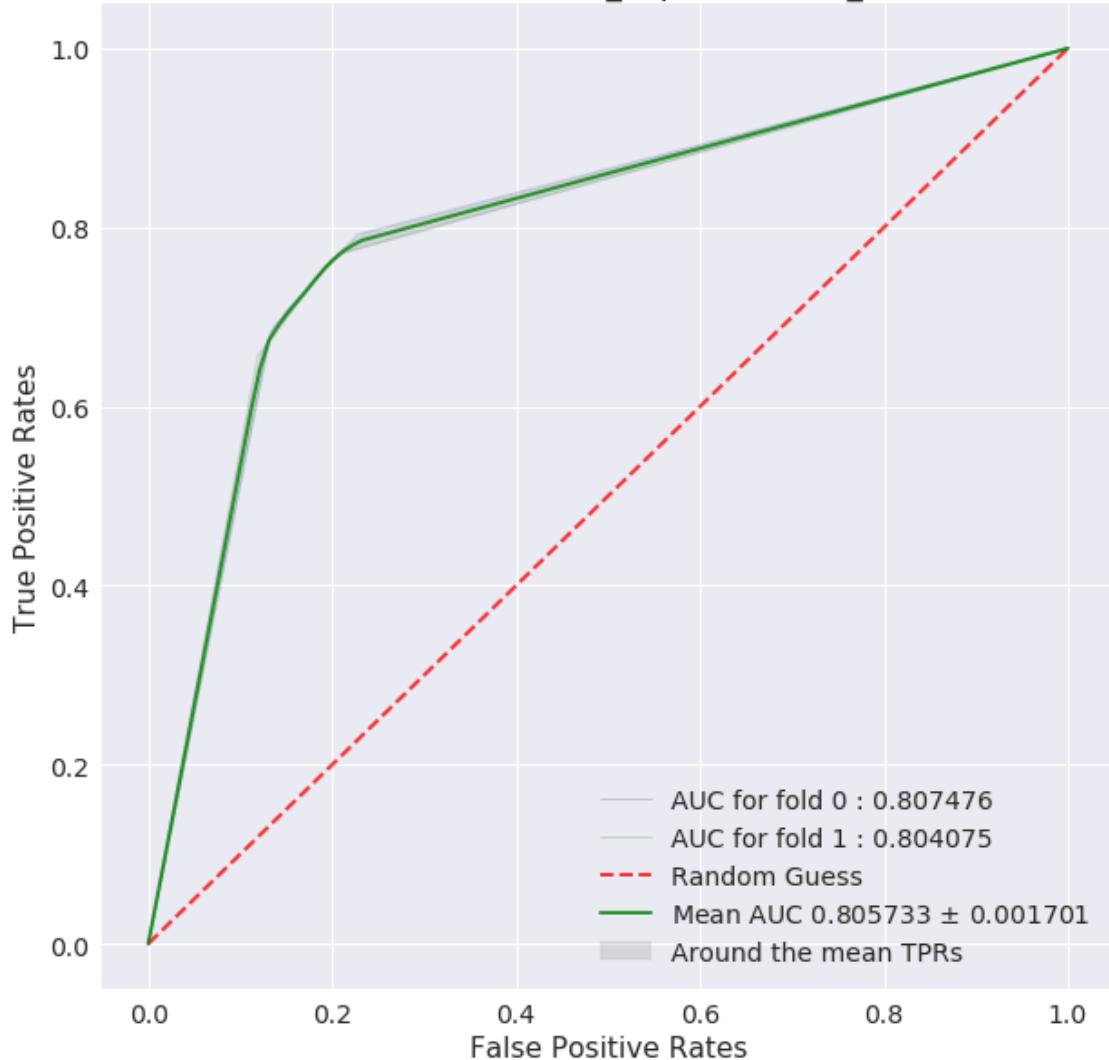
ROC - Validation Ensemble (max\_depth:20, num\_estimators:60)



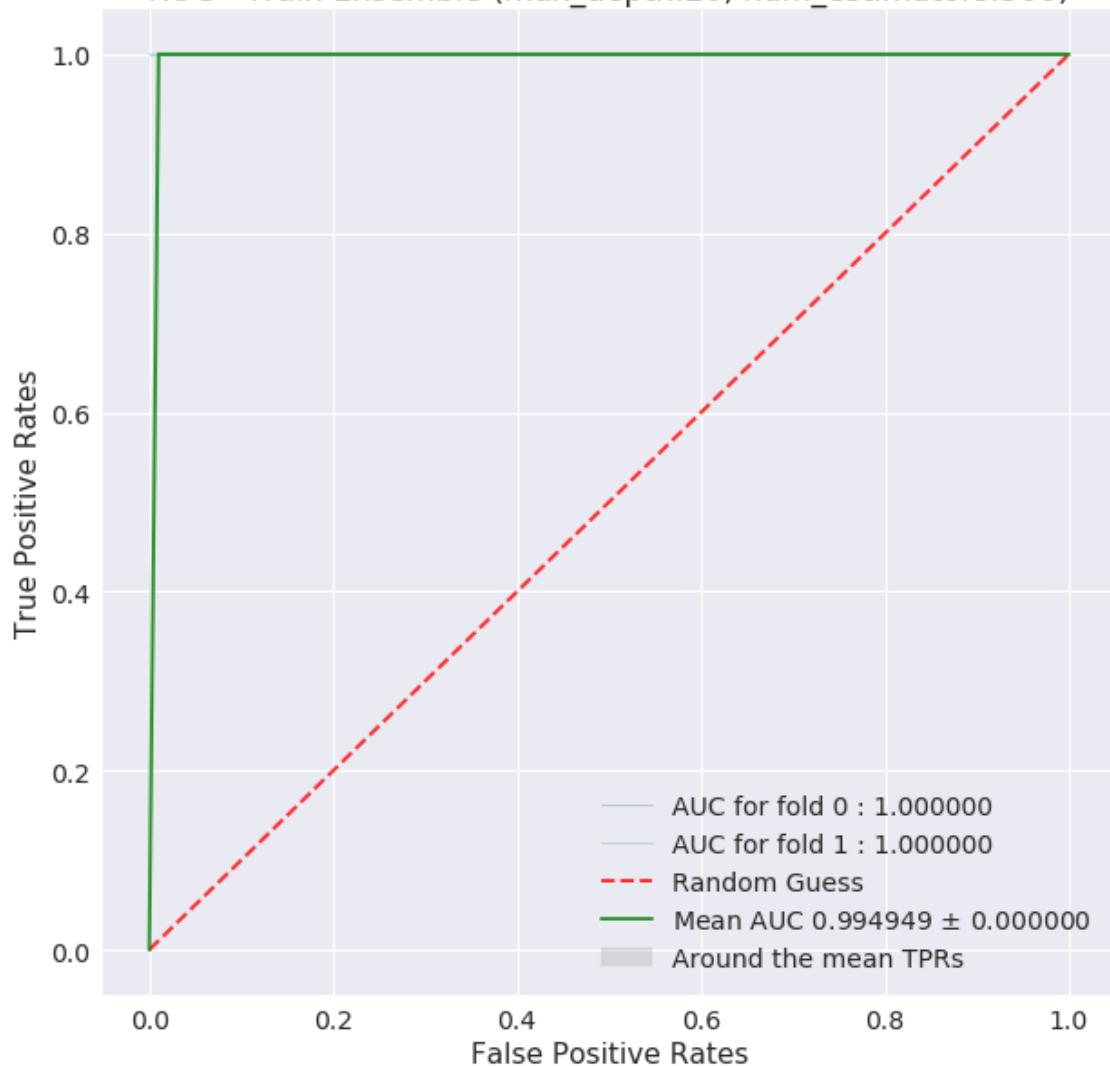
ROC - Train Ensemble (max\_depth:20, num\_estimators:120)



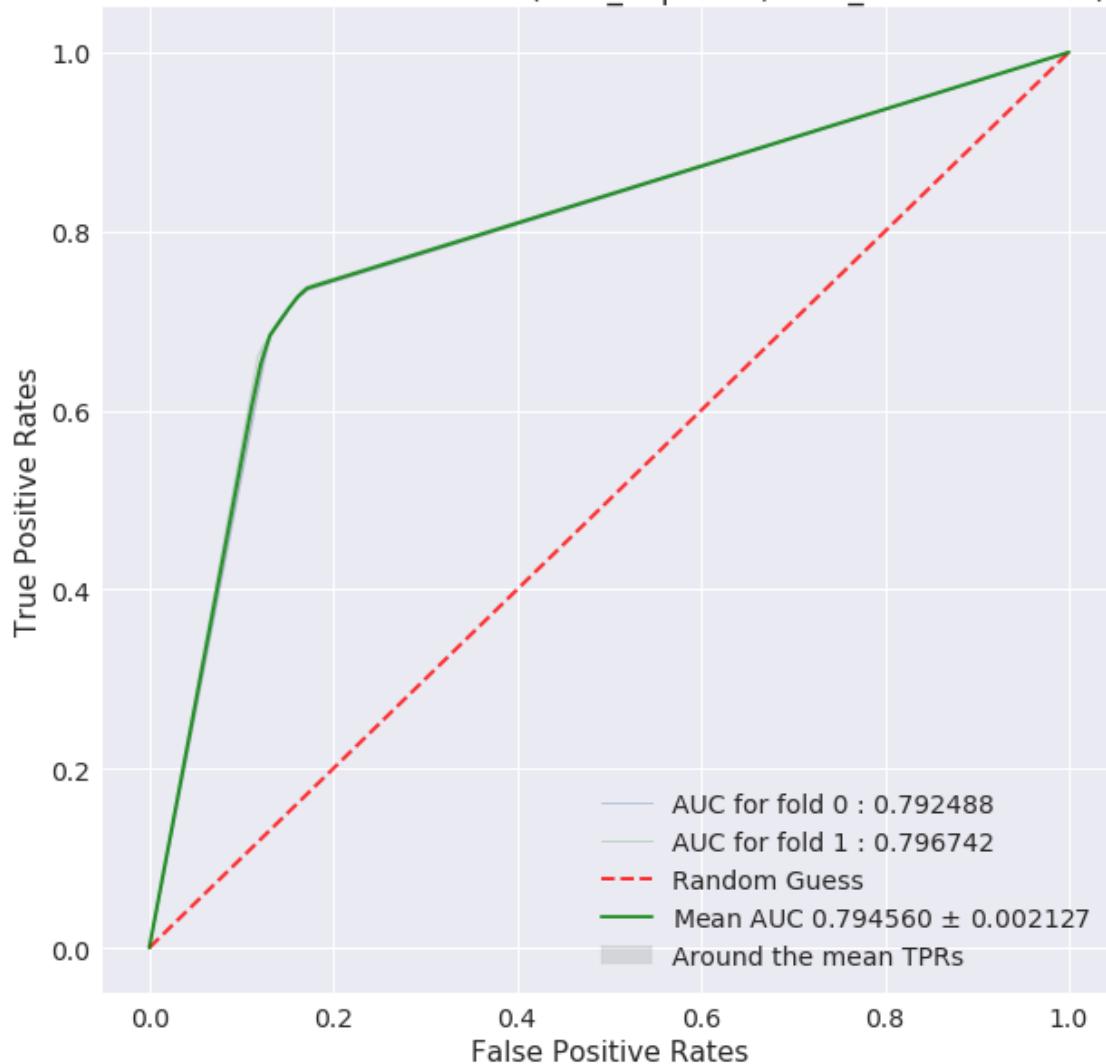
ROC - Validation Ensemble (max\_depth:20, num\_estimators:120)



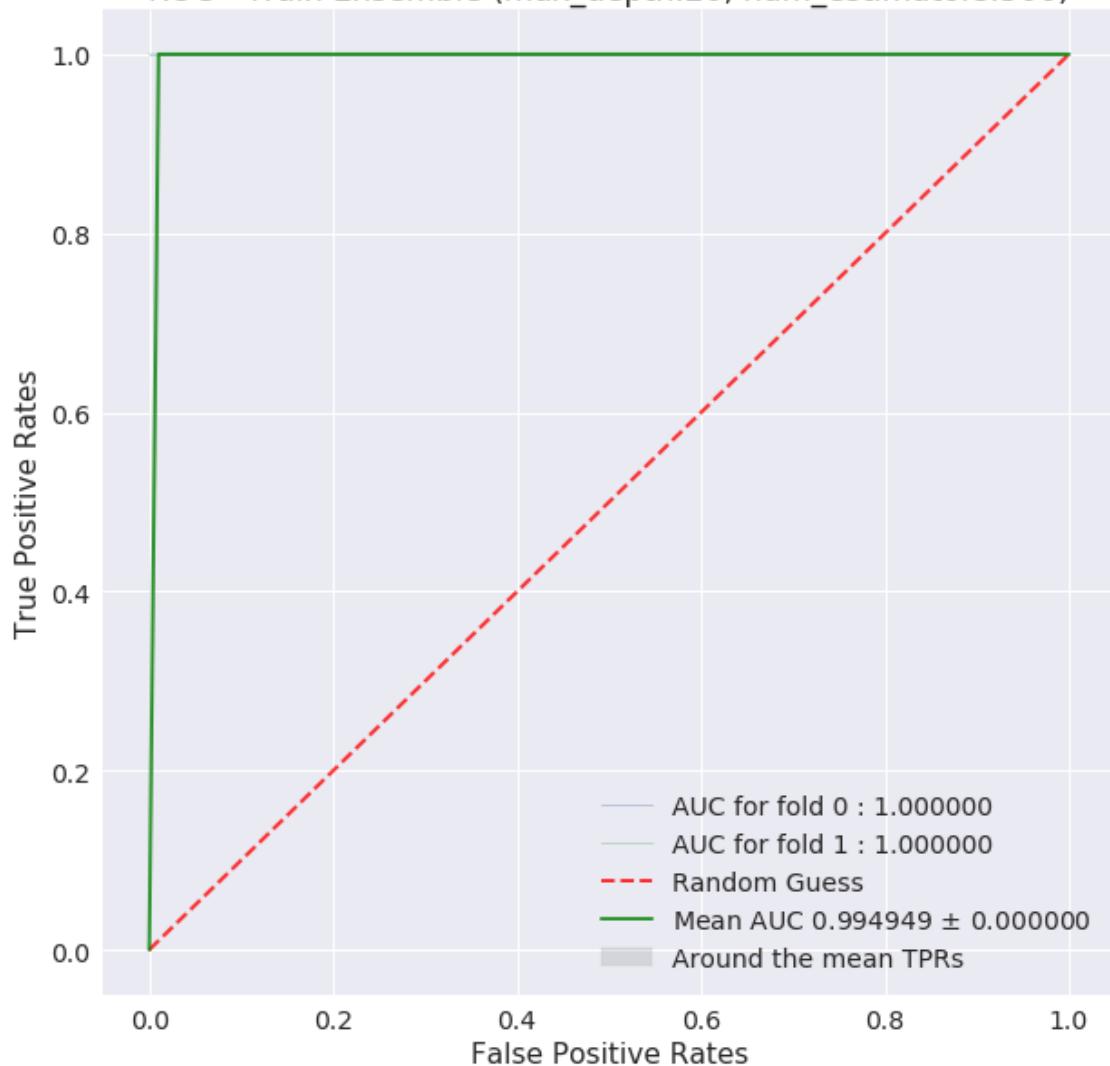
ROC - Train Ensemble (max\_depth:20, num\_estimators:300)



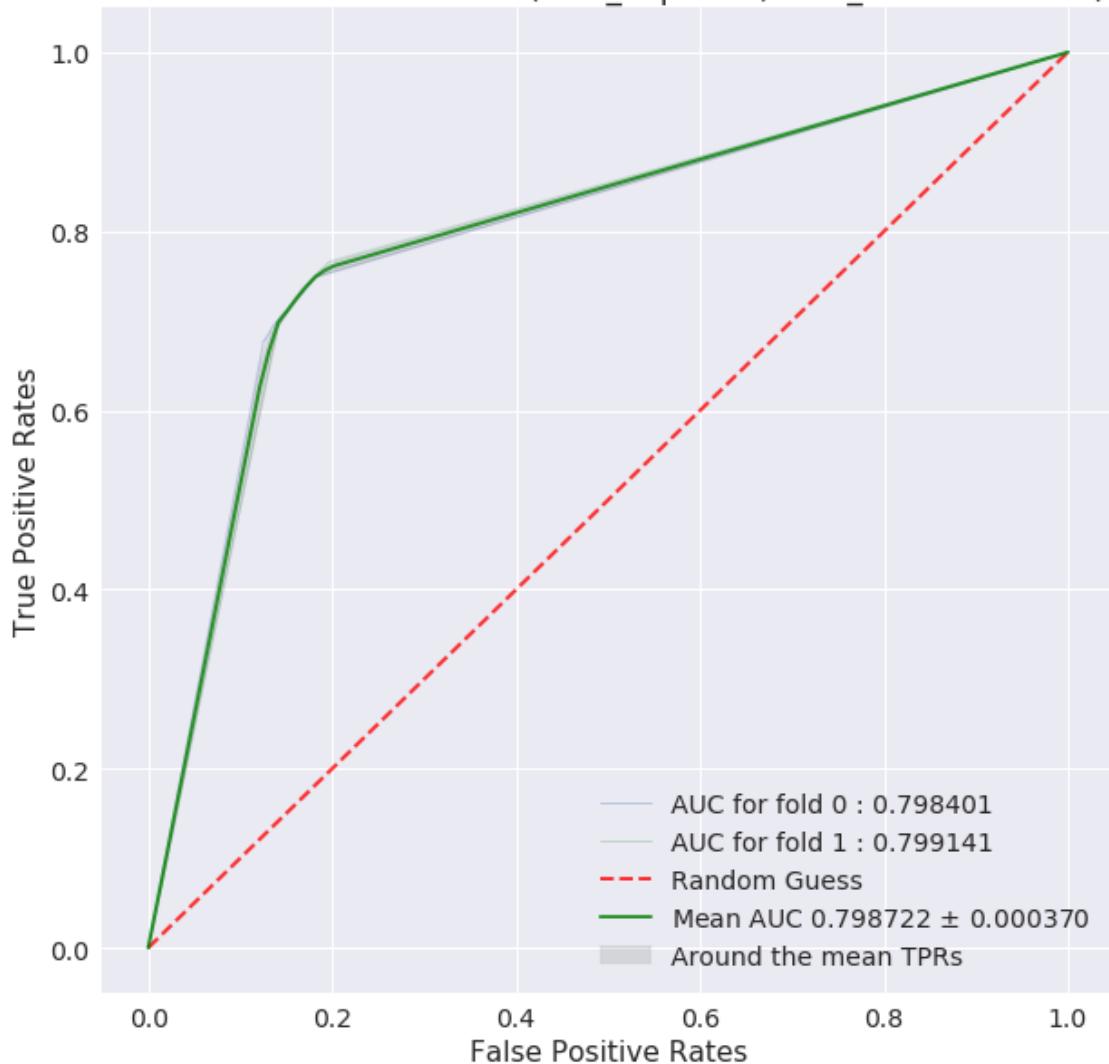
ROC - Validation Ensemble (max\_depth:20, num\_estimators:300)



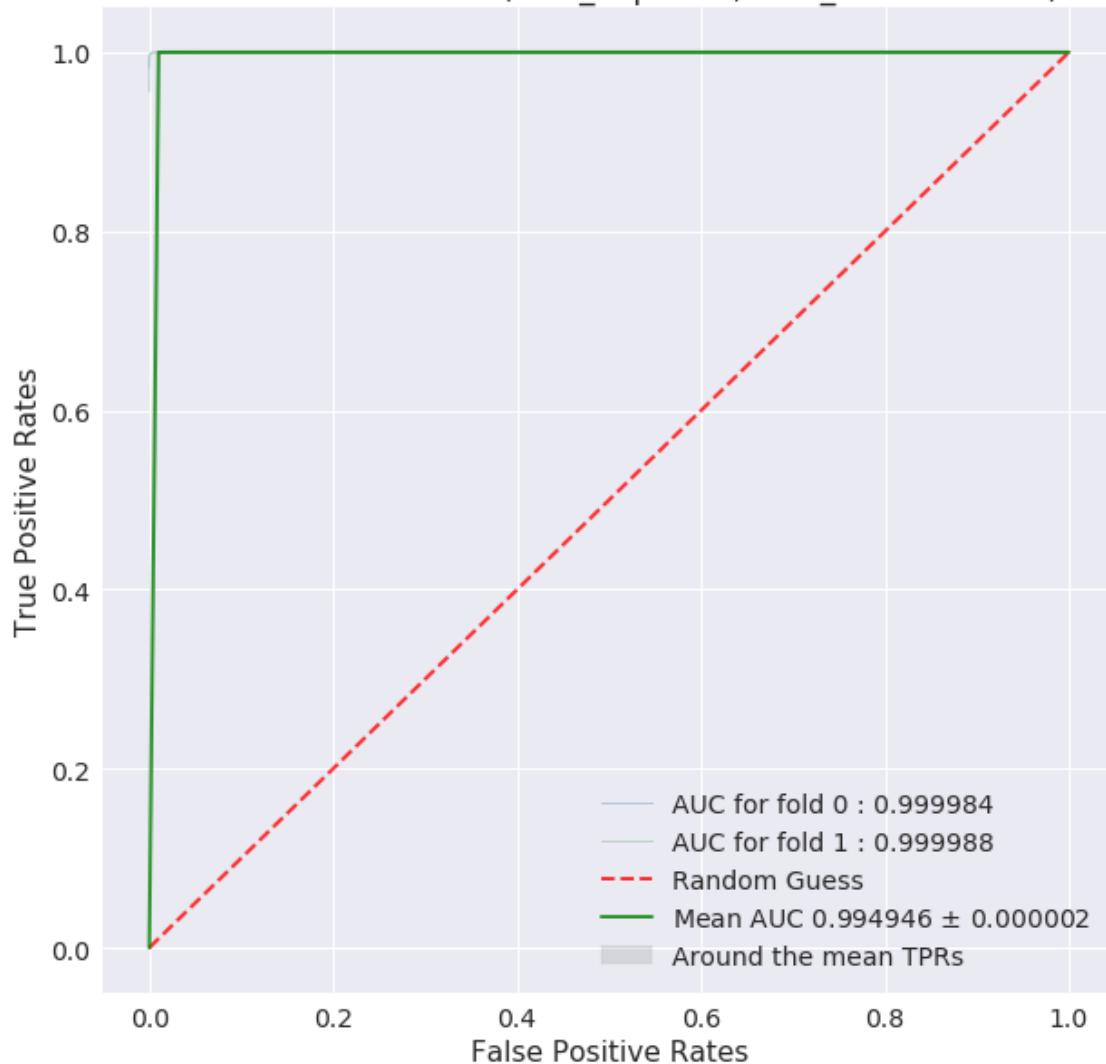
ROC - Train Ensemble (max\_depth:20, num\_estimators:500)



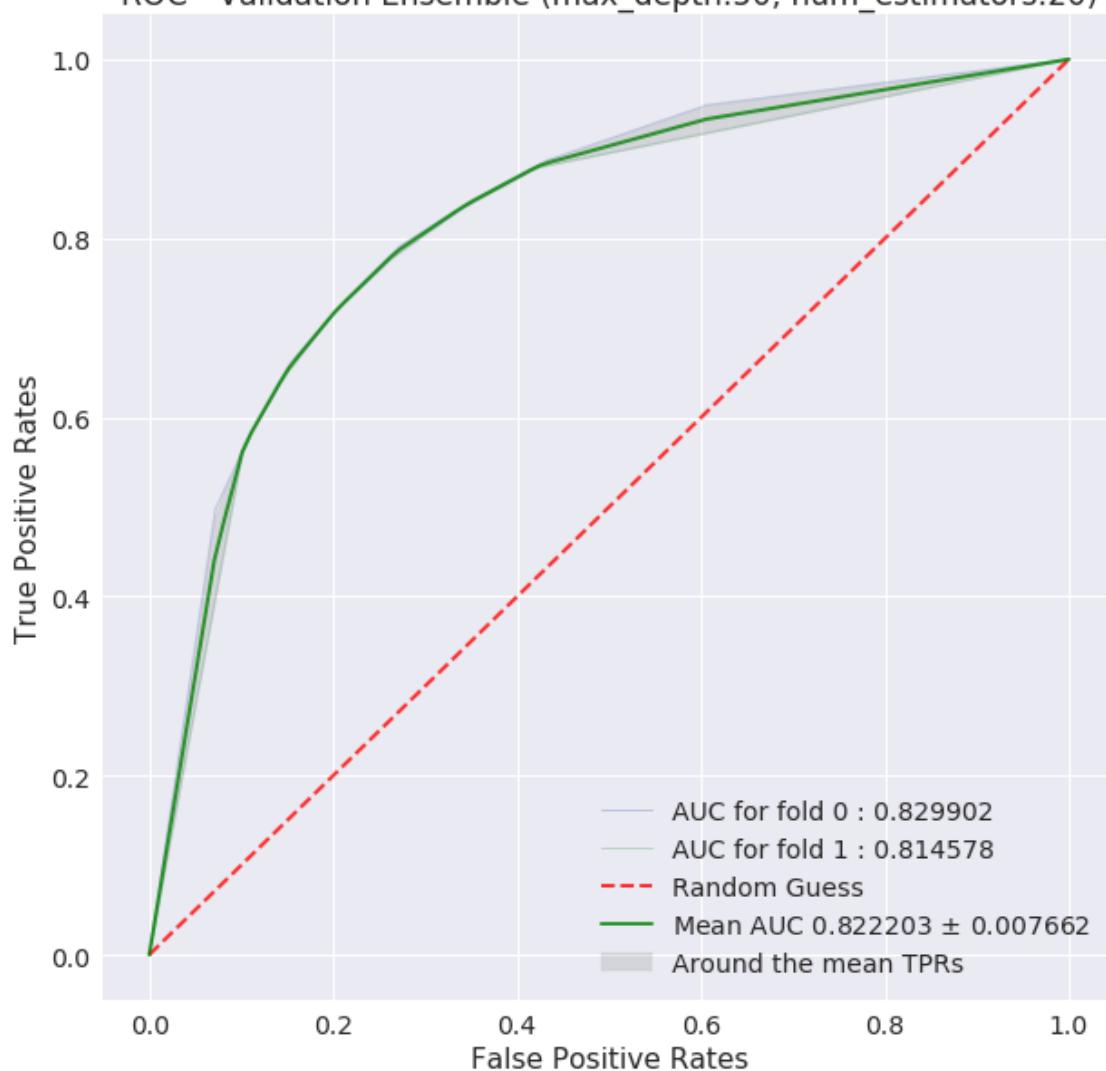
ROC - Validation Ensemble (max\_depth:20, num\_estimators:500)



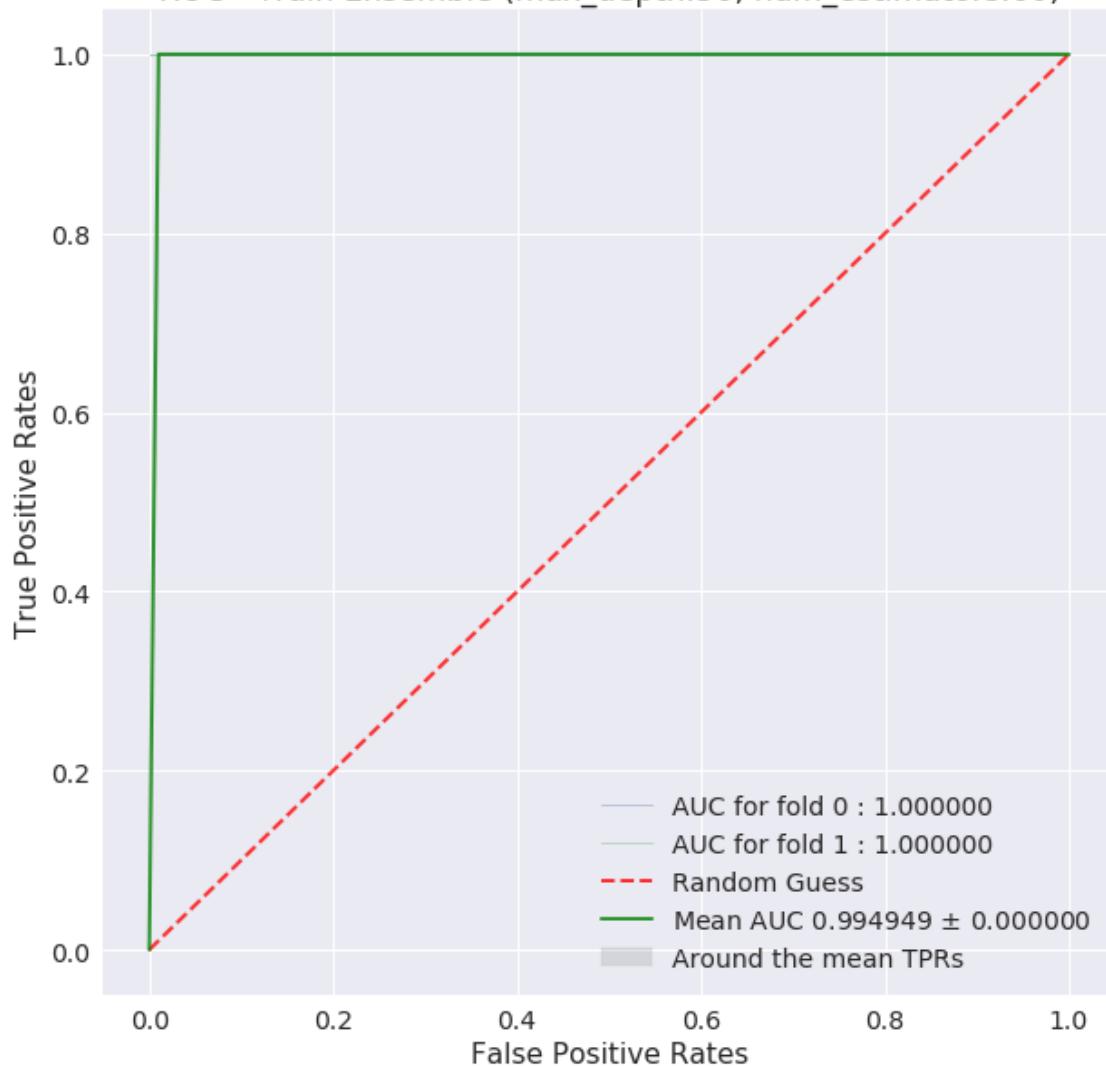
ROC - Train Ensemble (max\_depth:50, num\_estimators:20)



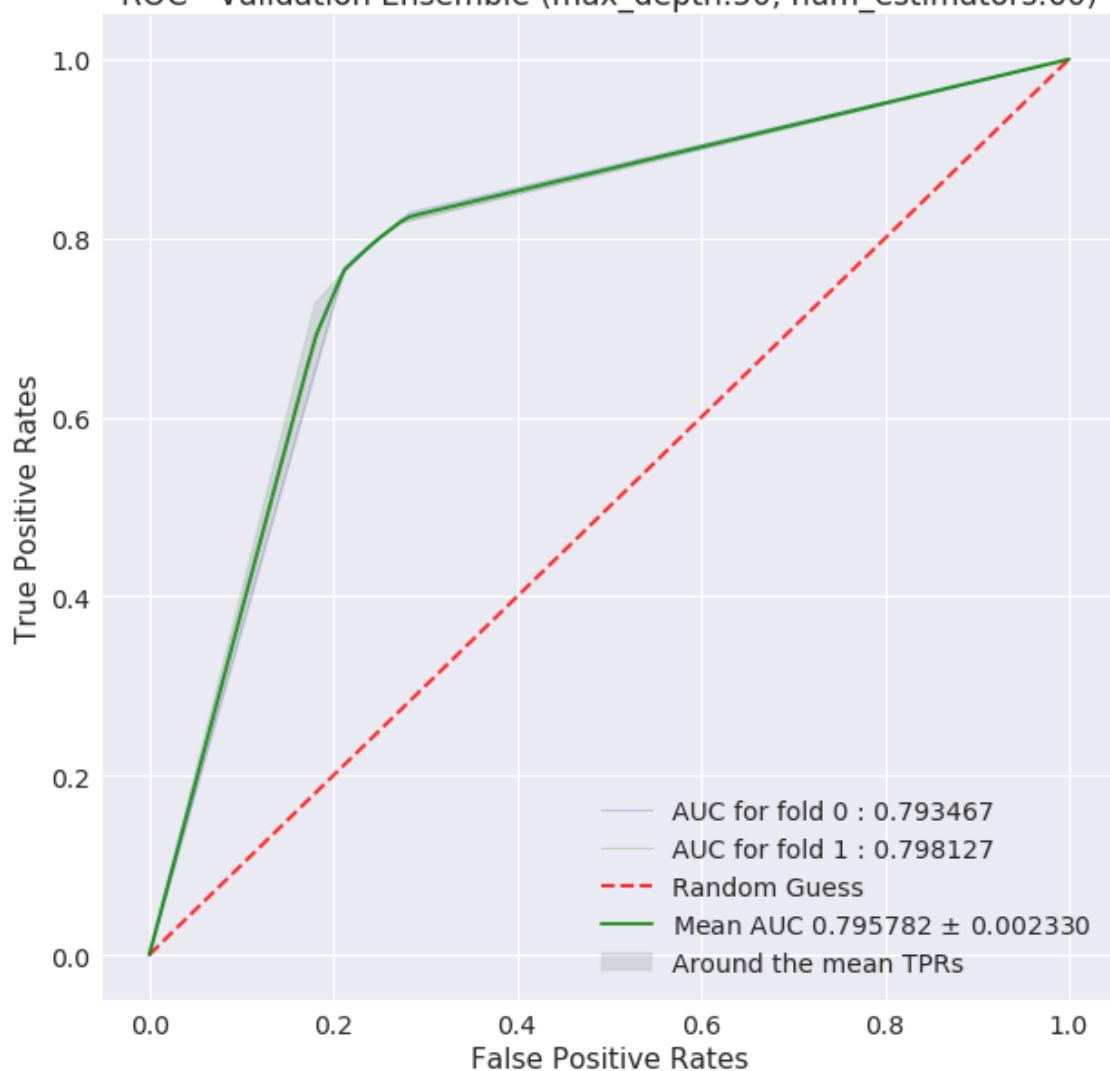
ROC - Validation Ensemble (max\_depth:50, num\_estimators:20)



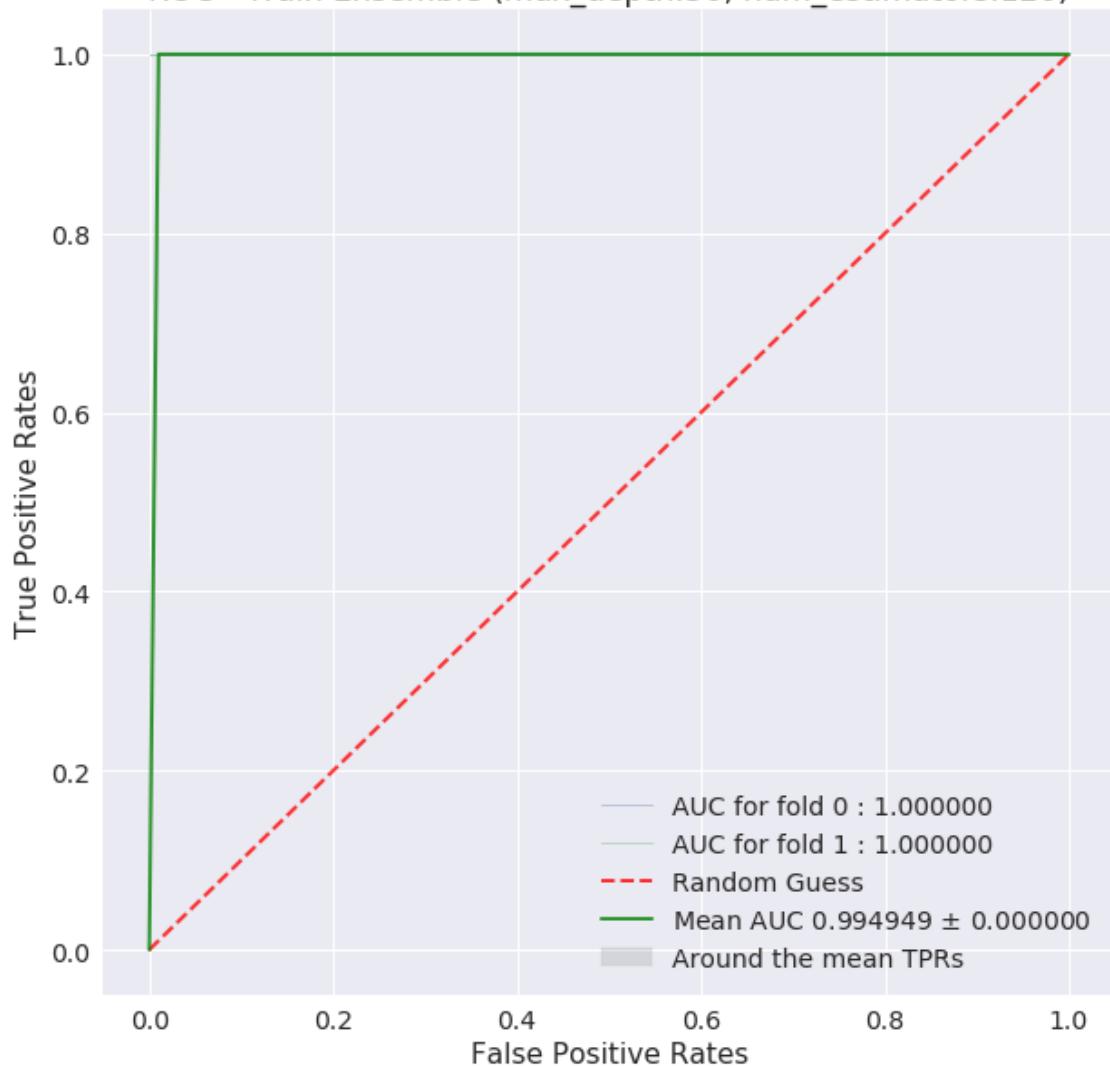
ROC - Train Ensemble (max\_depth:50, num\_estimators:60)



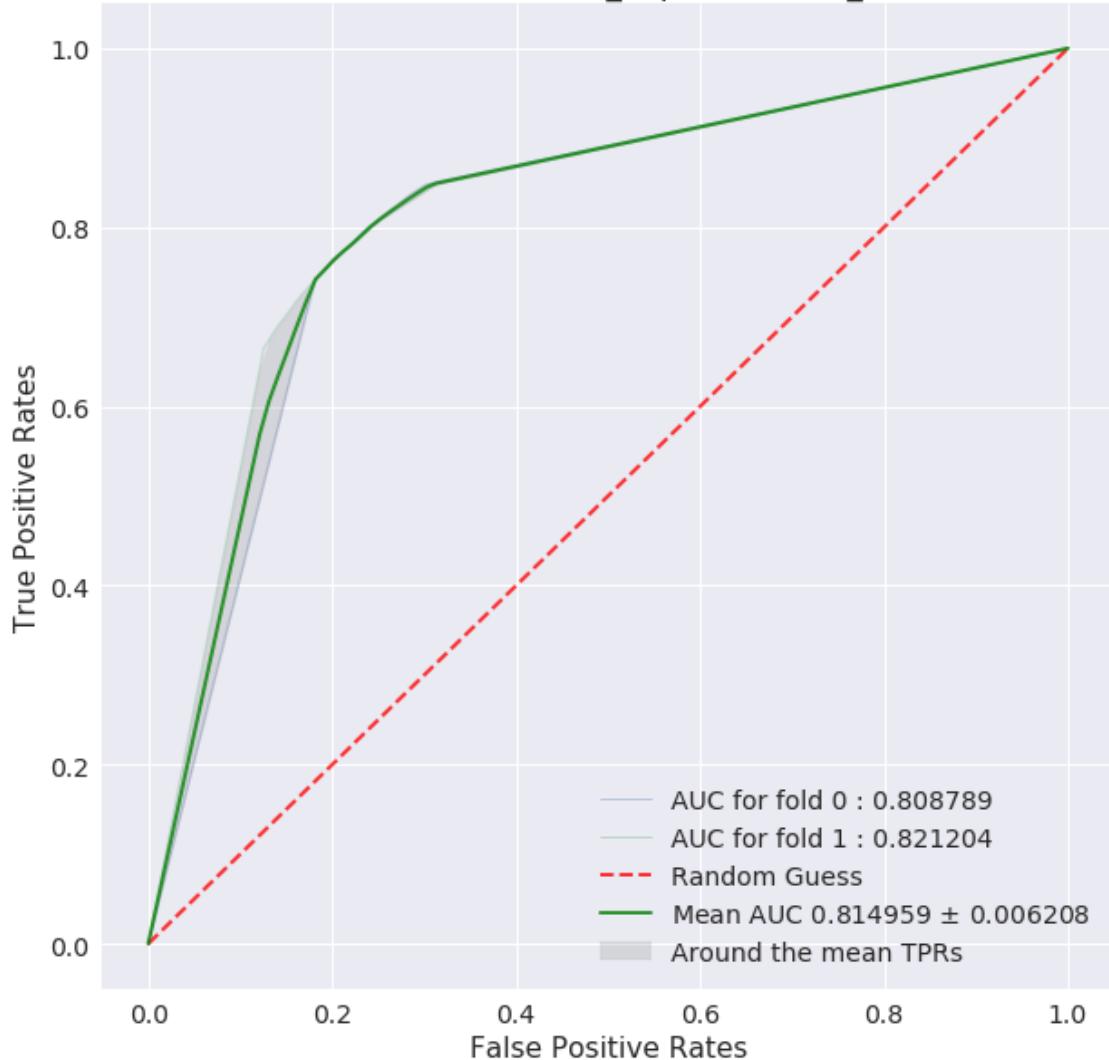
ROC - Validation Ensemble (max\_depth:50, num\_estimators:60)



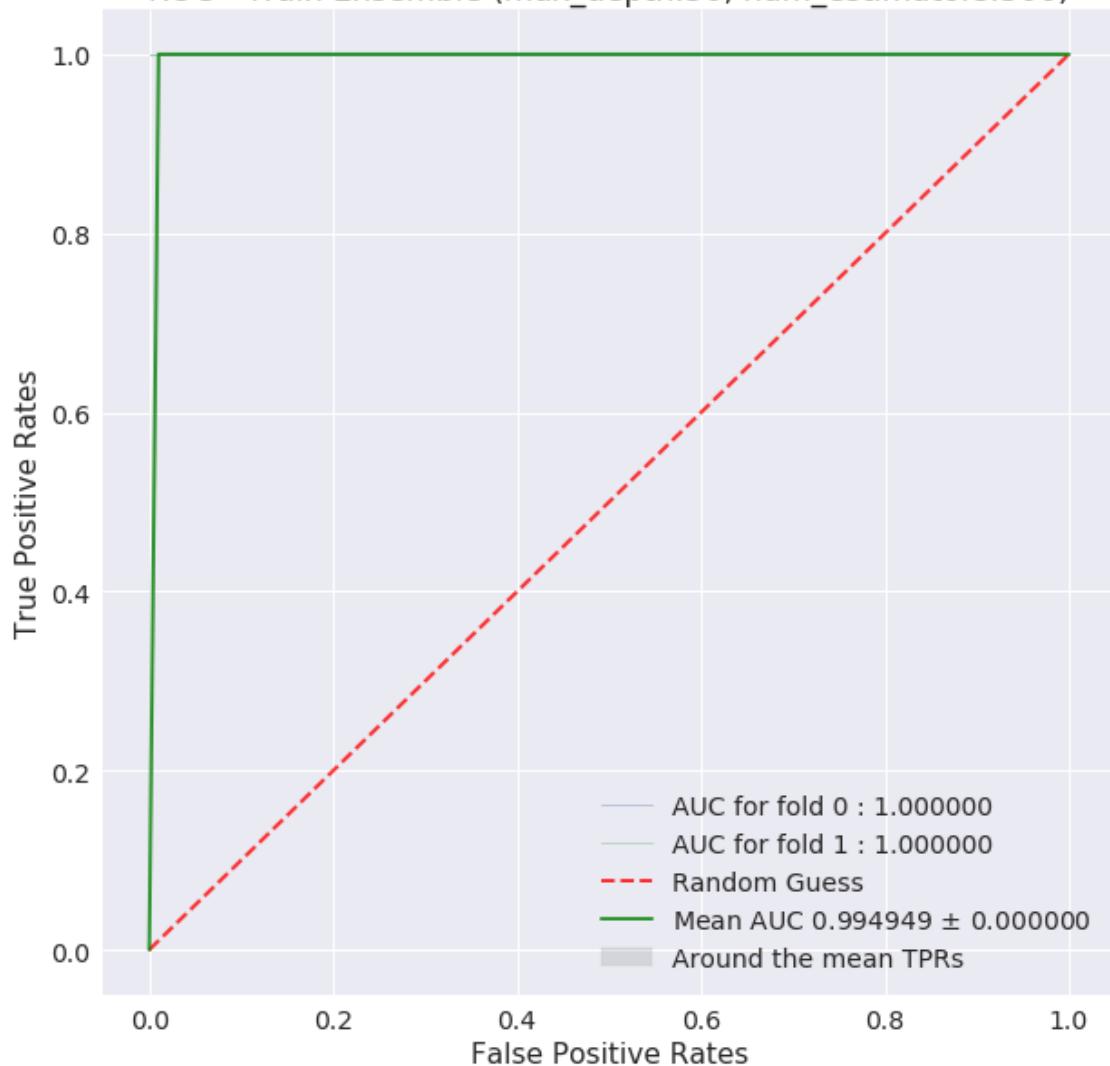
ROC - Train Ensemble (max\_depth:50, num\_estimators:120)



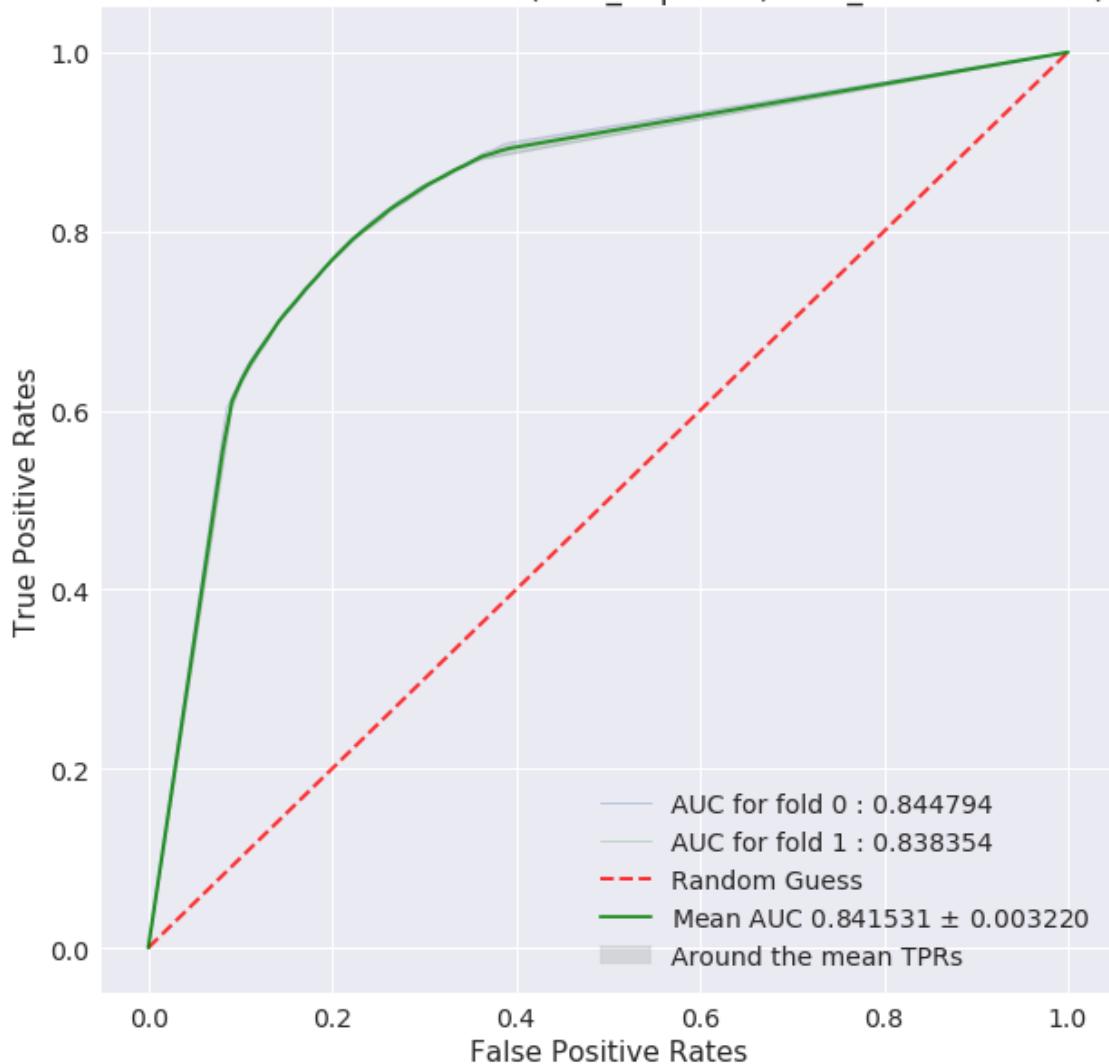
ROC - Validation Ensemble (max\_depth:50, num\_estimators:120)



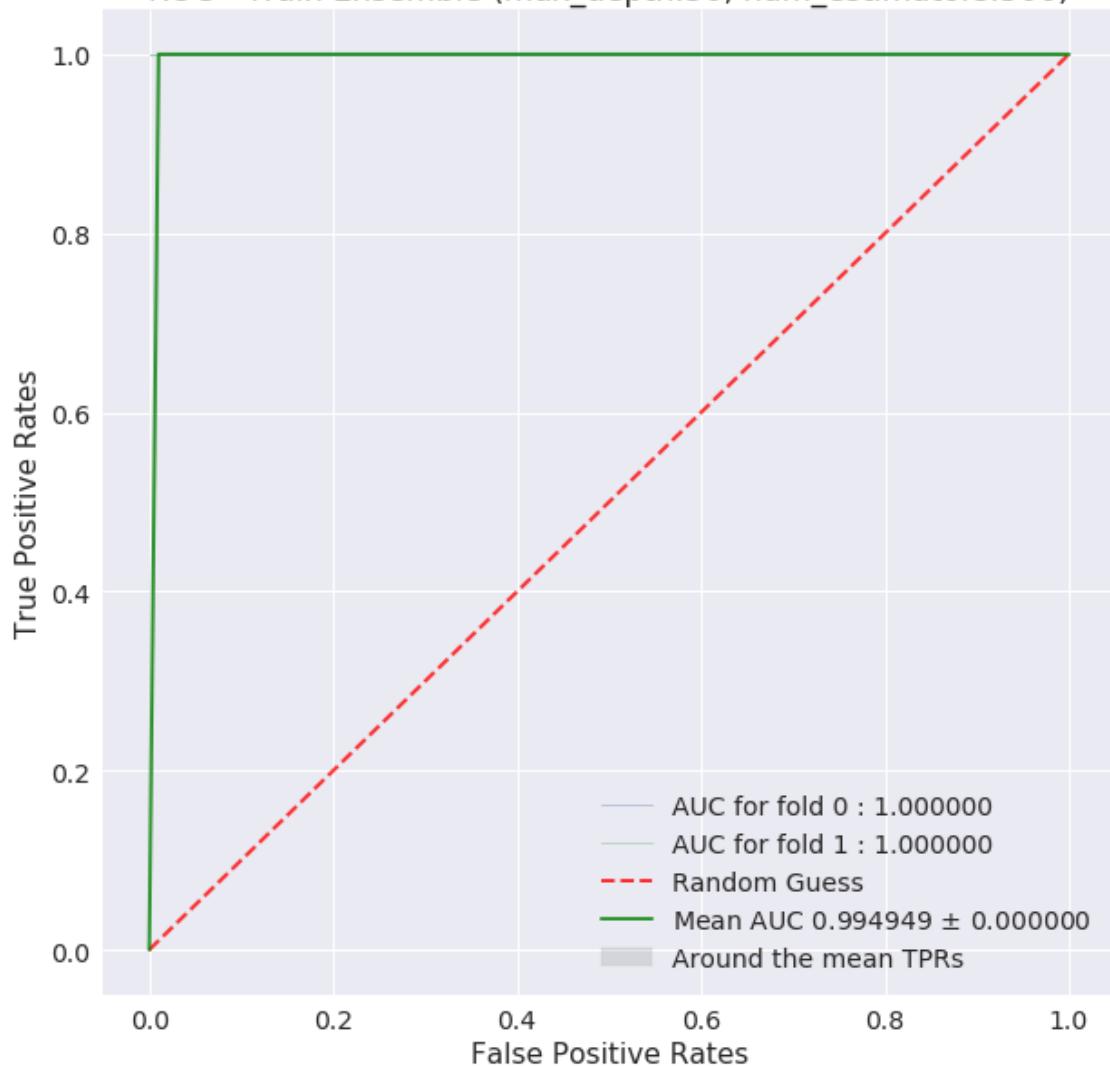
ROC - Train Ensemble (max\_depth:50, num\_estimators:300)



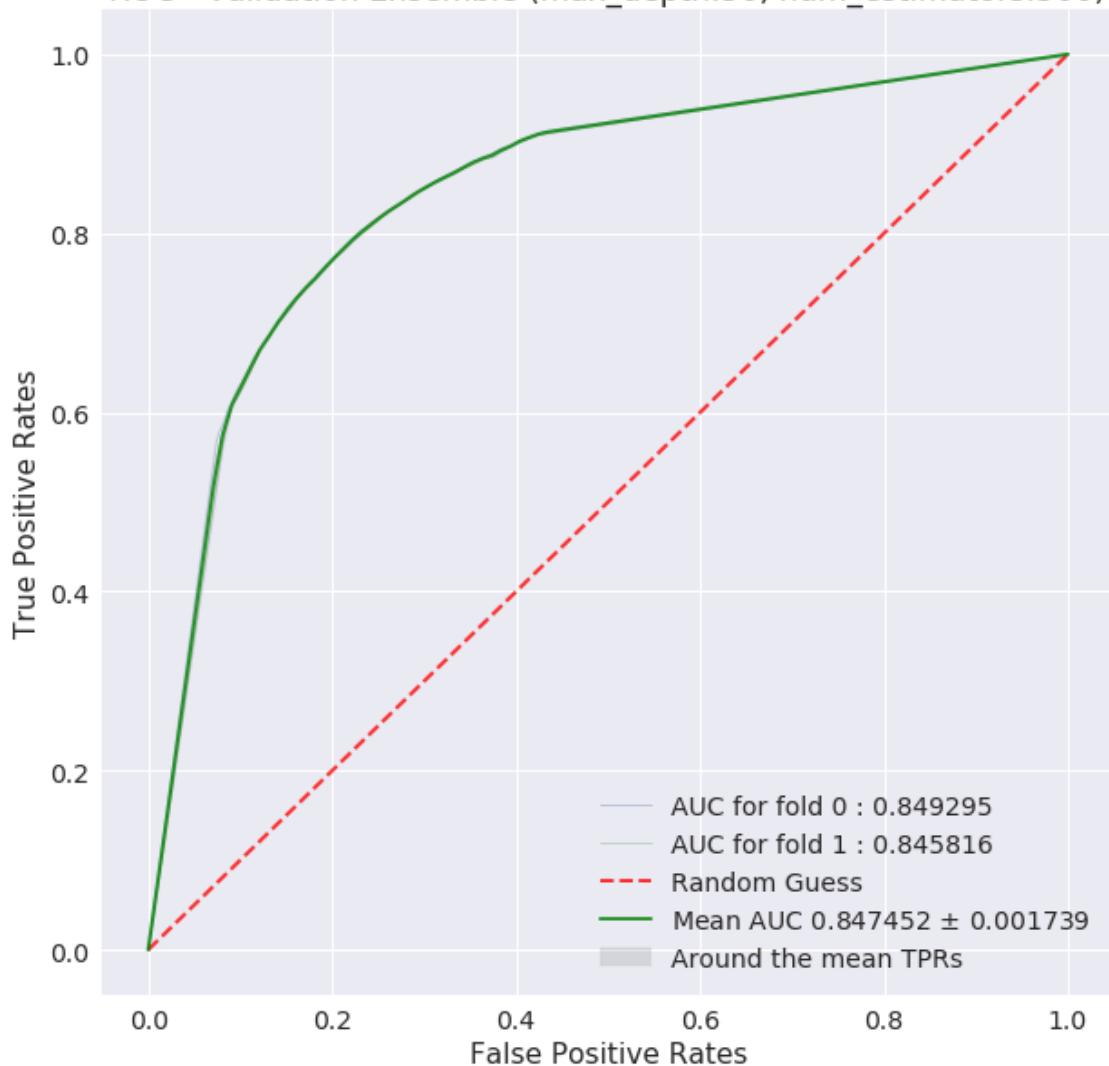
ROC - Validation Ensemble (max\_depth:50, num\_estimators:300)



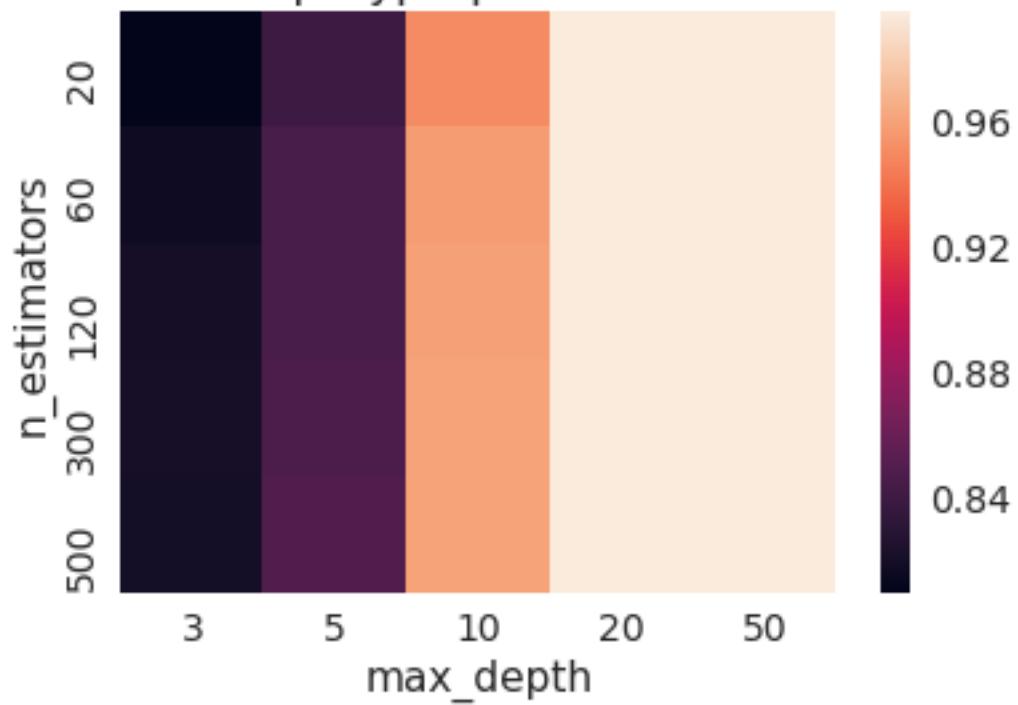
ROC - Train Ensemble (max\_depth:50, num\_estimators:500)



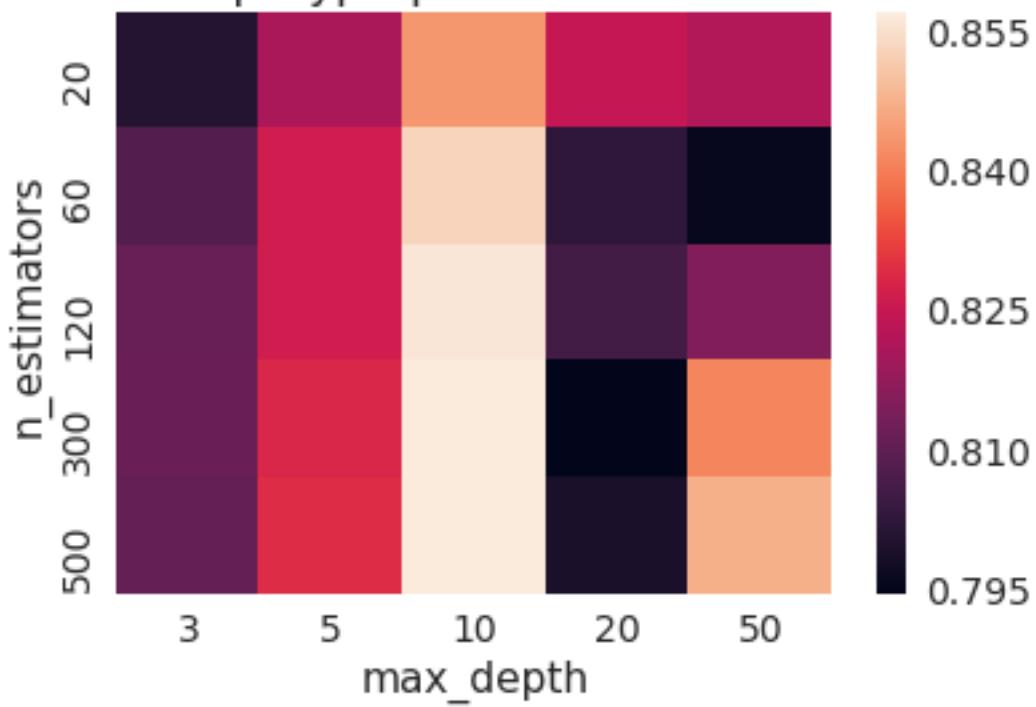
ROC - Validation Ensemble (max\_depth:50, num\_estimators:500)



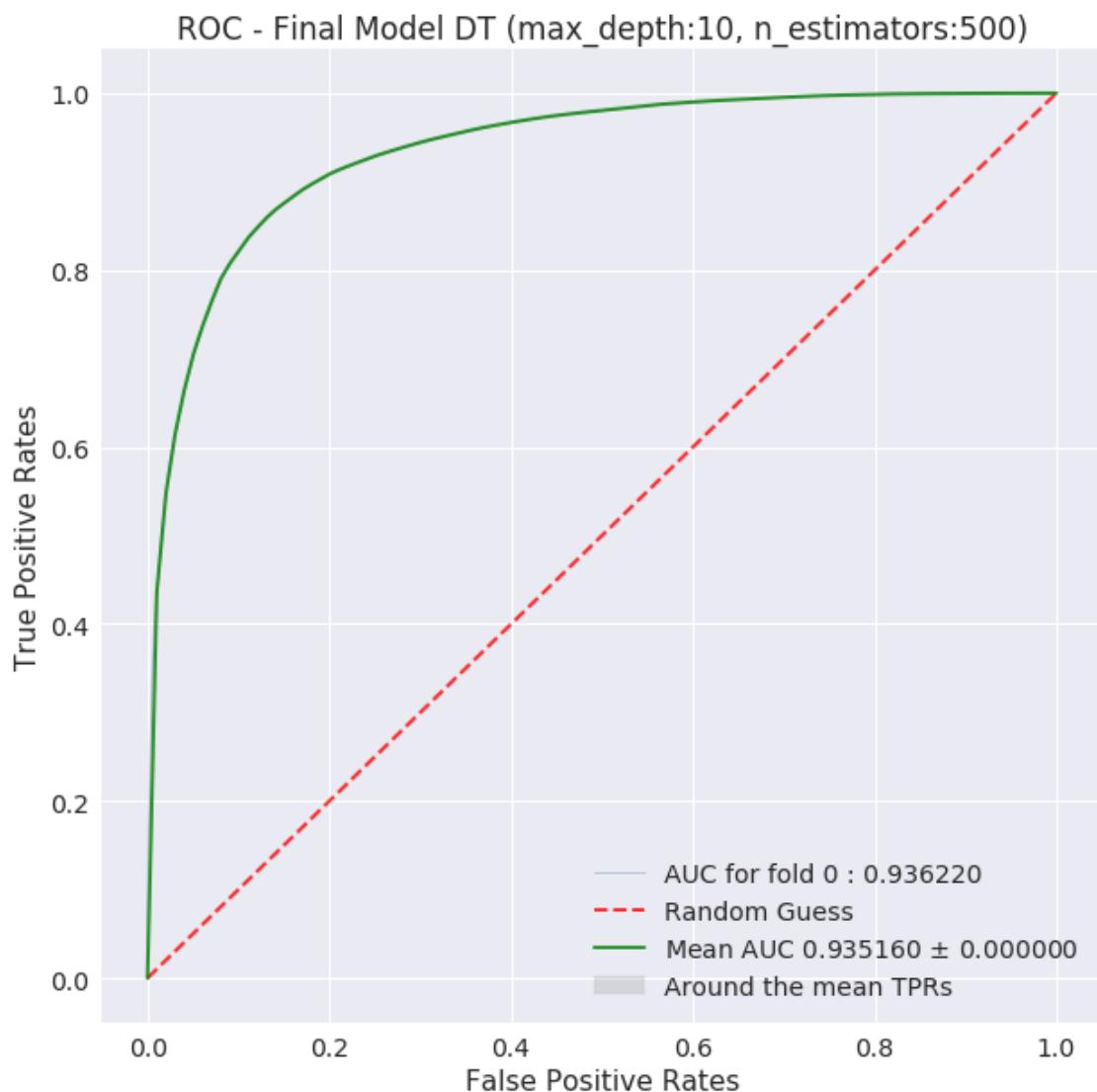
Heatmap Hyperparams for Train

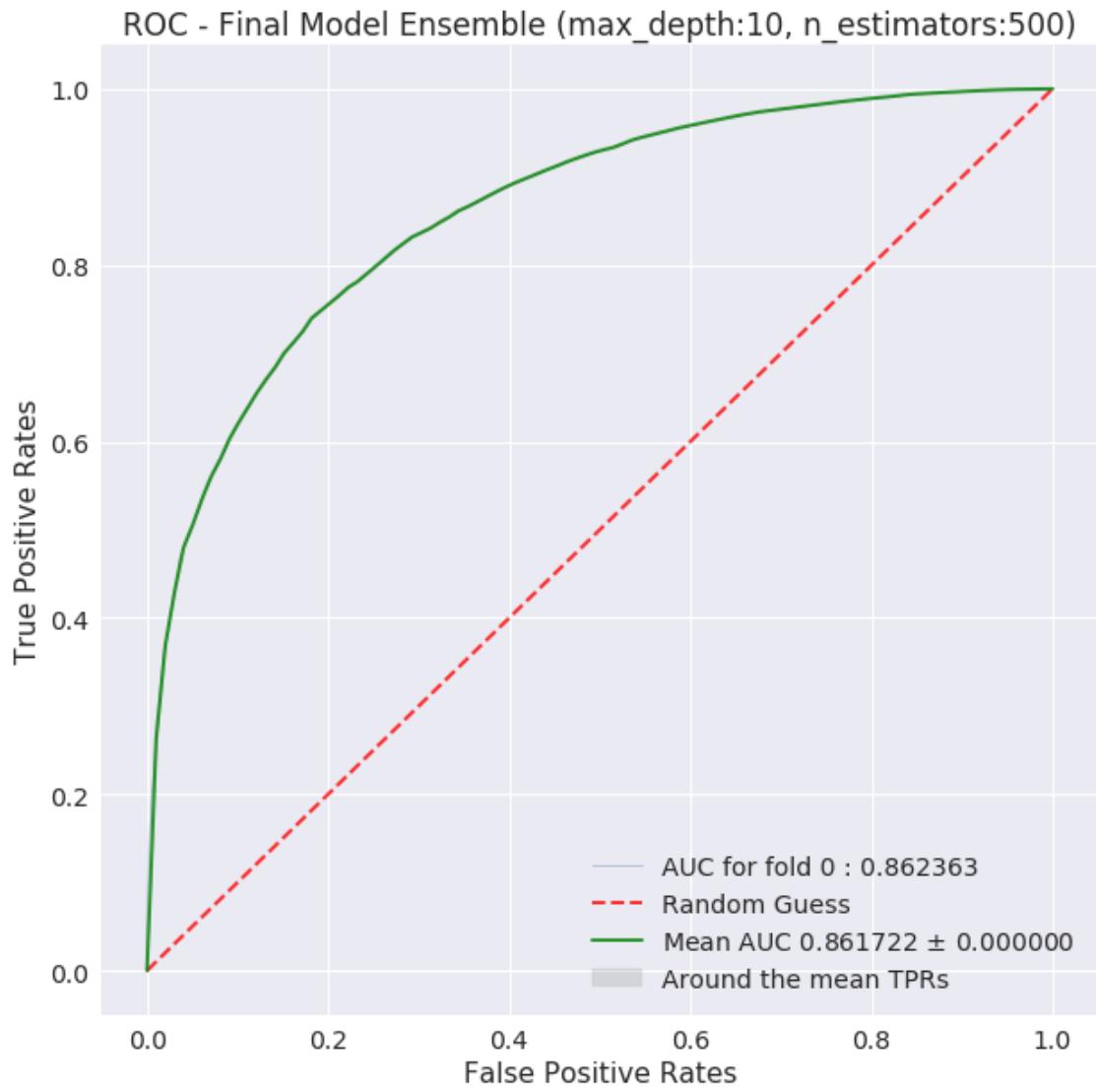


Heatmap Hyperparams for Validation

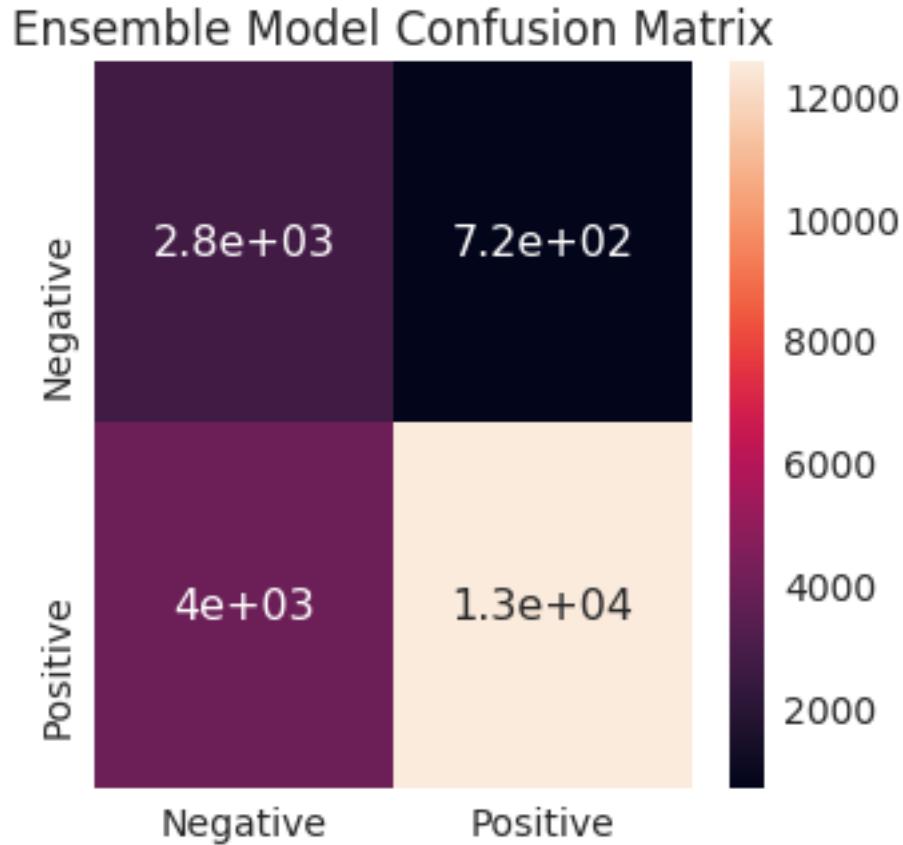


Best hyperparam value: (10, 500)





Test auc score 0.8617215981013538



	Negative	Positive
Precision	0.411310	0.946084
Recall	0.794253	0.760533
Fscore	0.541961	0.843221
Support	3480.000000	16520.000000

best hyper param identified is max\_depth = 10, and n\_estimators=500

## 4.5 [B] Applying GBDT using XGBOOST

```
In [21]: # In this work, Hyper parameter tuning is done only on the max_depth and n_estimators parameters
# restricting to two are, adding more parameters will increase the training time exponentially
# large dataset size. The other parameter values are left to its default values.
```

### 4.5.1 [B.1] Applying XGBOOST on BOW, SET 1

```
In [17]: # form two lists
depth_list = [2, 5, 20, 50] # depends on size of dataset
n_estimators_list = [40, 70, 120, 200] # depends on size of dataset
```

```

learning_rate_list = [0.1] # learning rate for XGB training

# create a configuartion dictionary
config_dict = {
    'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVW/BOW/training.csv',
    'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVW/BOW/test.csv',
    'train_size' : 40000,
    'test_size' : 15000,
    'hyperparam_list' : list(product(depth_list, n_estimators_list, learning_rate_list)),
    'implementation': 'xgb' # 'xgb' or 'rf'
}

In [18]: # read the train, test data and preprocess it
train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                                                      scaling=True,
                                                                      dim_reduction=True)

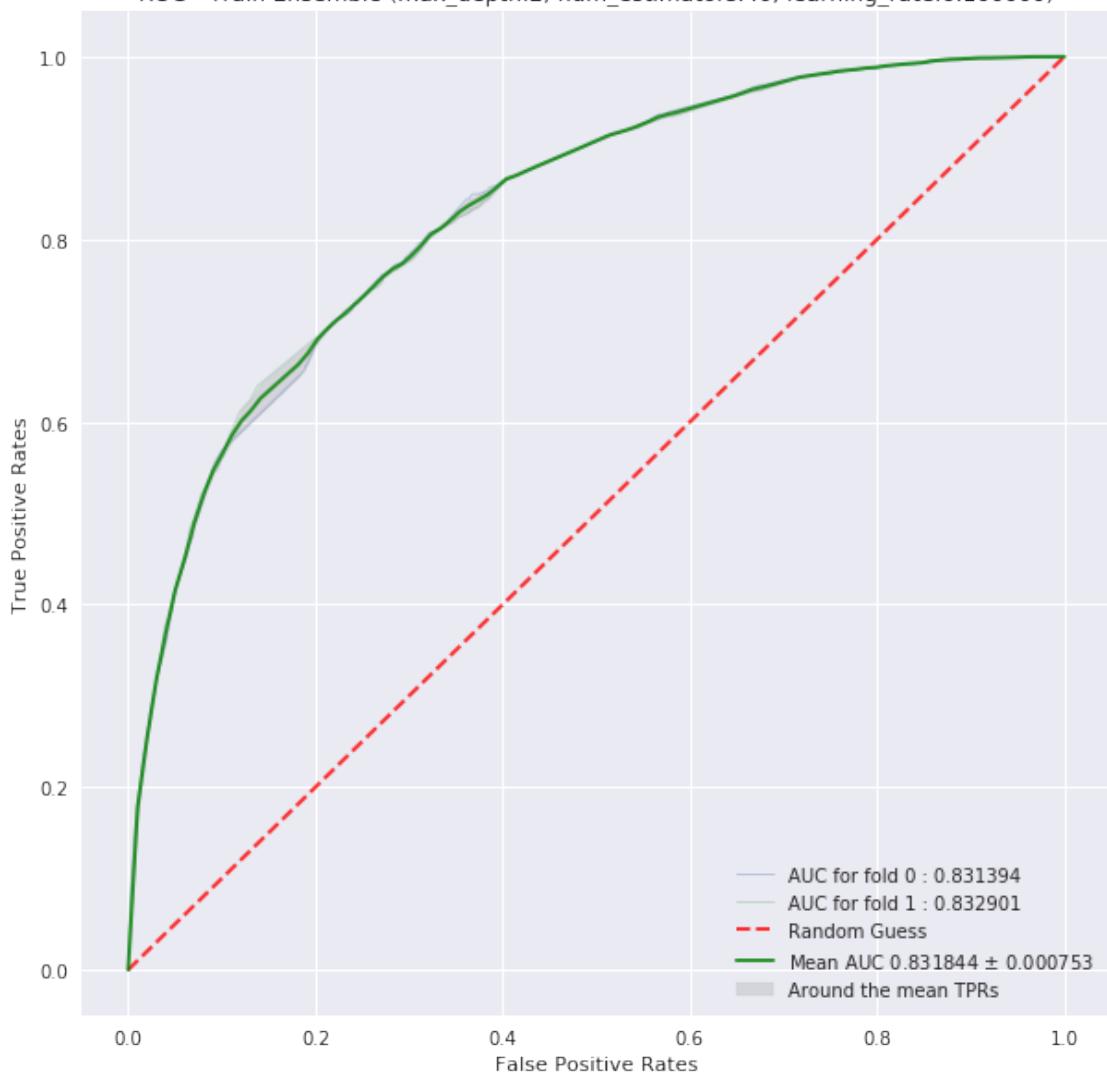
# train and validate the model
model = train_and_validate_model(config_dict, train_features, train_labels)

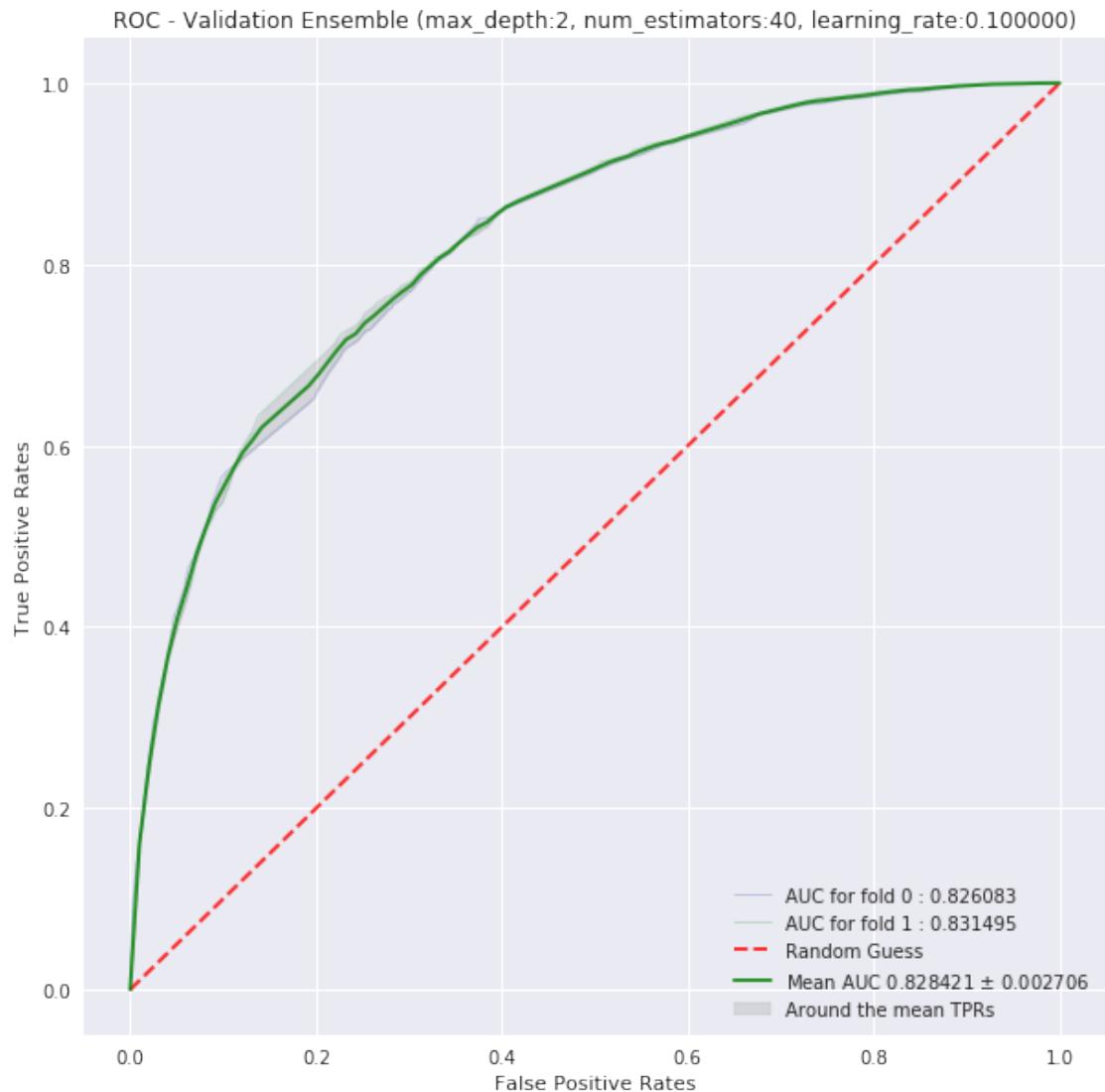
# test and evaluate the model
ptabe_entry_b1 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

Train df shape (40000, 503)
Class label distribution in train df:
0    20024
1    19976
Name: Label, dtype: int64
Test df shape (15000, 503)
Class label distribution in test df:
1    12386
0    2614
Name: Label, dtype: int64
Shape of -> train features :40000,501, test features: 15000,501
Shape of -> train labels :40000, test labels: 15000
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If you are using this function please consider passing a pandas Series instead of a list-like object.
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If you are using this function please consider passing a pandas Series instead of a list-like object.
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If you are using this function please consider passing a pandas Series instead of a list-like object.
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If you are using this function please consider passing a pandas Series instead of a list-like object.
  if diff:

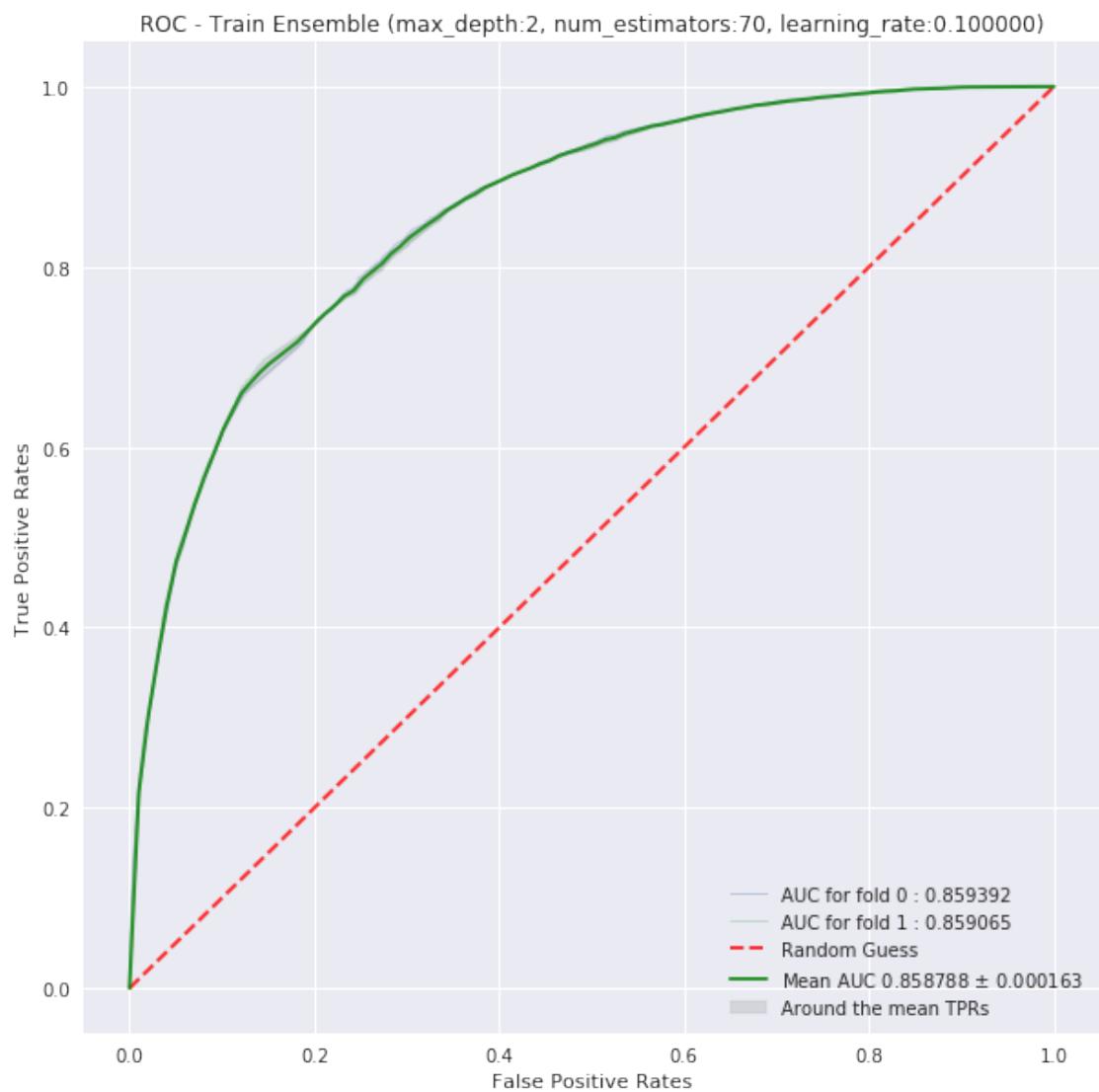
```

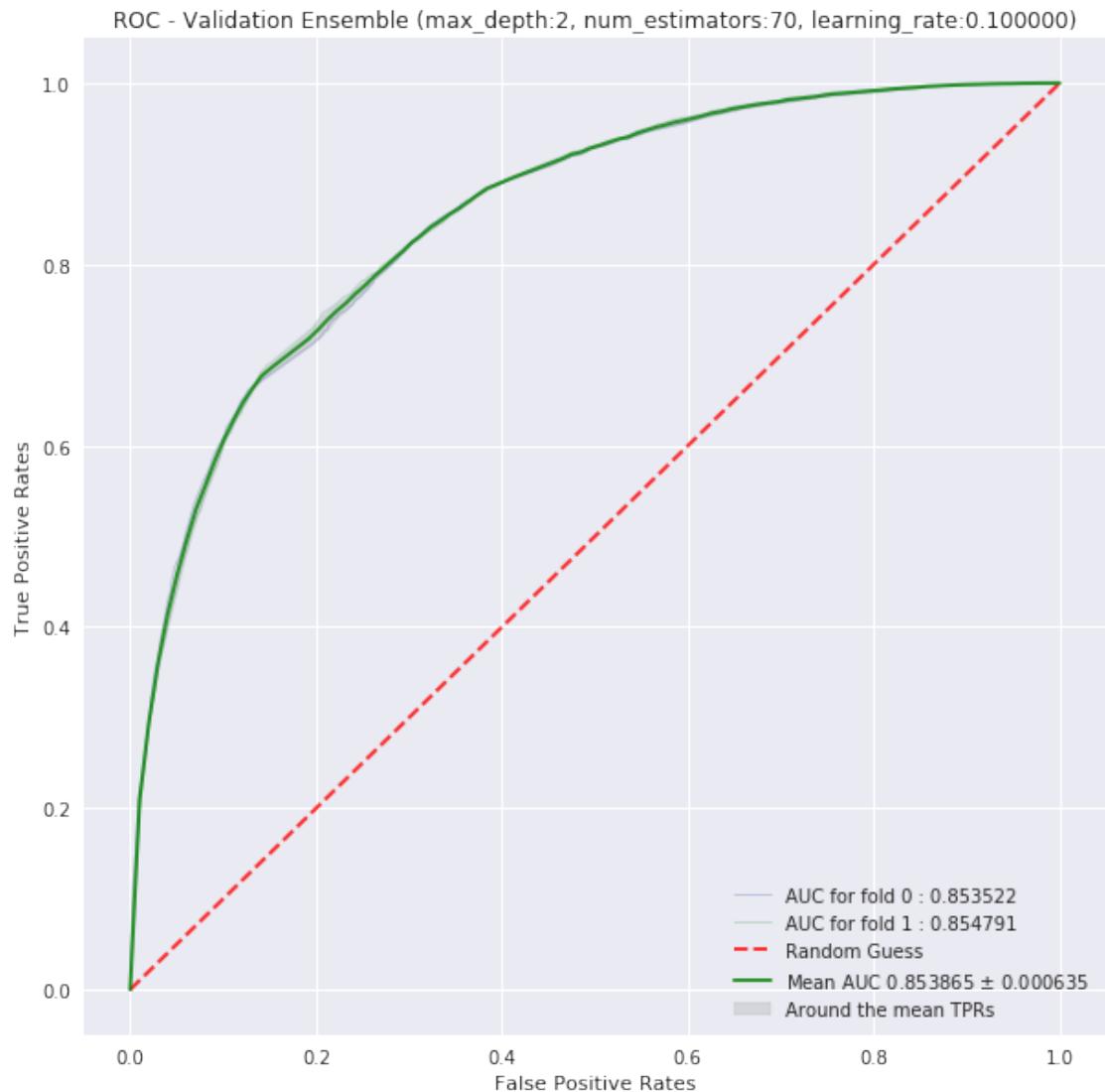
ROC - Train Ensemble (max\_depth:2, num\_estimators:40, learning\_rate:0.100000)



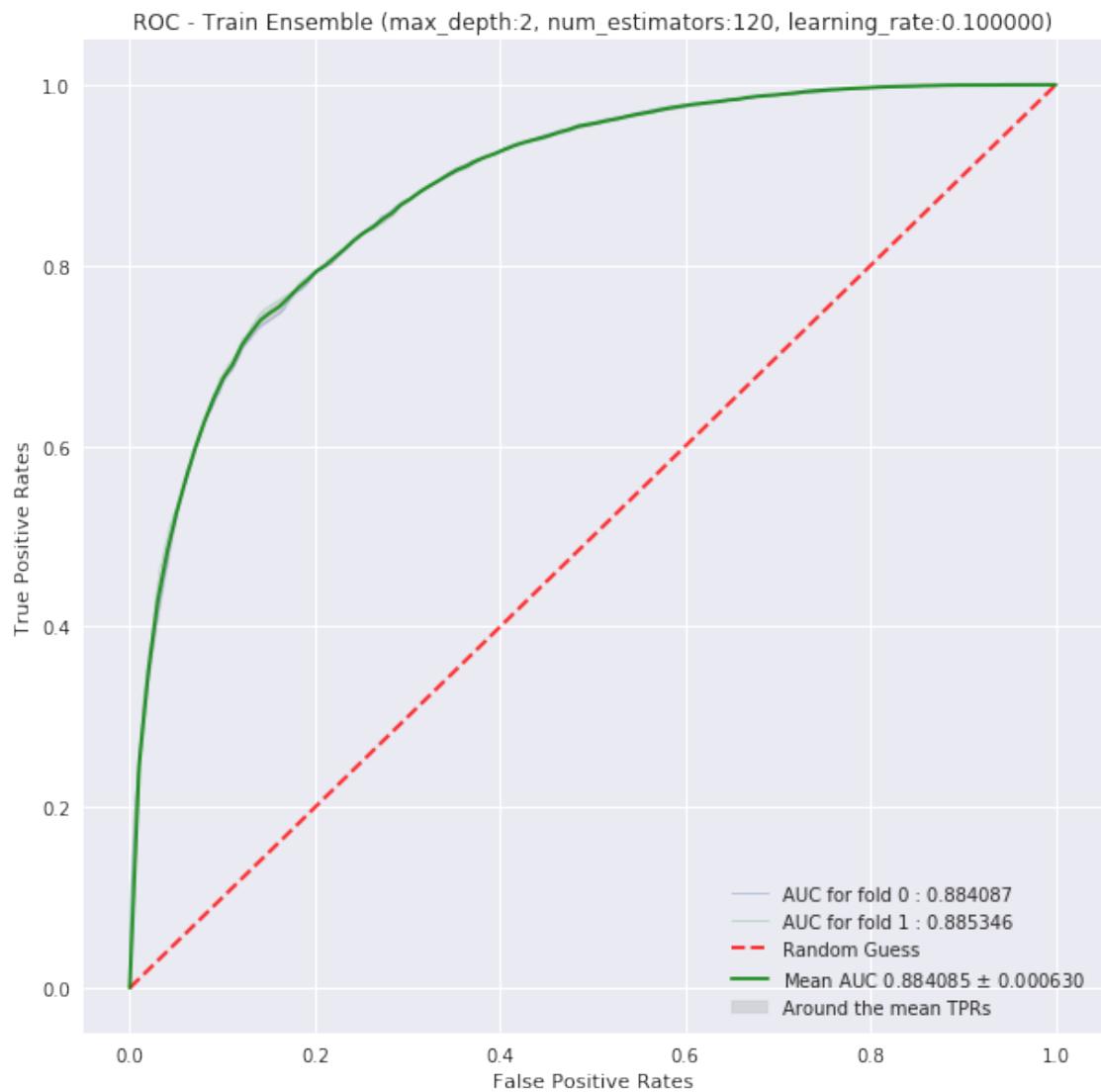


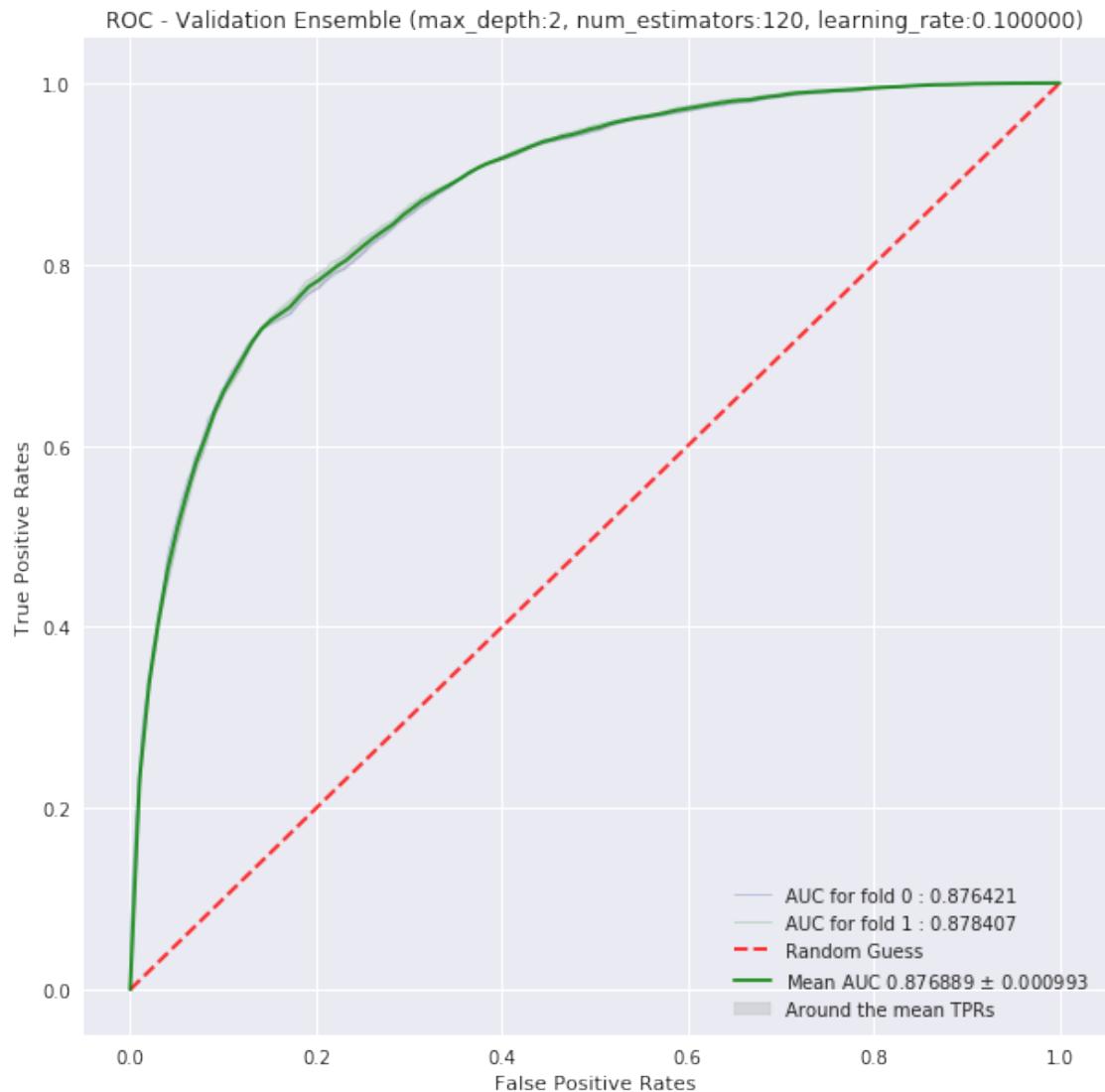
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```



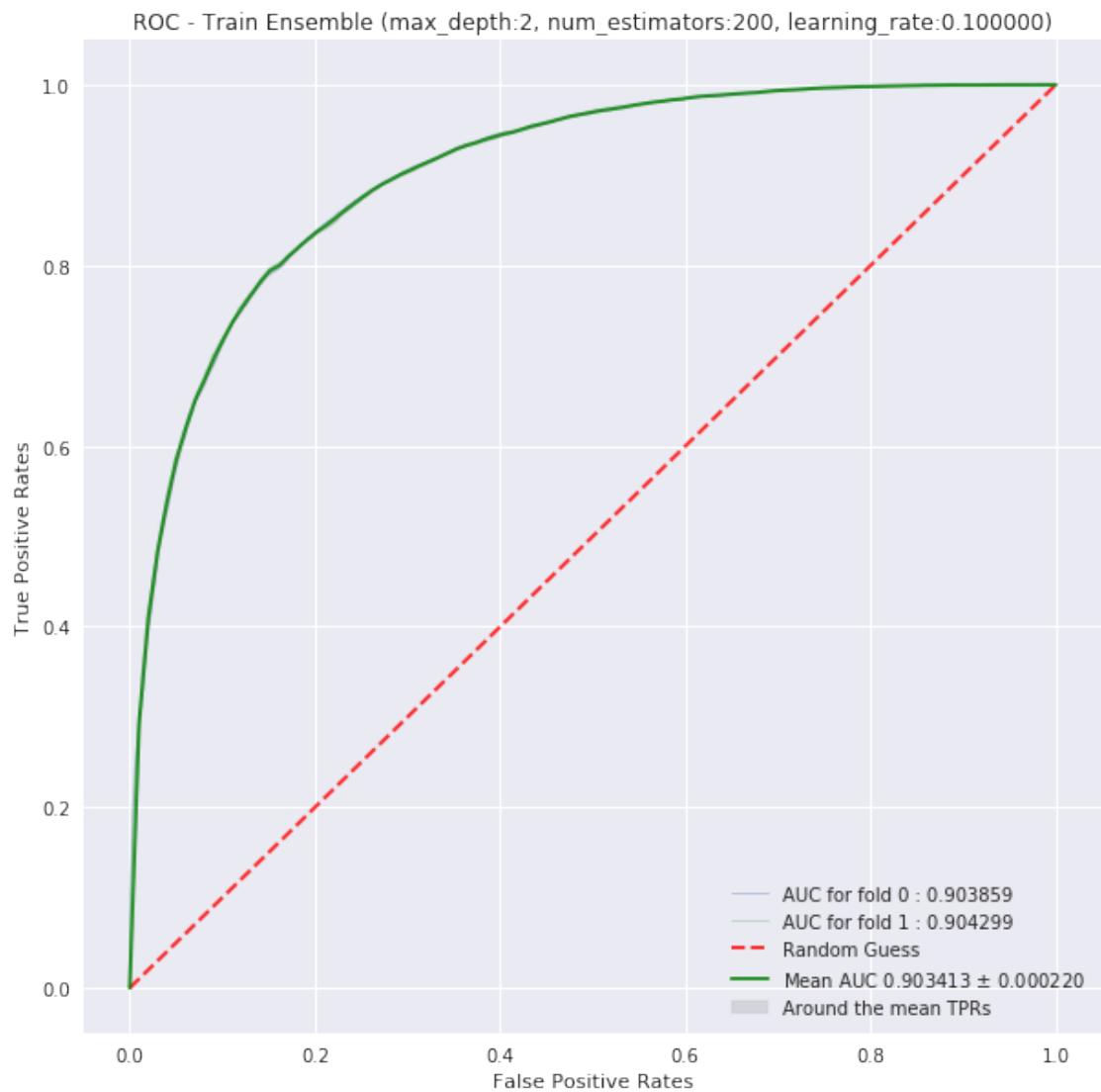


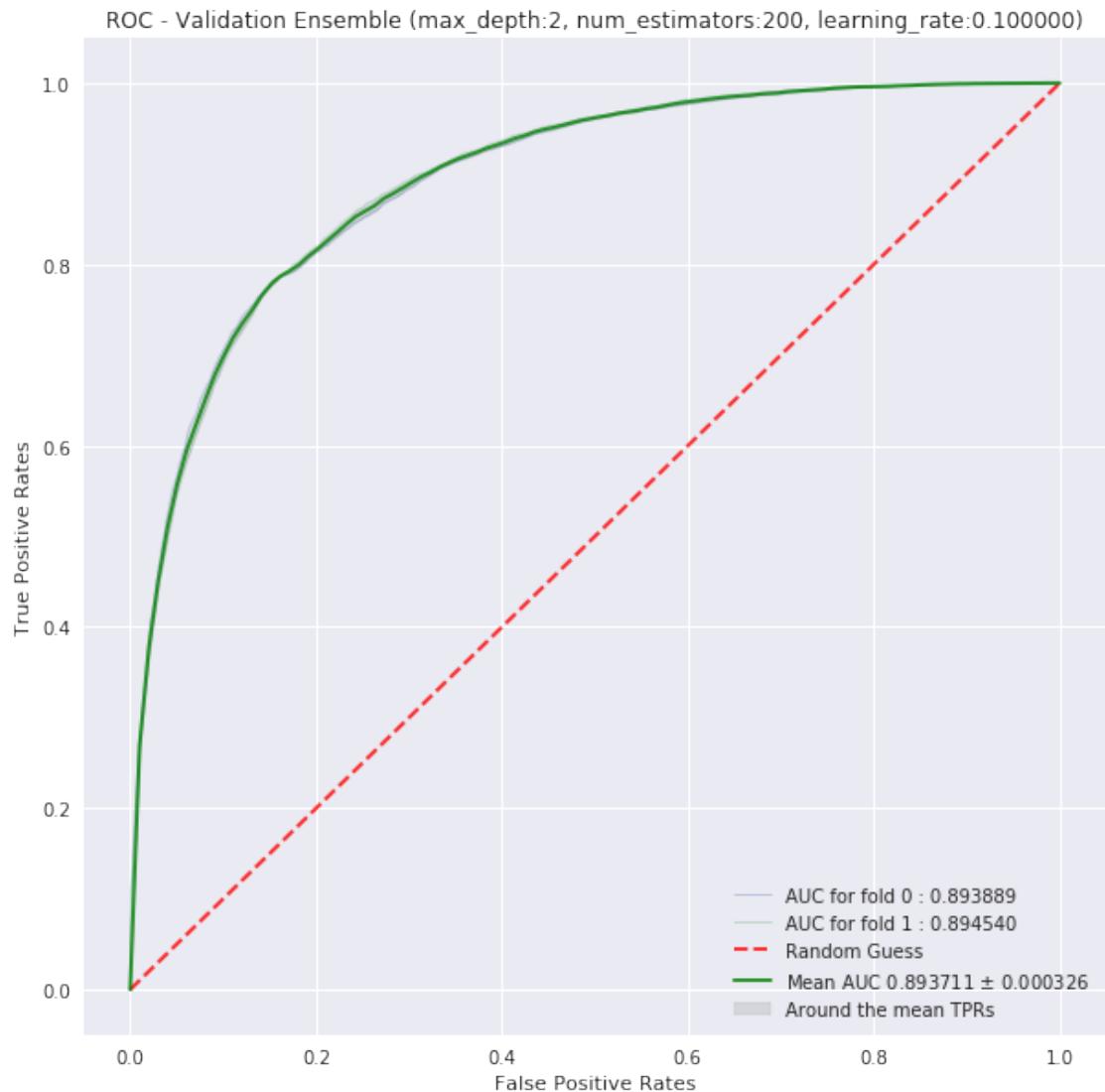
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```



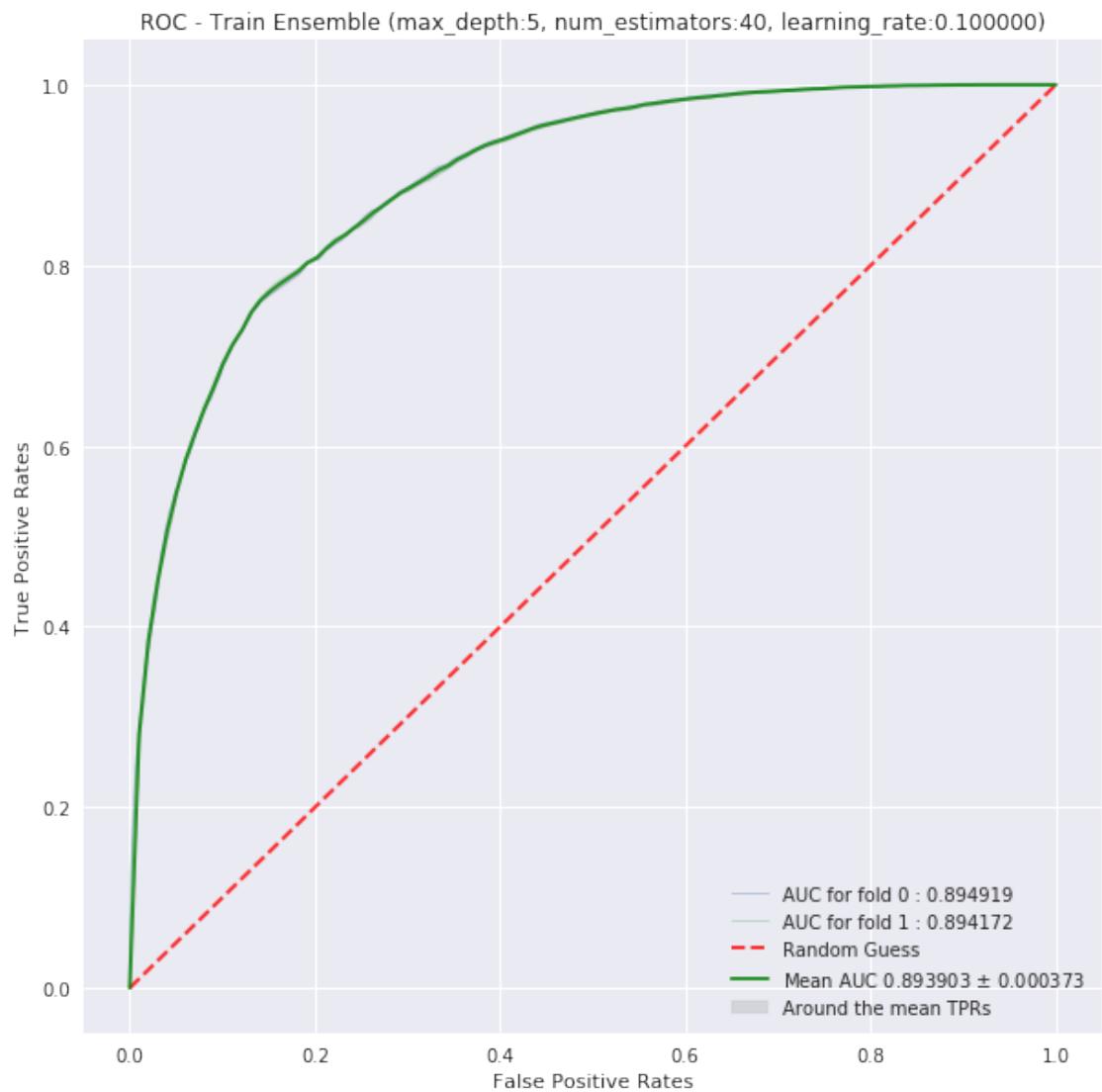


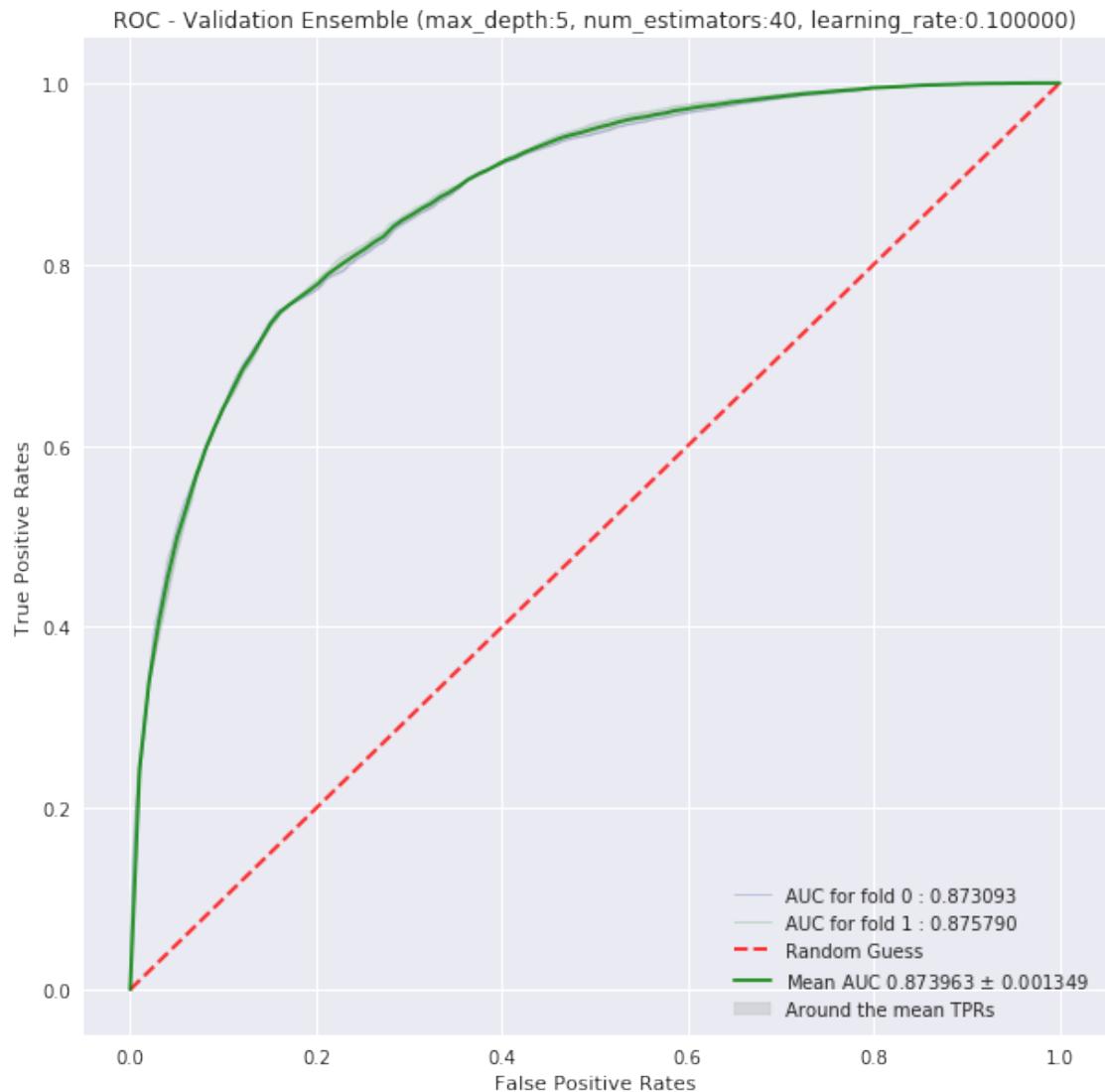
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```



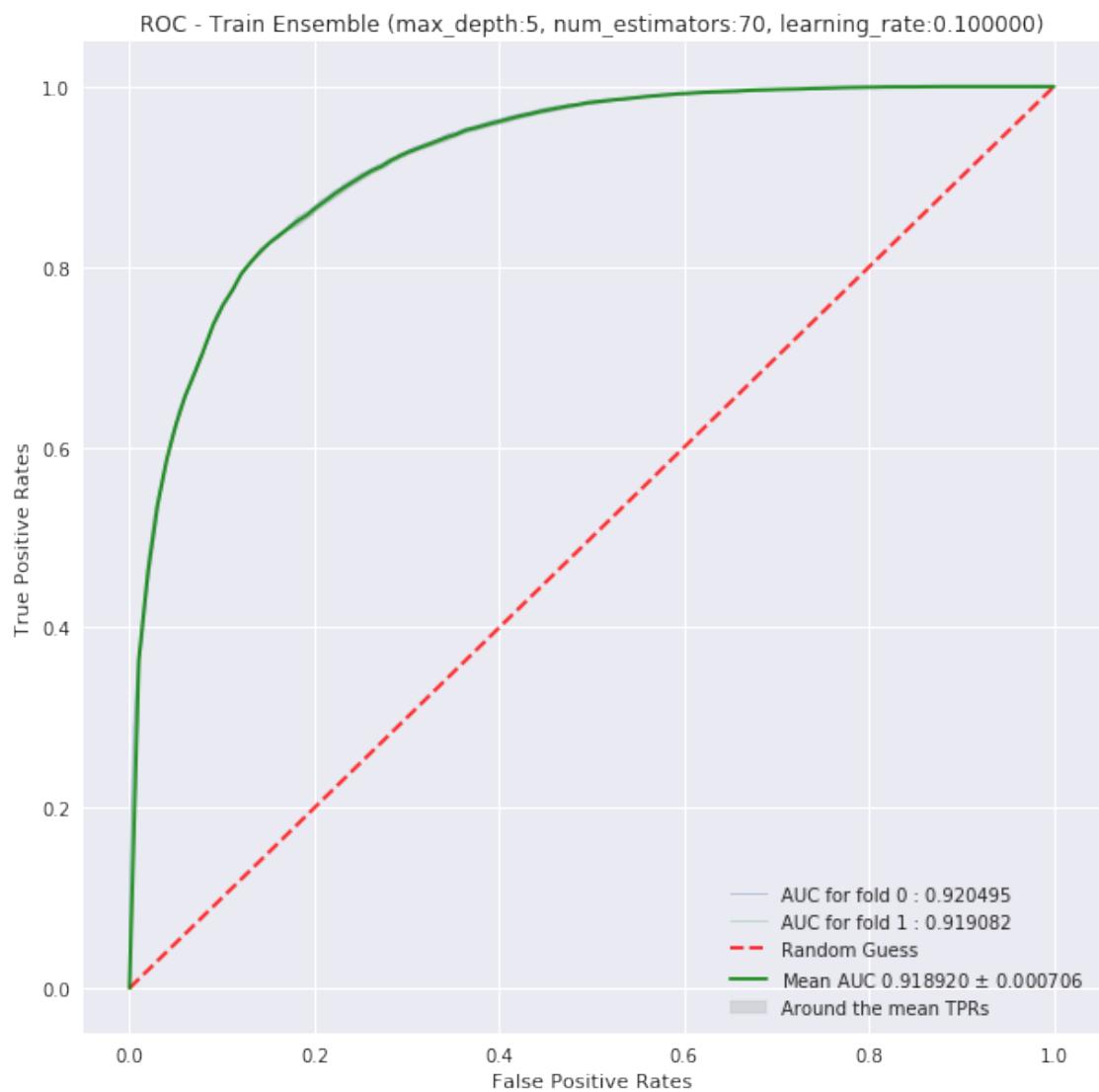


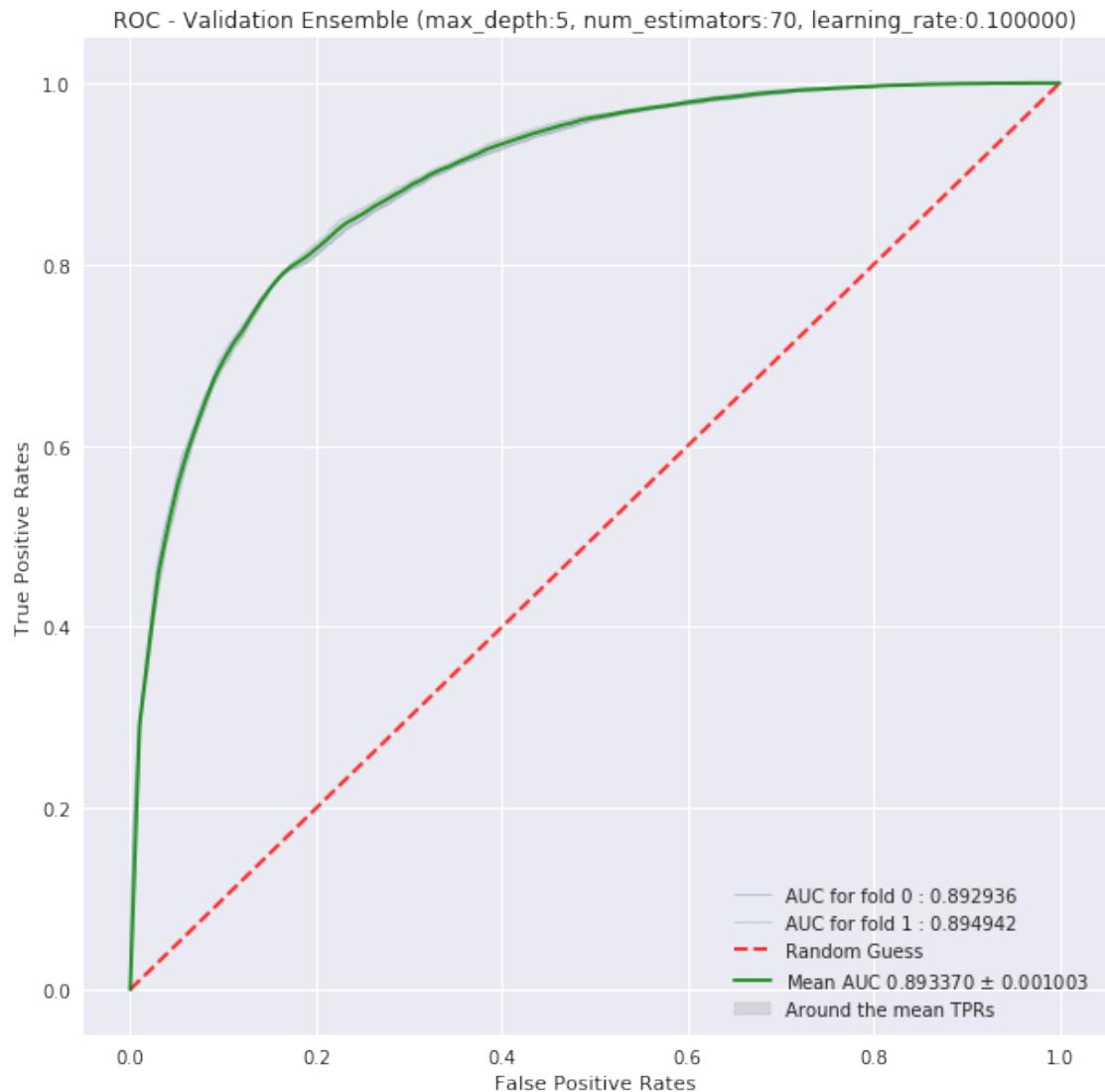
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```



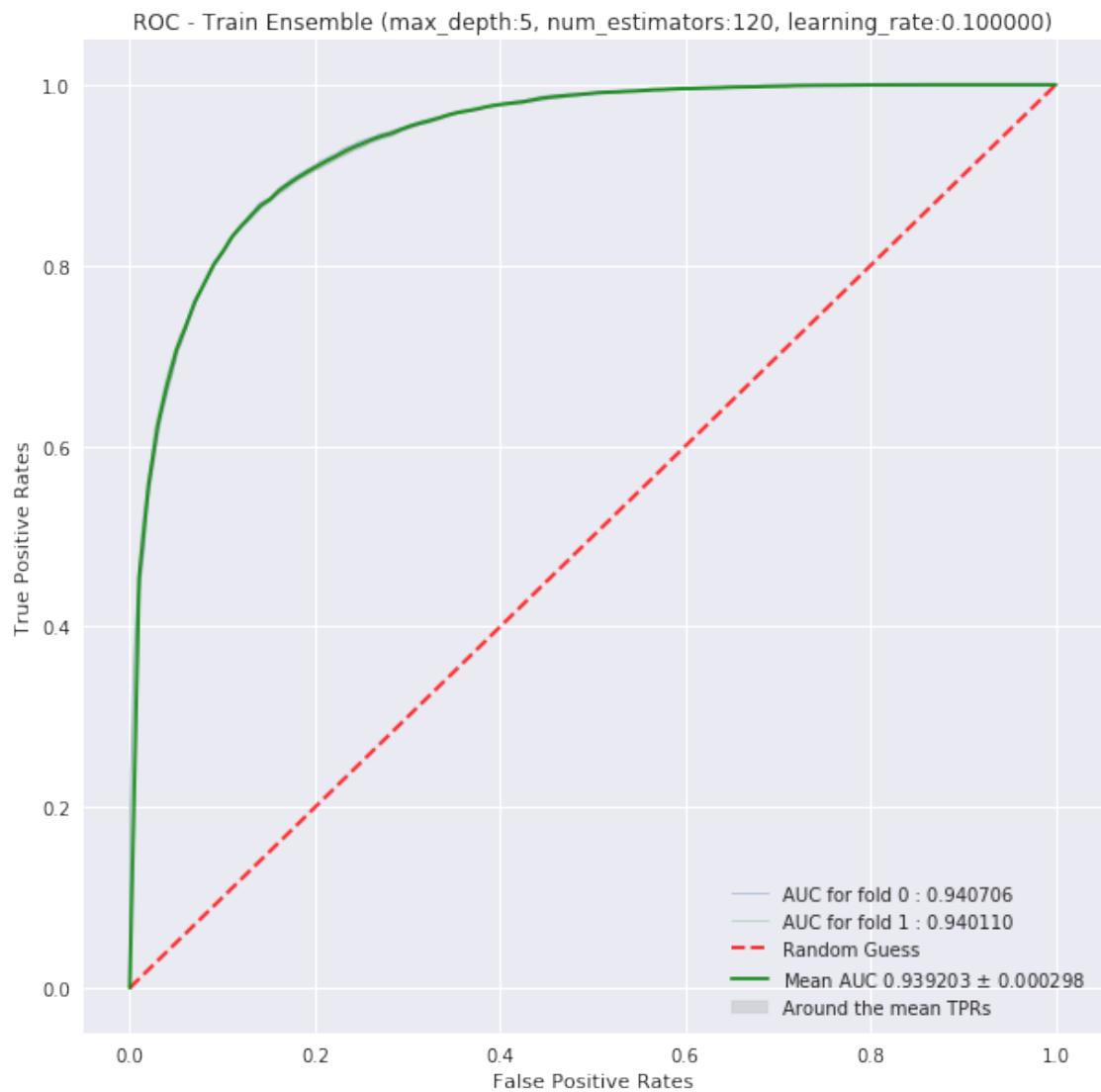


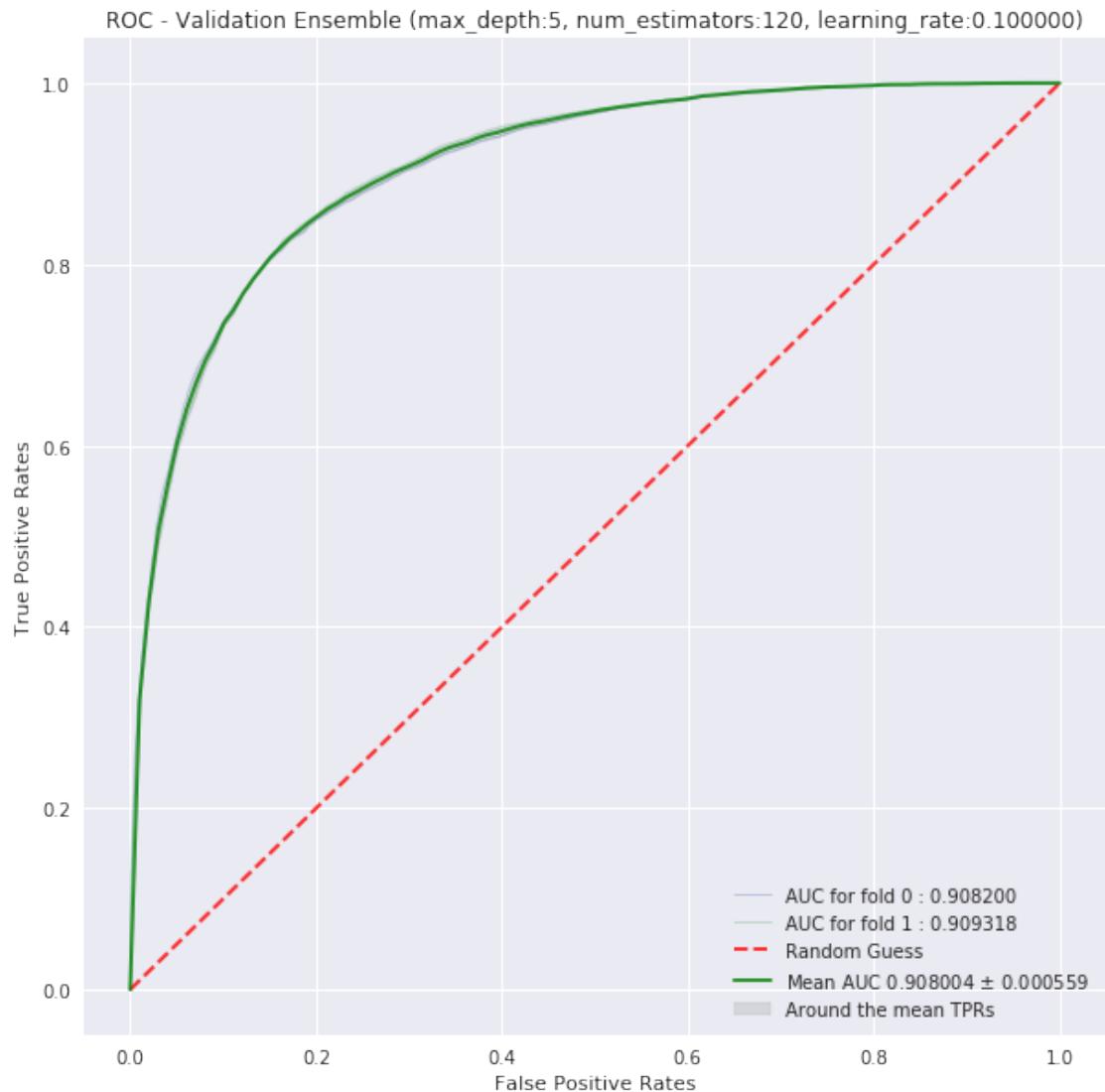
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```





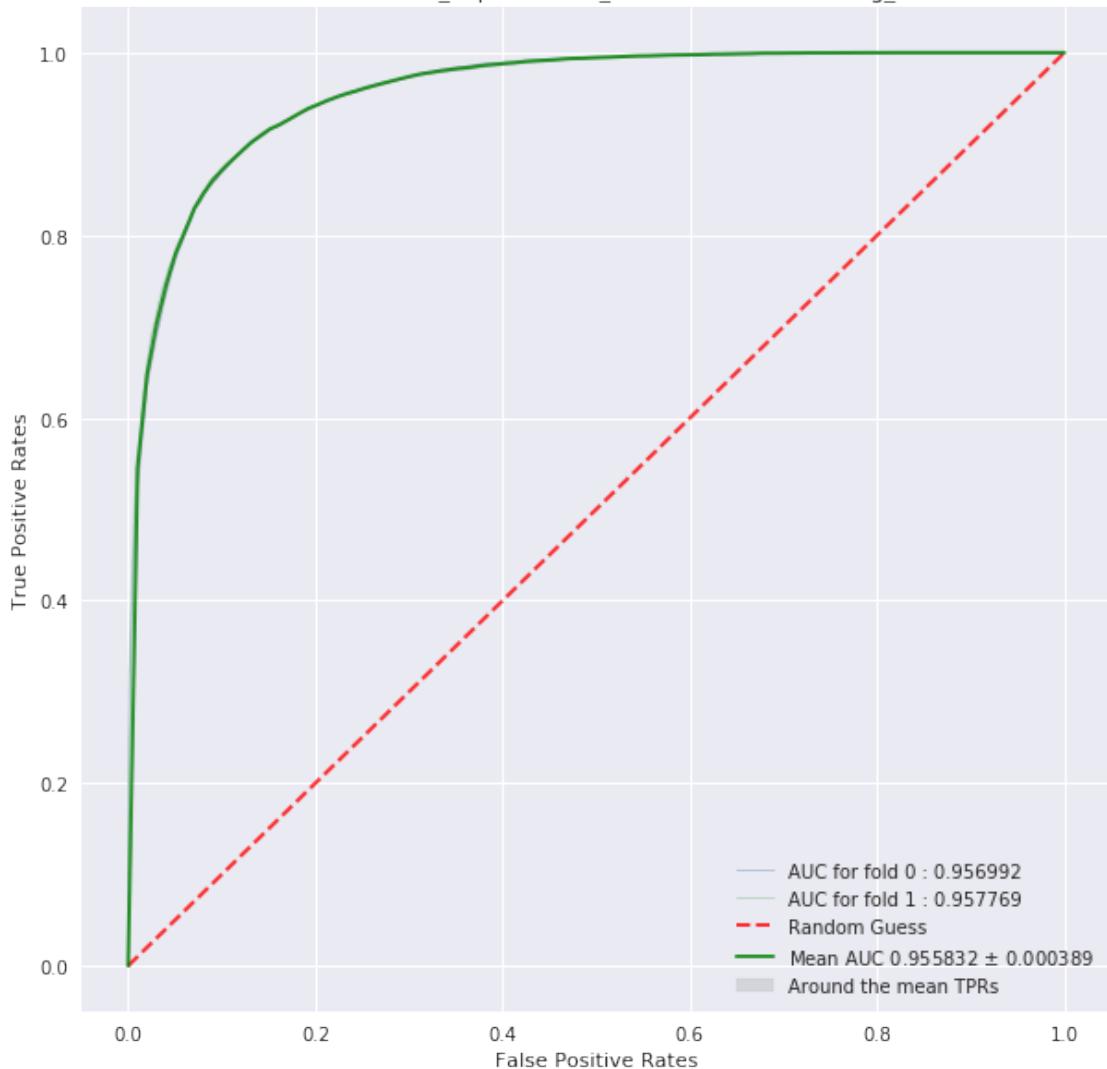
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

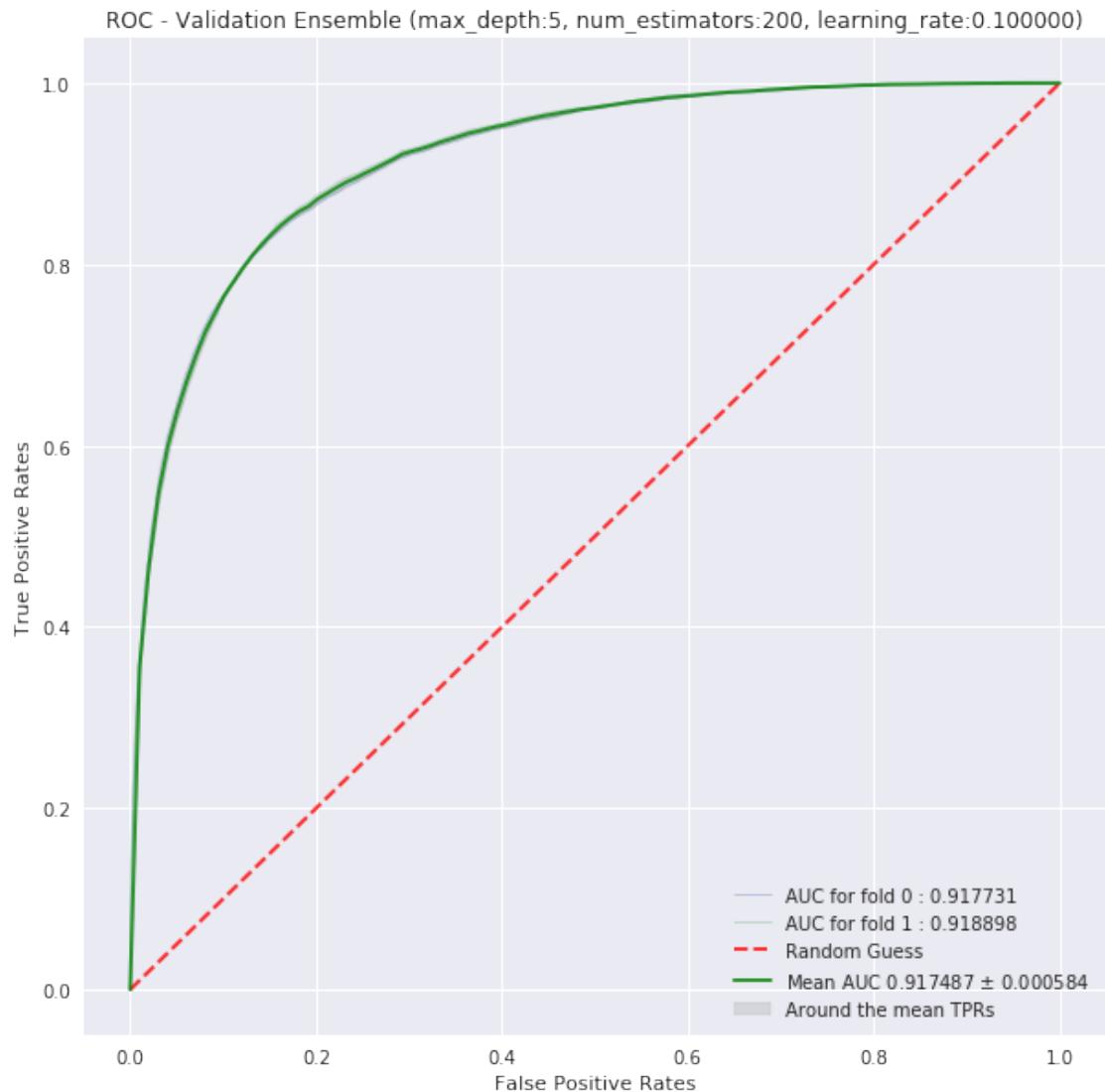




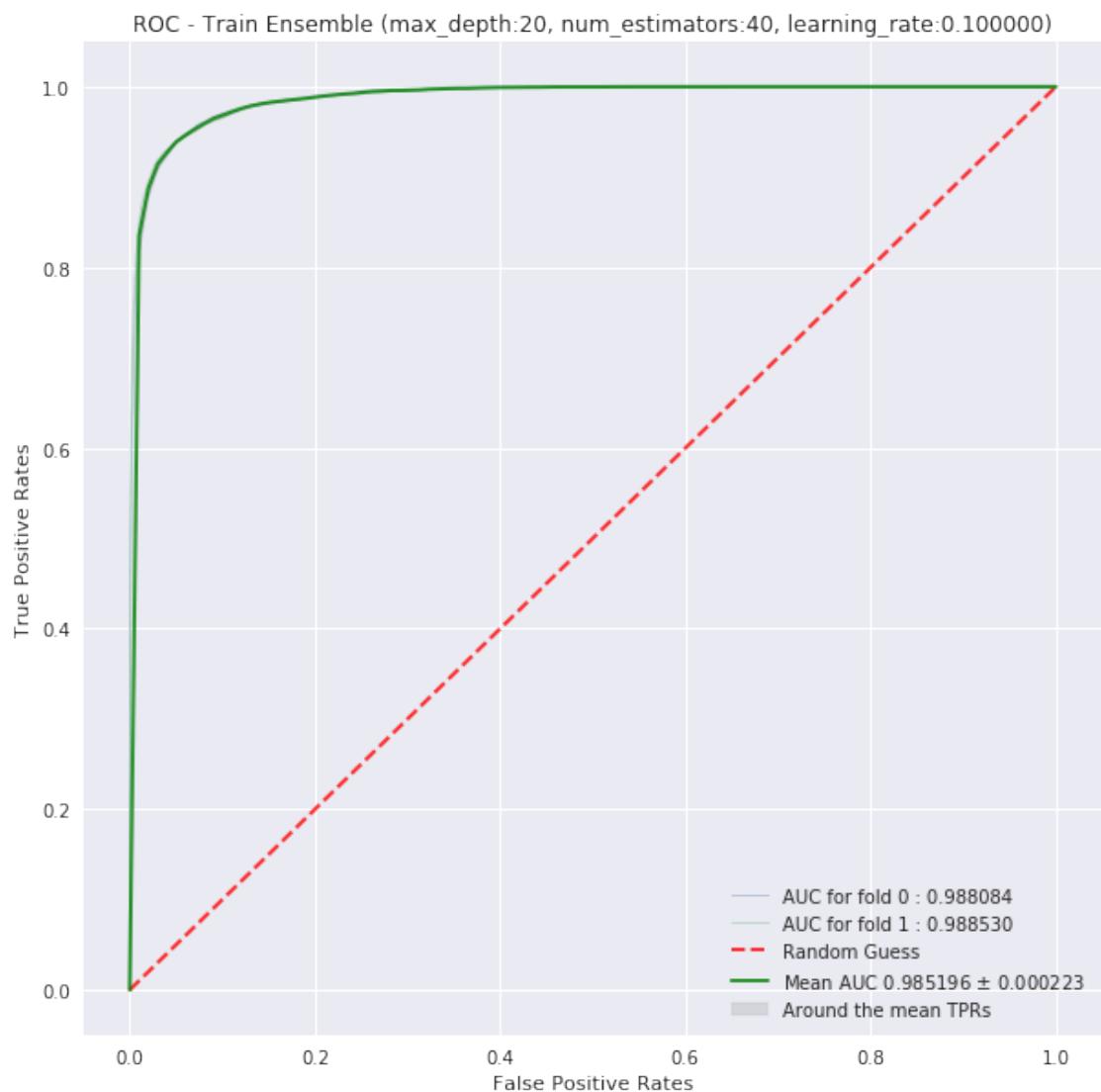
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

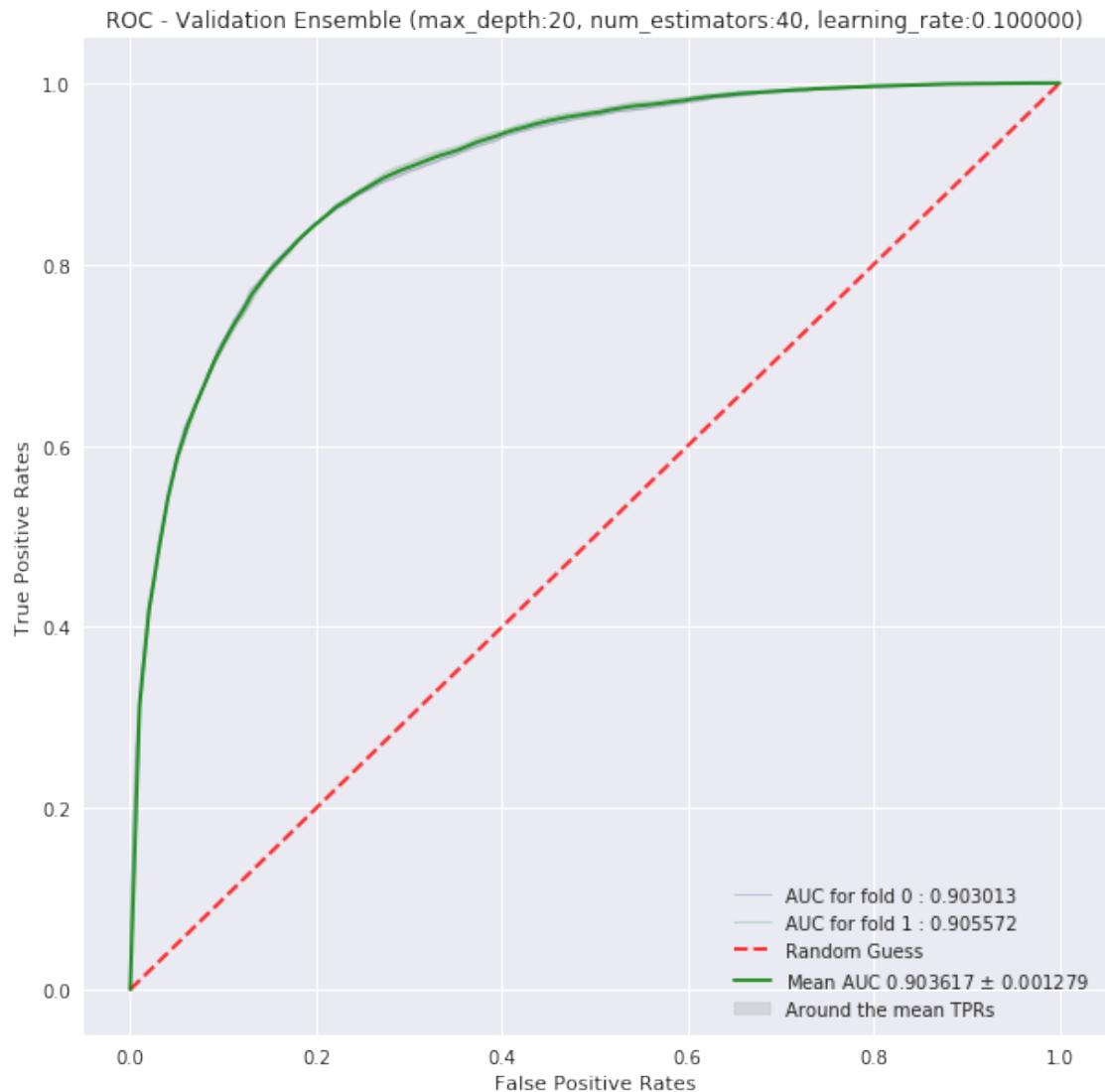
ROC - Train Ensemble (max\_depth:5, num\_estimators:200, learning\_rate:0.100000)





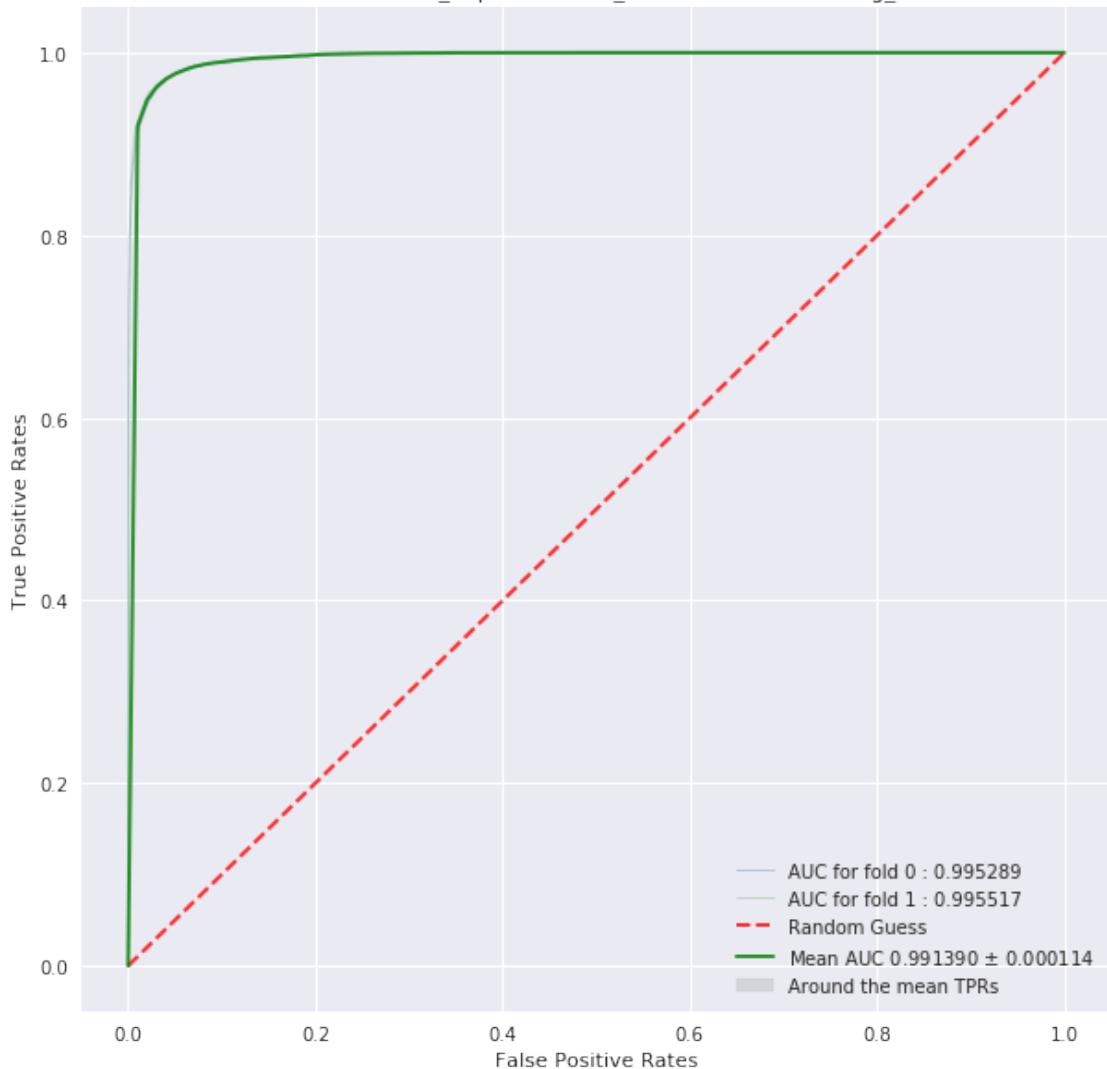
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

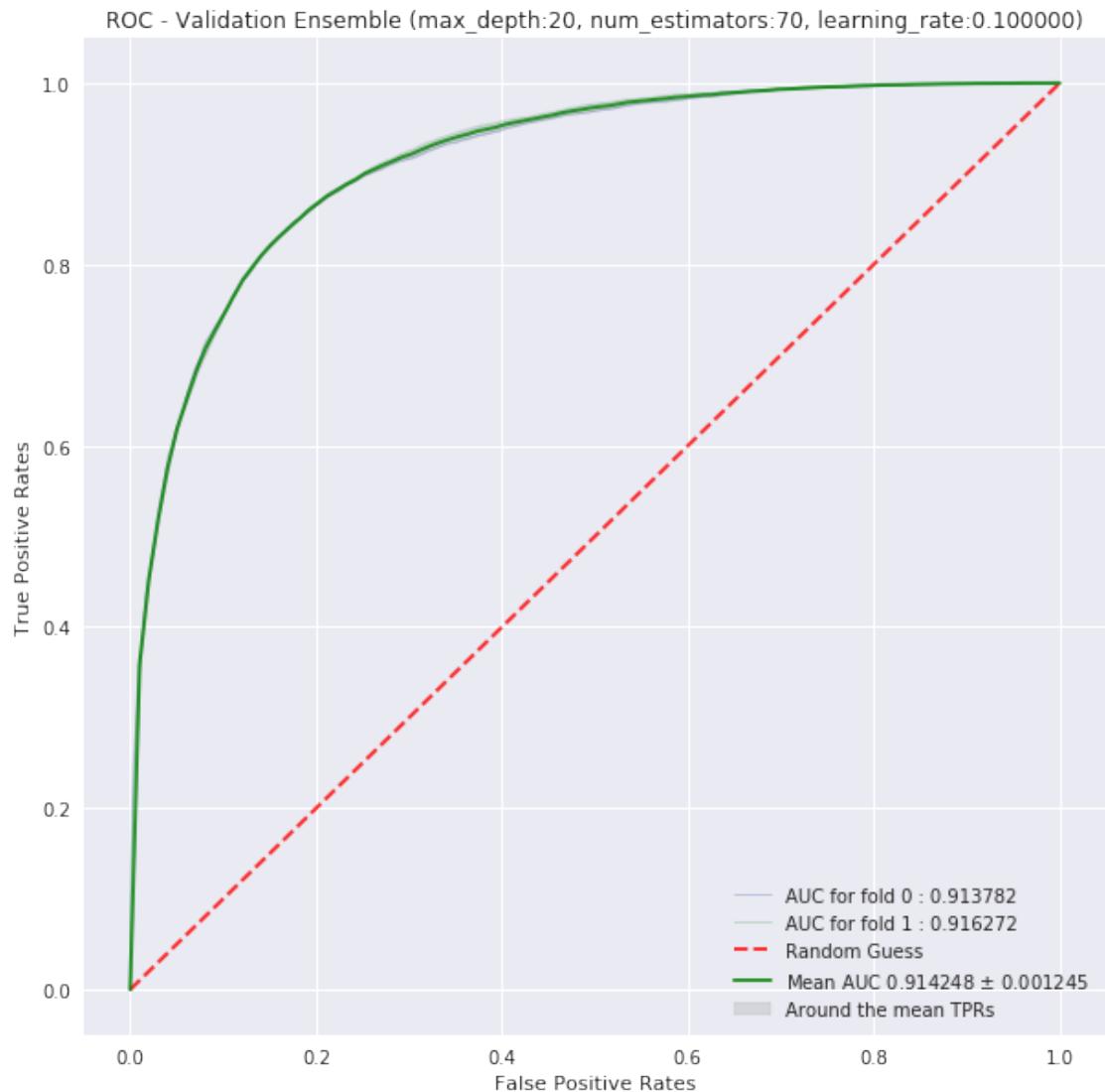




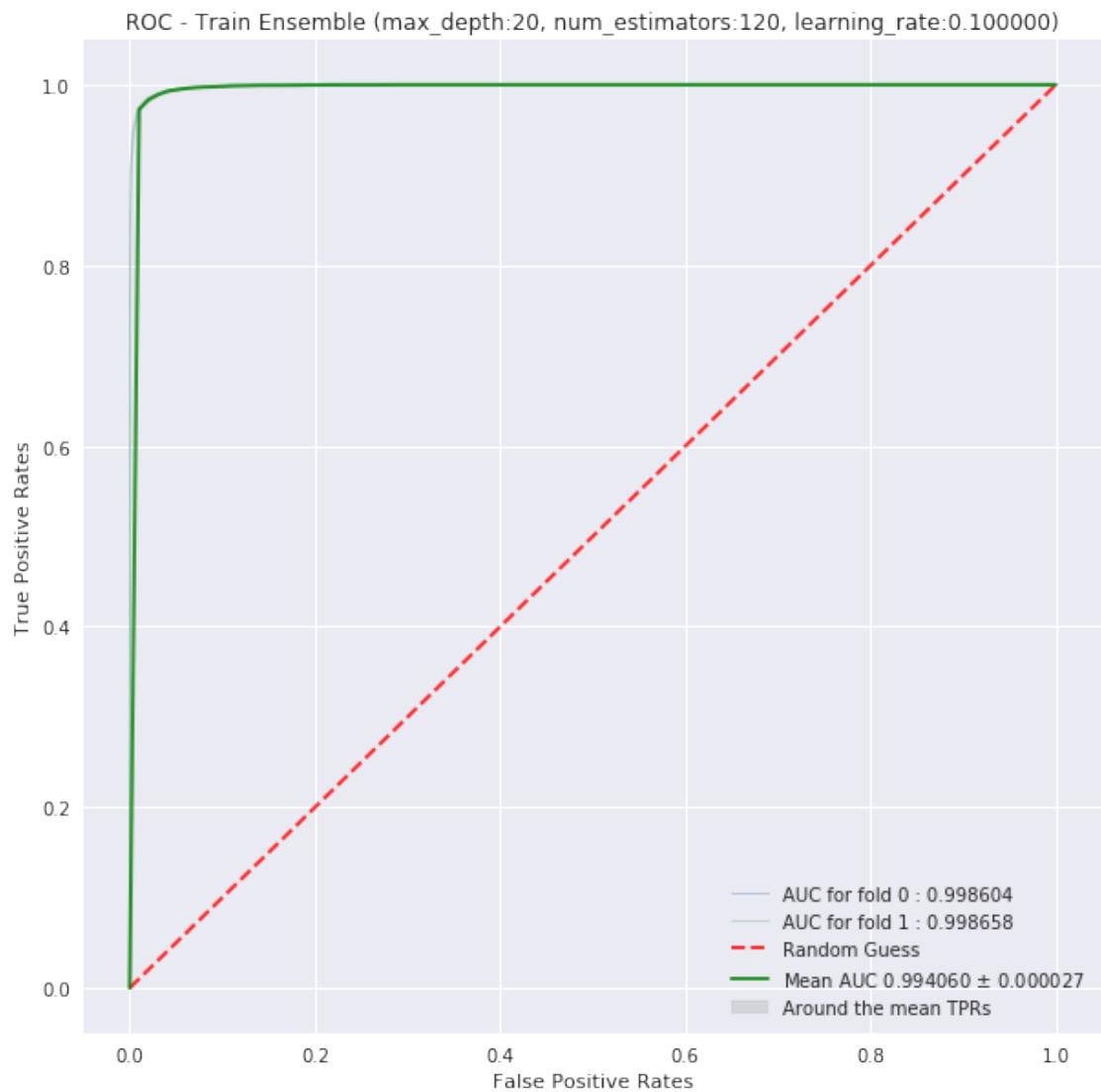
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

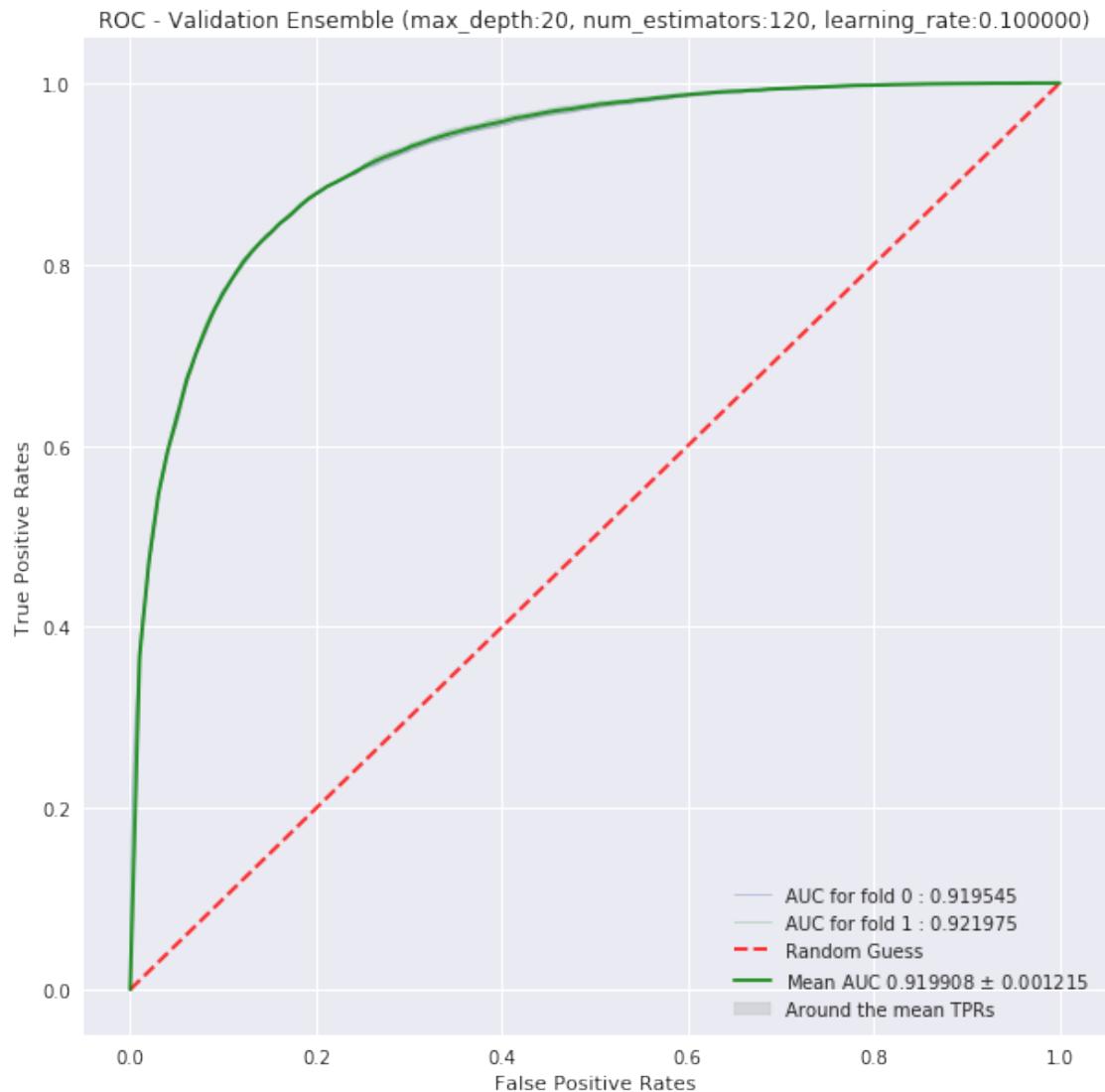
ROC - Train Ensemble (max\_depth:20, num\_estimators:70, learning\_rate:0.100000)





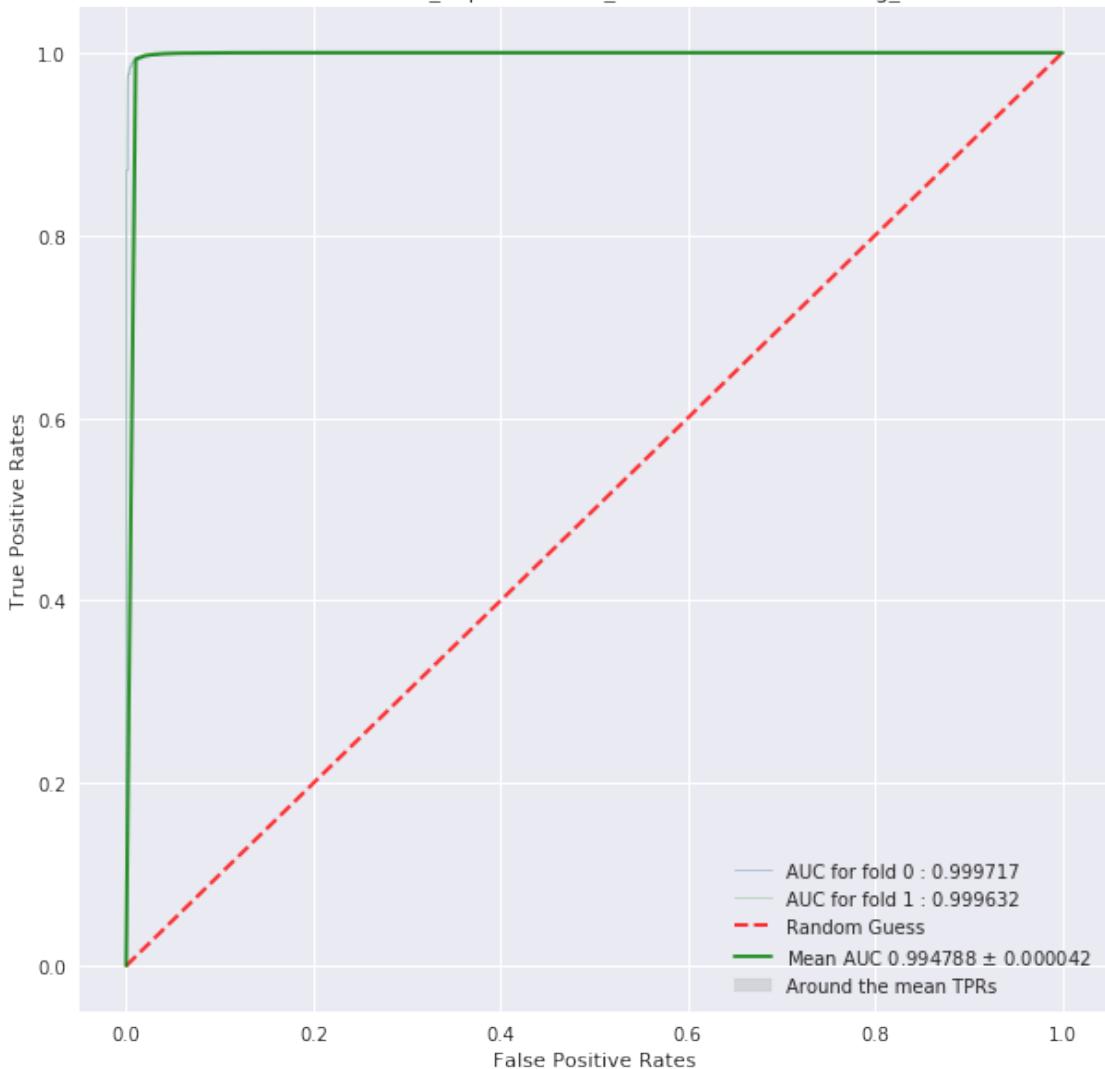
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

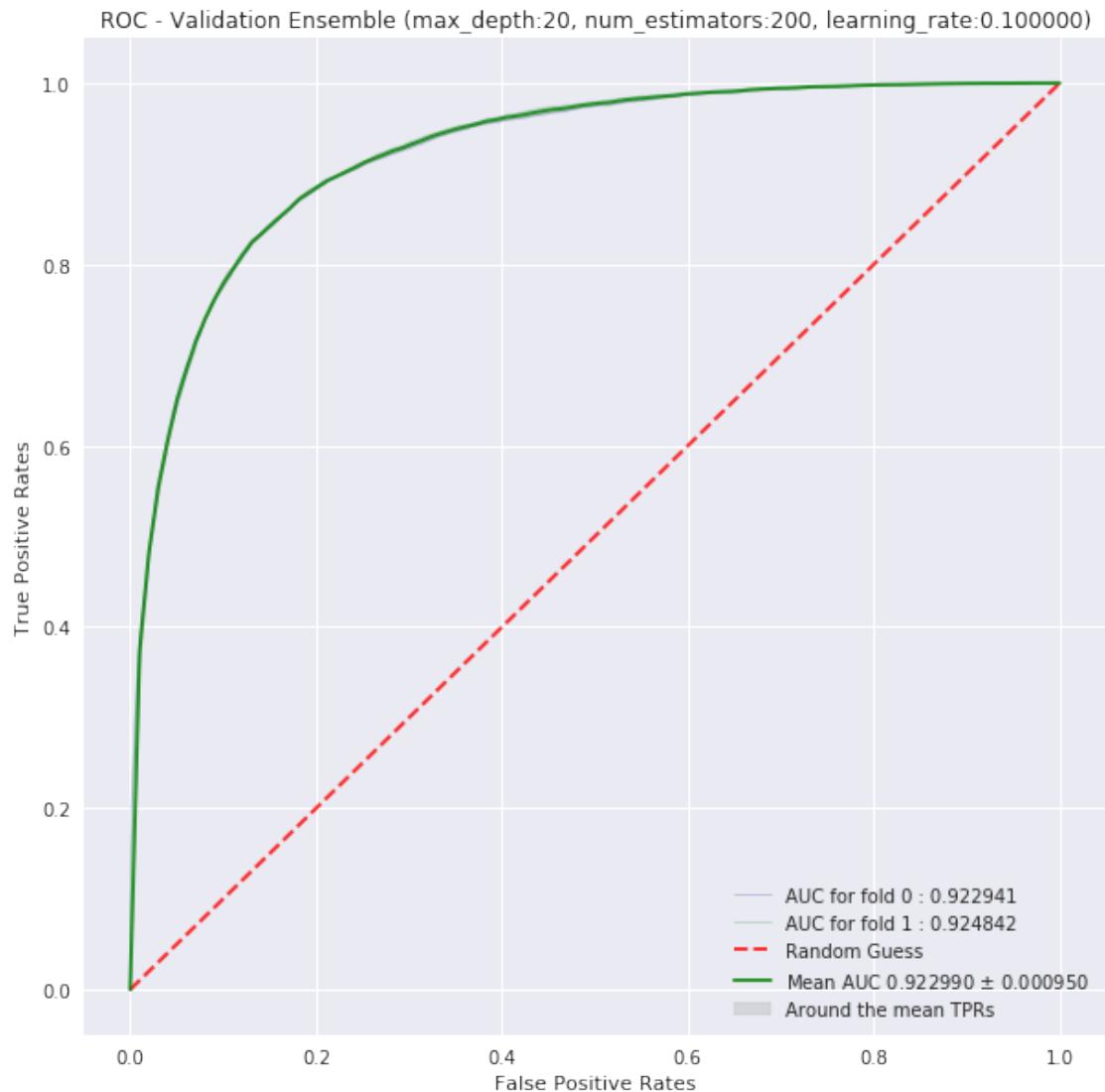




```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

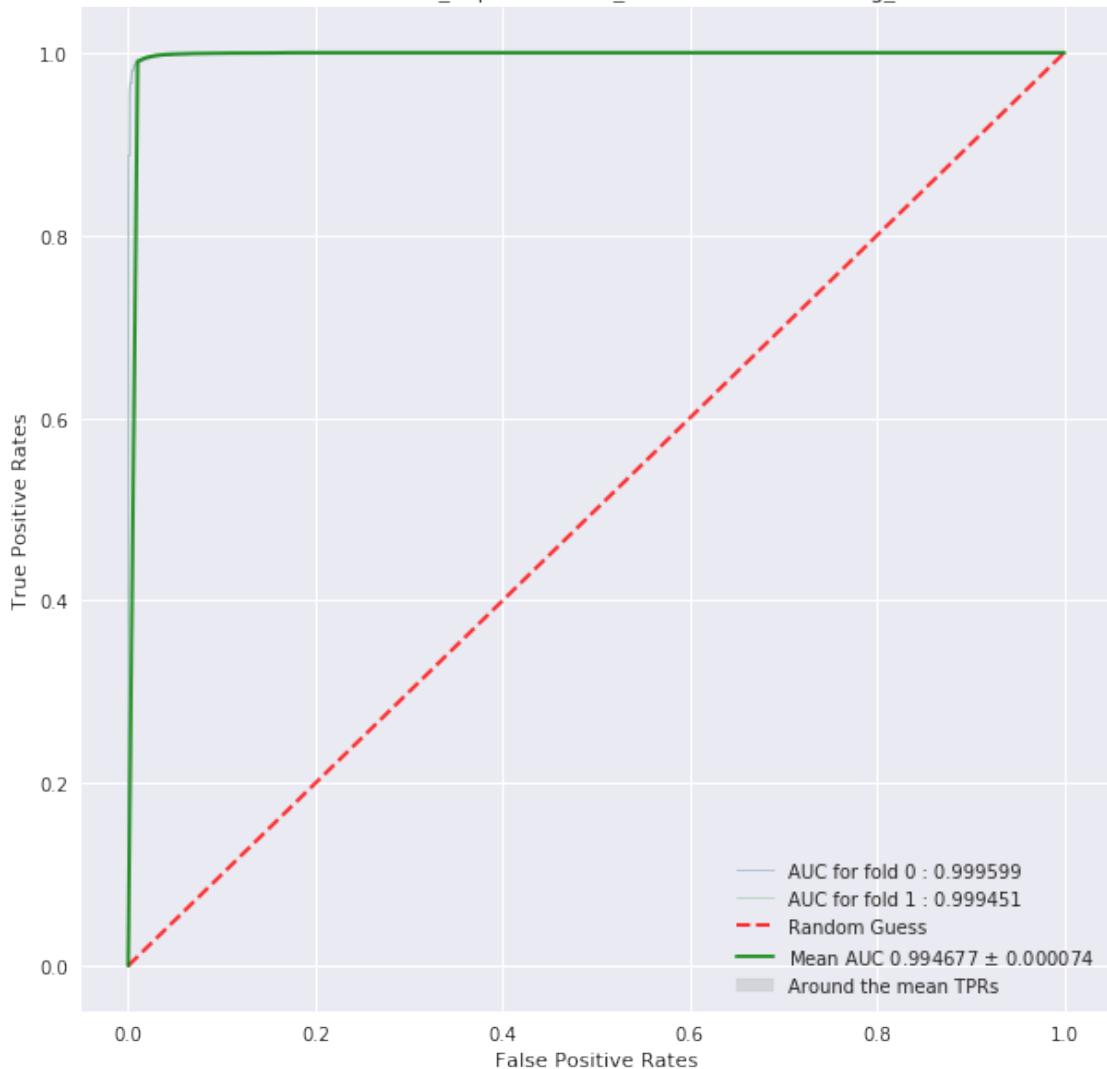
ROC - Train Ensemble (max\_depth:20, num\_estimators:200, learning\_rate:0.100000)

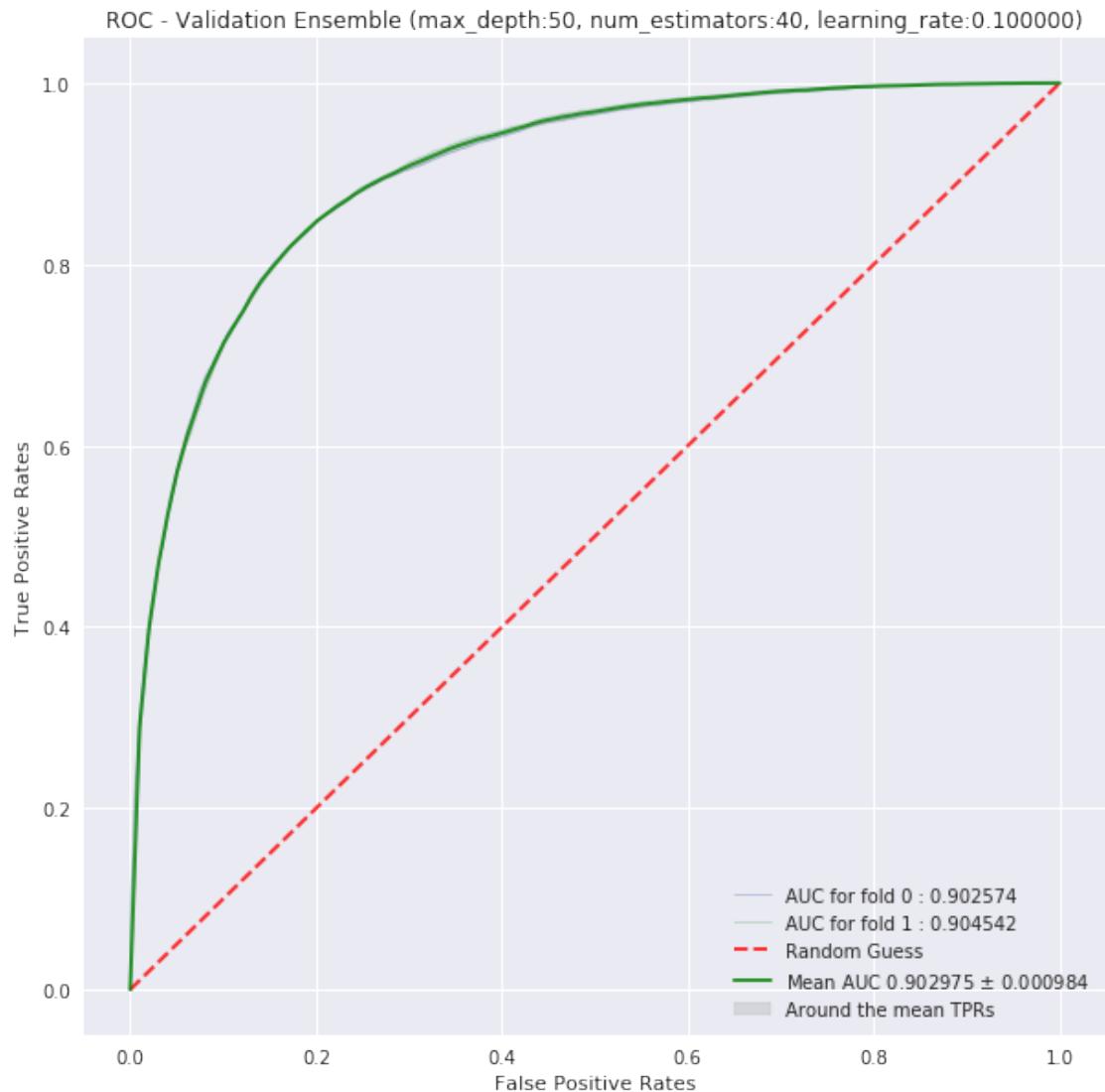




```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

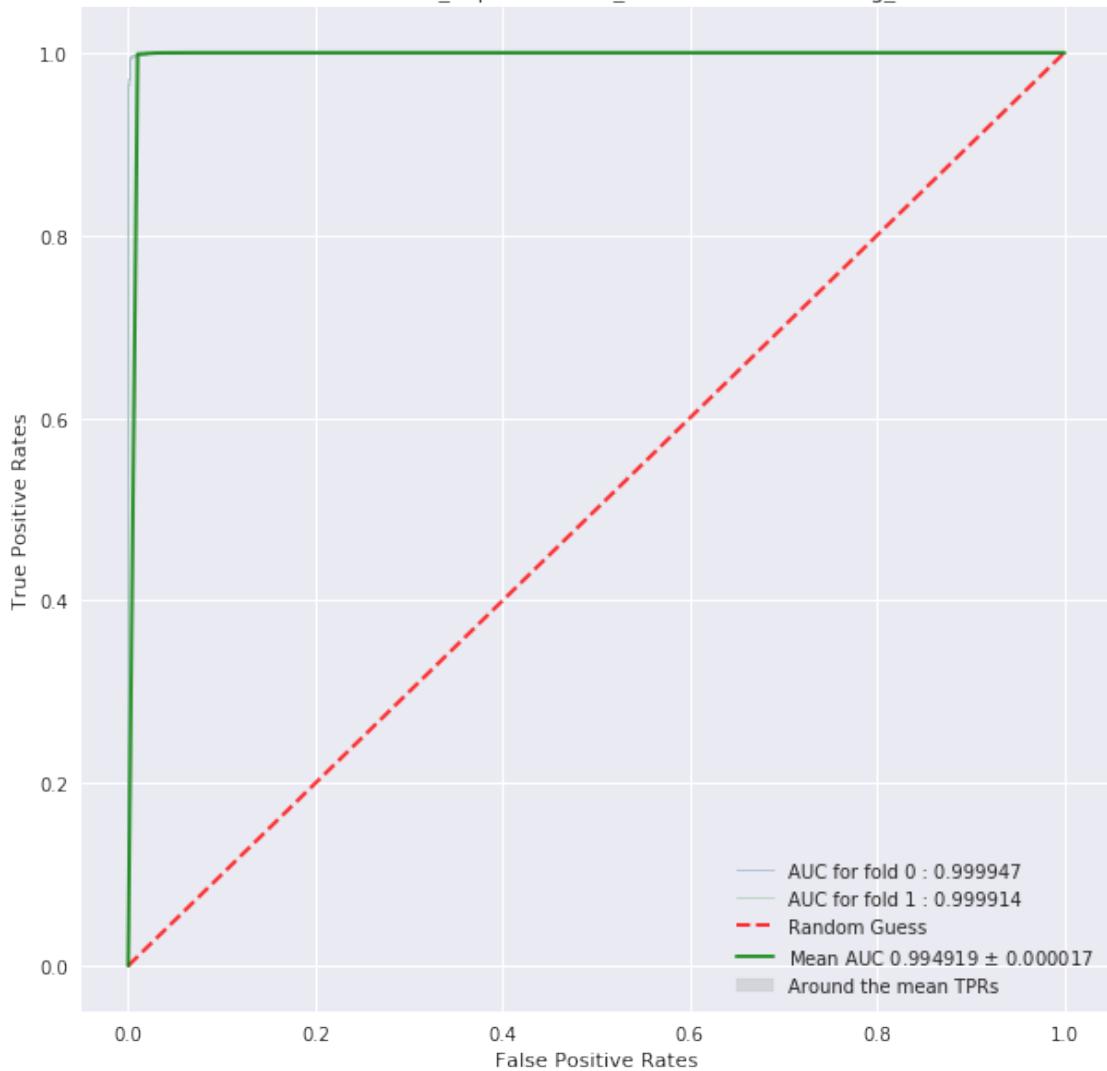
ROC - Train Ensemble (max\_depth:50, num\_estimators:40, learning\_rate:0.100000)

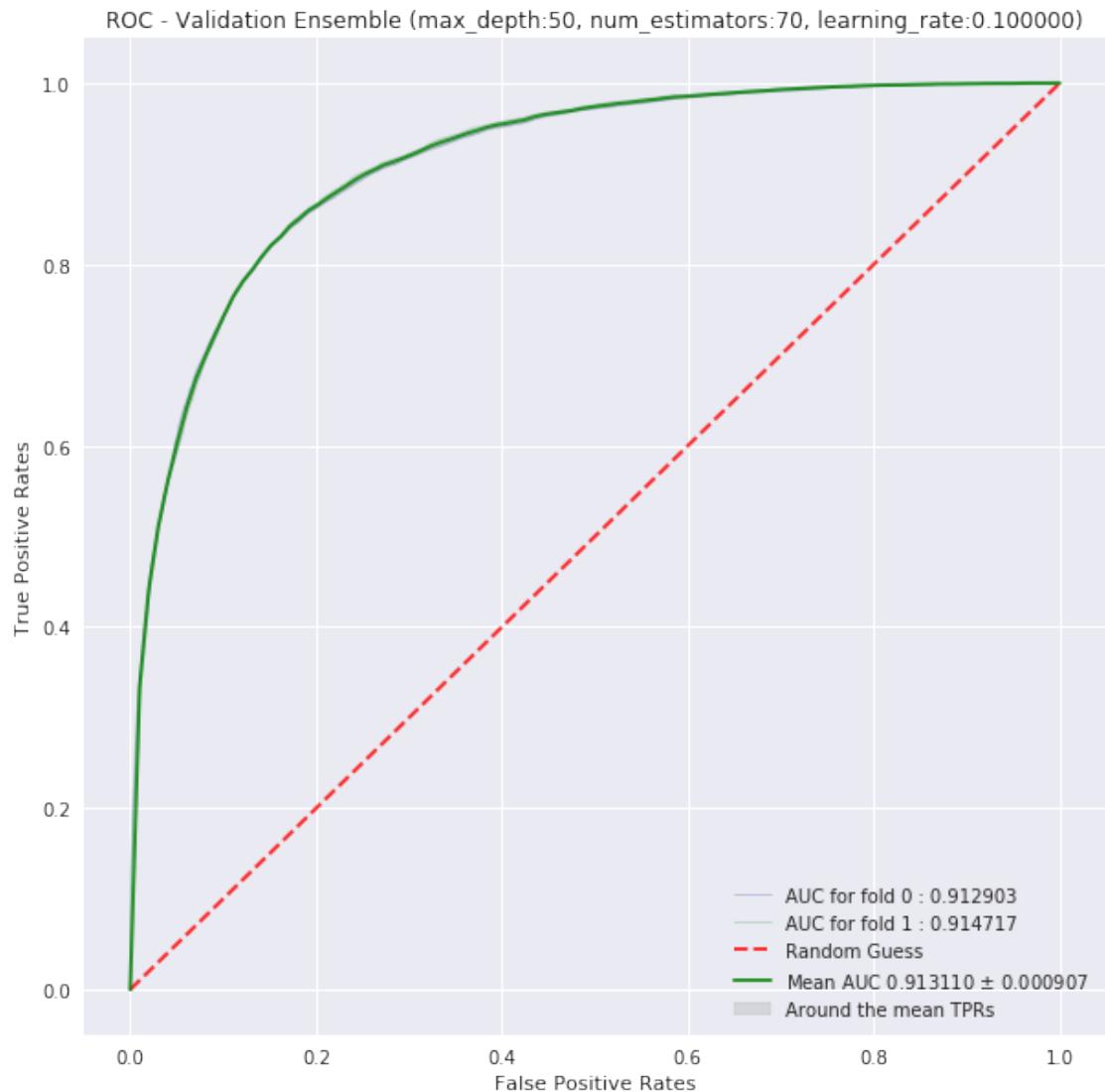




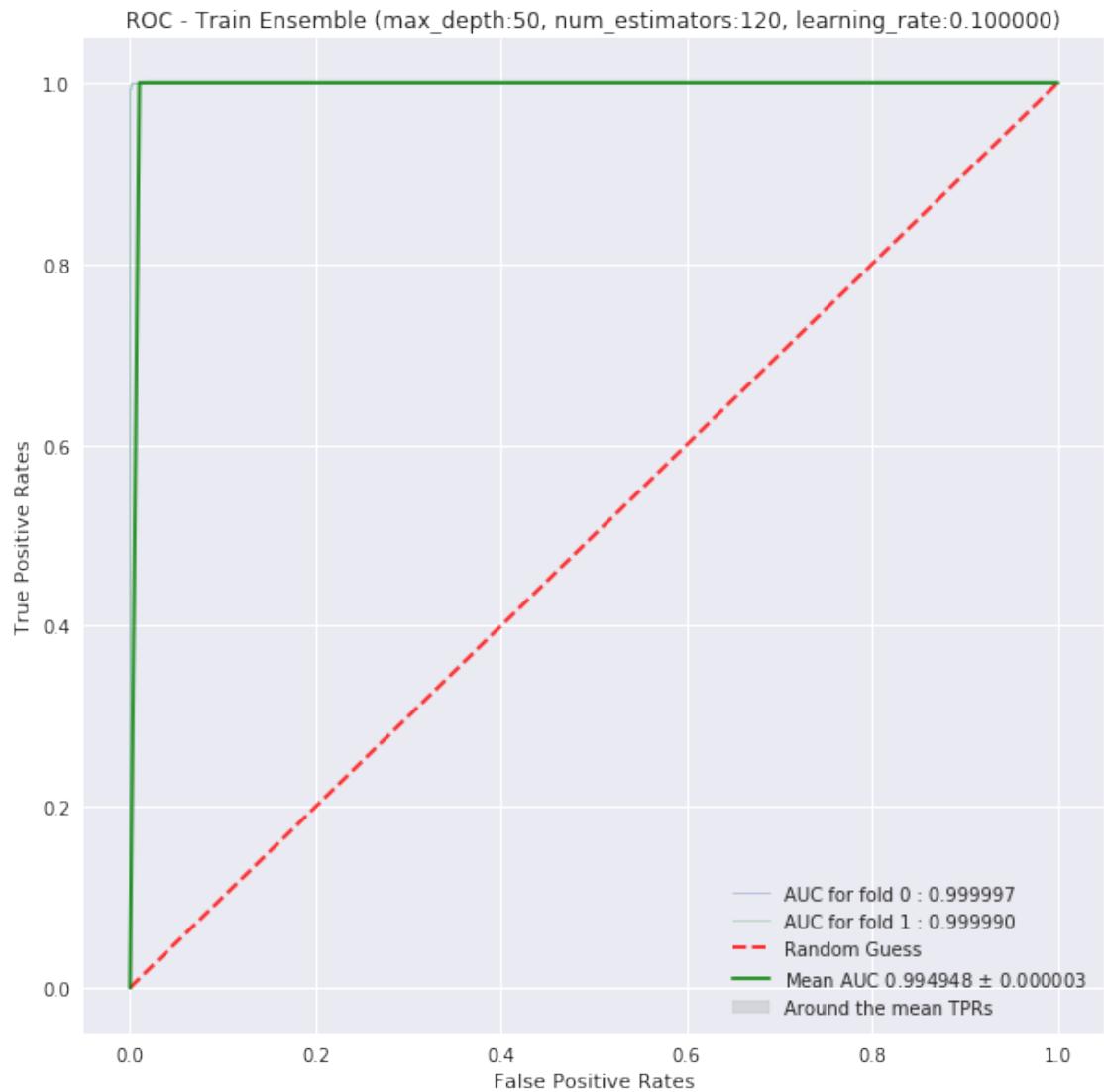
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

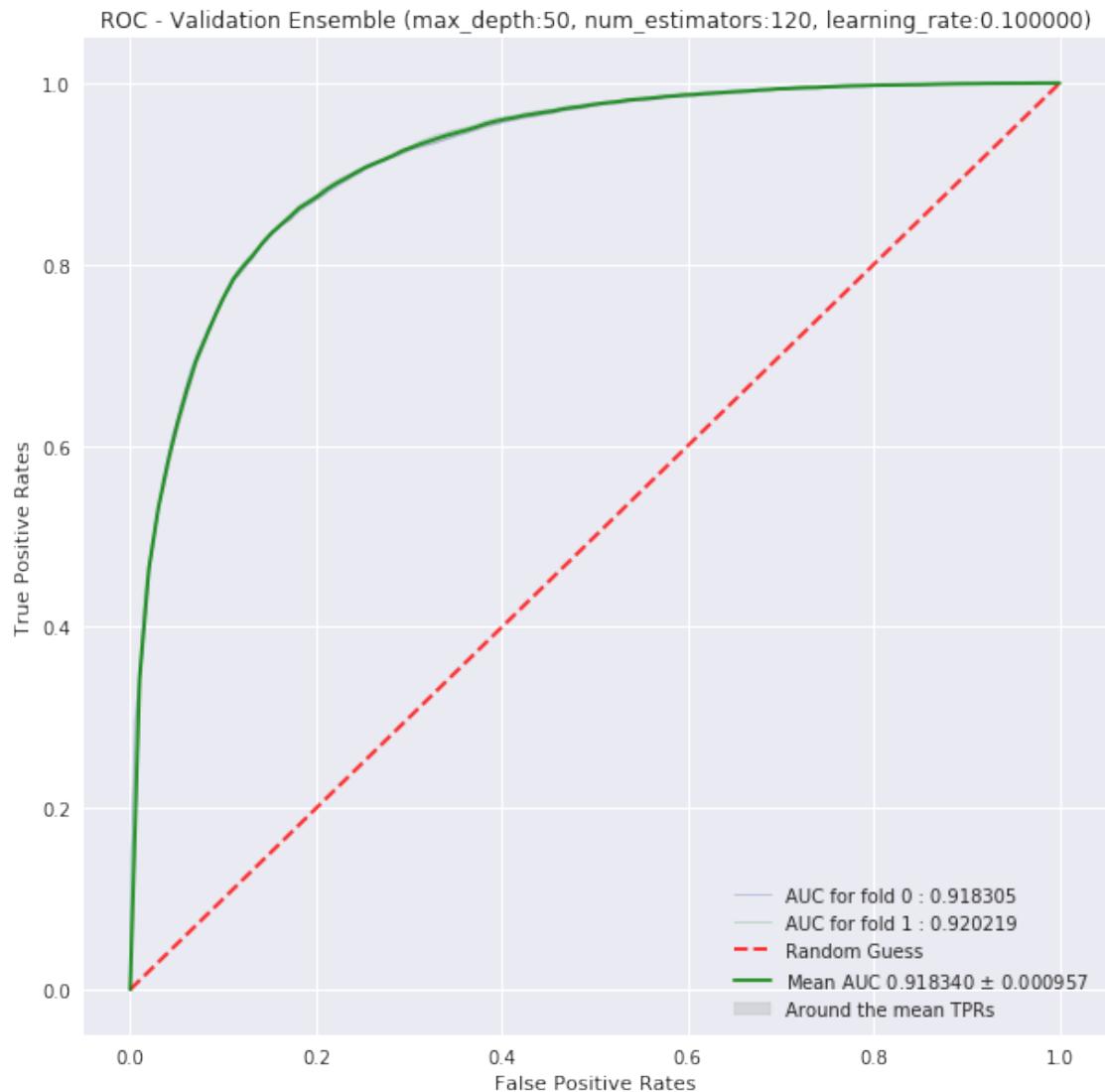
ROC - Train Ensemble (max\_depth:50, num\_estimators:70, learning\_rate:0.100000)





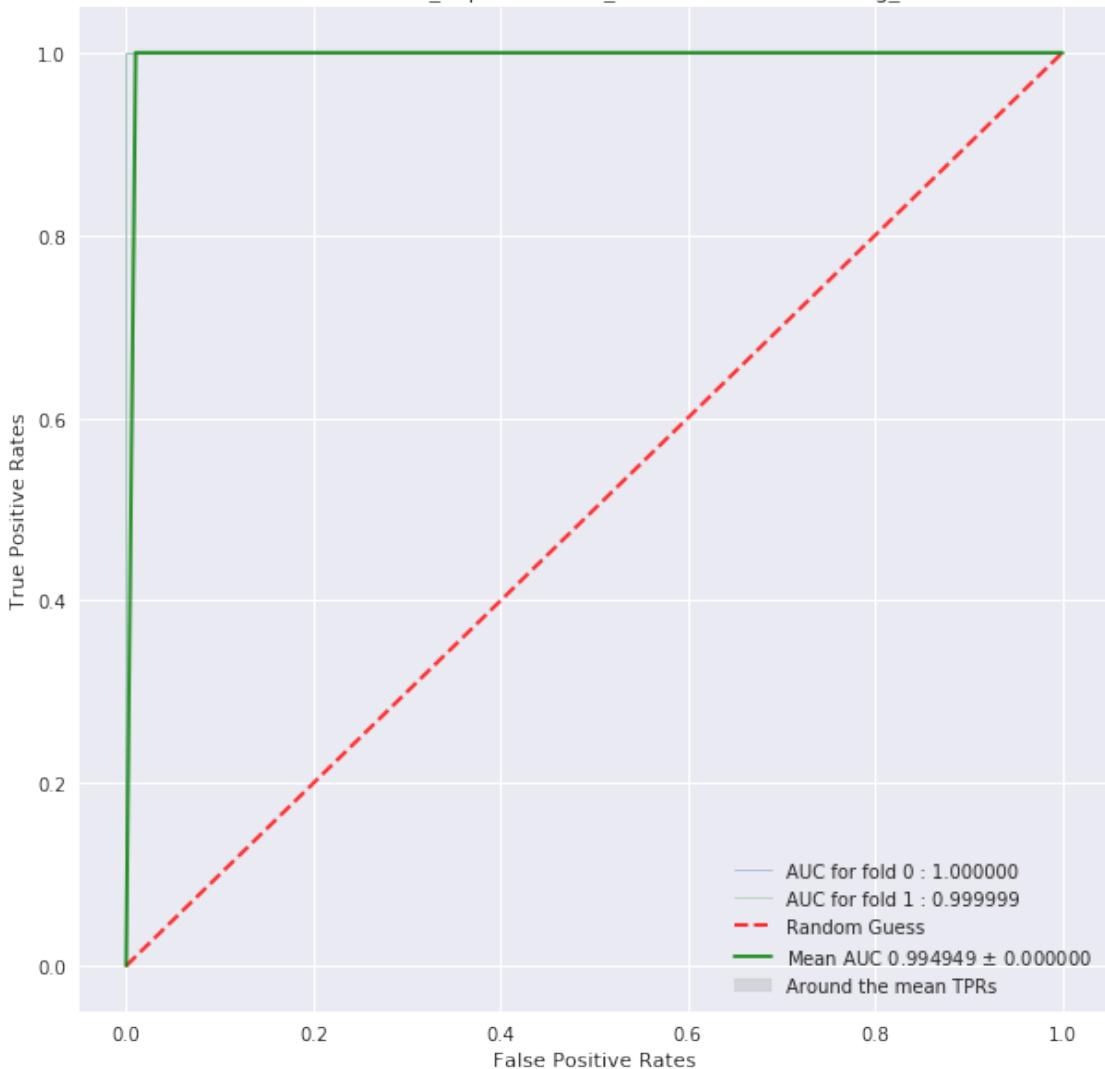
```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

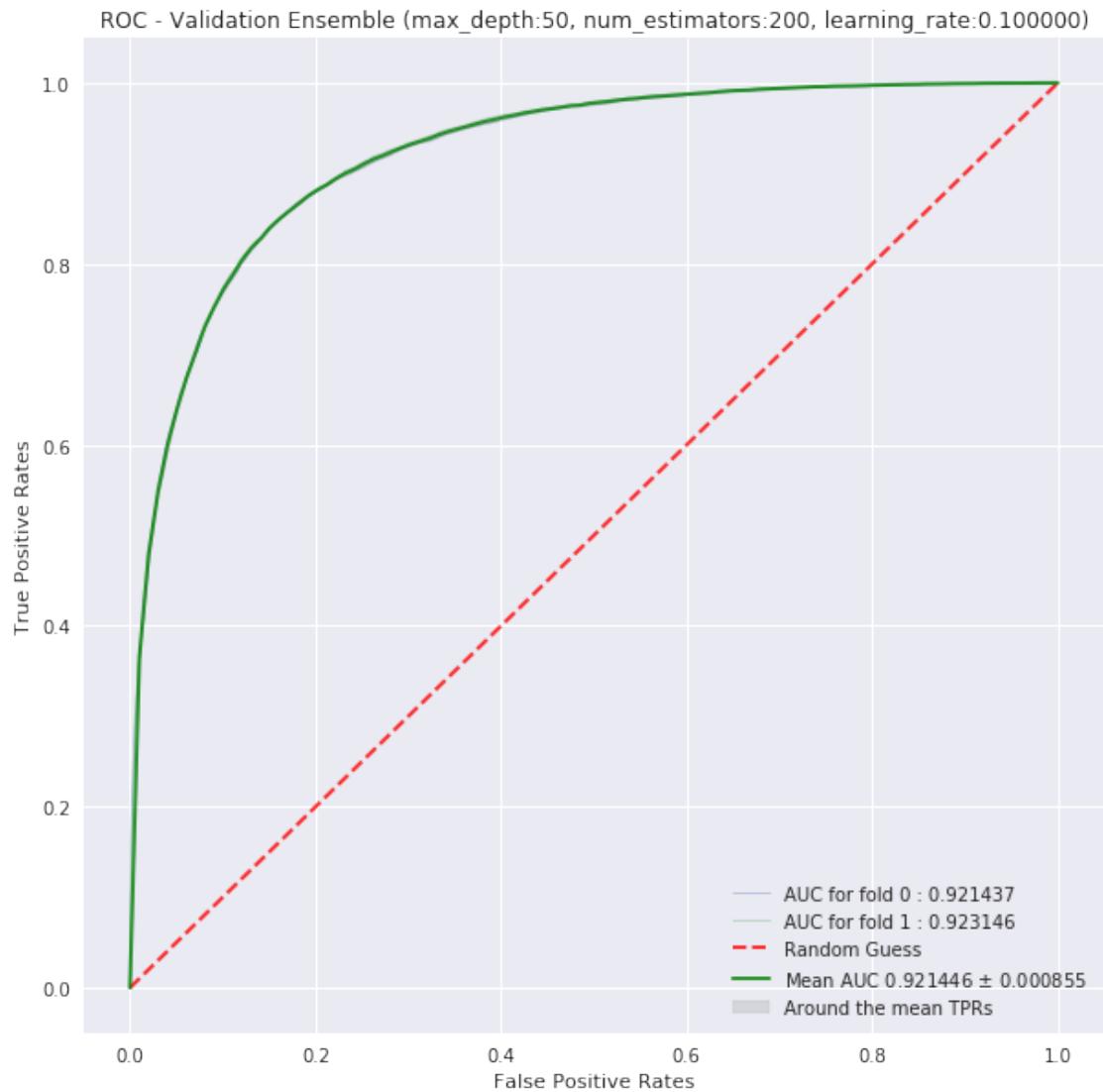




```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:200, learning\_rate:0.100000)






---



---



---

Train hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.831844
1	(2, 70, 0.1)	0.858788
2	(2, 120, 0.1)	0.884085
3	(2, 200, 0.1)	0.903413
4	(5, 40, 0.1)	0.893903
5	(5, 70, 0.1)	0.918920
6	(5, 120, 0.1)	0.939203
7	(5, 200, 0.1)	0.955832

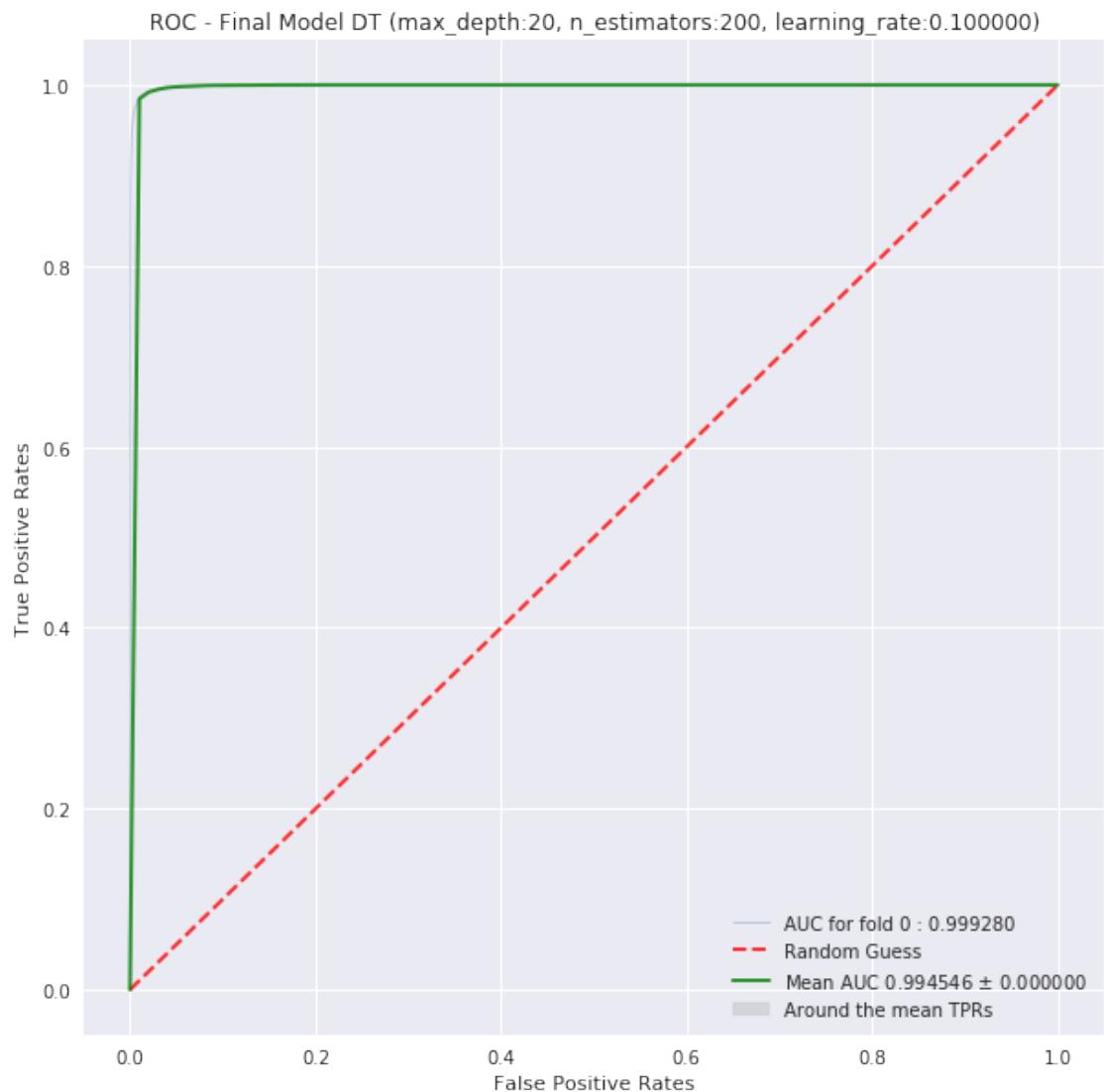
```
8   (20, 40, 0.1) 0.985196
9   (20, 70, 0.1) 0.991390
10  (20, 120, 0.1) 0.994060
11  (20, 200, 0.1) 0.994788
12  (50, 40, 0.1) 0.994677
13  (50, 70, 0.1) 0.994919
14  (50, 120, 0.1) 0.994948
15  (50, 200, 0.1) 0.994949
```

Validation hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.828421
1	(2, 70, 0.1)	0.853865
2	(2, 120, 0.1)	0.876889
3	(2, 200, 0.1)	0.893711
4	(5, 40, 0.1)	0.873963
5	(5, 70, 0.1)	0.893370
6	(5, 120, 0.1)	0.908004
7	(5, 200, 0.1)	0.917487
8	(20, 40, 0.1)	0.903617
9	(20, 70, 0.1)	0.914248
10	(20, 120, 0.1)	0.919908
11	(20, 200, 0.1)	0.922990
12	(50, 40, 0.1)	0.902975
13	(50, 70, 0.1)	0.913110
14	(50, 120, 0.1)	0.918340
15	(50, 200, 0.1)	0.921446

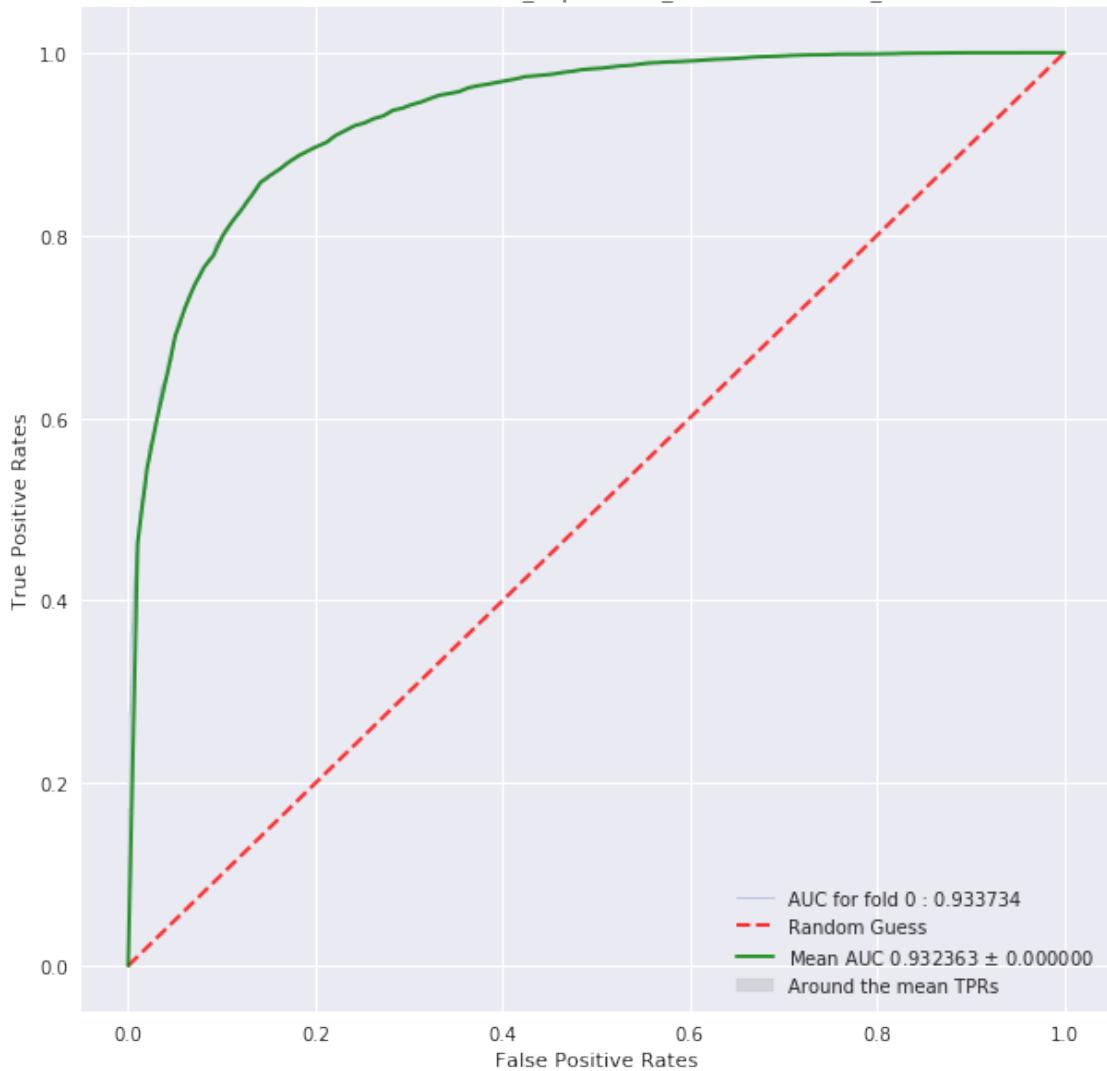
Best hyperparam value: (20, 200, 0.1)

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
if diff:
```

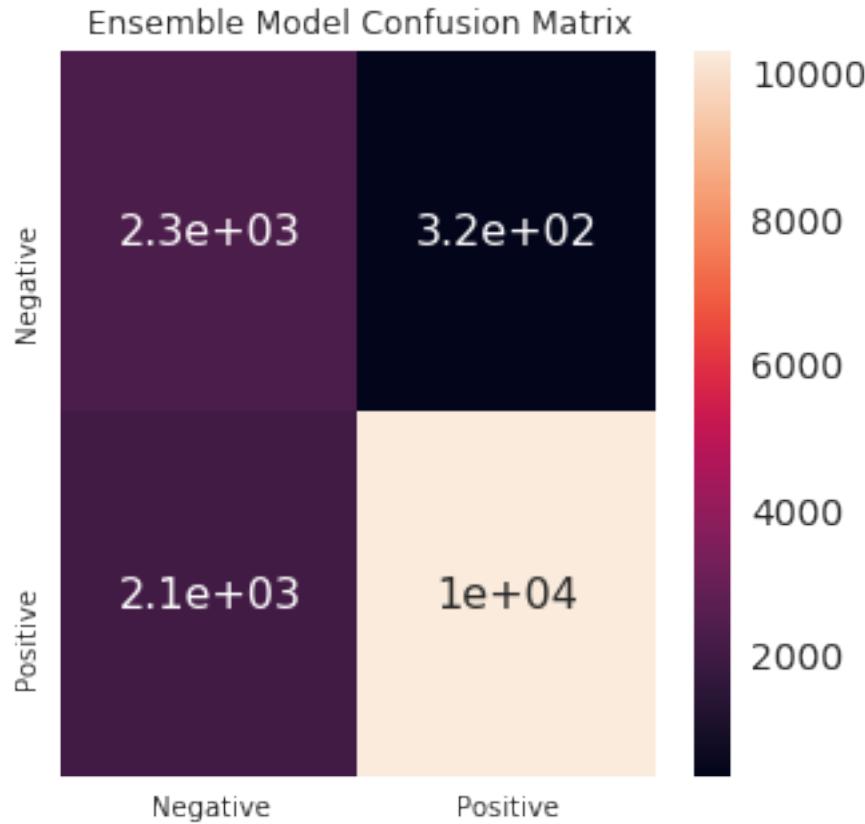


```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
if diff:
```

ROC - Final Model Ensemble (max\_depth:20, n\_estimators:200, lr\_rate:0.100000)



Test auc score 0.932363355825329



	Negative	Positive
Precision	0.525229	0.969549
Recall	0.876052	0.832876
Fscore	0.656725	0.896031
Support	2614.000000	12386.000000

#### 4.5.2 [B.2] Applying XGBOOST on TFIDF, SET 2

```
In [19]: # form two lists
    depth_list = [2, 5, 20, 50] # depends on size of dataset
    n_estimators_listt = [40, 70, 120, 200] # depends on size of dataset
    learning_rate_list = [0.1] # learning rate for XGB training

    # create a configuartion dictionary
    config_dict = {
        'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF/',
        'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF/t',
        'train_size' : 40000,
        'test_size' : 15000,
        'hyperparam_list' : list(product(depth_list, n_estimators_list, learning_rate_list))}
```

```

        'implementation': 'xgb' # 'xgb' or 'rf'
    }

In [20]: # read the train, test data and preprocess it
train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                                               scaling=True,
                                                               dim_reduction=True)

# train and validate the model
model = train_and_validate_model(config_dict, train_features, train_labels)

# test and evaluate the model
ptabe_entry_b2 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

Train df shape (40000, 503)
Class label distribution in train df:
0    20024
1    19976
Name: Label, dtype: int64
Test df shape (15000, 503)
Class label distribution in test df:
1    12386
0    2614
Name: Label, dtype: int64
Shape of -> train features :40000,501, test features: 15000,501
Shape of -> train labels :40000, test labels: 15000
=====

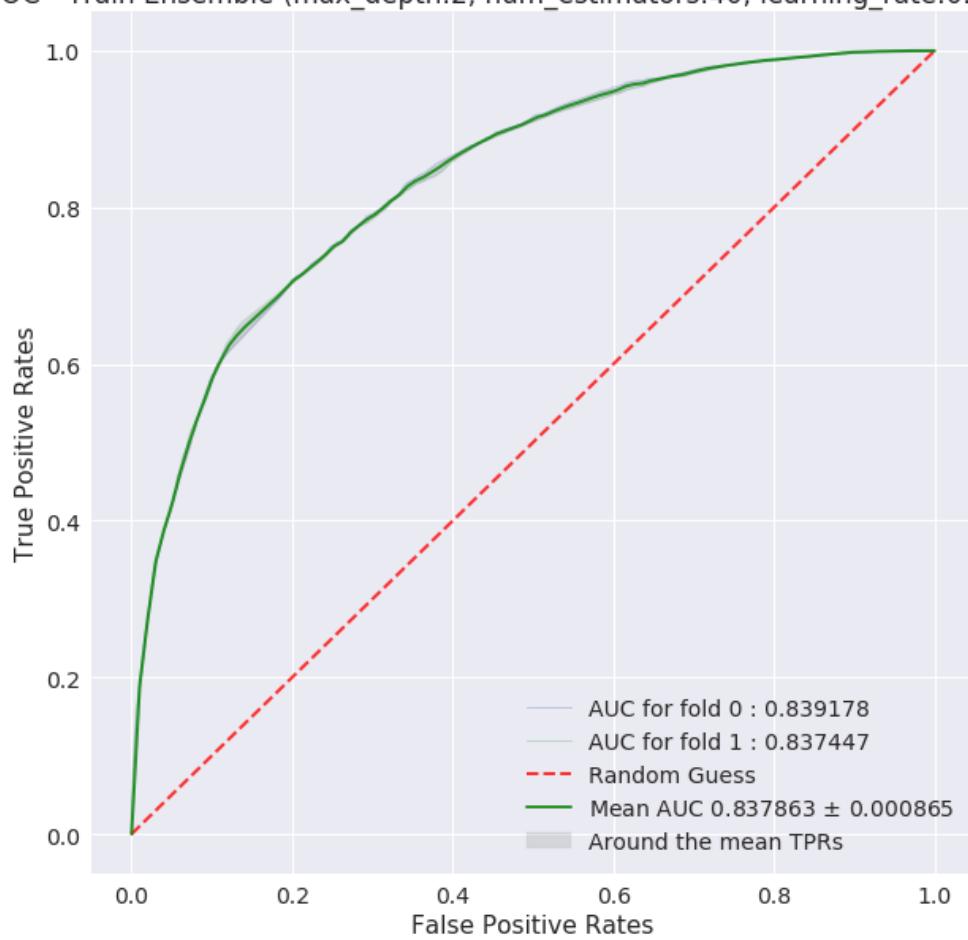
```

```

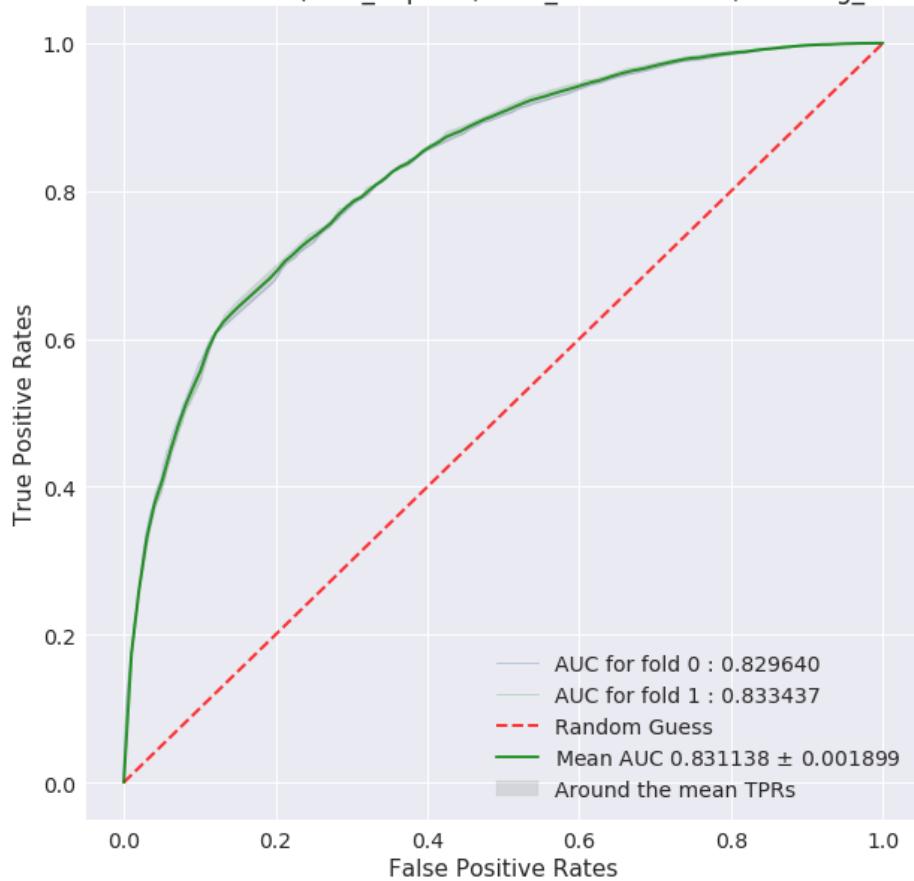
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The 'labels' parameter is deprecated. Use 'y' instead.
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The 'labels' parameter is deprecated. Use 'y' instead.
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The 'labels' parameter is deprecated. Use 'y' instead.
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: The 'labels' parameter is deprecated. Use 'y' instead.
  if diff:

```

ROC - Train Ensemble (max\_depth:2, num\_estimators:40, learning\_rate:0.100000)

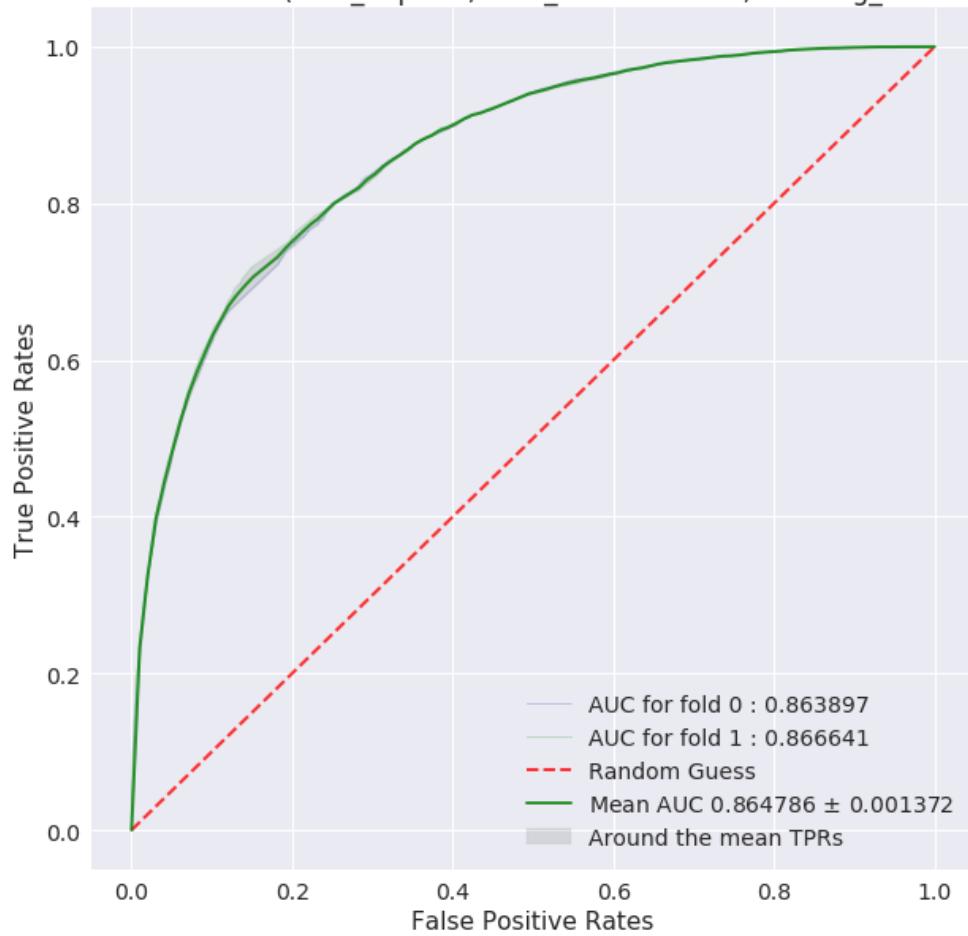


ROC - Validation Ensemble (max\_depth:2, num\_estimators:40, learning\_rate:0.100000)

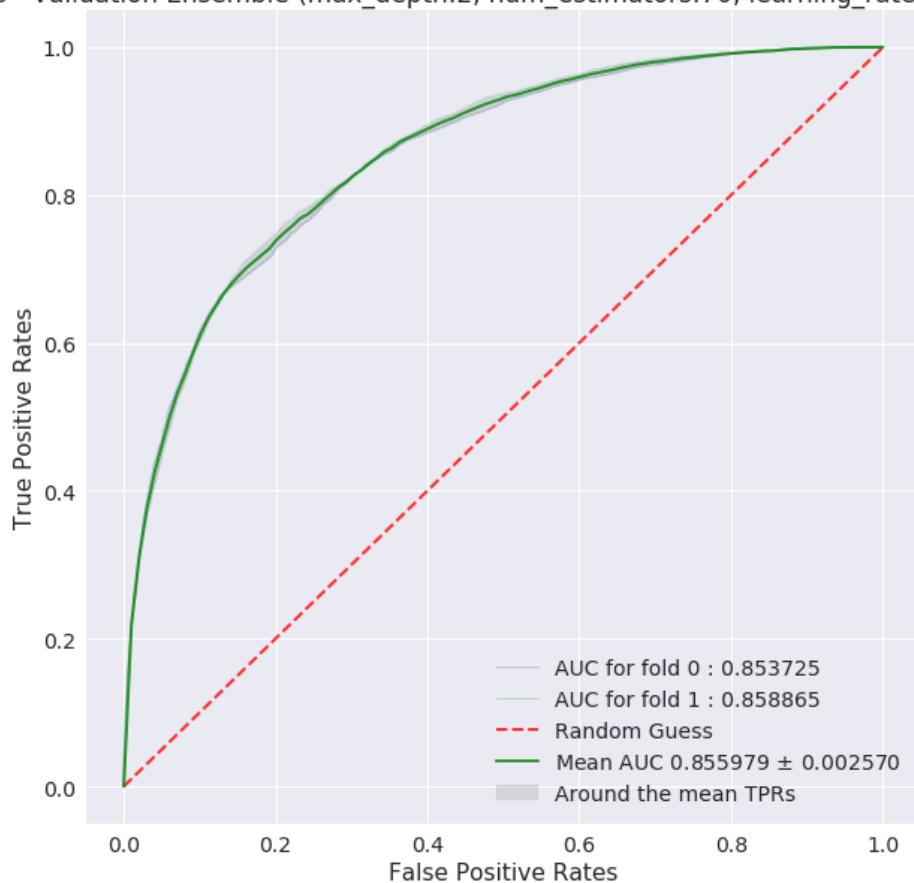


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:70, learning\_rate:0.100000)

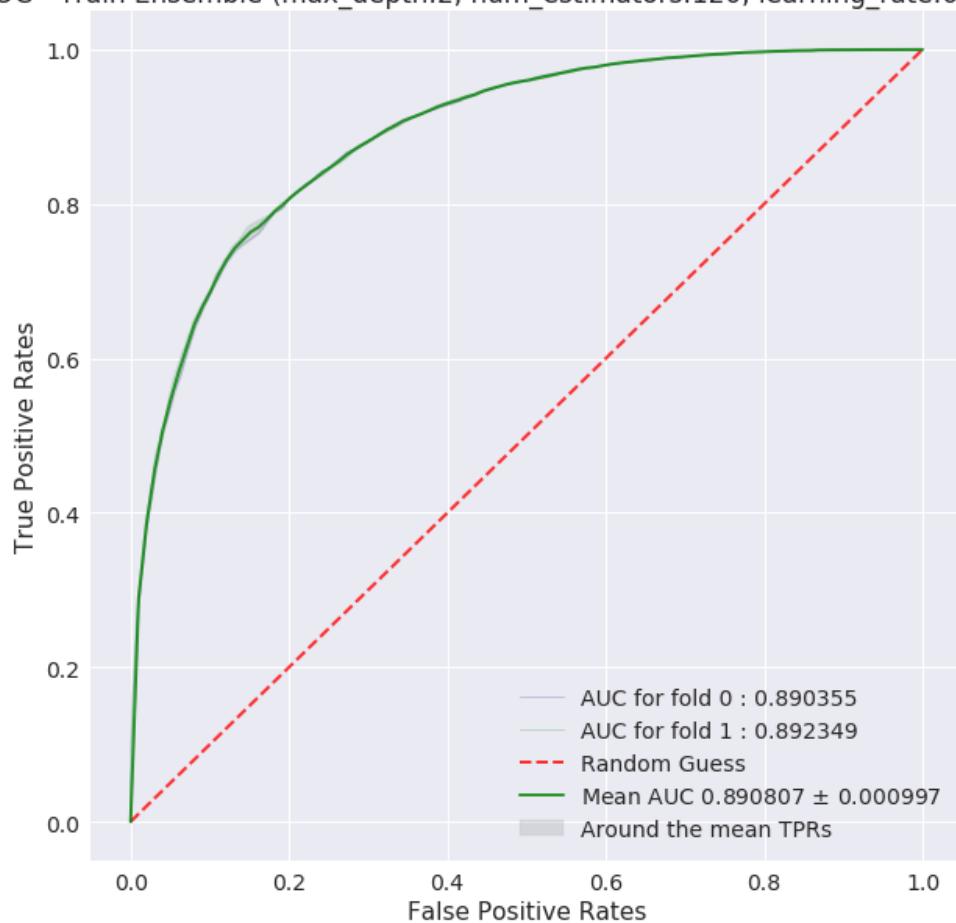


ROC - Validation Ensemble (max\_depth:2, num\_estimators:70, learning\_rate:0.100000)

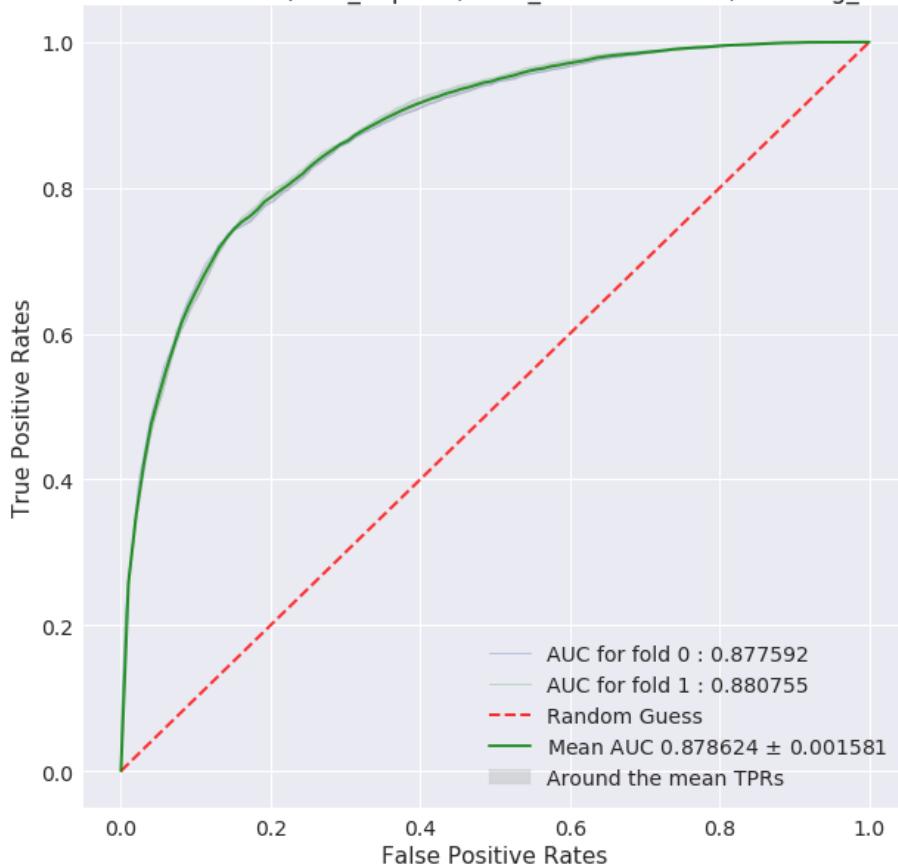


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:120, learning\_rate:0.100000)

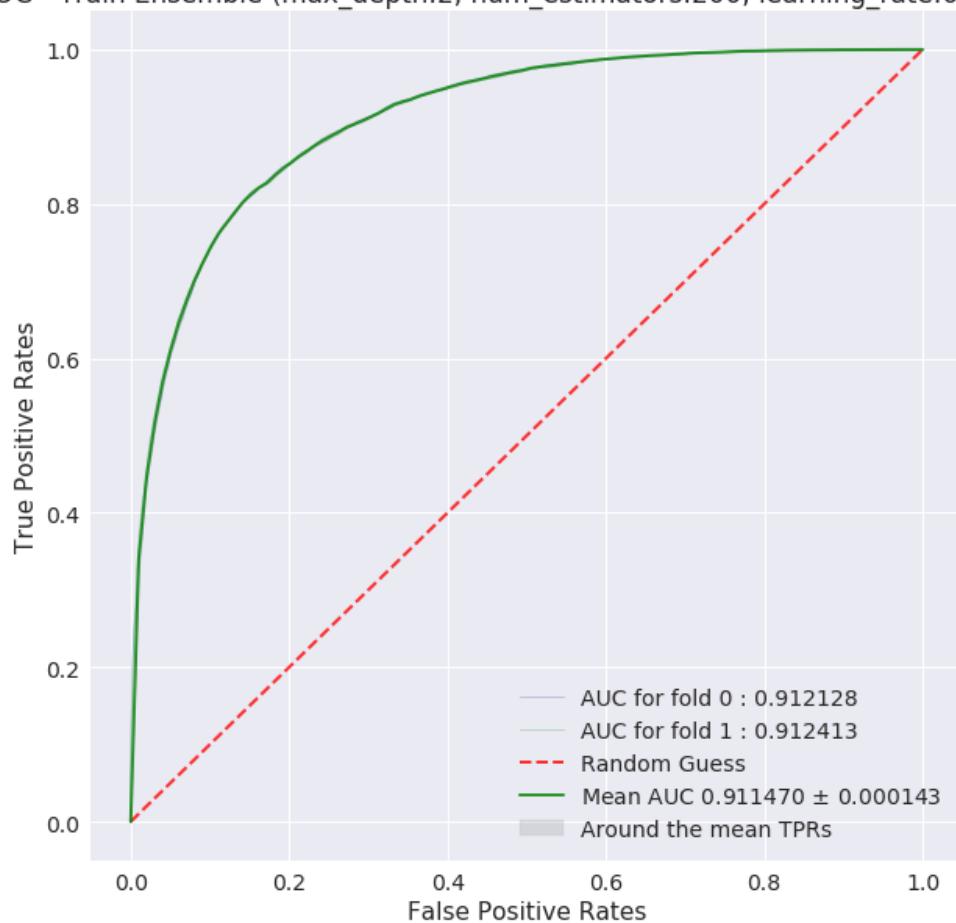


ROC - Validation Ensemble (max\_depth:2, num\_estimators:120, learning\_rate:0.100000)

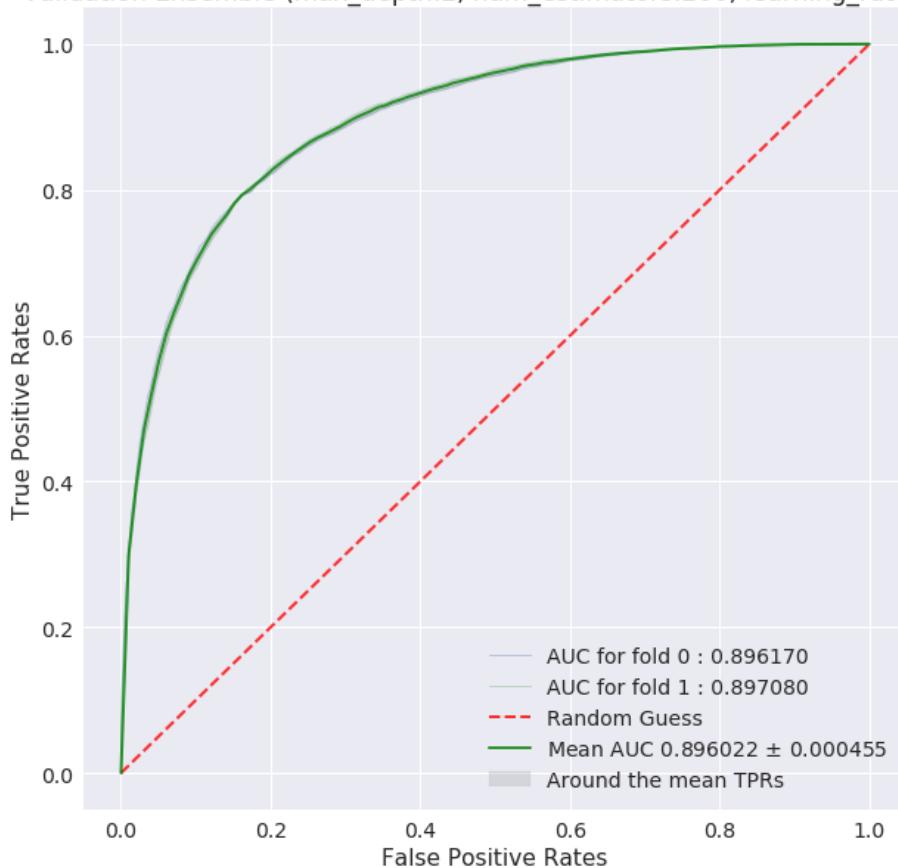


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:200, learning\_rate:0.100000)

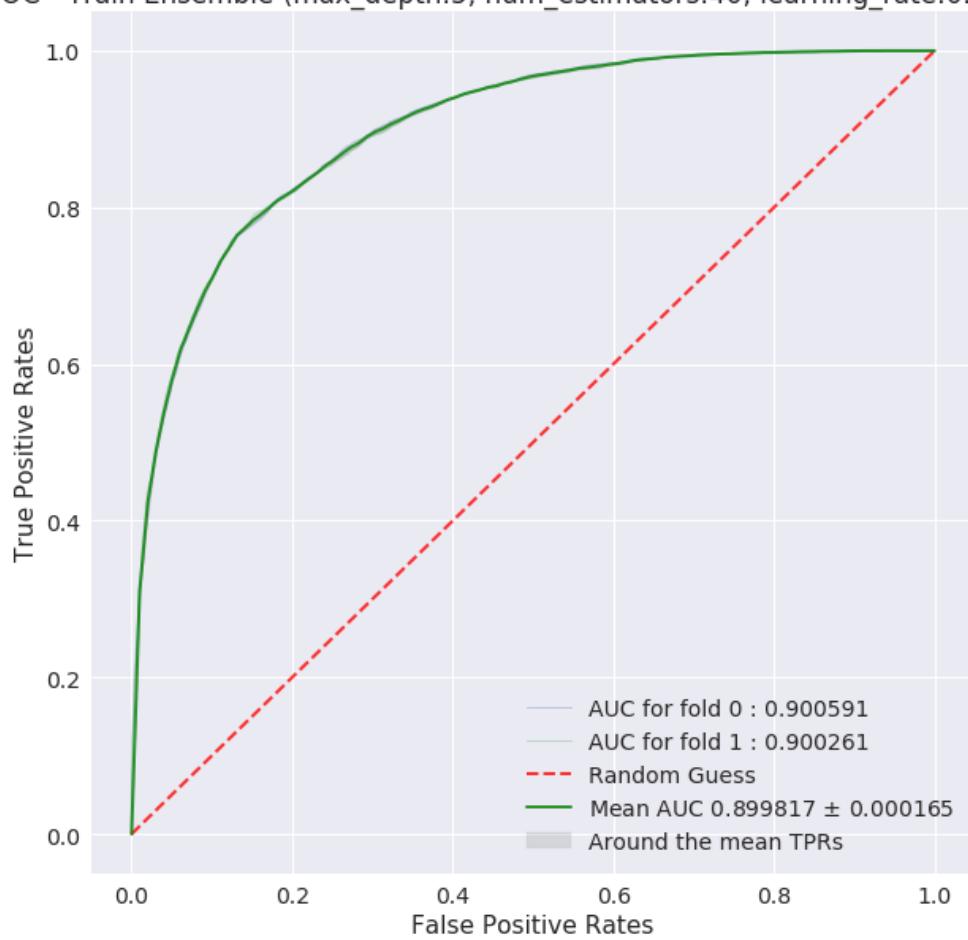


ROC - Validation Ensemble (max\_depth:2, num\_estimators:200, learning\_rate:0.100000)

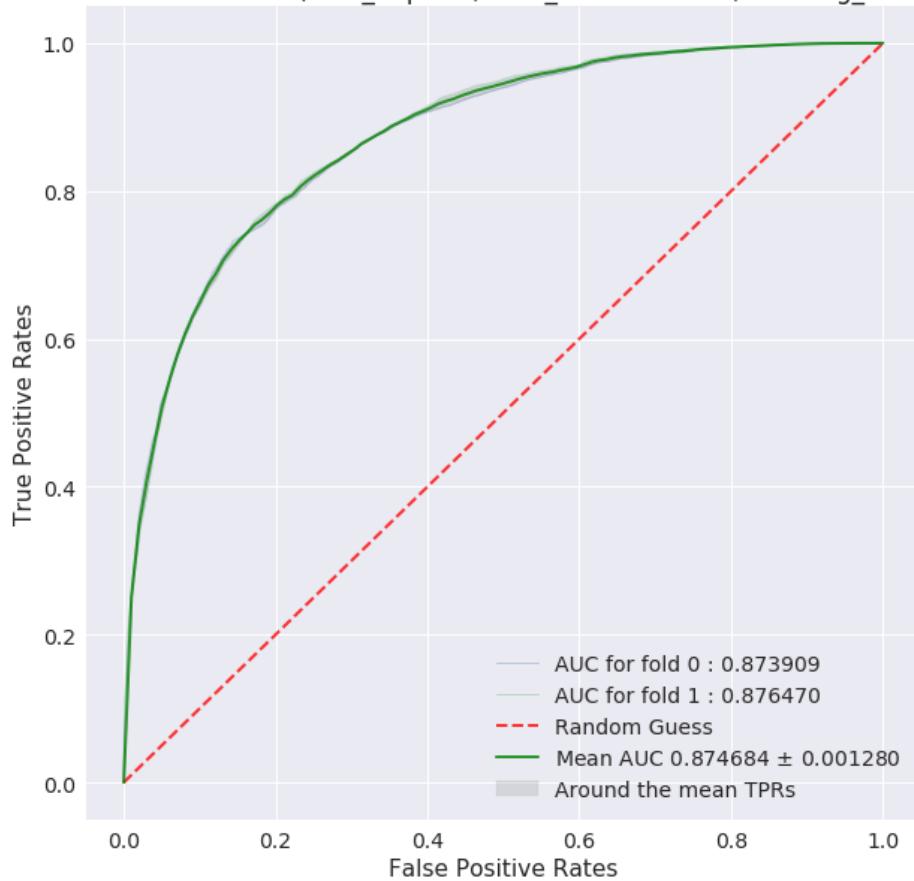


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:40, learning\_rate:0.100000)

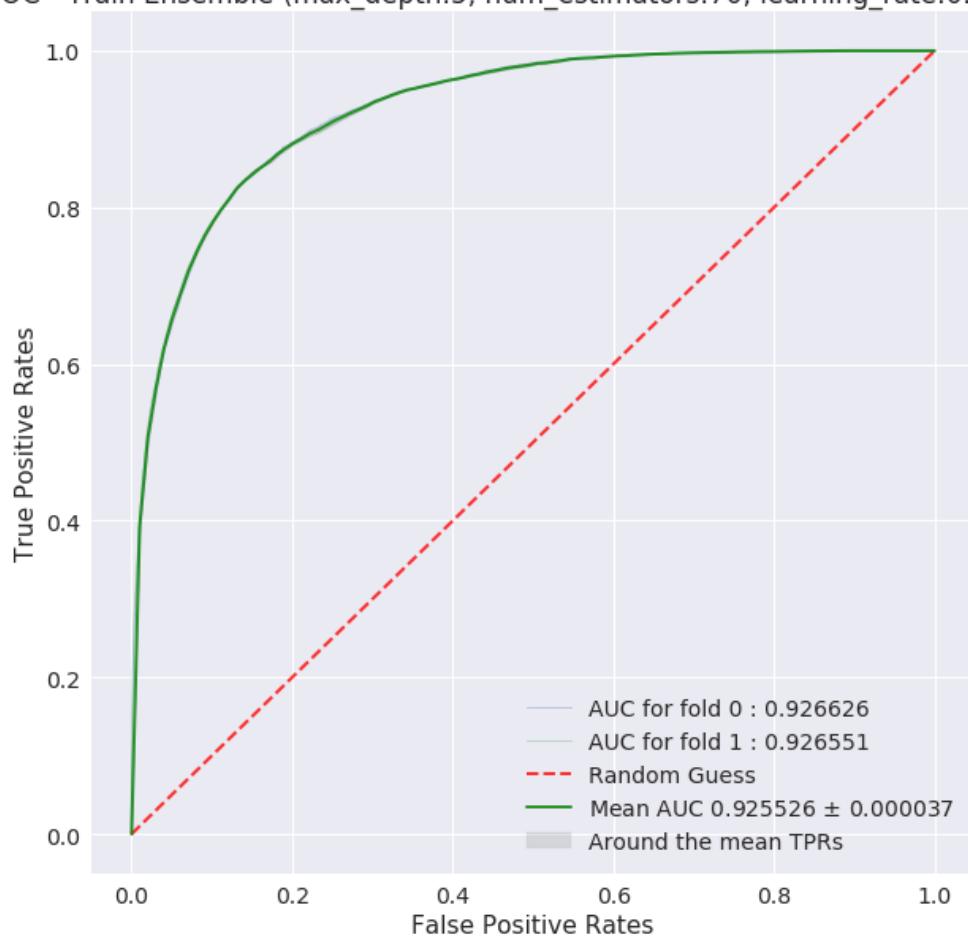


ROC - Validation Ensemble (max\_depth:5, num\_estimators:40, learning\_rate:0.100000)

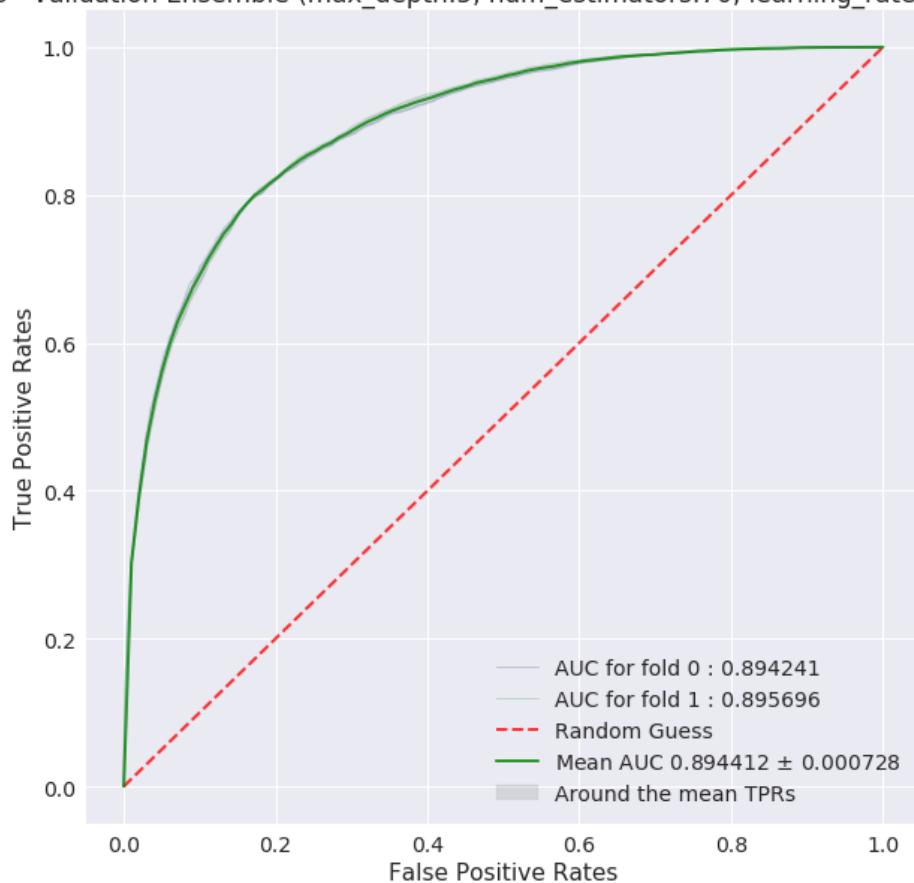


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:70, learning\_rate:0.100000)

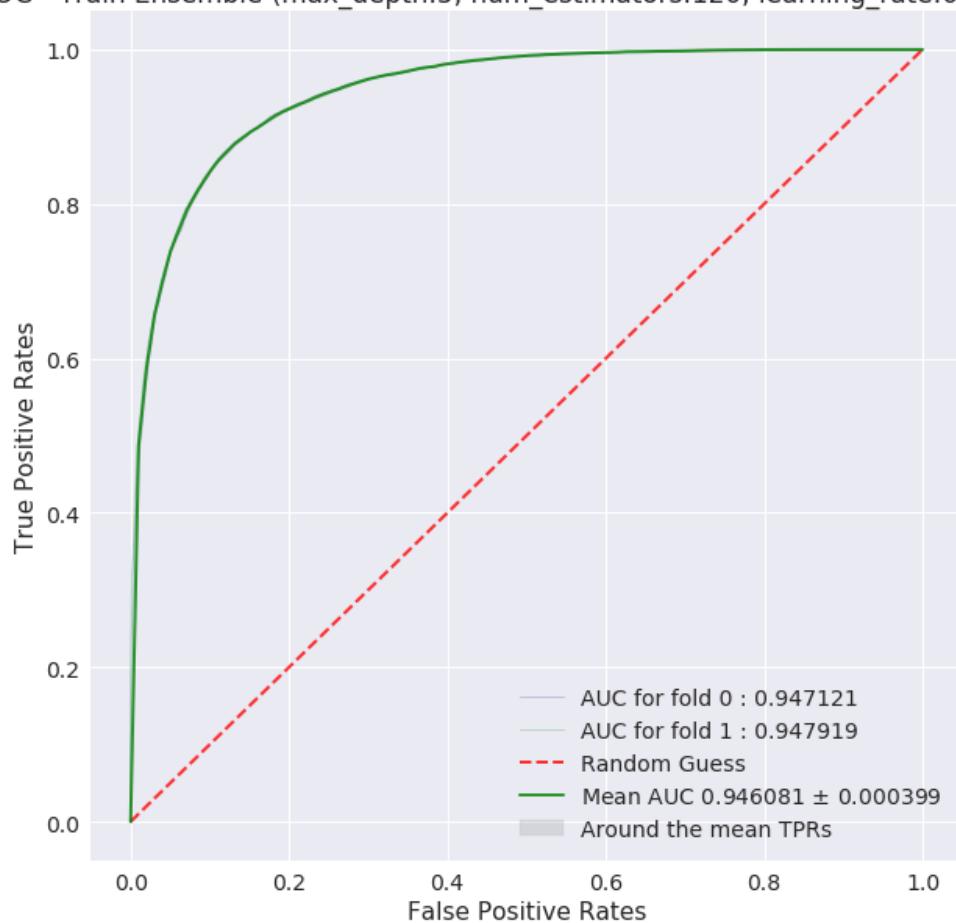


ROC - Validation Ensemble (max\_depth:5, num\_estimators:70, learning\_rate:0.100000)

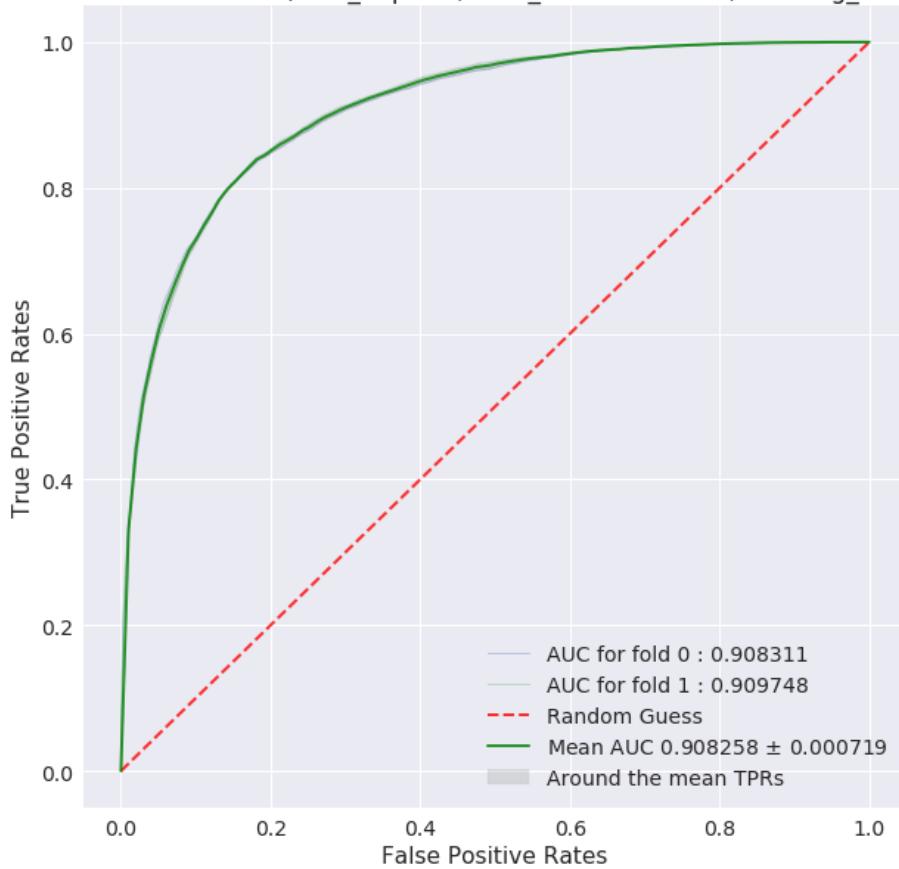


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:120, learning\_rate:0.100000)

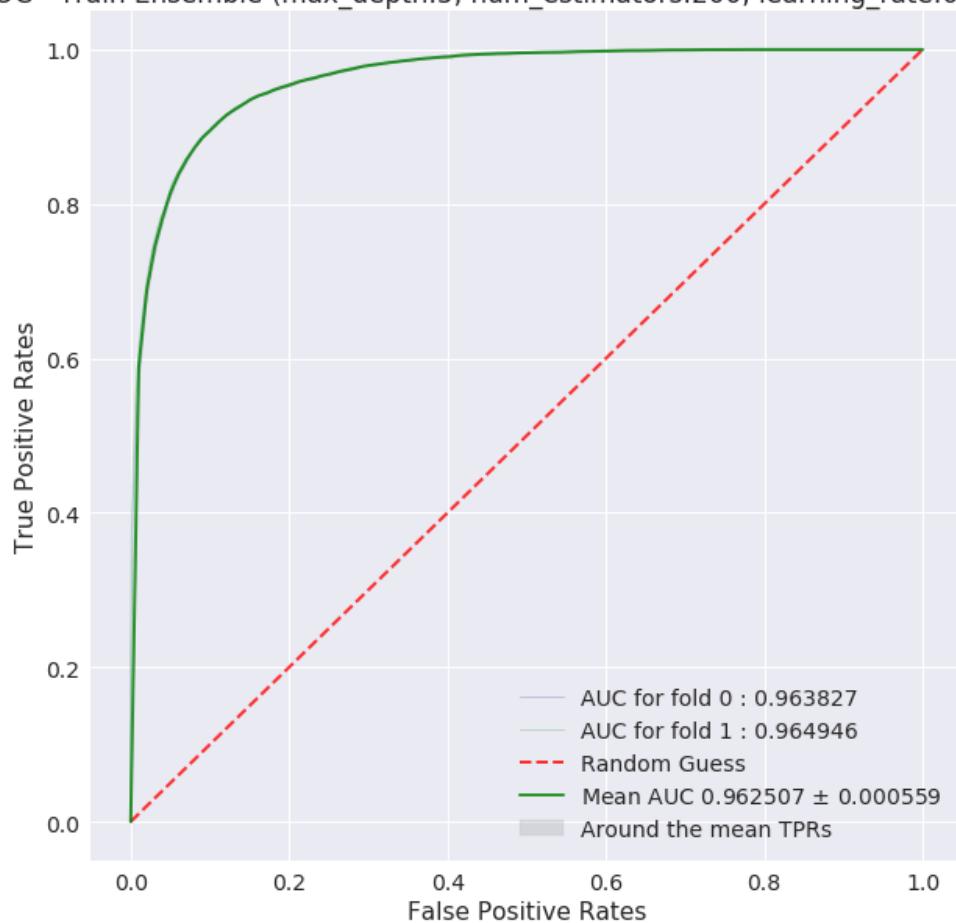


ROC - Validation Ensemble (max\_depth:5, num\_estimators:120, learning\_rate:0.100000)

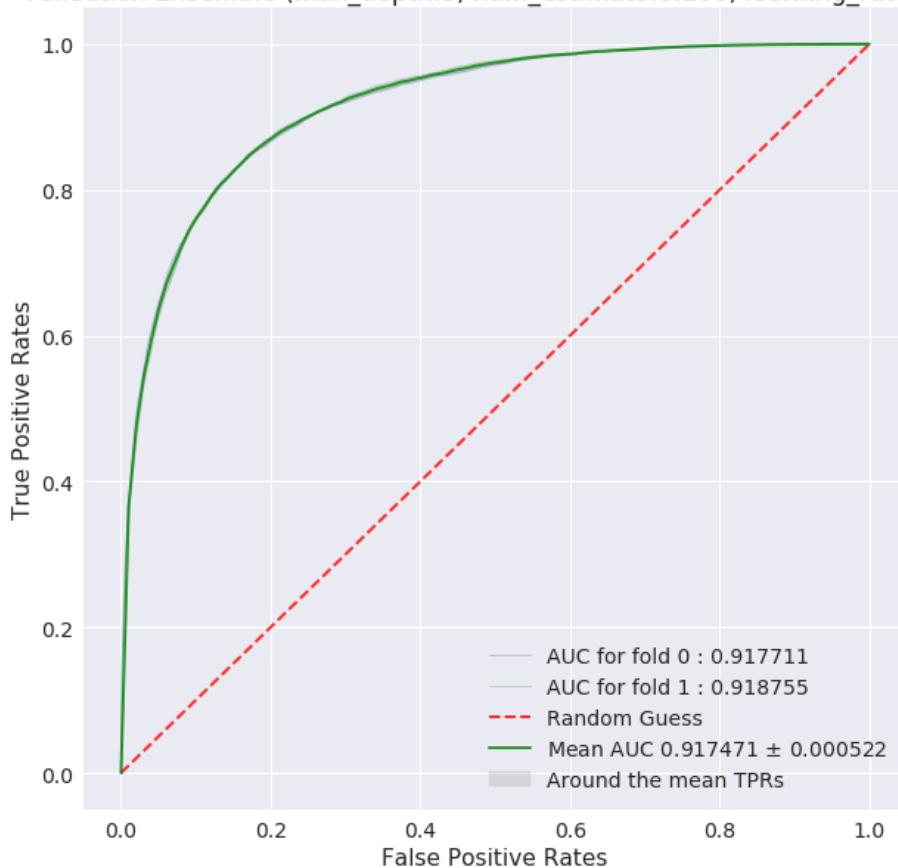


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:200, learning\_rate:0.100000)

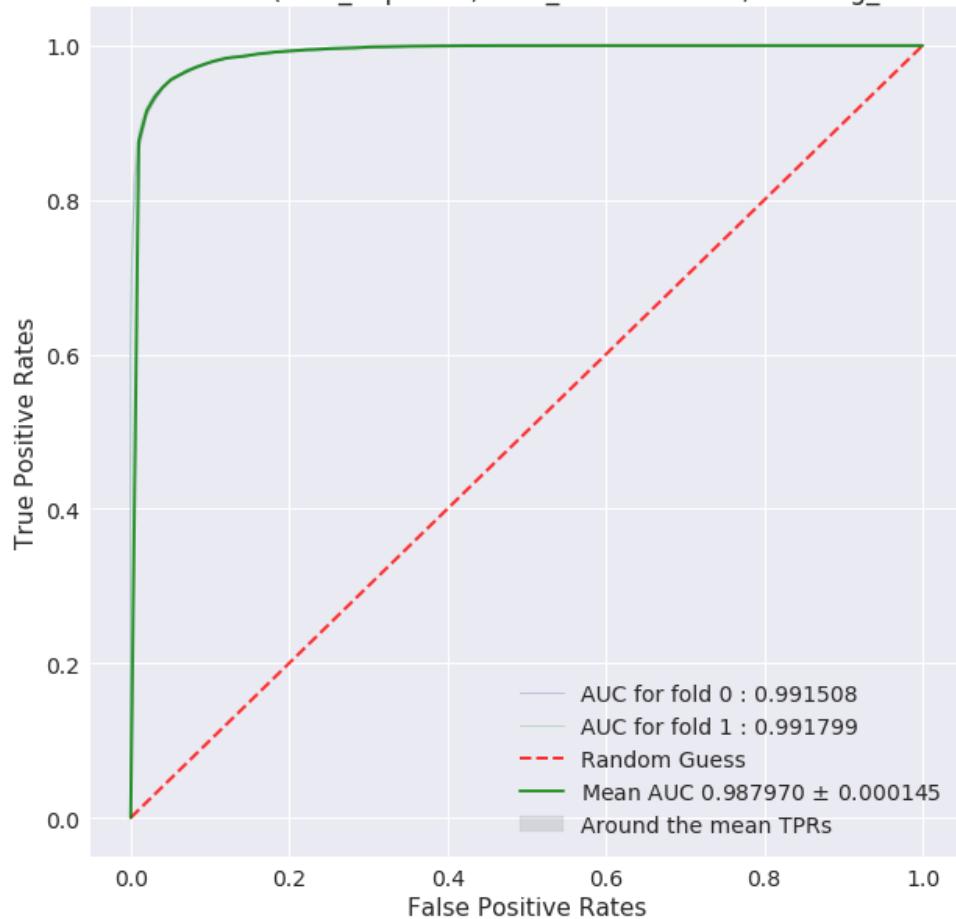


ROC - Validation Ensemble (max\_depth:5, num\_estimators:200, learning\_rate:0.100000)

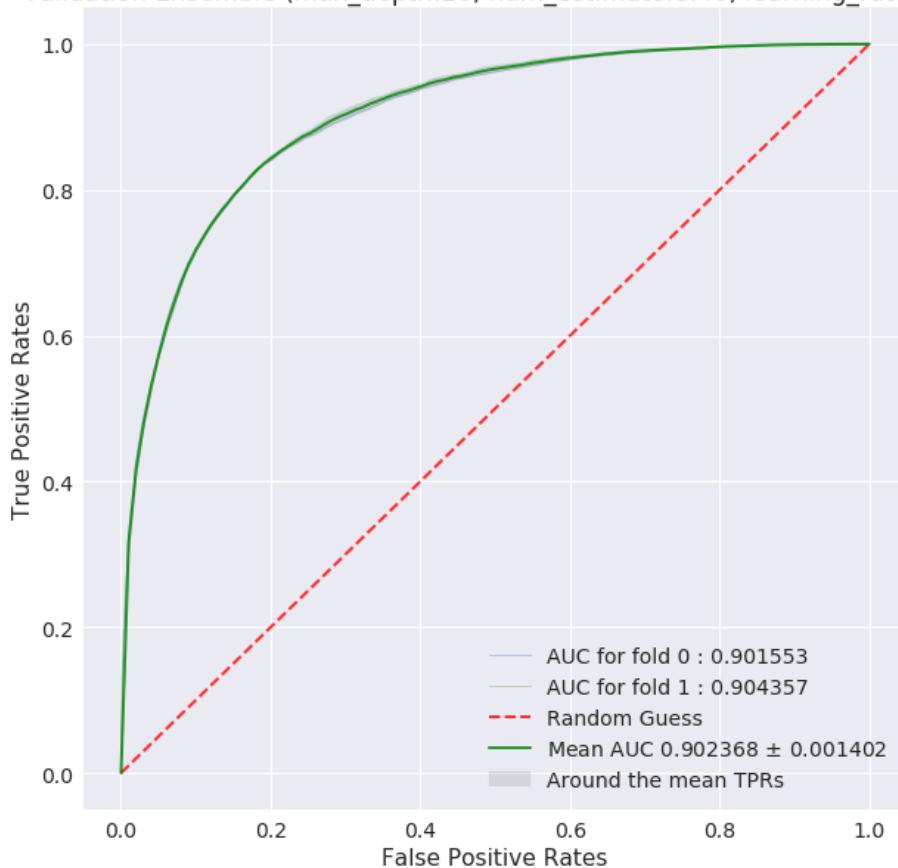


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:40, learning\_rate:0.100000)

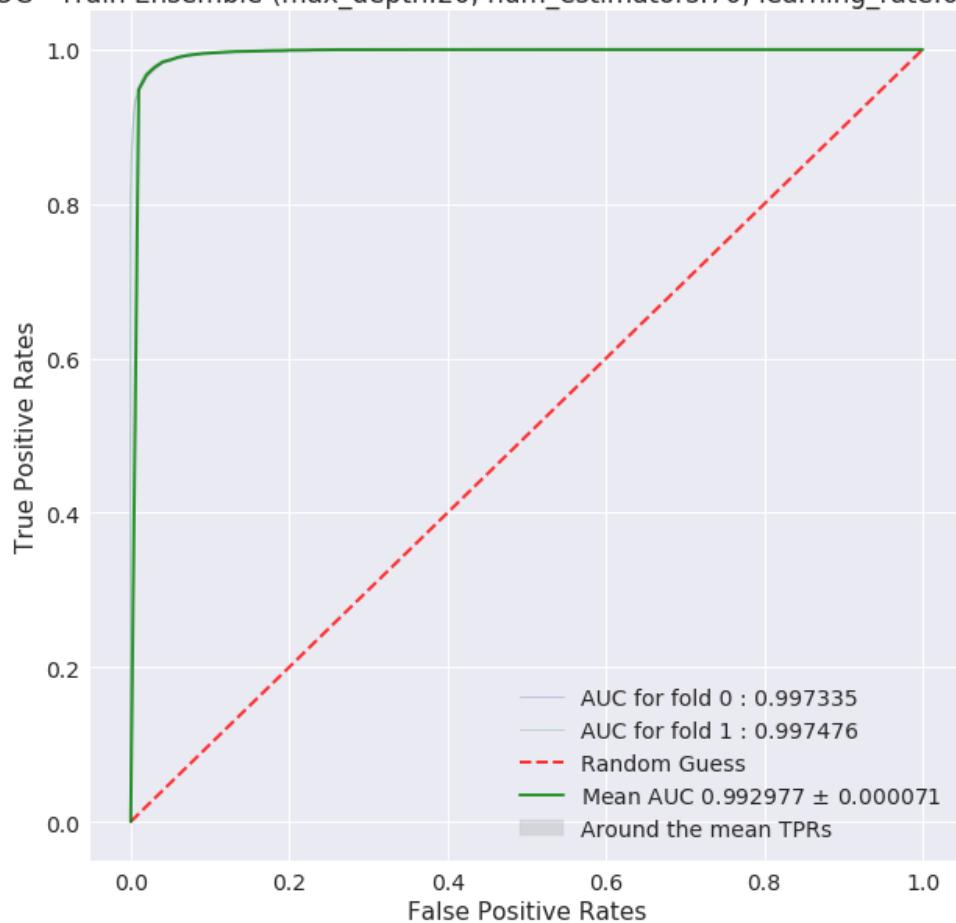


ROC - Validation Ensemble (max\_depth:20, num\_estimators:40, learning\_rate:0.100000)

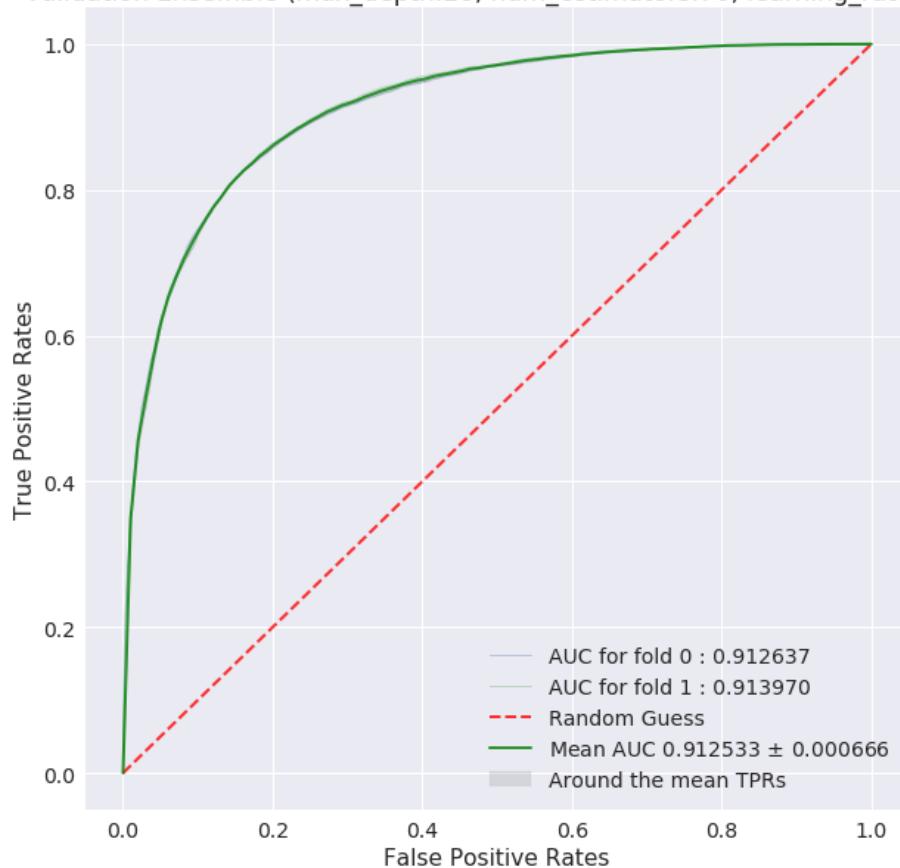


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:70, learning\_rate:0.100000)

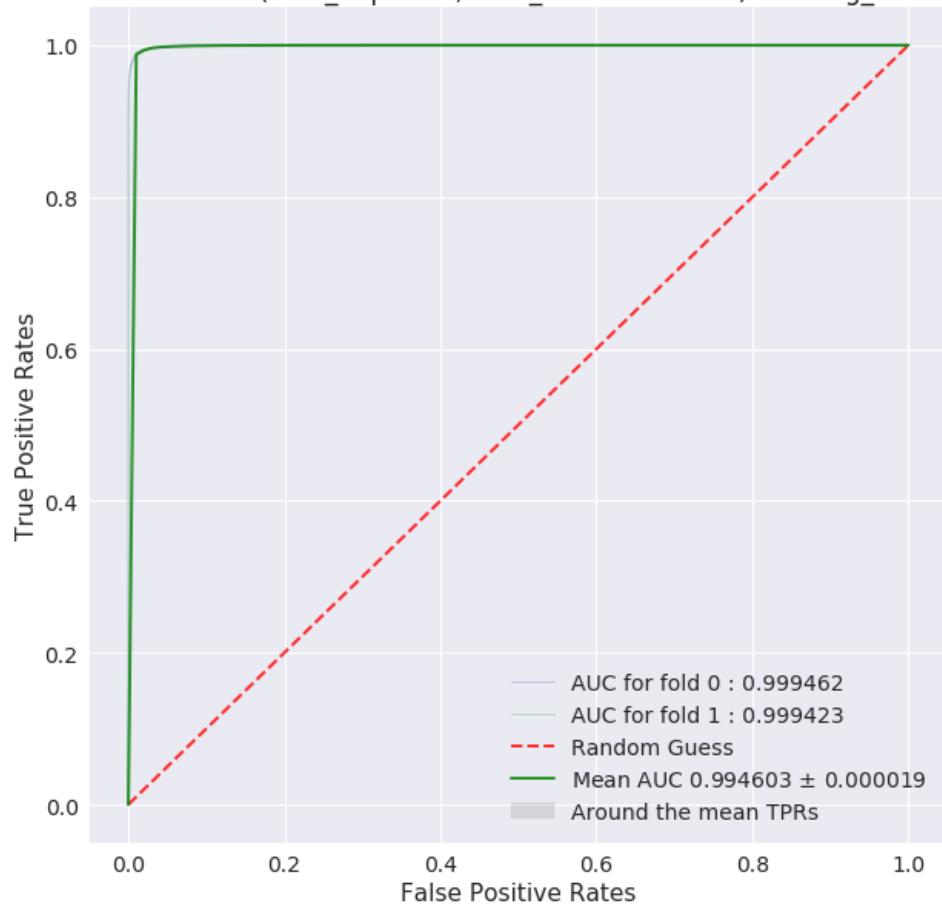


ROC - Validation Ensemble (max\_depth:20, num\_estimators:70, learning\_rate:0.100000)

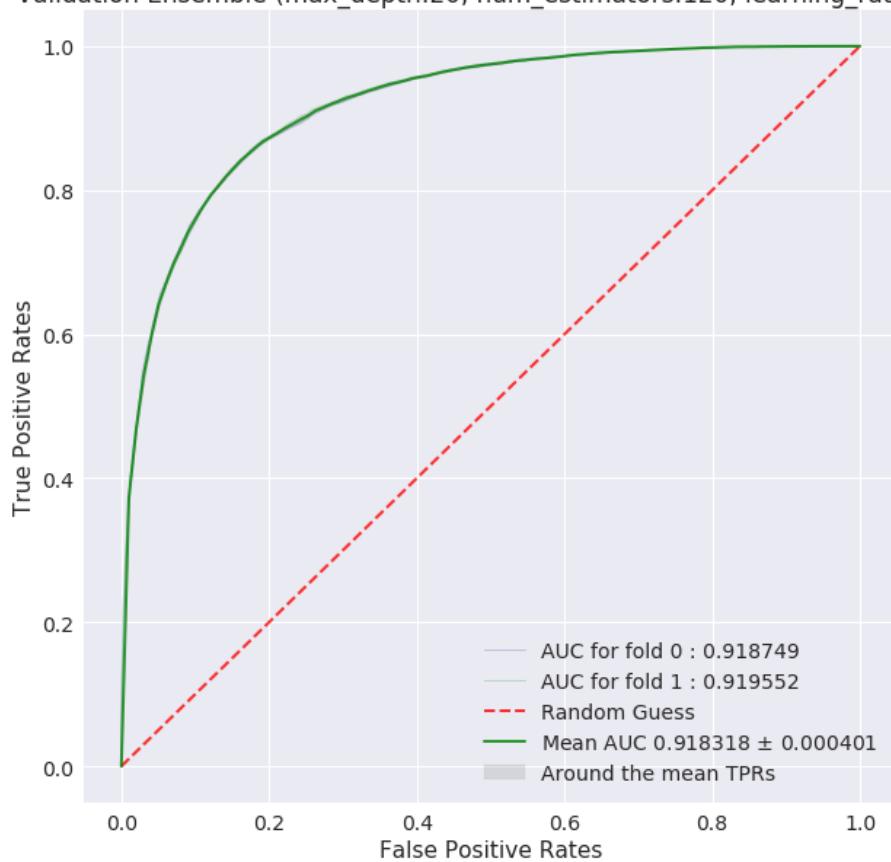


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:120, learning\_rate:0.100000)



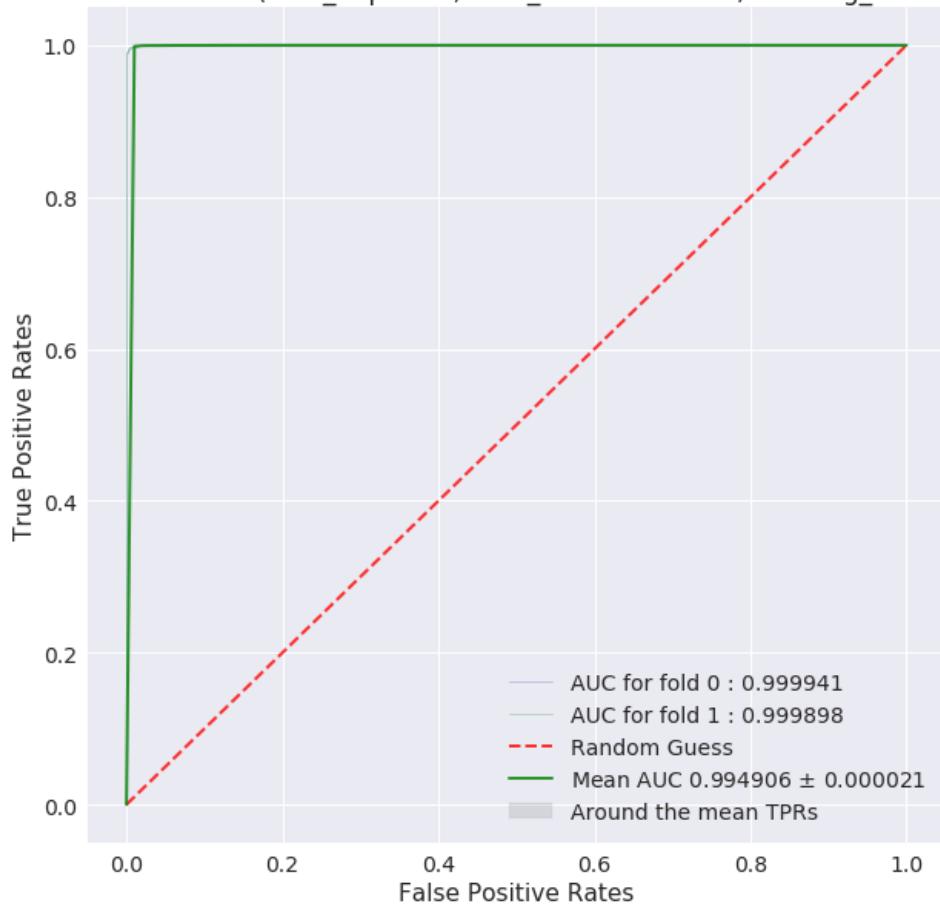
ROC - Validation Ensemble (max\_depth:20, num\_estimators:120, learning\_rate:0.100000)



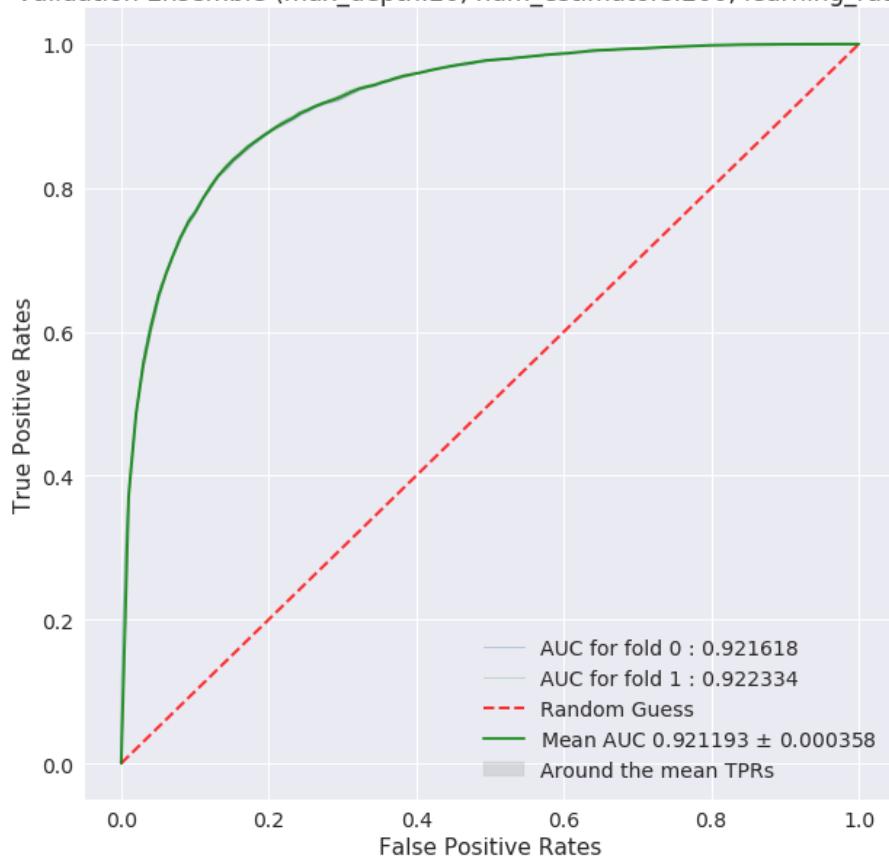
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If labels are present as a column in X and y, their order in X must be consistent across all columns. This will be an error in a future version of pandas. To silence this warning, use pd.concat to make sure labels are consistently aligned.
    if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If labels are present as a column in X and y, their order in X must be consistent across all columns. This will be an error in a future version of pandas. To silence this warning, use pd.concat to make sure labels are consistently aligned.
    if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If labels are present as a column in X and y, their order in X must be consistent across all columns. This will be an error in a future version of pandas. To silence this warning, use pd.concat to make sure labels are consistently aligned.
    if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: If labels are present as a column in X and y, their order in X must be consistent across all columns. This will be an error in a future version of pandas. To silence this warning, use pd.concat to make sure labels are consistently aligned.
    if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:200, learning\_rate:0.100000)



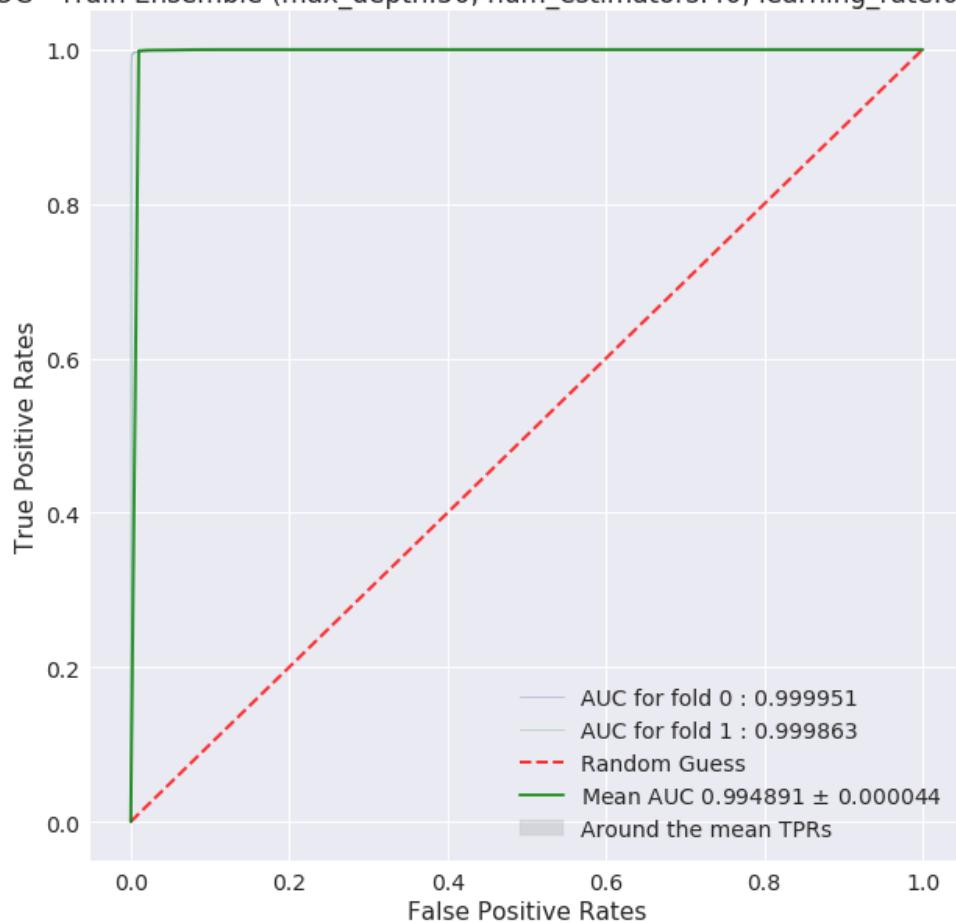
ROC - Validation Ensemble (max\_depth:20, num\_estimators:200, learning\_rate:0.100000)



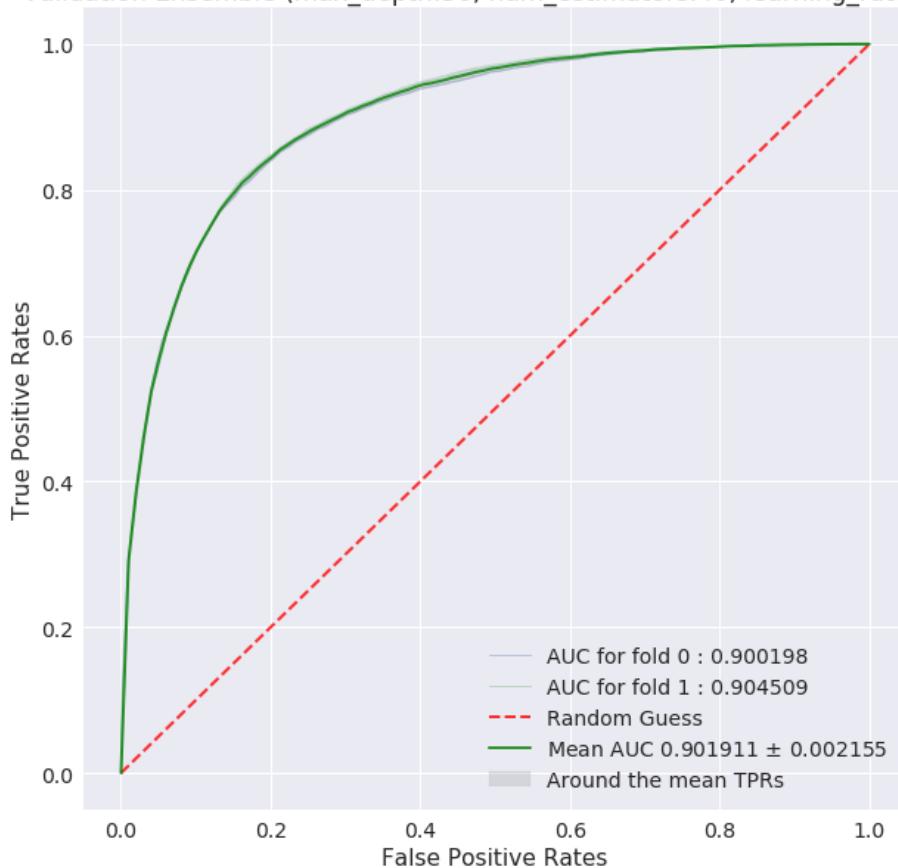
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:40, learning\_rate:0.100000)

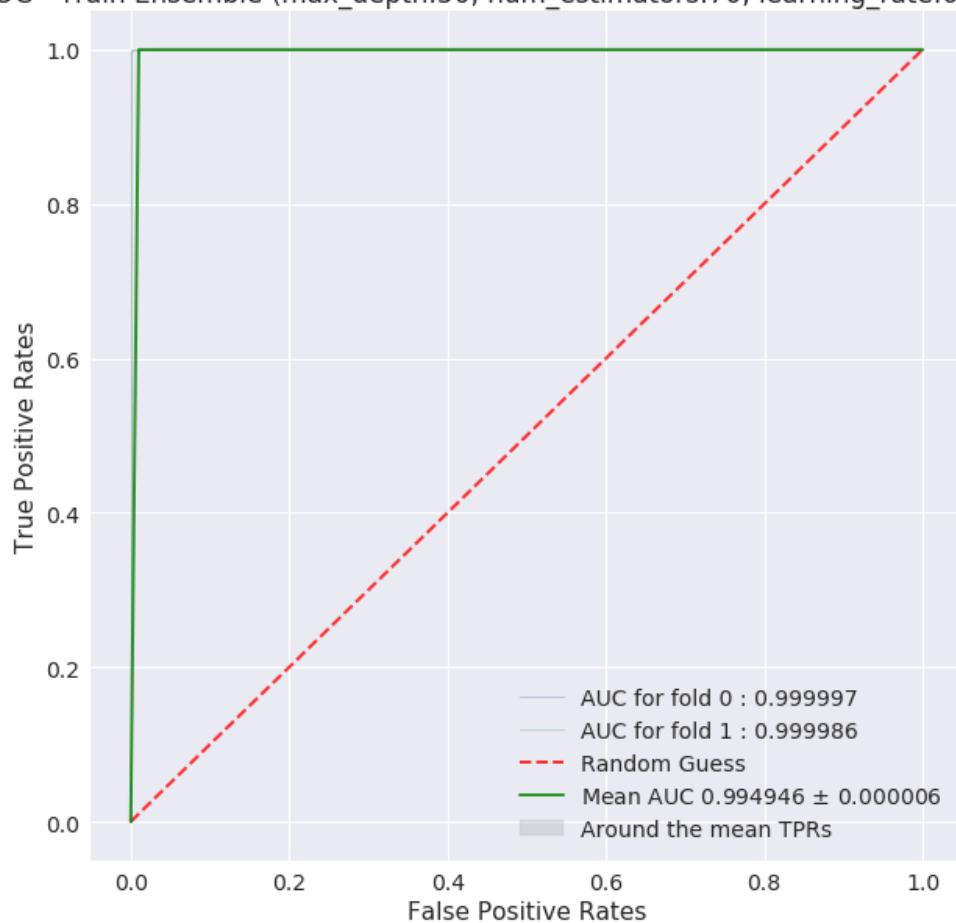


ROC - Validation Ensemble (max\_depth:50, num\_estimators:40, learning\_rate:0.100000)

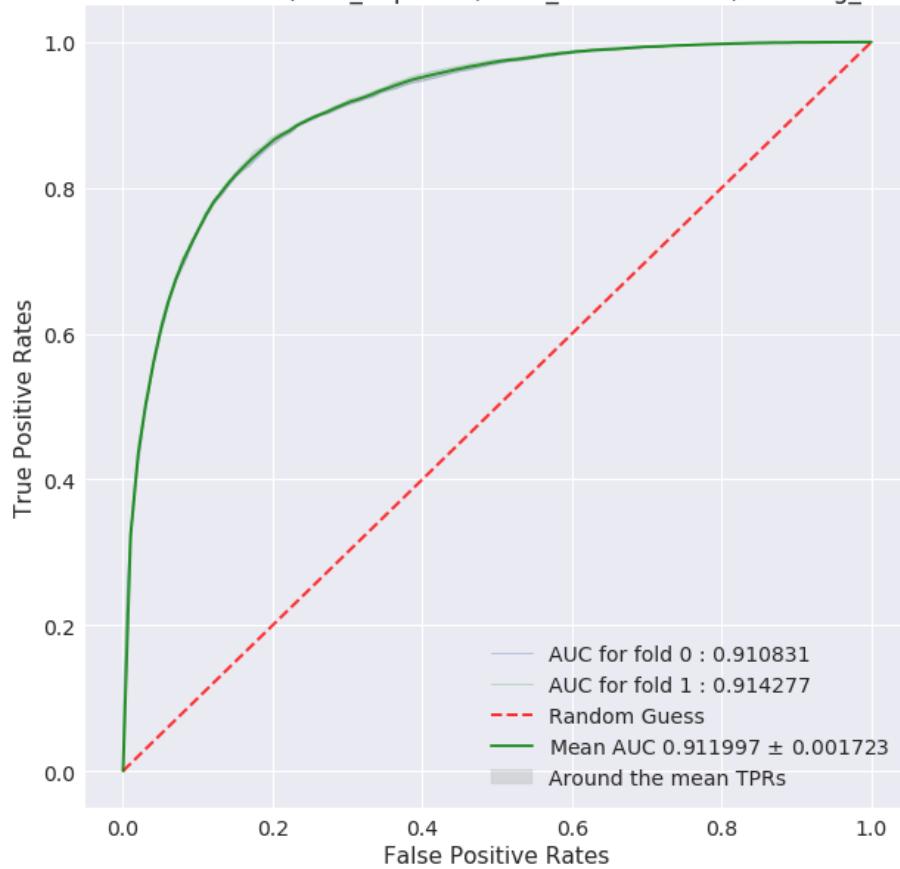


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:70, learning\_rate:0.100000)

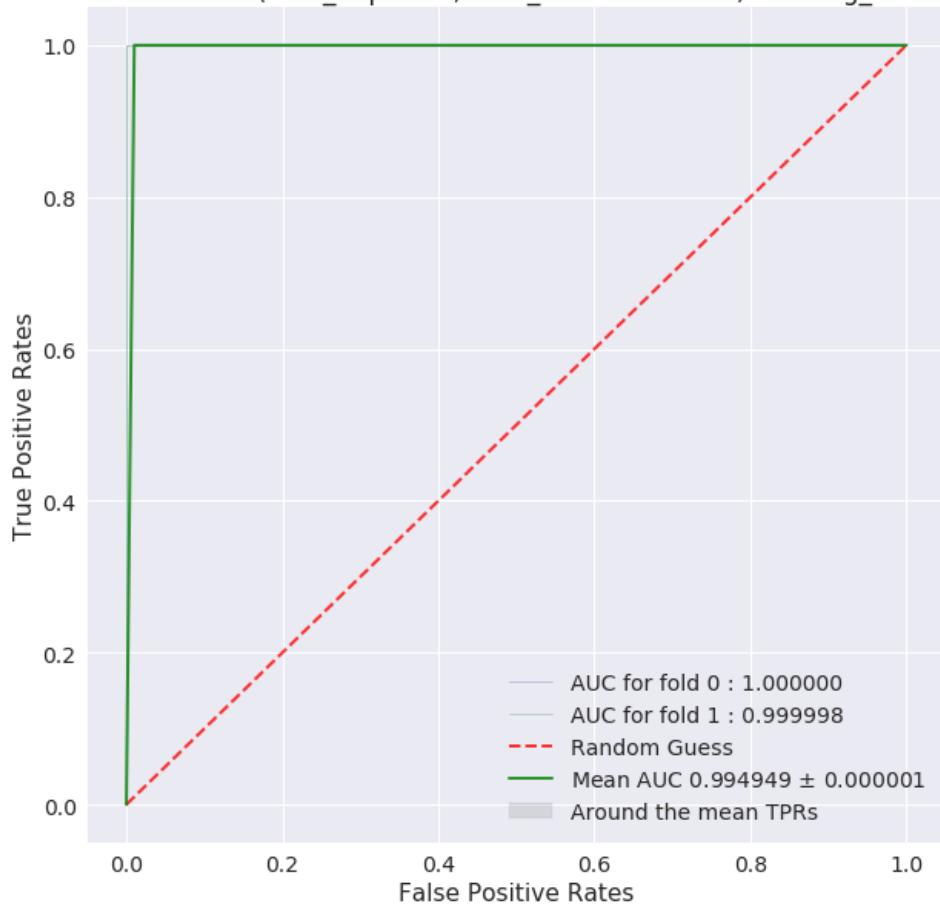


ROC - Validation Ensemble (max\_depth:50, num\_estimators:70, learning\_rate:0.100000)

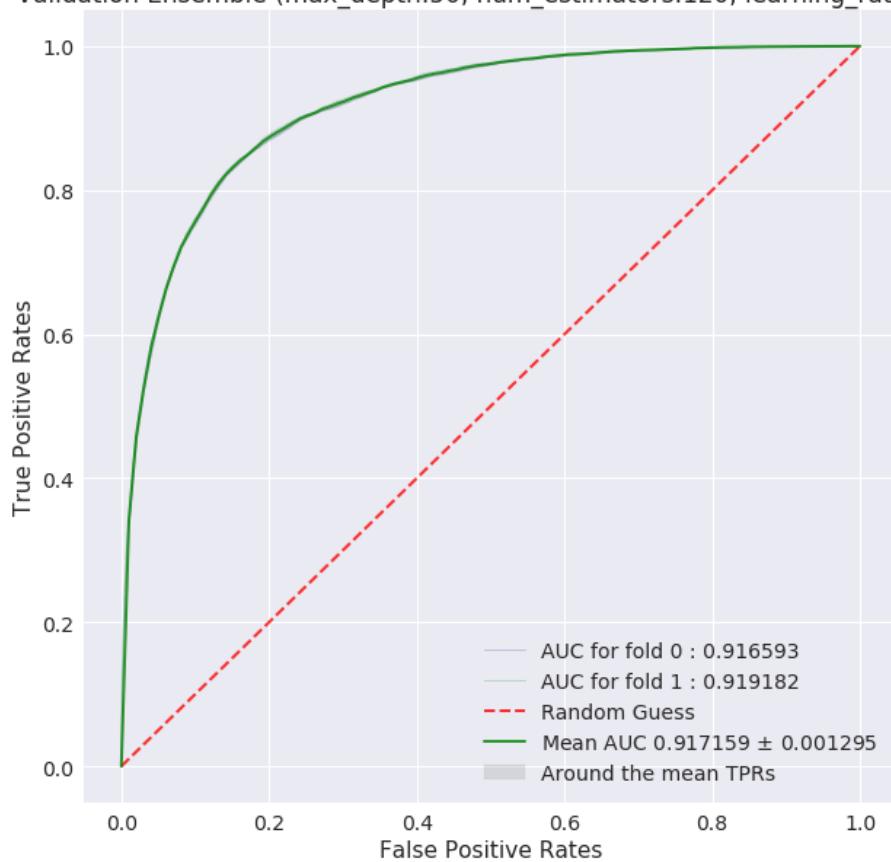


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:120, learning\_rate:0.100000)



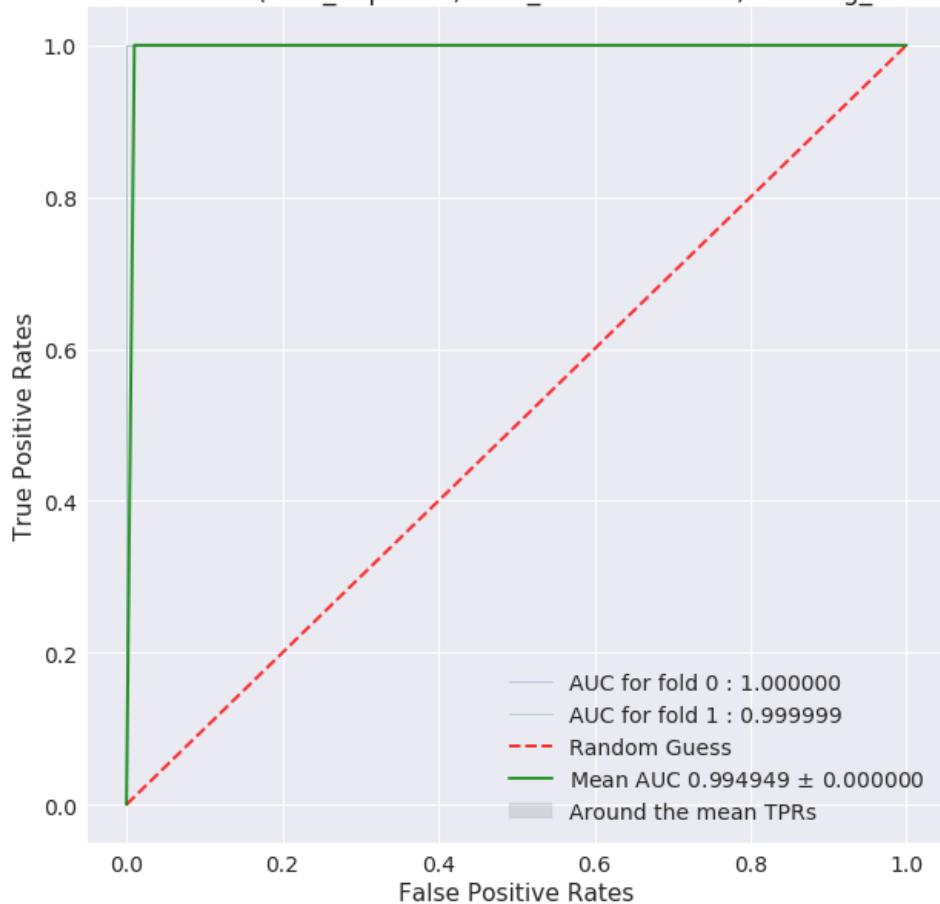
ROC - Validation Ensemble (max\_depth:50, num\_estimators:120, learning\_rate:0.100000)



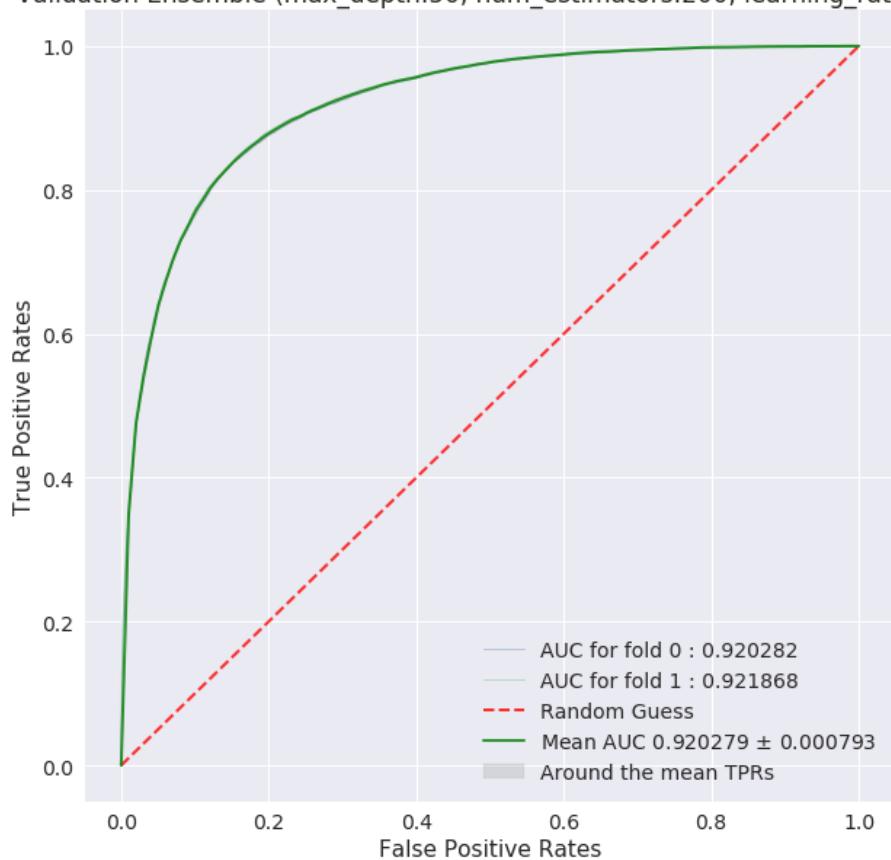
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:200, learning\_rate:0.100000)



ROC - Validation Ensemble (max\_depth:50, num\_estimators:200, learning\_rate:0.100000)



---

=====  
Train hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.837863
1	(2, 70, 0.1)	0.864786
2	(2, 120, 0.1)	0.890807
3	(2, 200, 0.1)	0.911470
4	(5, 40, 0.1)	0.899817
5	(5, 70, 0.1)	0.925526
6	(5, 120, 0.1)	0.946081
7	(5, 200, 0.1)	0.962507
8	(20, 40, 0.1)	0.987970
9	(20, 70, 0.1)	0.992977
10	(20, 120, 0.1)	0.994603
11	(20, 200, 0.1)	0.994906
12	(50, 40, 0.1)	0.994891
13	(50, 70, 0.1)	0.994946

```
14 (50, 120, 0.1) 0.994949
15 (50, 200, 0.1) 0.994949
```

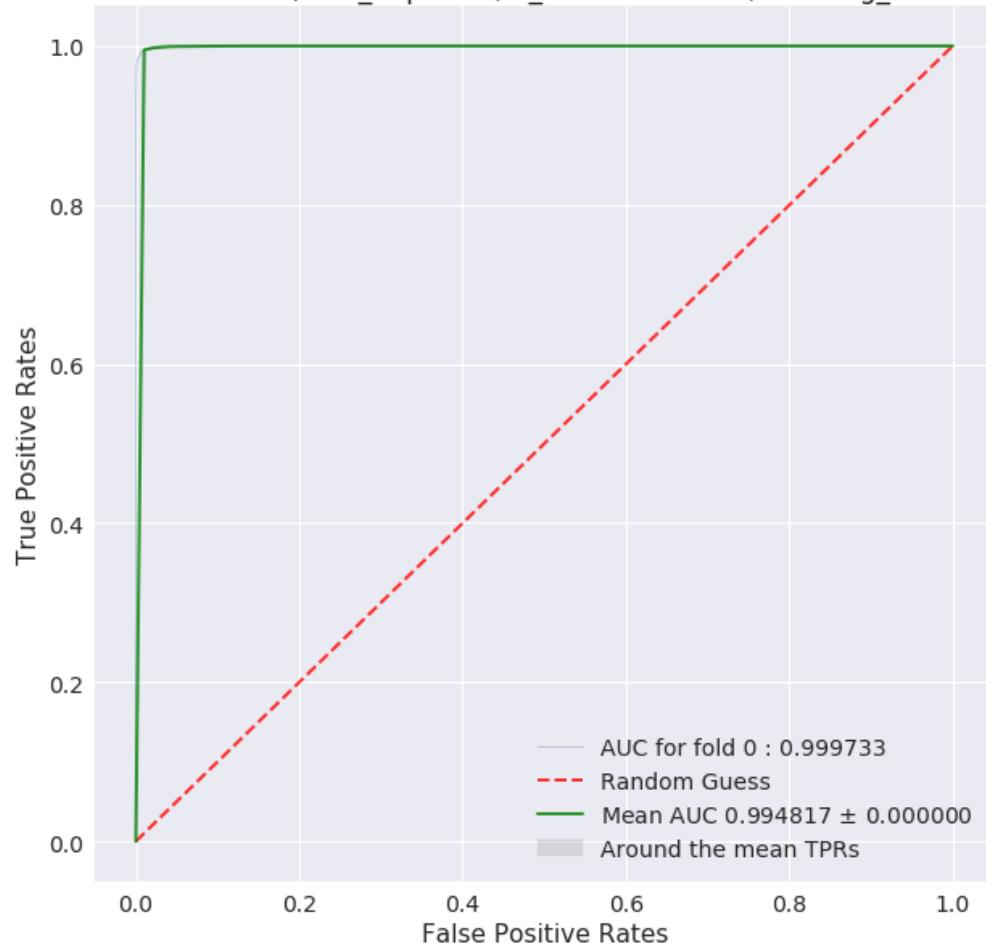
Validation hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.831138
1	(2, 70, 0.1)	0.855979
2	(2, 120, 0.1)	0.878624
3	(2, 200, 0.1)	0.896022
4	(5, 40, 0.1)	0.874684
5	(5, 70, 0.1)	0.894412
6	(5, 120, 0.1)	0.908258
7	(5, 200, 0.1)	0.917471
8	(20, 40, 0.1)	0.902368
9	(20, 70, 0.1)	0.912533
10	(20, 120, 0.1)	0.918318
11	(20, 200, 0.1)	0.921193
12	(50, 40, 0.1)	0.901911
13	(50, 70, 0.1)	0.911997
14	(50, 120, 0.1)	0.917159
15	(50, 200, 0.1)	0.920279

Best hyperparam value: (20, 200, 0.1)

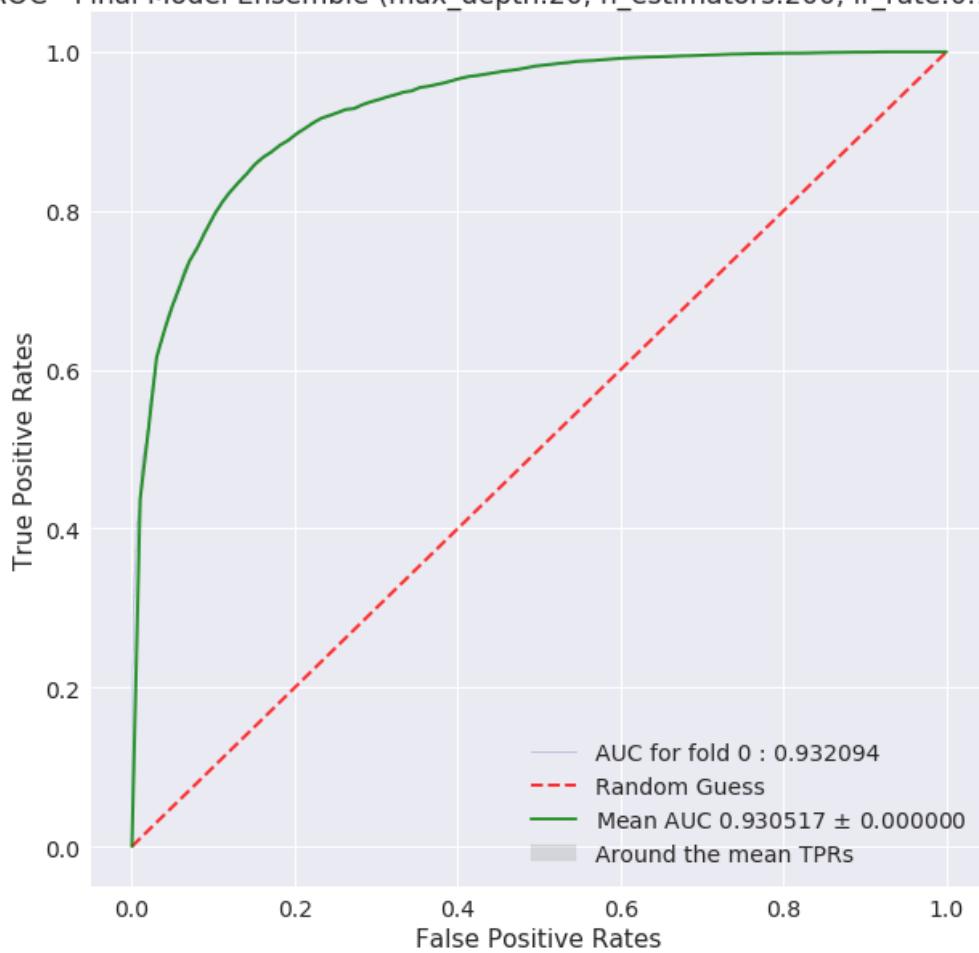
```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
```

ROC - Final Model DT (max\_depth:20, n\_estimators:200, learning\_rate:0.100000)

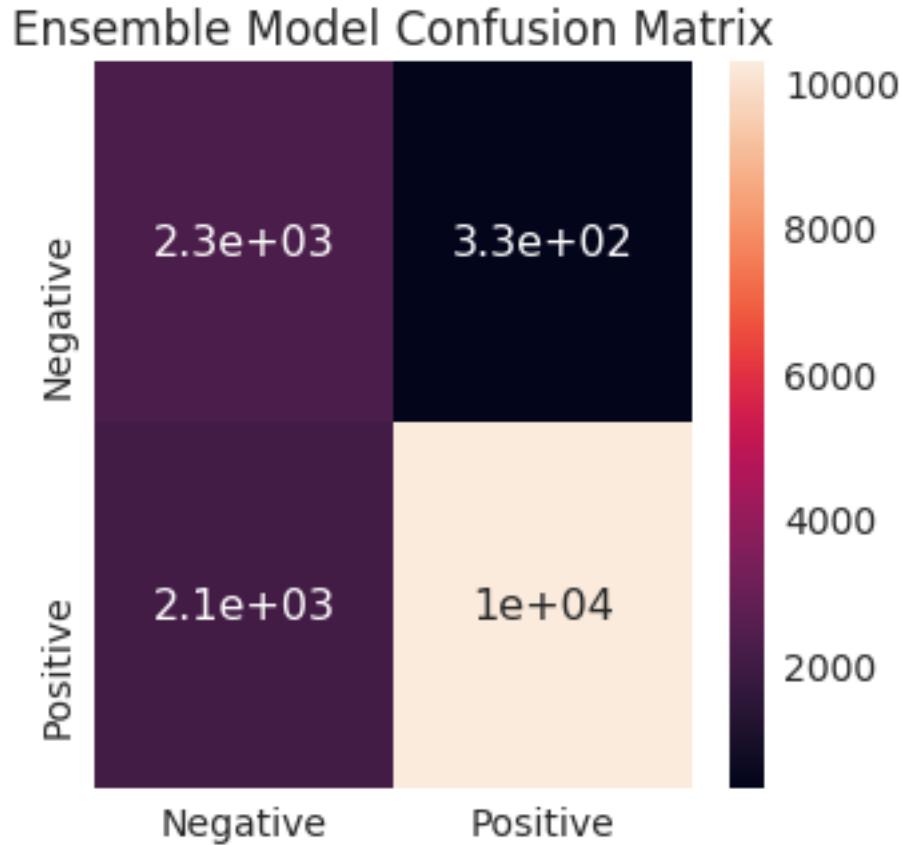


```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Final Model Ensemble (max\_depth:20, n\_estimators:200, lr\_rate:0.100000)



Test auc score 0.9305170223142127



	Negative	Positive
Precision	0.521719	0.968756
Recall	0.872992	0.831100
Fscore	0.653120	0.894664
Support	2614.000000	12386.000000

#### 4.5.3 [B.3] Applying XGBOOST on AVG W2V, SET 3

```
In [21]: # form two lists
depth_list = [2, 5, 20, 50] # depends on size of dataset
n_estimators_list = [40, 70, 120, 200] # depends on size of dataset
learning_rate_list = [0.1] # learning rate for XGB training

# create a configuartion dictionary
config_dict = {
    'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/AVG_W2V',
    'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/AVG_W2V',
    'train_size' : 40000,
    'test_size' : 15000,
```

```

'hyperparam_list' : list(product(depth_list, n_estimators_list, learning_rate_list))
'implementation': 'xgb' # 'xgb' or 'rf'
}

In [22]: # read the train, test data and preprocess it
train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                                               scaling=True
                                                               dim_reductio

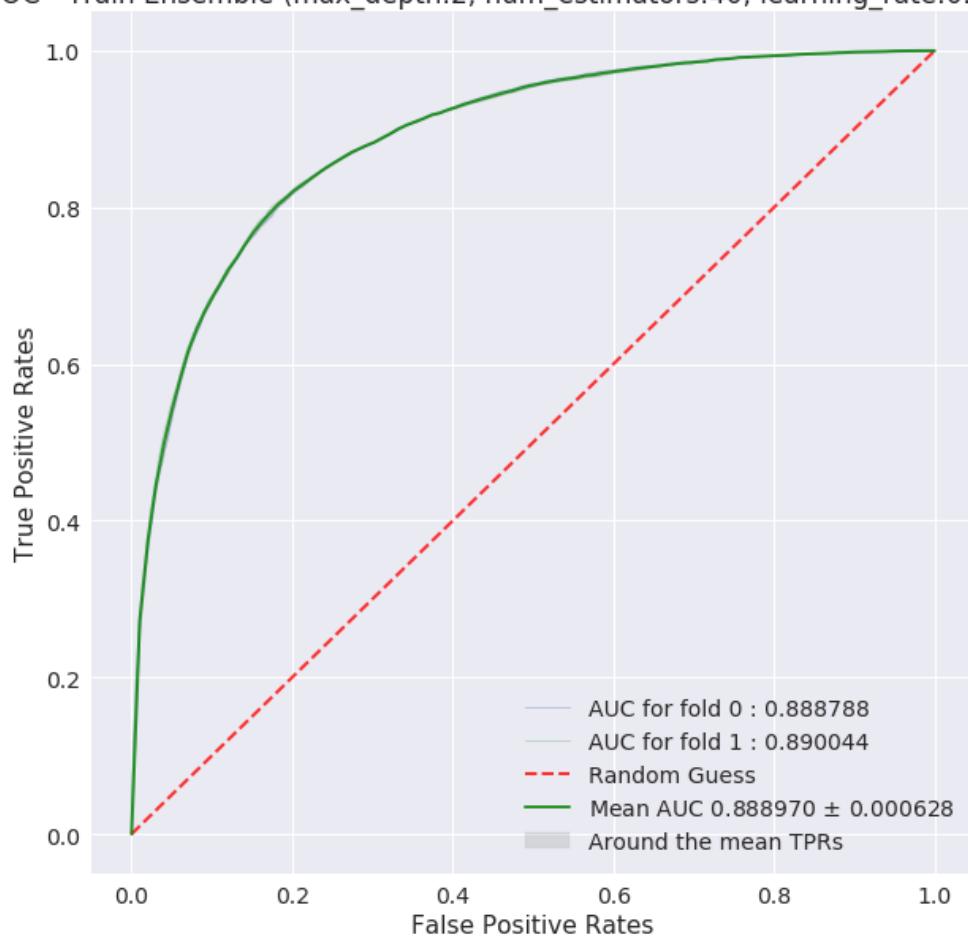
# train and validate the model
model = train_and_validate_model(config_dict, train_features, train_labels)

# test and evaluate the model
ptabe_entry_b3 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

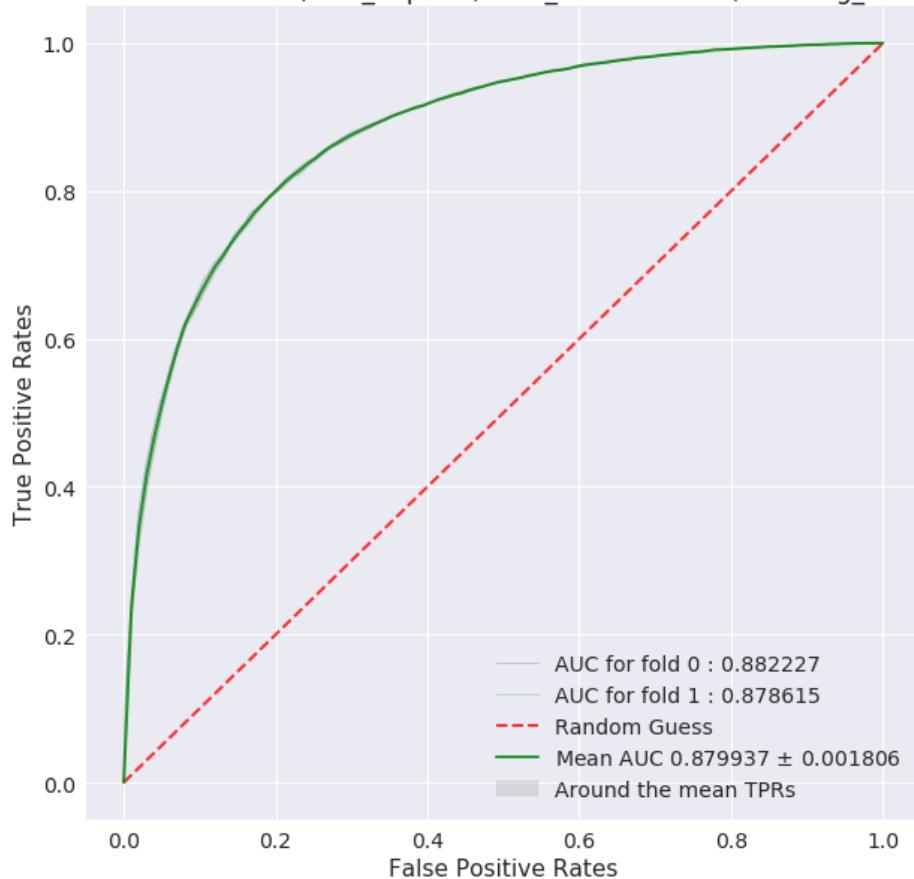
Train df shape (40000, 52)
Class label distribution in train df:
0    20024
1    19976
Name: Label, dtype: int64
Test df shape (15000, 52)
Class label distribution in test df:
1    12386
0    2614
Name: Label, dtype: int64
Shape of -> train features :40000,50, test features: 15000,50
Shape of -> train labels :40000, test labels: 15000
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:

```

ROC - Train Ensemble (max\_depth:2, num\_estimators:40, learning\_rate:0.100000)

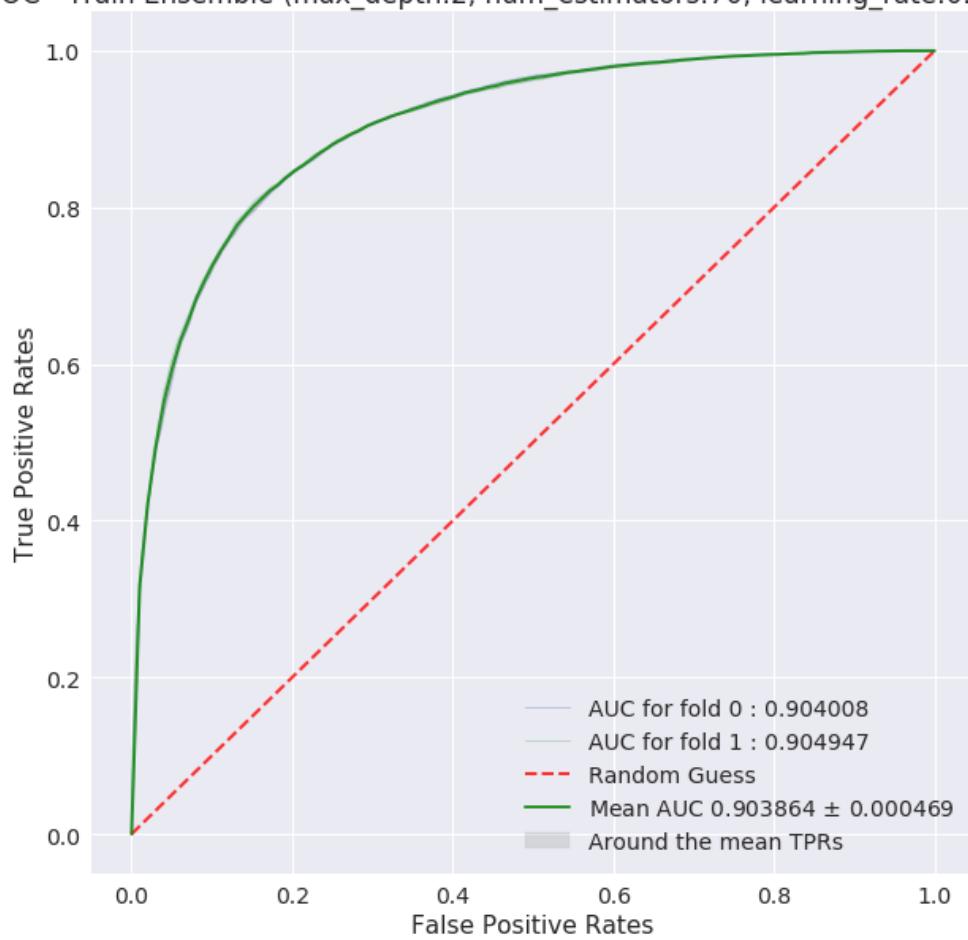


ROC - Validation Ensemble (max\_depth:2, num\_estimators:40, learning\_rate:0.100000)

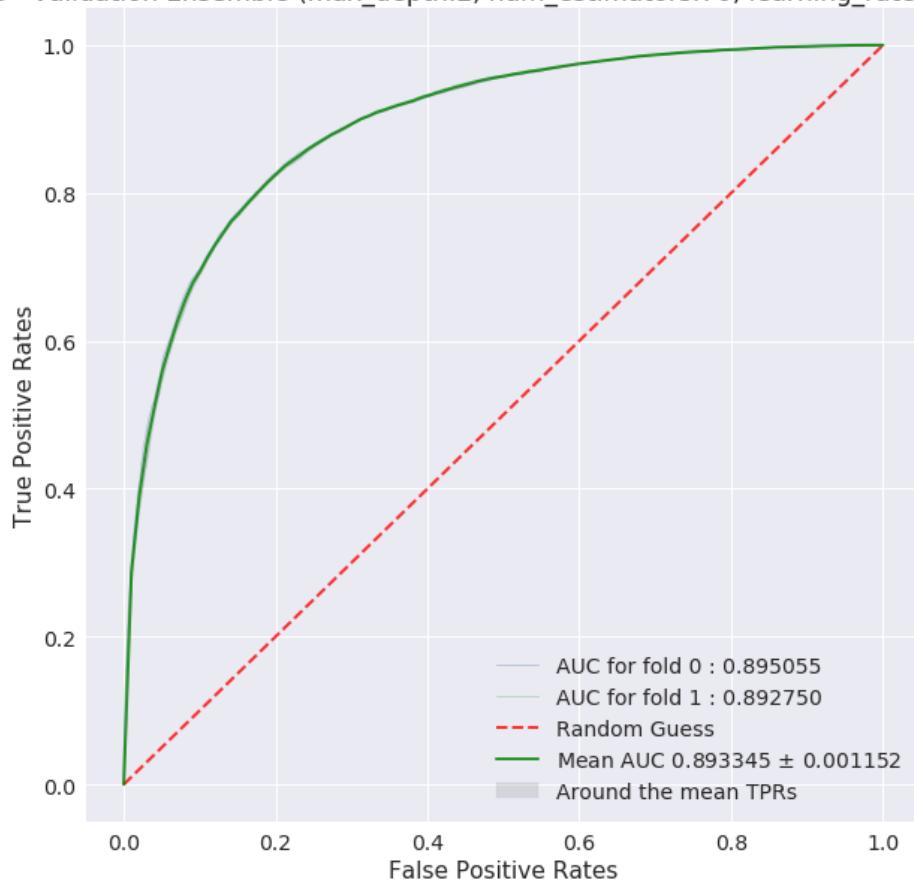


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:70, learning\_rate:0.100000)

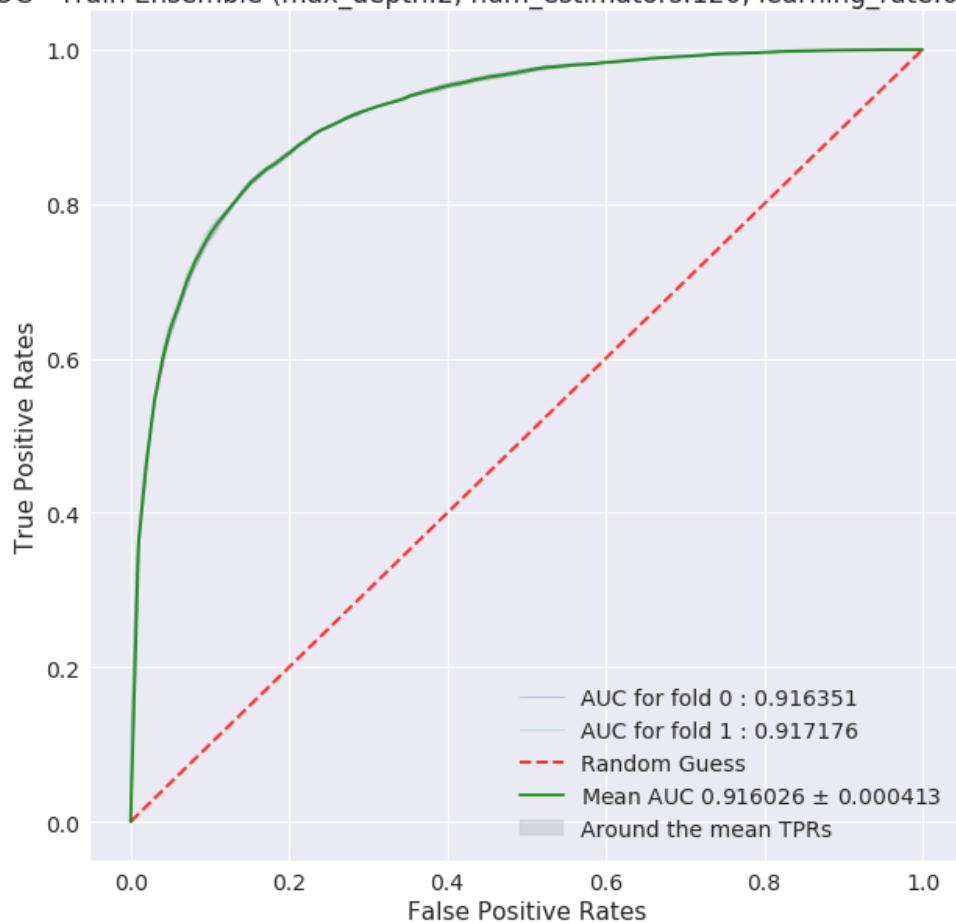


ROC - Validation Ensemble (max\_depth:2, num\_estimators:70, learning\_rate:0.100000)

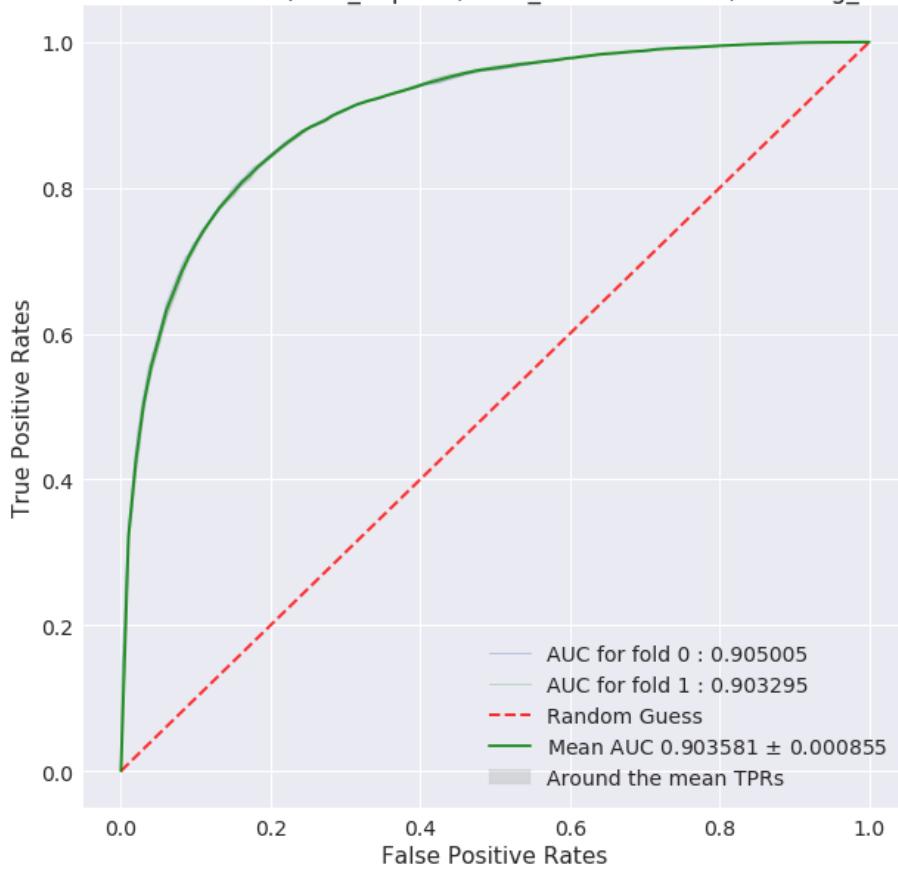


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:120, learning\_rate:0.100000)

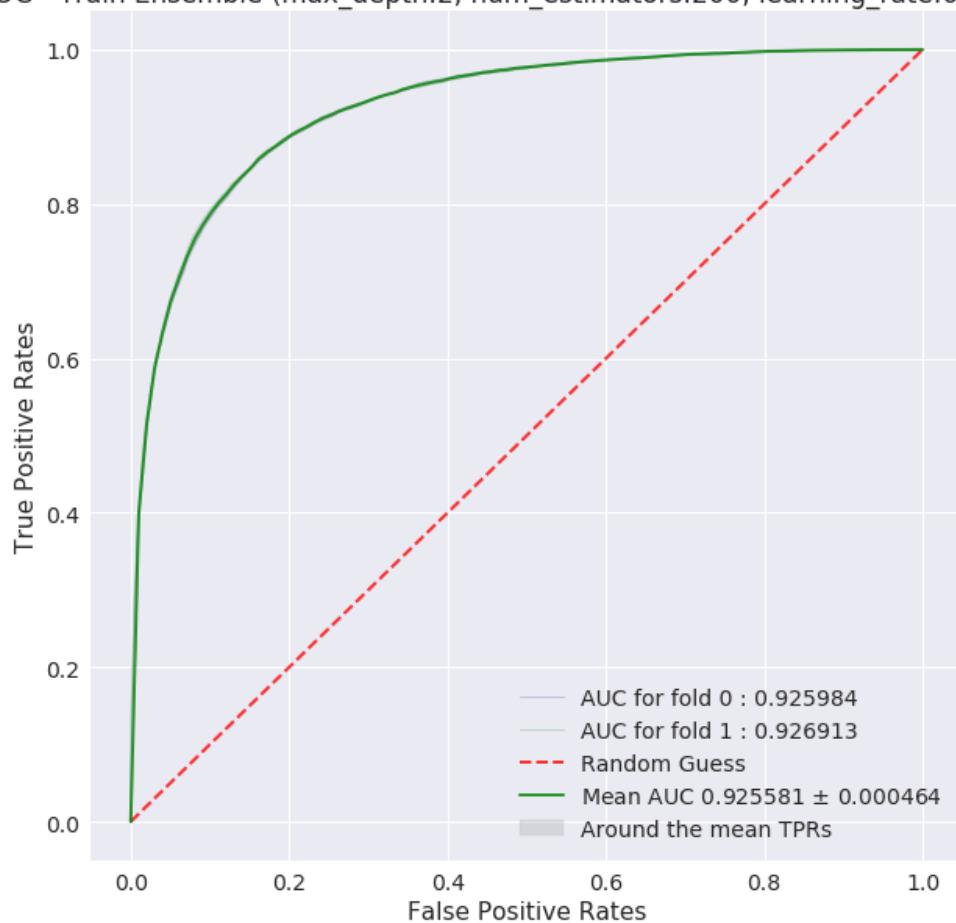


ROC - Validation Ensemble (max\_depth:2, num\_estimators:120, learning\_rate:0.100000)

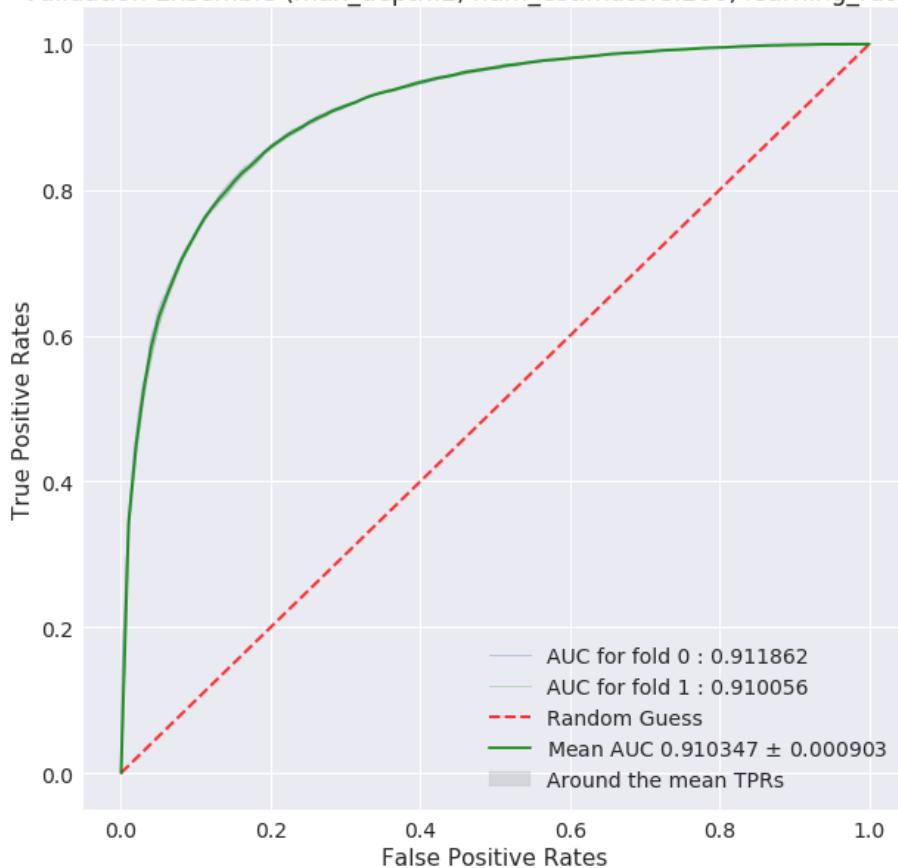


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:200, learning\_rate:0.100000)

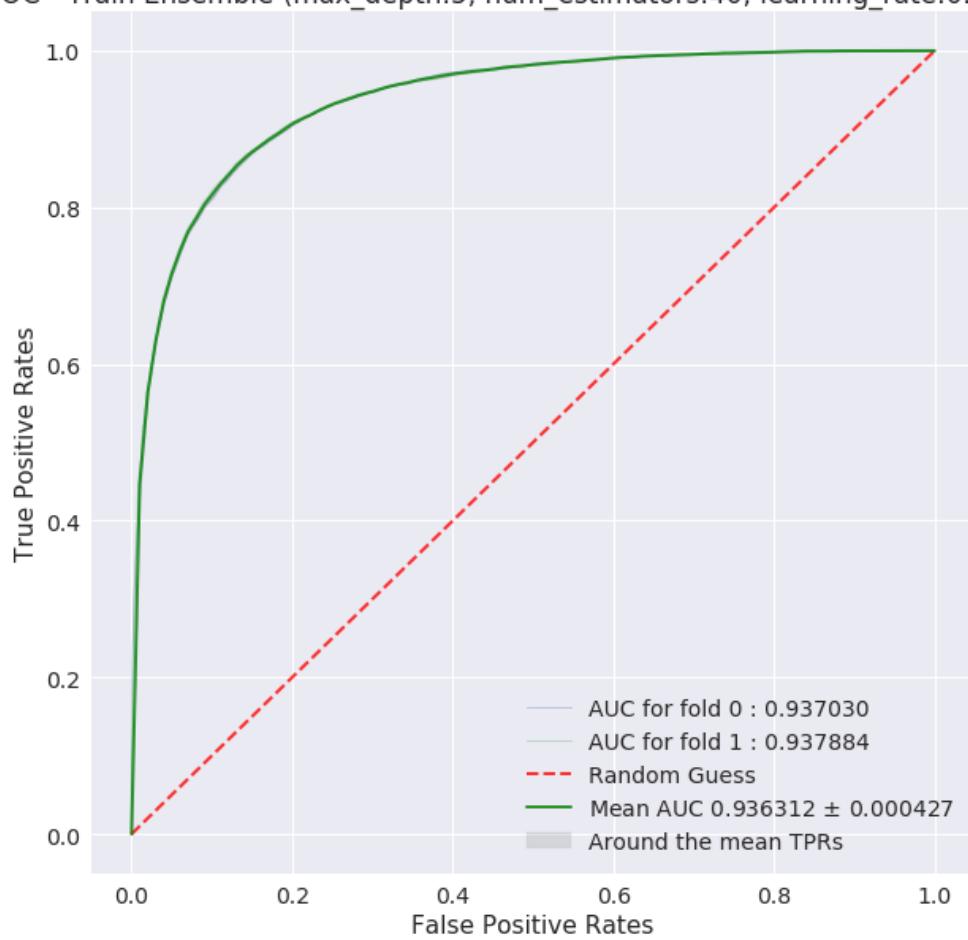


ROC - Validation Ensemble (max\_depth:2, num\_estimators:200, learning\_rate:0.100000)

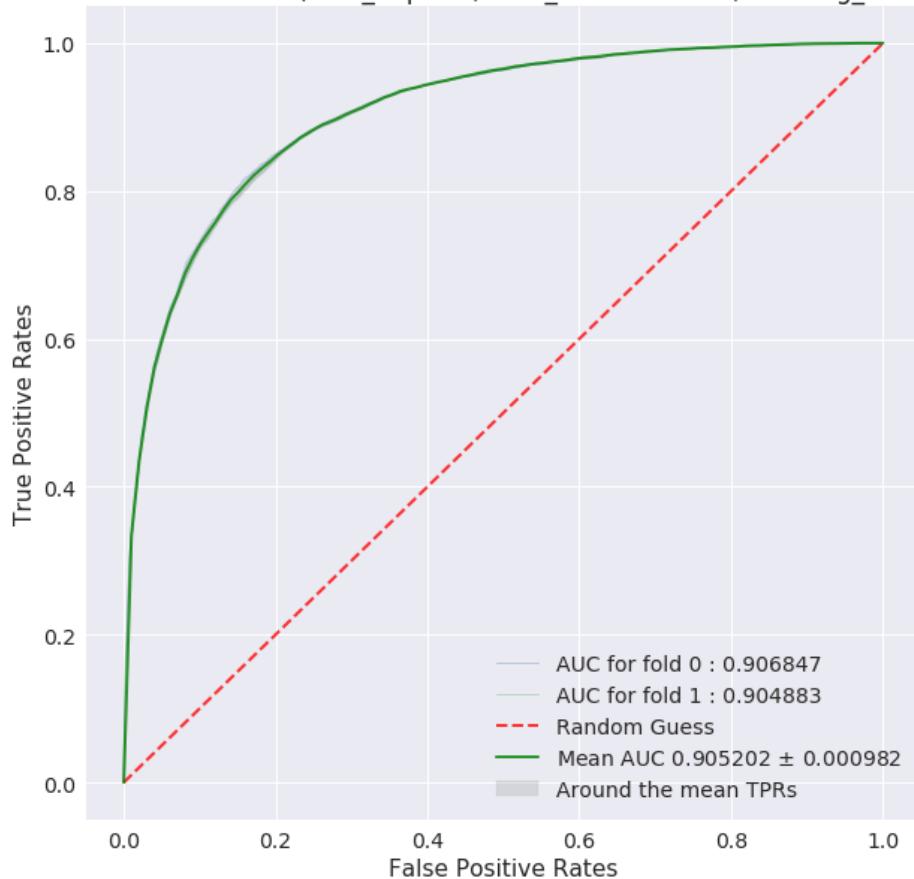


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:40, learning\_rate:0.100000)

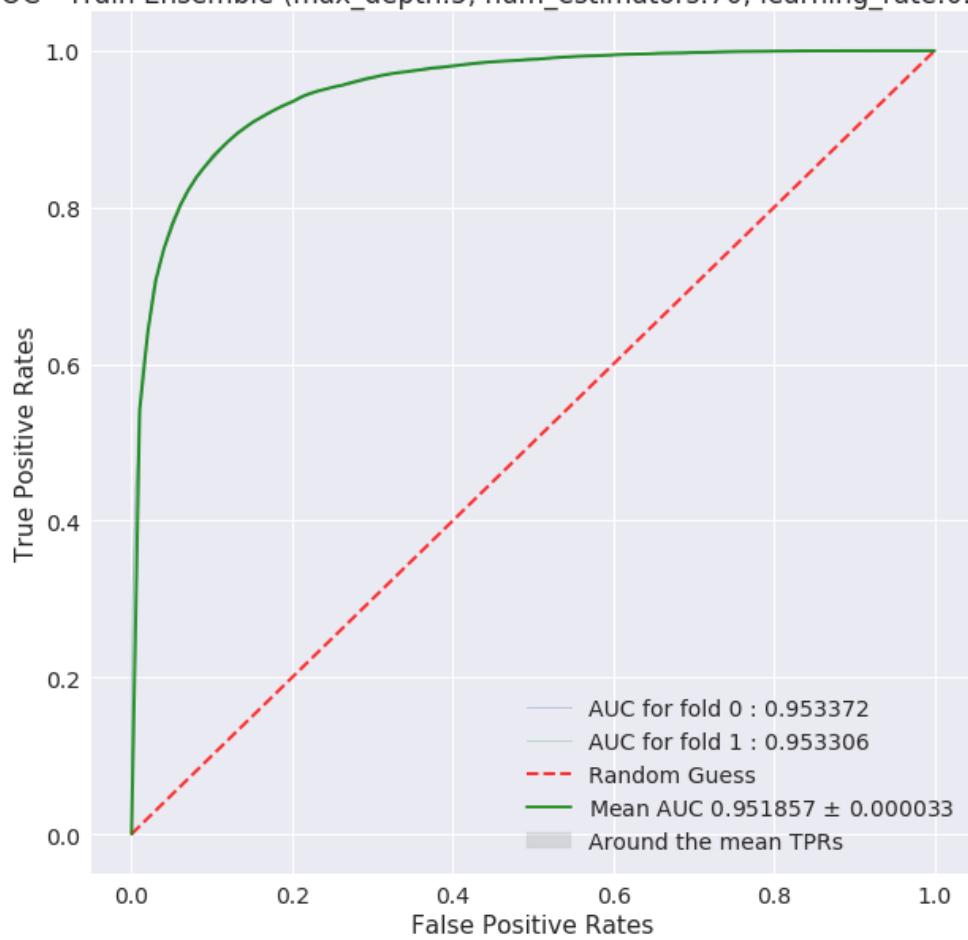


ROC - Validation Ensemble (max\_depth:5, num\_estimators:40, learning\_rate:0.100000)

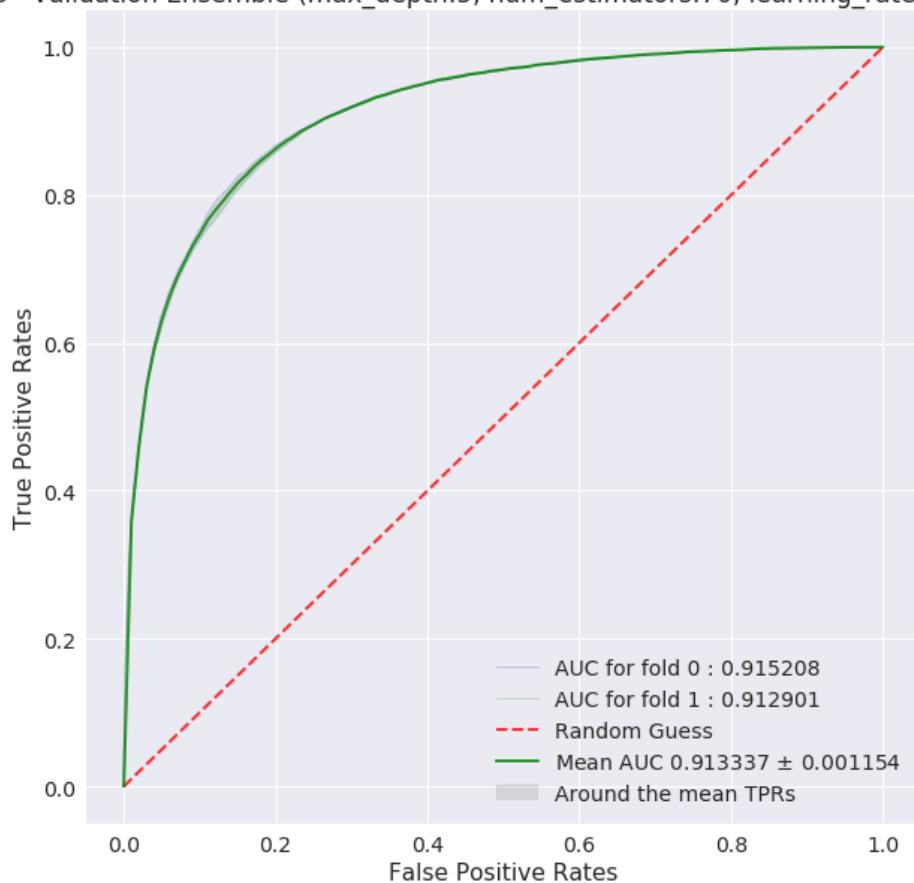


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:70, learning\_rate:0.100000)

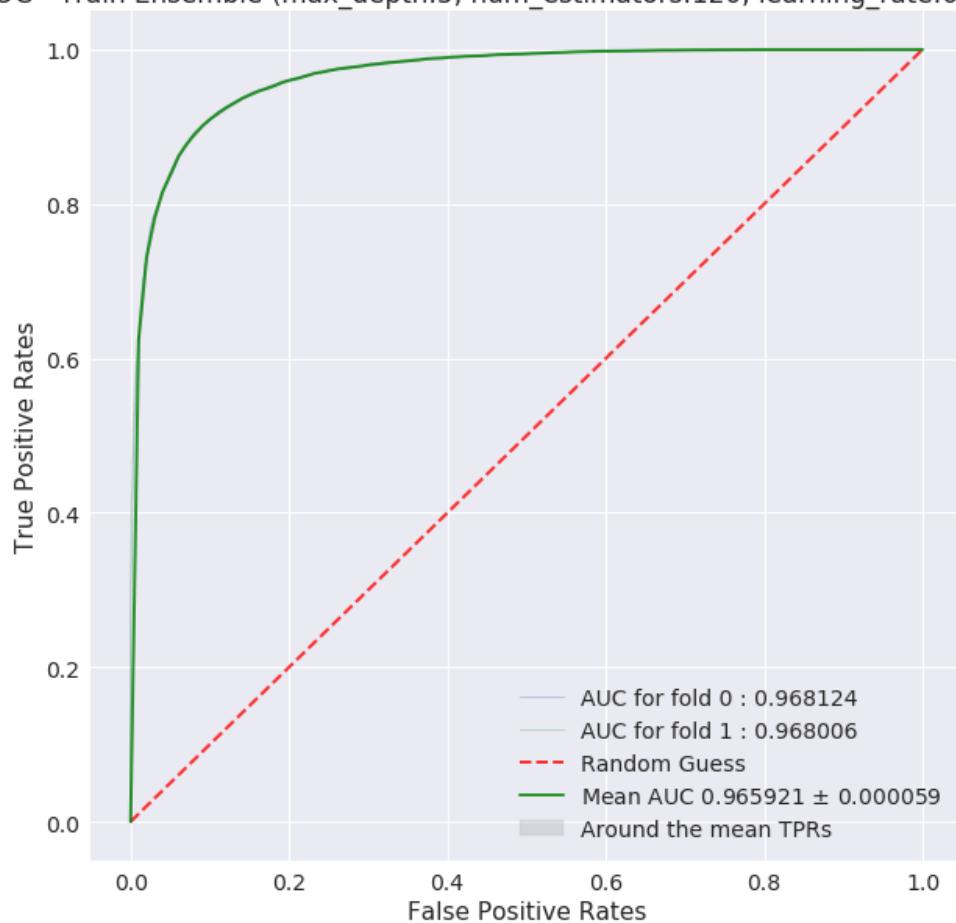


ROC - Validation Ensemble (max\_depth:5, num\_estimators:70, learning\_rate:0.100000)

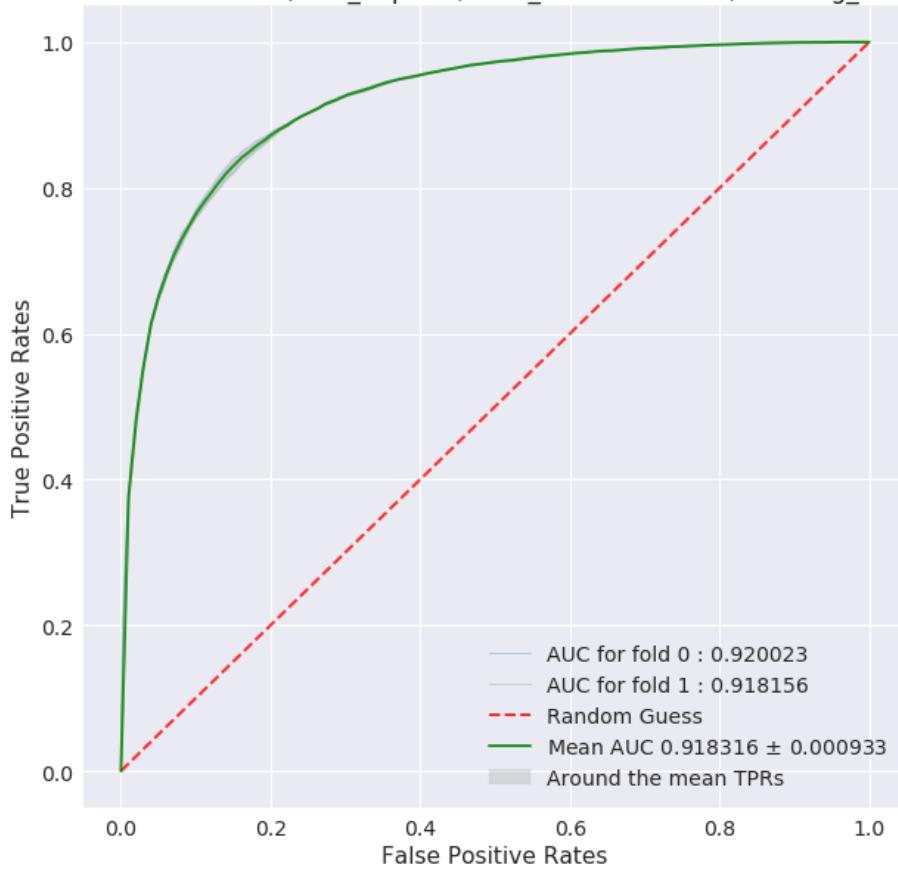


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:120, learning\_rate:0.100000)

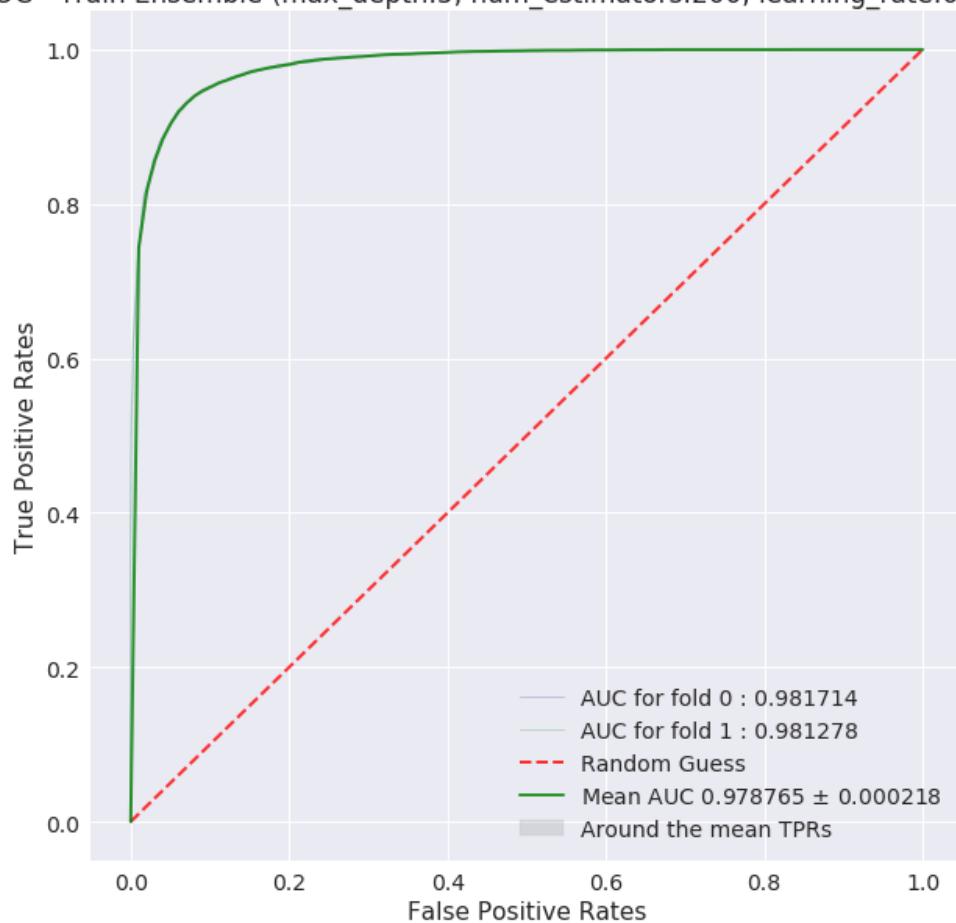


ROC - Validation Ensemble (max\_depth:5, num\_estimators:120, learning\_rate:0.100000)

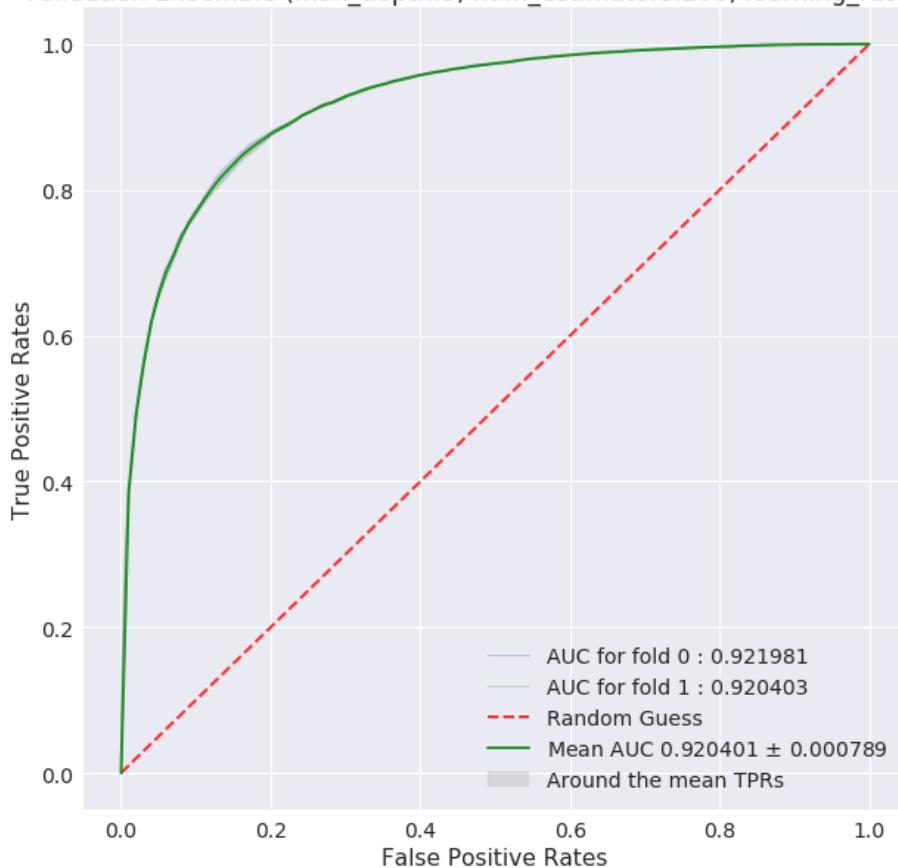


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:200, learning\_rate:0.100000)

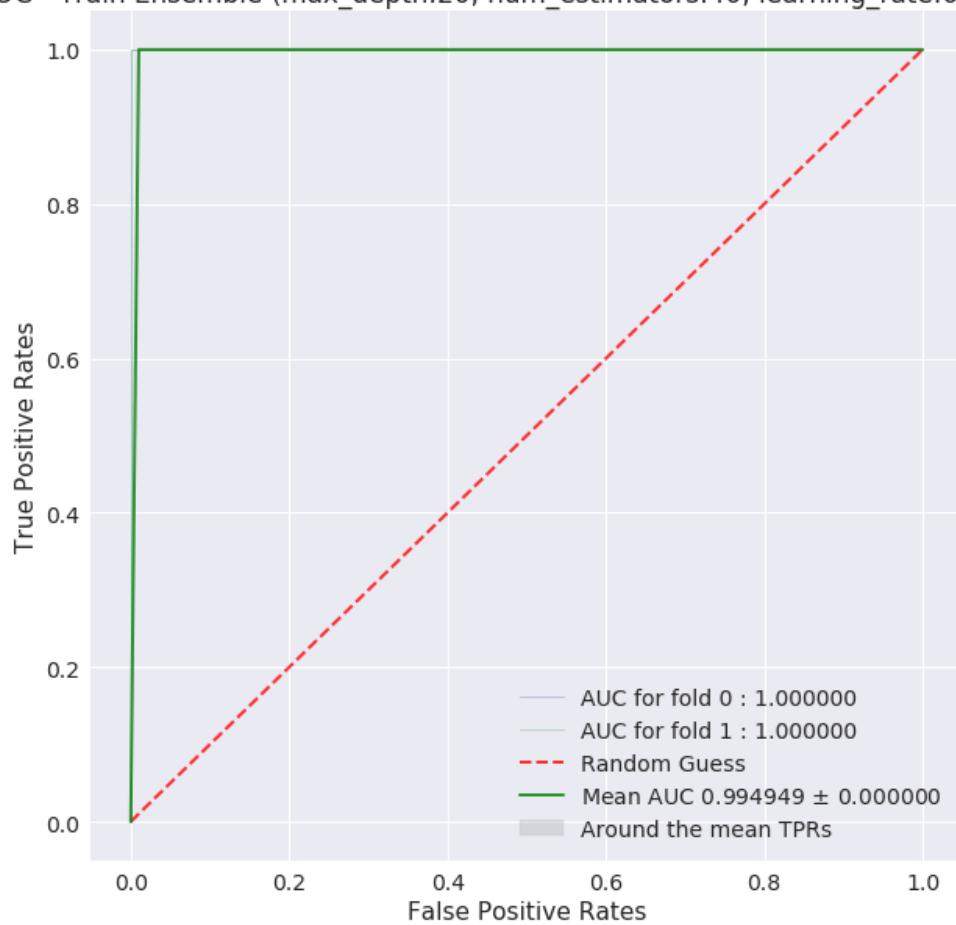


ROC - Validation Ensemble (max\_depth:5, num\_estimators:200, learning\_rate:0.100000)

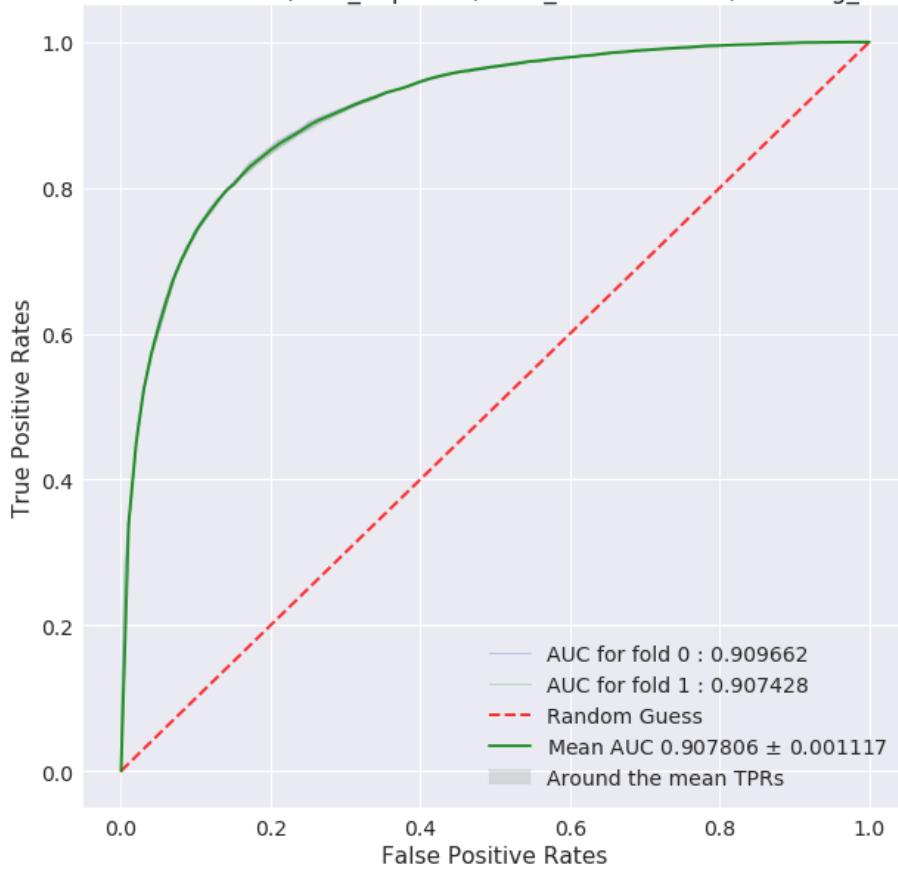


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:40, learning\_rate:0.100000)

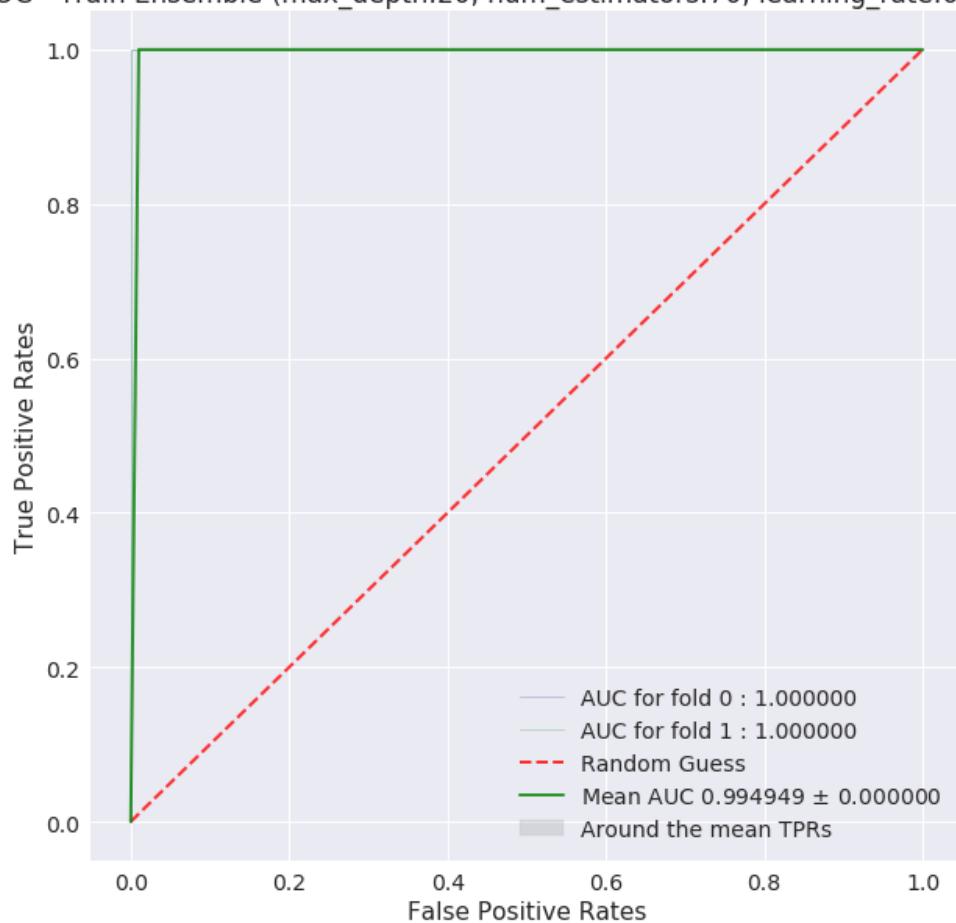


ROC - Validation Ensemble (max\_depth:20, num\_estimators:40, learning\_rate:0.100000)

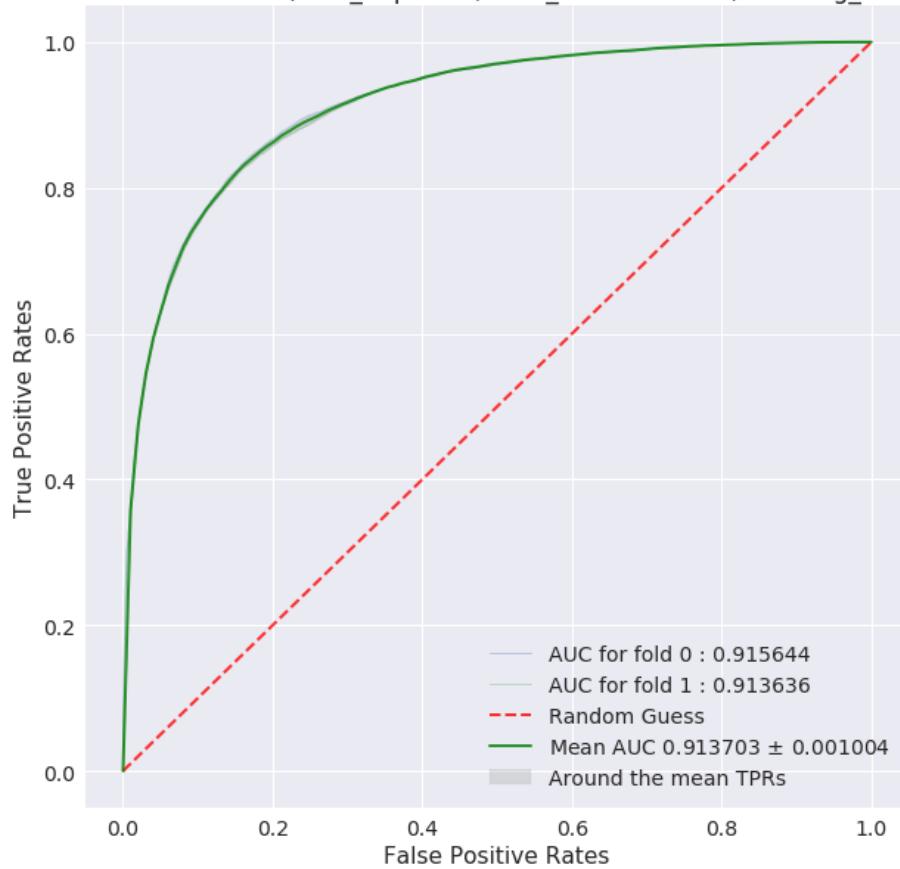


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:70, learning\_rate:0.100000)

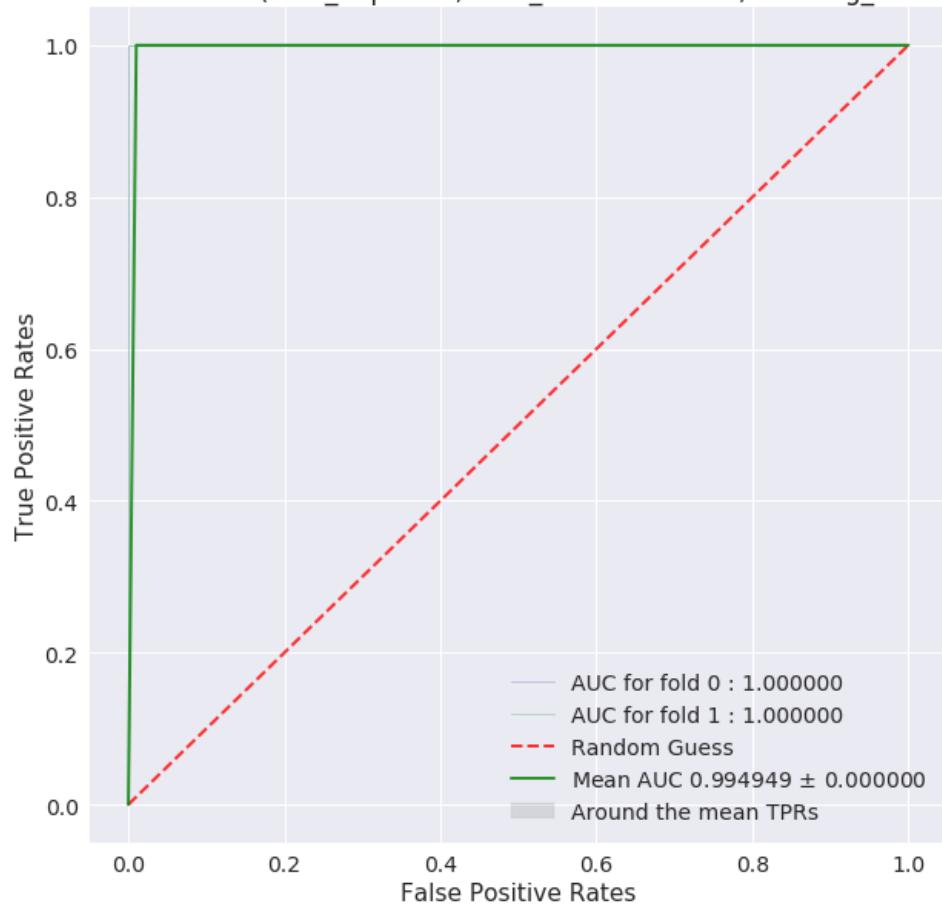


ROC - Validation Ensemble (max\_depth:20, num\_estimators:70, learning\_rate:0.100000)

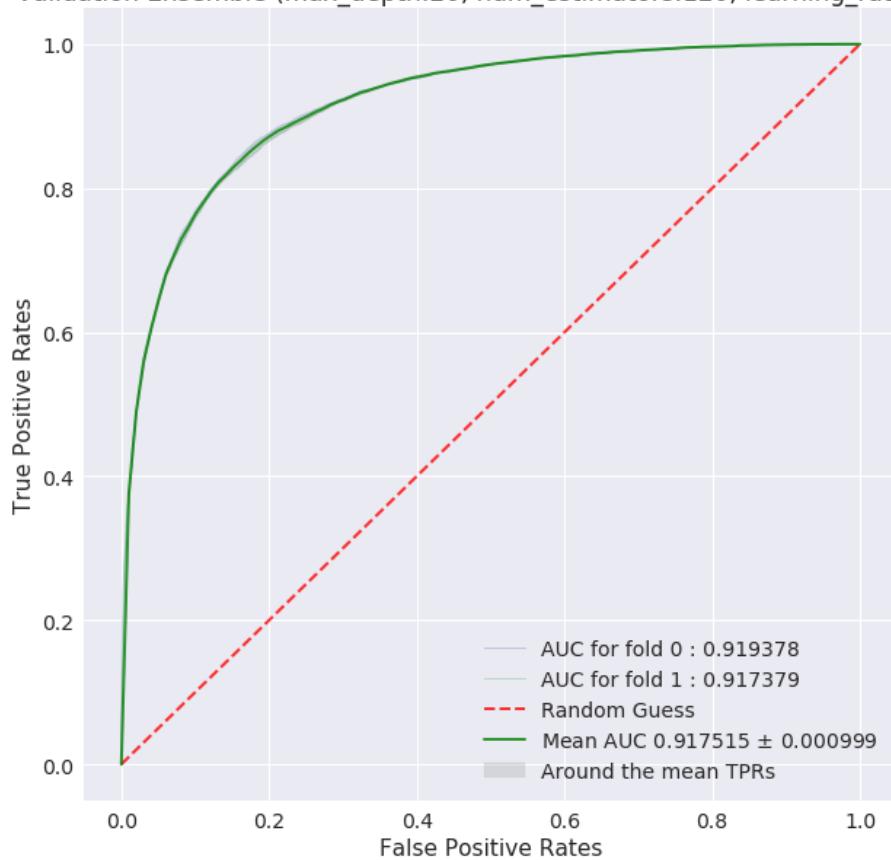


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:120, learning\_rate:0.100000)



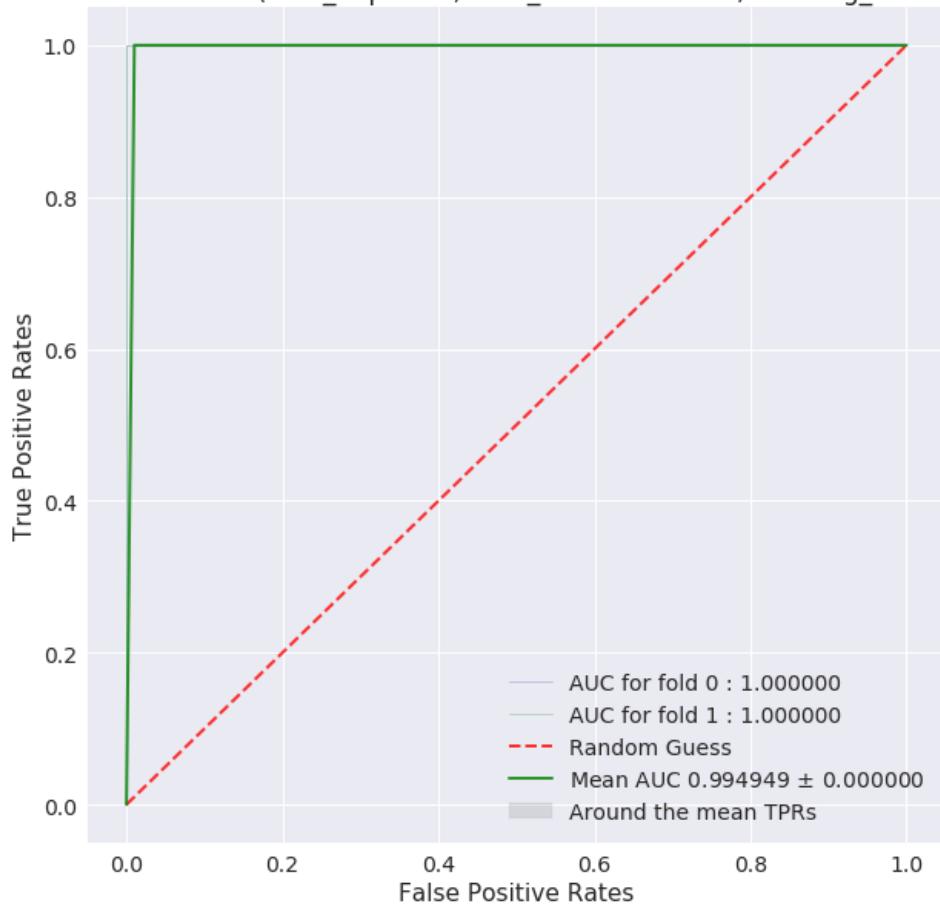
ROC - Validation Ensemble (max\_depth:20, num\_estimators:120, learning\_rate:0.100000)



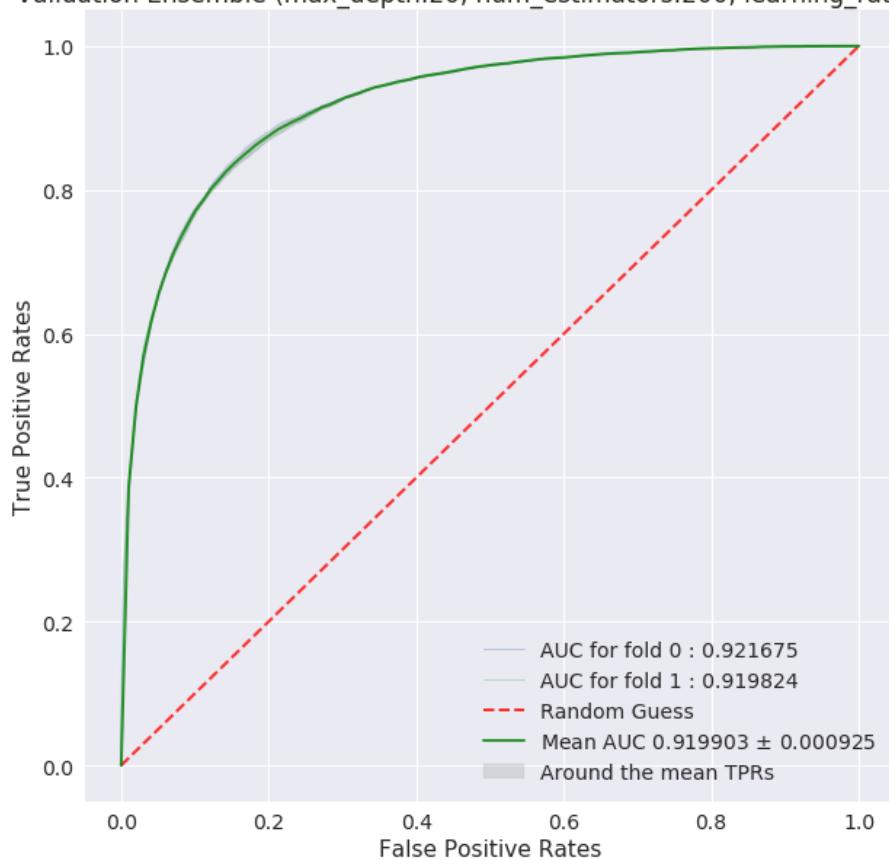
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:200, learning\_rate:0.100000)



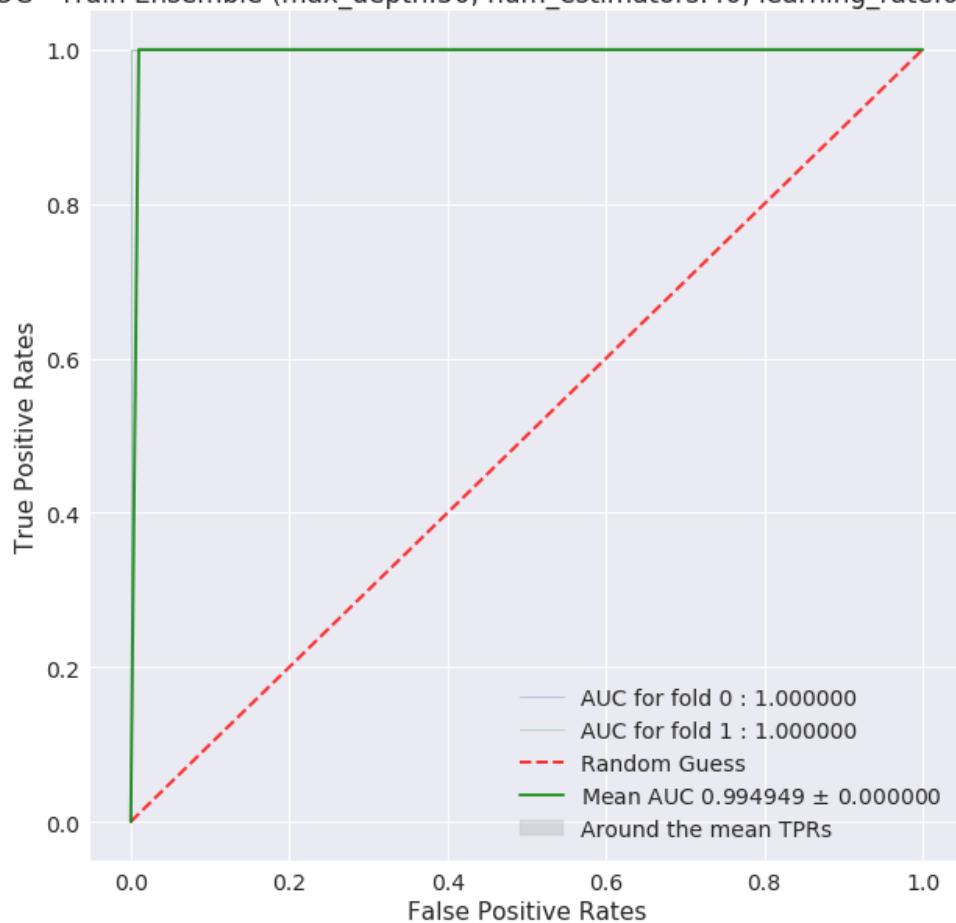
ROC - Validation Ensemble (max\_depth:20, num\_estimators:200, learning\_rate:0.100000)



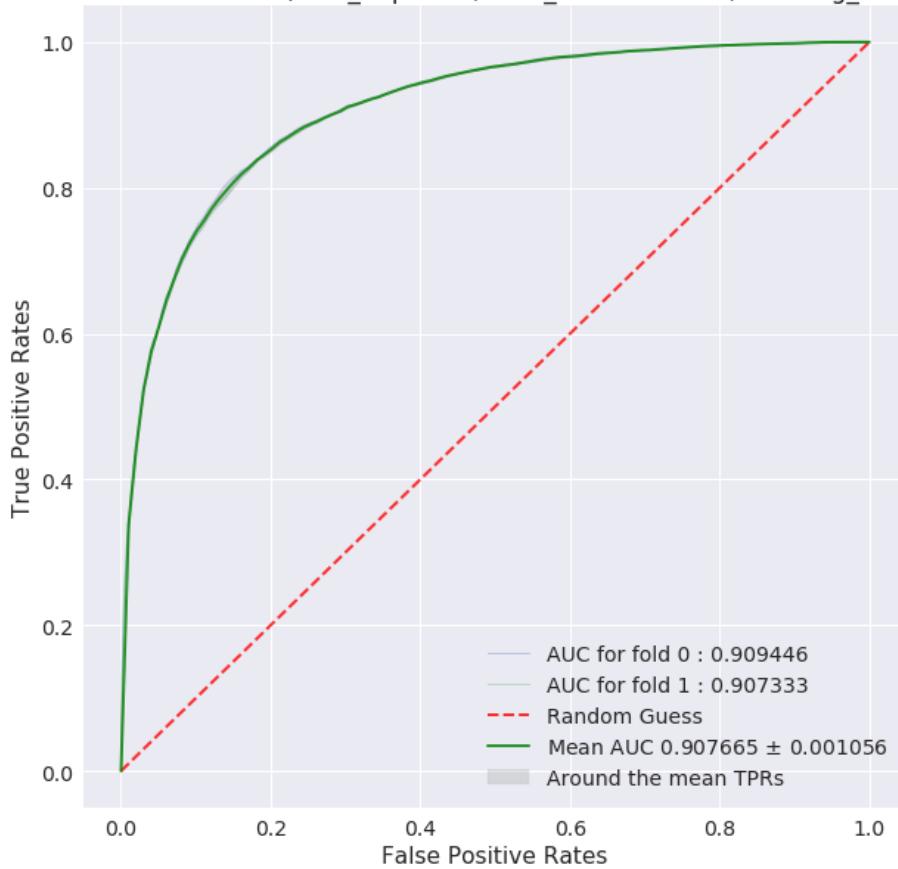
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:40, learning\_rate:0.100000)

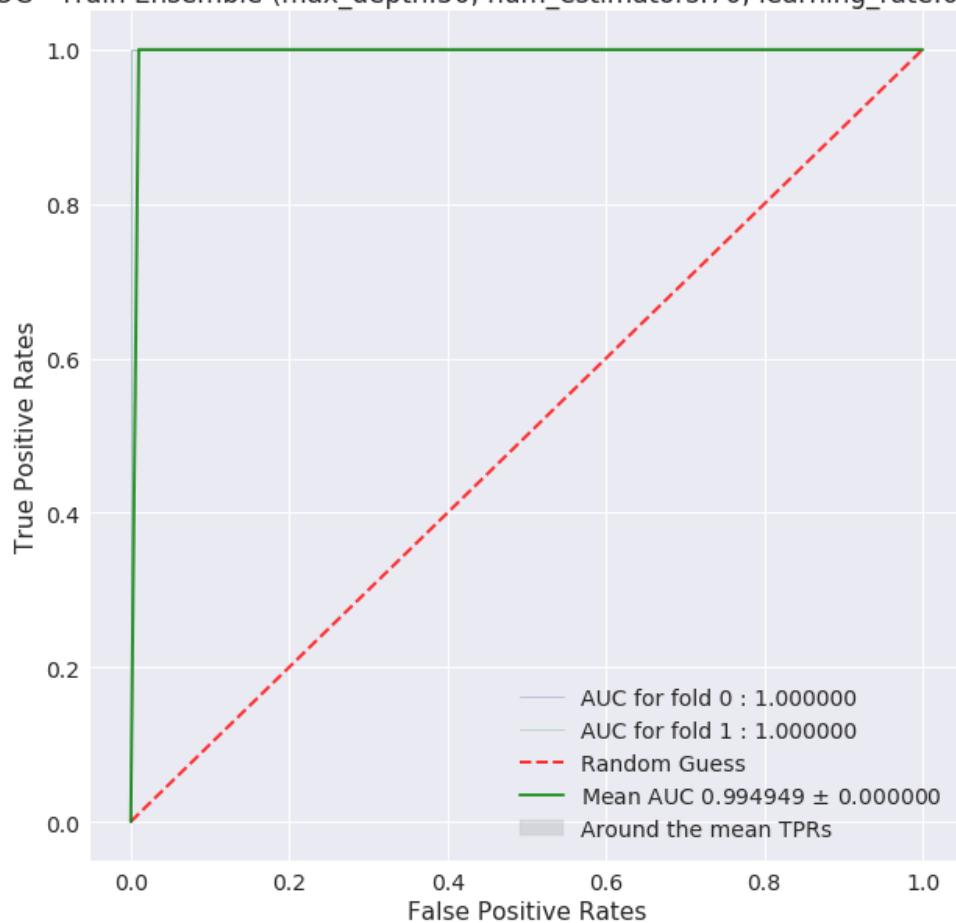


ROC - Validation Ensemble (max\_depth:50, num\_estimators:40, learning\_rate:0.100000)

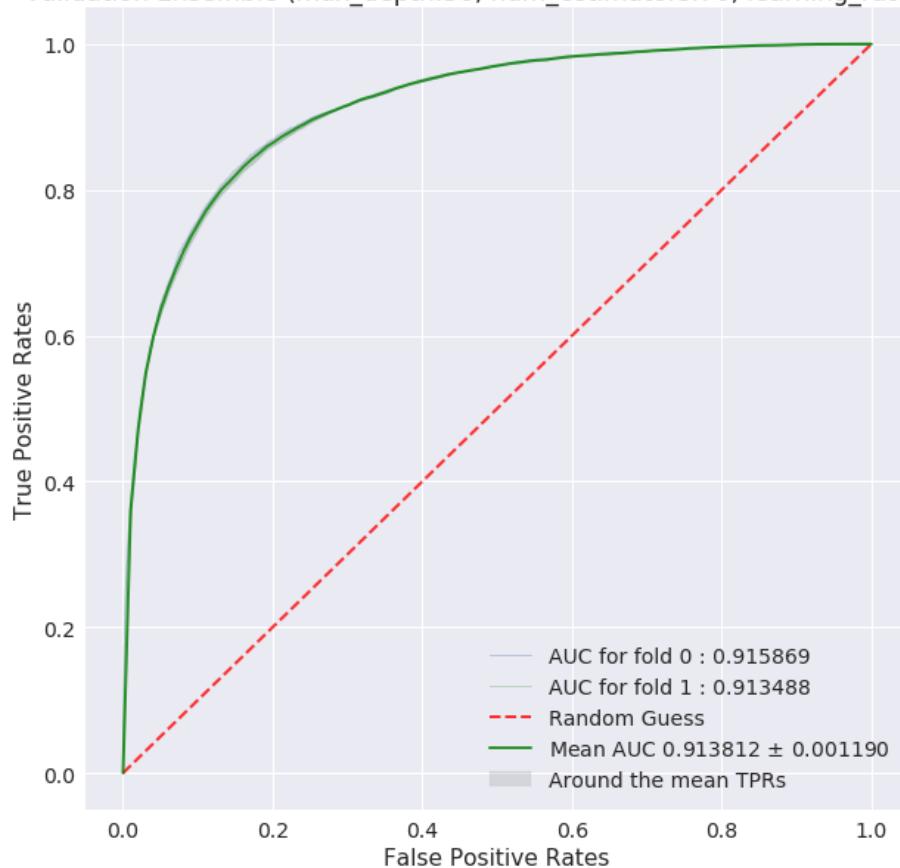


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:70, learning\_rate:0.100000)

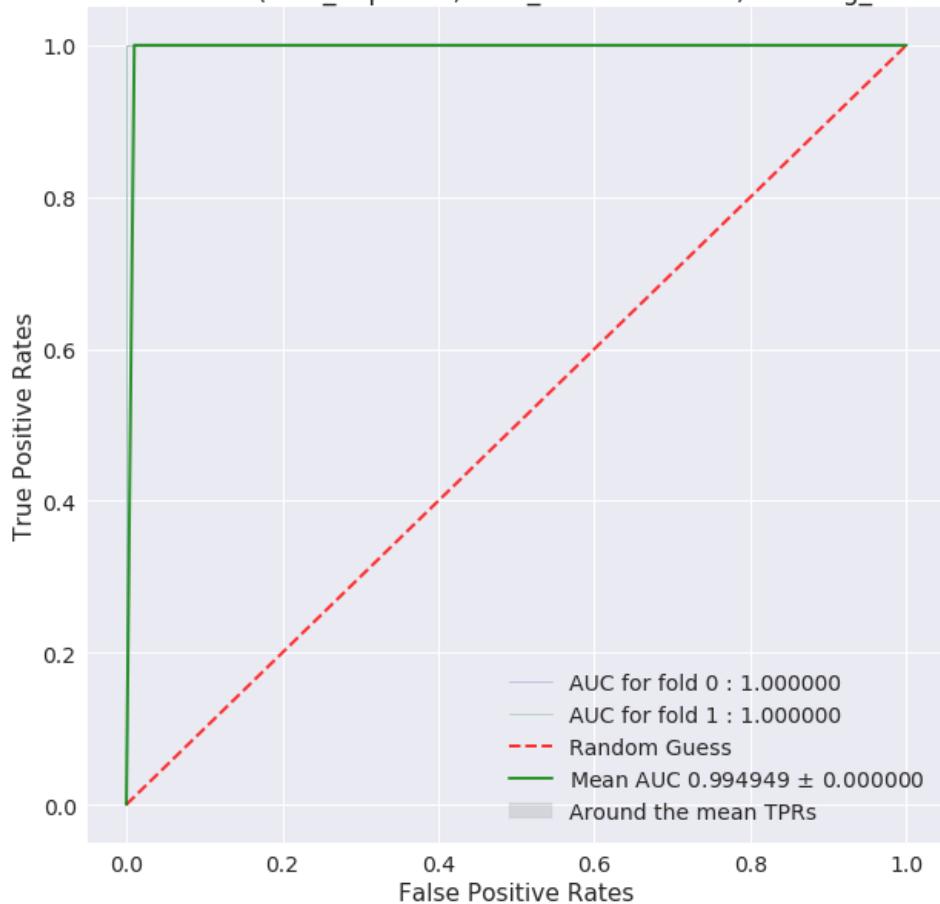


ROC - Validation Ensemble (max\_depth:50, num\_estimators:70, learning\_rate:0.100000)

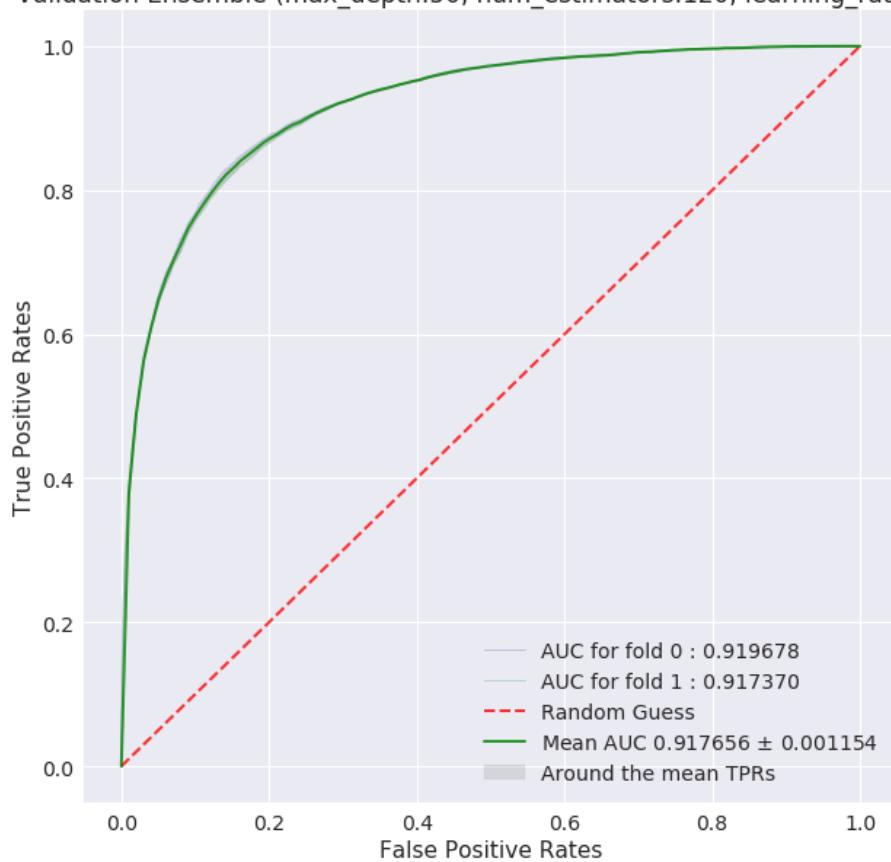


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:120, learning\_rate:0.100000)



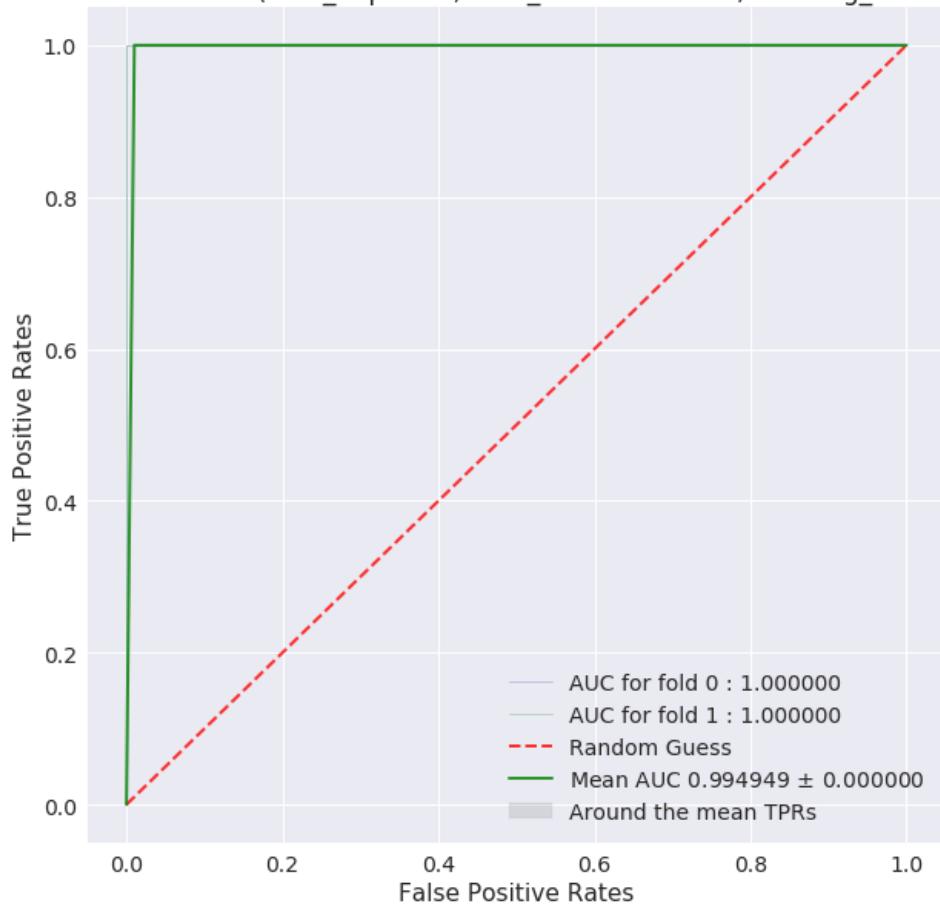
ROC - Validation Ensemble (max\_depth:50, num\_estimators:120, learning\_rate:0.100000)



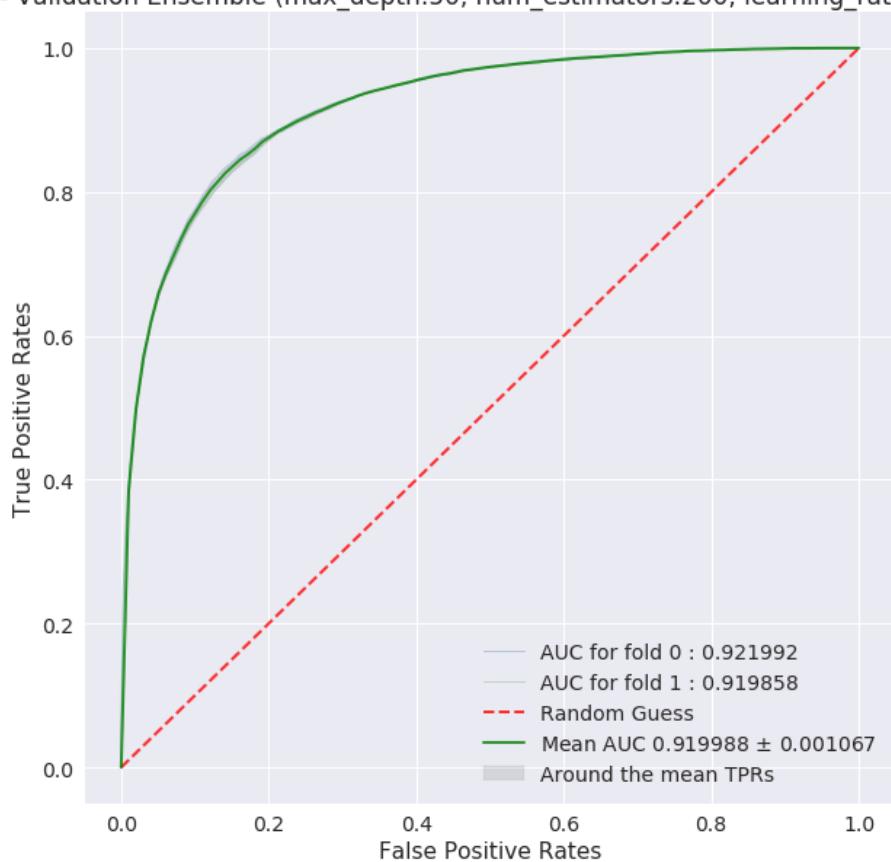
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:200, learning\_rate:0.100000)



ROC - Validation Ensemble (max\_depth:50, num\_estimators:200, learning\_rate:0.100000)



=====  
Train hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.888970
1	(2, 70, 0.1)	0.903864
2	(2, 120, 0.1)	0.916026
3	(2, 200, 0.1)	0.925581
4	(5, 40, 0.1)	0.936312
5	(5, 70, 0.1)	0.951857
6	(5, 120, 0.1)	0.965921
7	(5, 200, 0.1)	0.978765
8	(20, 40, 0.1)	0.994949
9	(20, 70, 0.1)	0.994949
10	(20, 120, 0.1)	0.994949
11	(20, 200, 0.1)	0.994949
12	(50, 40, 0.1)	0.994949
13	(50, 70, 0.1)	0.994949

```
14 (50, 120, 0.1) 0.994949
15 (50, 200, 0.1) 0.994949
```

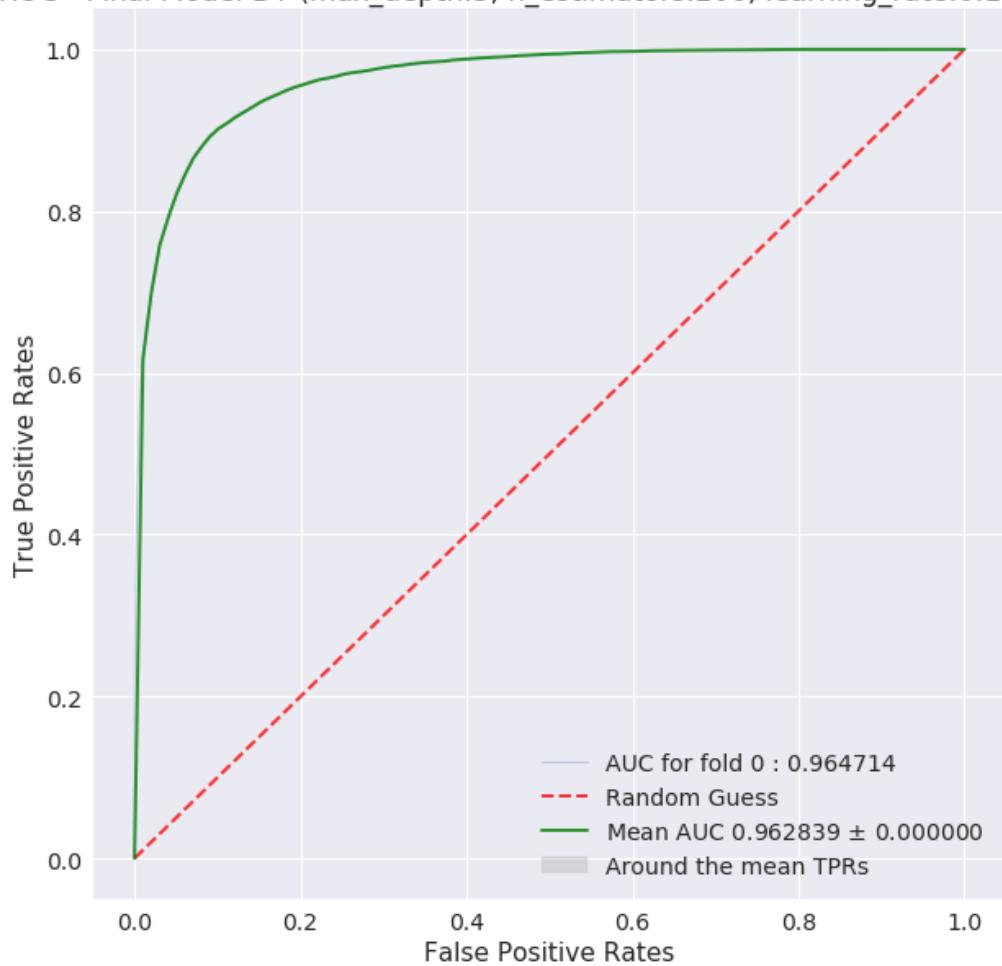
Validation hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.879937
1	(2, 70, 0.1)	0.893345
2	(2, 120, 0.1)	0.903581
3	(2, 200, 0.1)	0.910347
4	(5, 40, 0.1)	0.905202
5	(5, 70, 0.1)	0.913337
6	(5, 120, 0.1)	0.918316
7	(5, 200, 0.1)	0.920401
8	(20, 40, 0.1)	0.907806
9	(20, 70, 0.1)	0.913703
10	(20, 120, 0.1)	0.917515
11	(20, 200, 0.1)	0.919903
12	(50, 40, 0.1)	0.907665
13	(50, 70, 0.1)	0.913812
14	(50, 120, 0.1)	0.917656
15	(50, 200, 0.1)	0.919988

Best hyperparam value: (5, 200, 0.1)

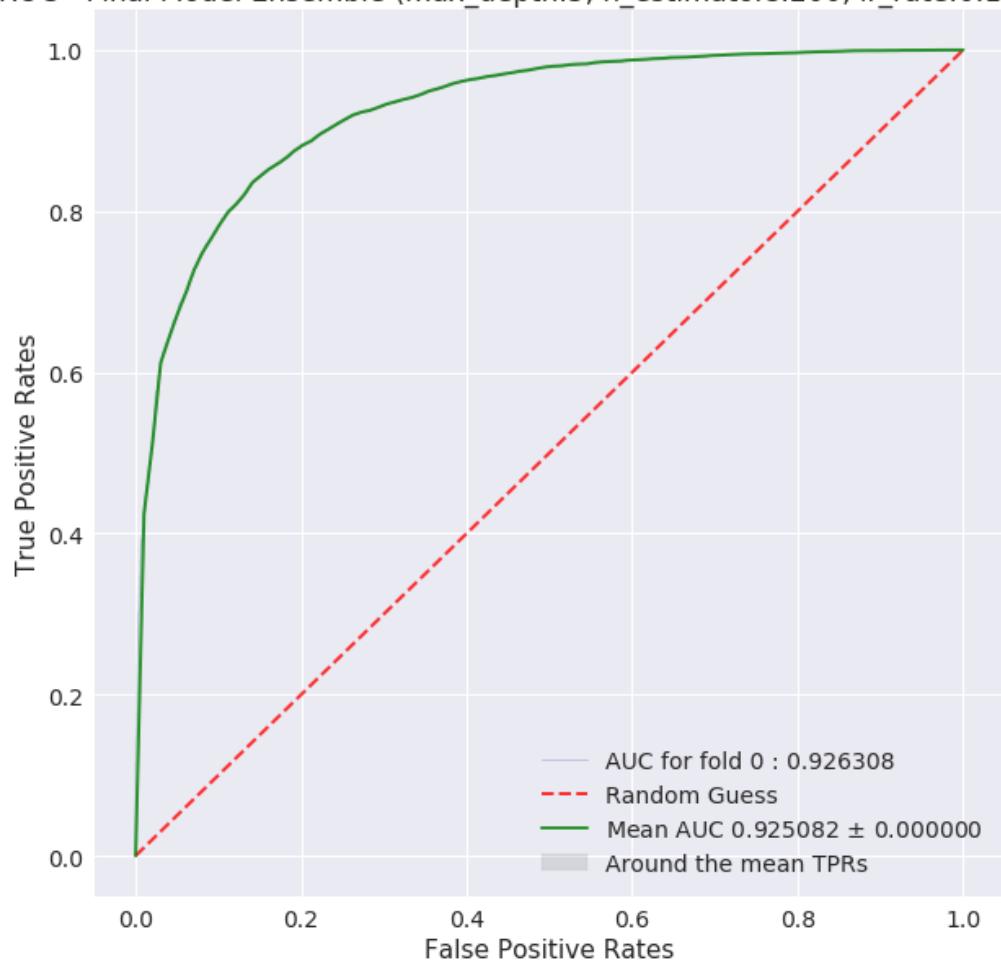
```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
```

ROC - Final Model DT (max\_depth:5, n\_estimators:200, learning\_rate:0.100000)

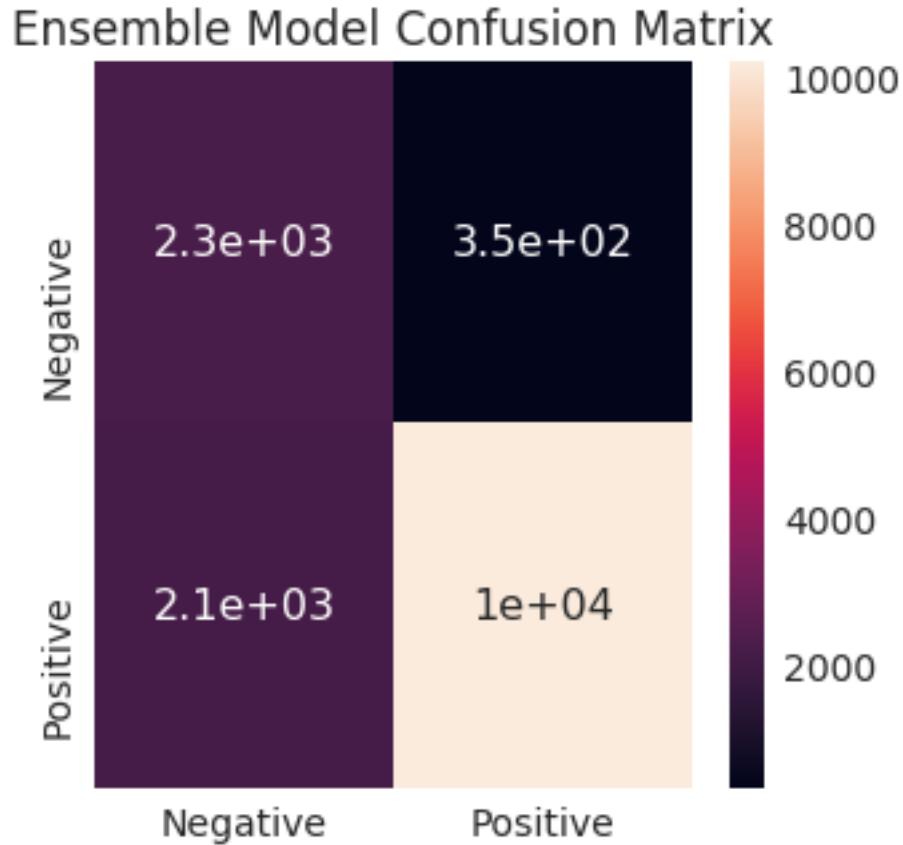


```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning
  if diff:
```

ROC - Final Model Ensemble (max\_depth:5, n\_estimators:200, lr\_rate:0.100000)



Test auc score 0.9250815926094467



	Negative	Positive
Precision	0.512698	0.966667
Recall	0.864958	0.826498
Fscore	0.643793	0.891104
Support	2614.000000	12386.000000

#### 4.5.4 [B.4] Applying XGBOOST on TFIDF W2V, SET 4

```
In [23]: # form two lists
depth_list = [2, 5, 20, 50] # depends on size of dataset
n_estimators_list = [40, 70, 120, 200] # depends on size of dataset
learning_rate_list = [0.1] # learning rate for XGB training

# create a configuration dictionary
config_dict = {
    'train_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF_W2V',
    'test_csv_path' : '/home/amd_3/AAIC/ASM_REPO/Processed_data/AMZN_FOOD_REVIEWS/TFIDF_W2V',
    'train_size' : 40000,
    'test_size' : 15000,
```

```

'hyperparam_list' : list(product(depth_list, n_estimators_list, learning_rate_list))
'implementation': 'xgb' # 'xgb' or 'rf'
}

In [24]: # read the train, test data and preprocess it
train_features, train_labels, test_features, test_labels = preprocess_data(config_dict,
                                                               scaling=True
                                                               dim_reductio

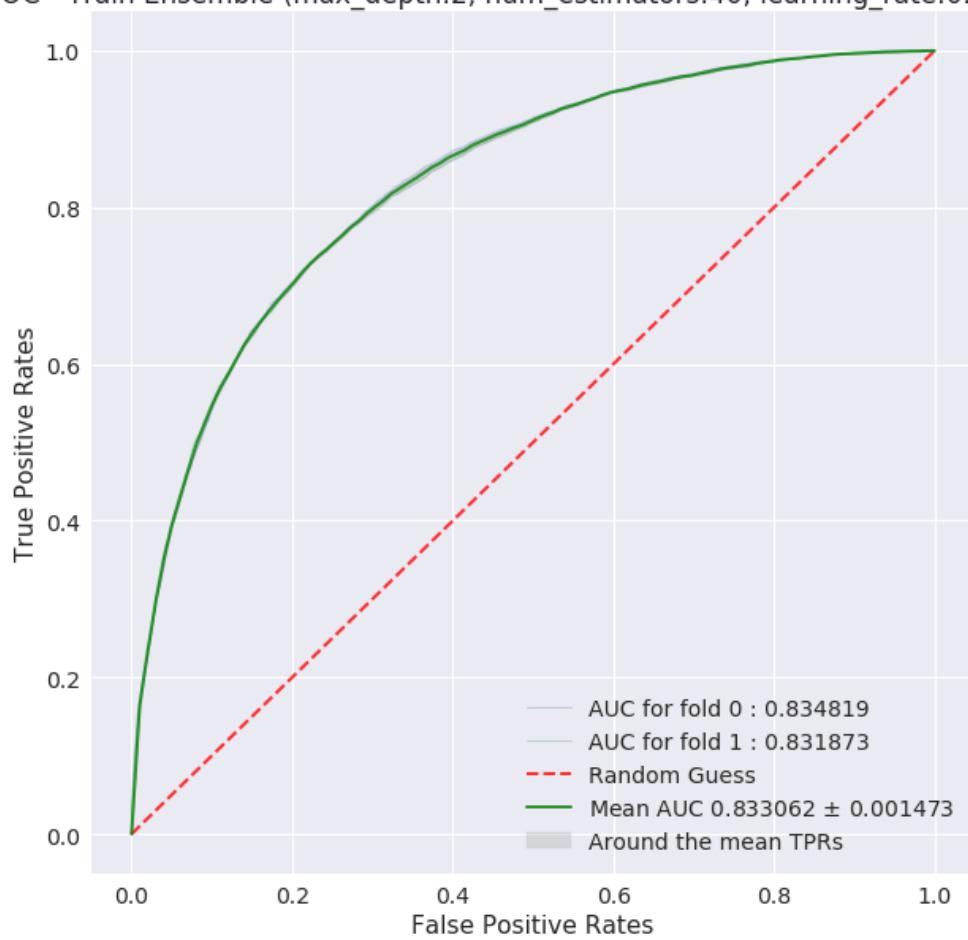
# train and validate the model
model = train_and_validate_model(config_dict, train_features, train_labels)

# test and evaluate the model
ptabe_entry_b4 = test_and_evaluate_model(config_dict, model, test_features, test_labels)

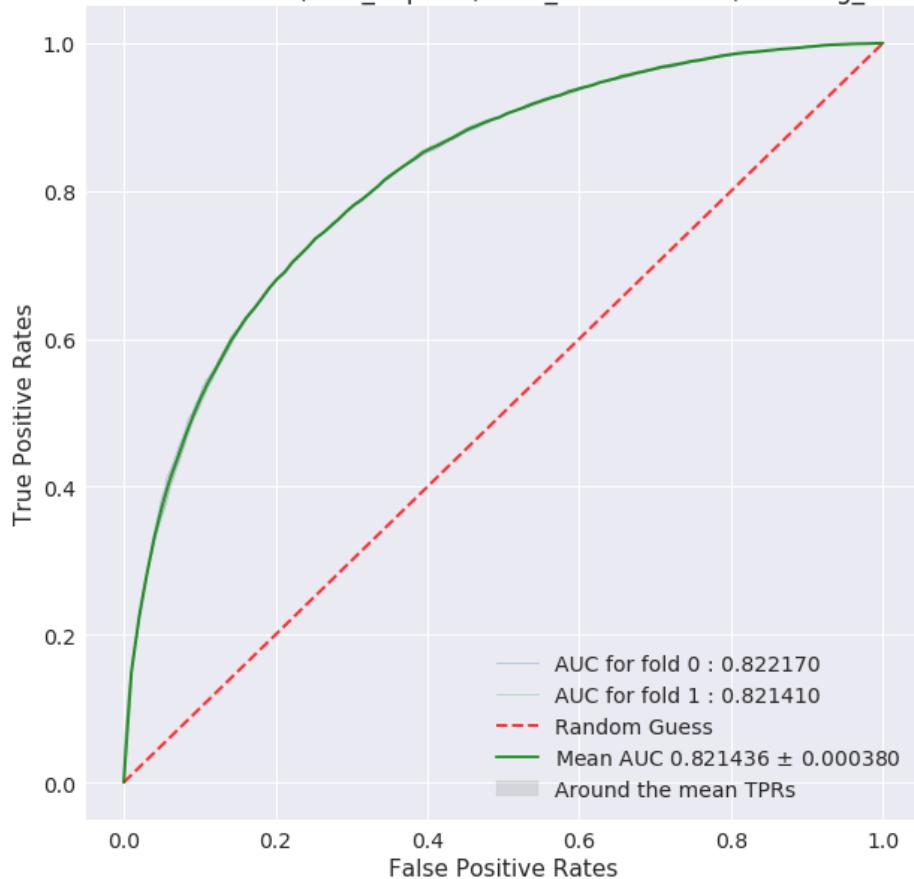
Train df shape (40000, 52)
Class label distribution in train df:
0    20024
1    19976
Name: Label, dtype: int64
Test df shape (15000, 52)
Class label distribution in test df:
1    12386
0    2614
Name: Label, dtype: int64
Shape of -> train features :40000,50, test features: 15000,50
Shape of -> train labels :40000, test labels: 15000
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:

```

ROC - Train Ensemble (max\_depth:2, num\_estimators:40, learning\_rate:0.100000)

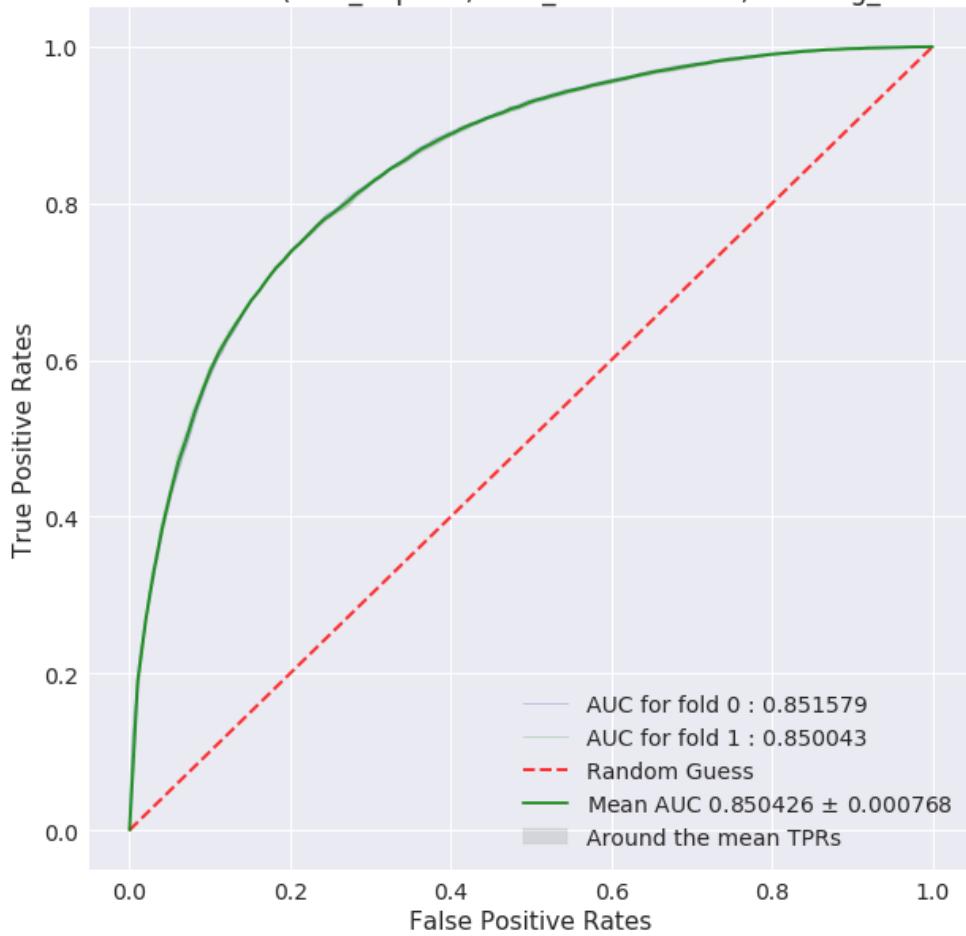


ROC - Validation Ensemble (max\_depth:2, num\_estimators:40, learning\_rate:0.100000)

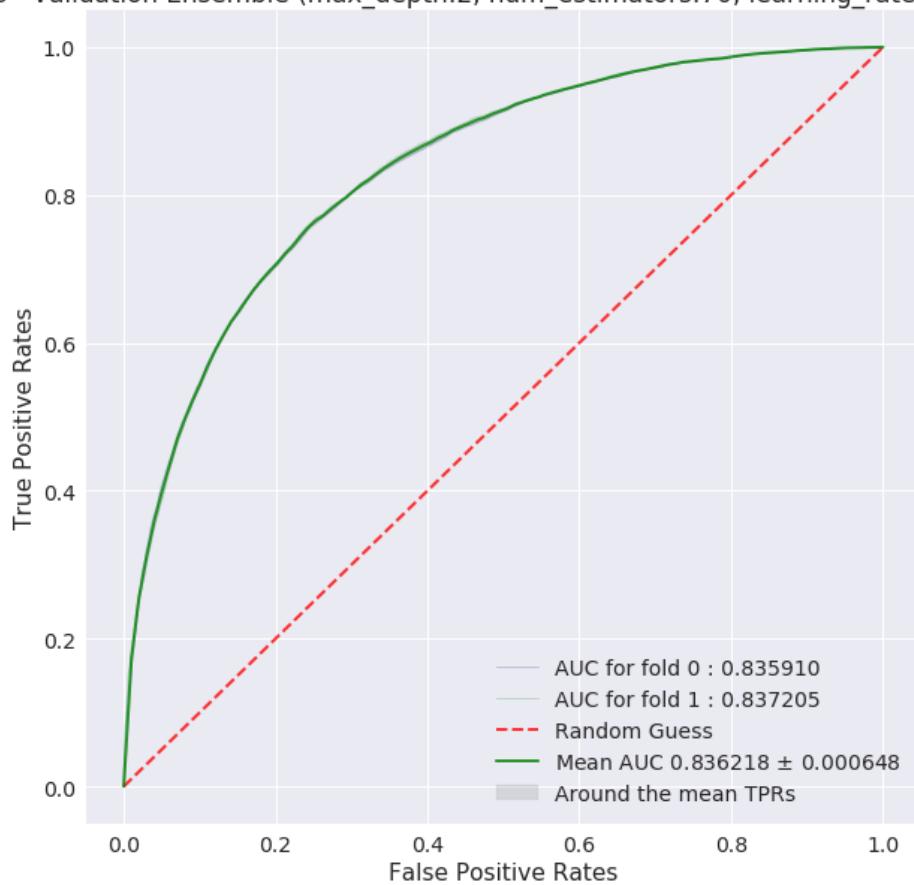


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:70, learning\_rate:0.100000)

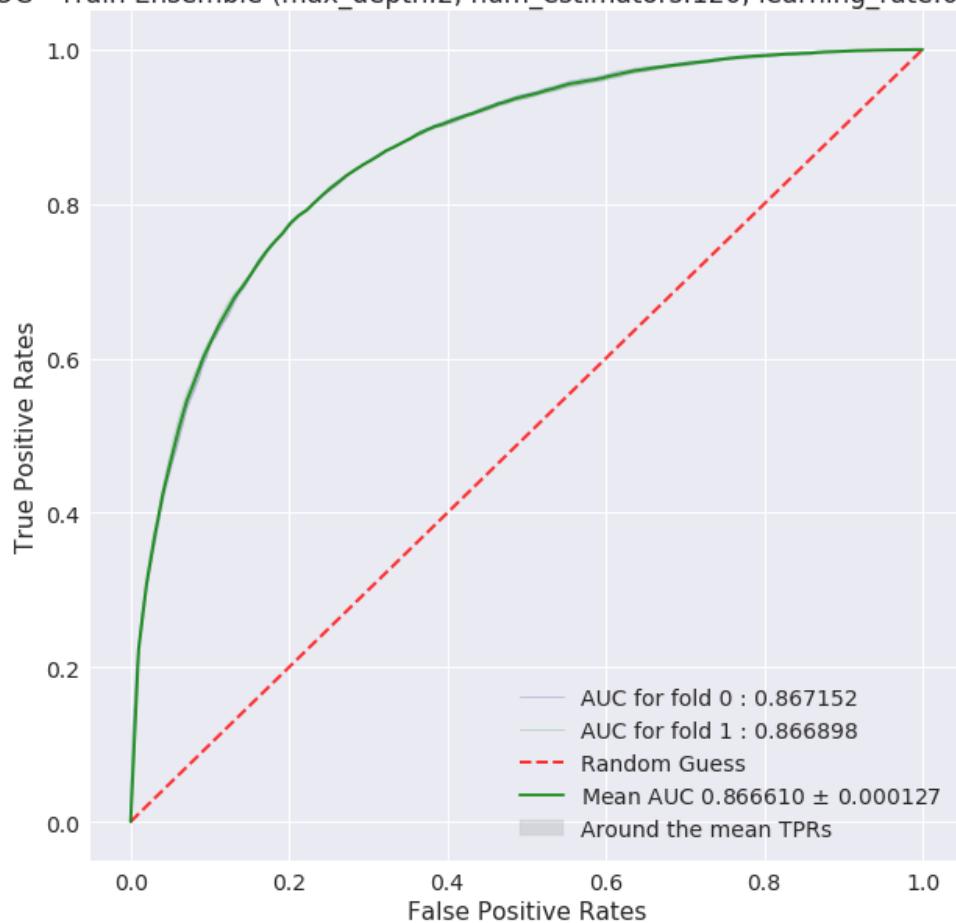


ROC - Validation Ensemble (max\_depth:2, num\_estimators:70, learning\_rate:0.100000)

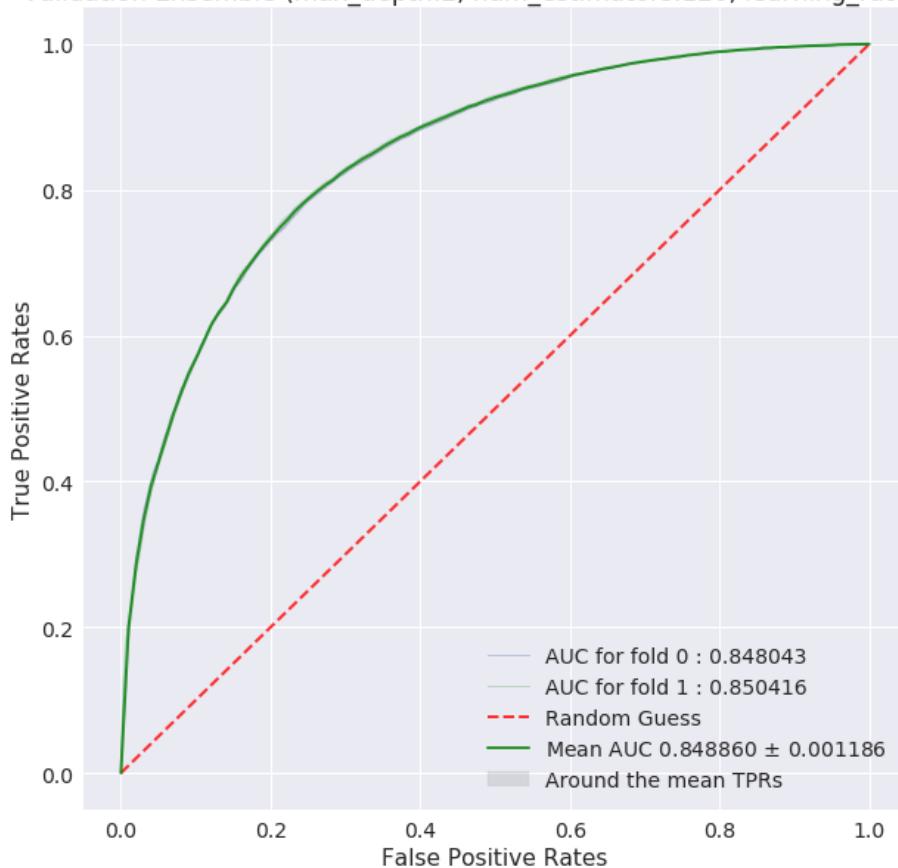


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:120, learning\_rate:0.100000)

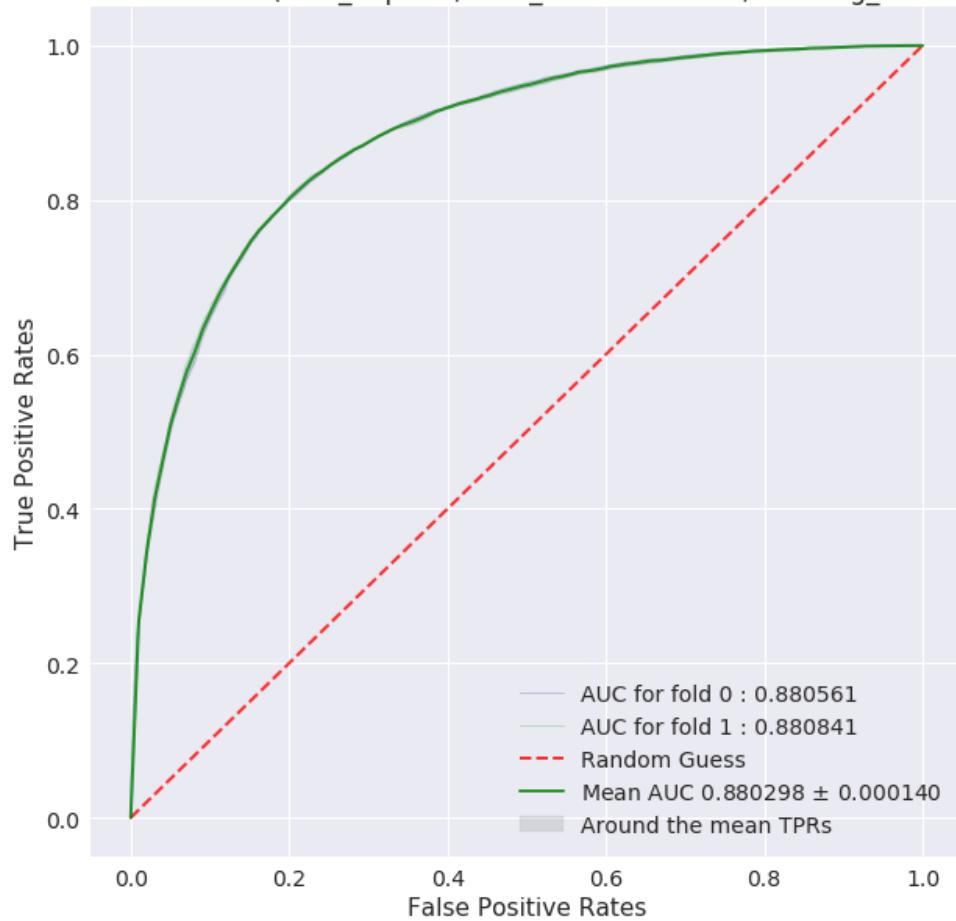


ROC - Validation Ensemble (max\_depth:2, num\_estimators:120, learning\_rate:0.100000)

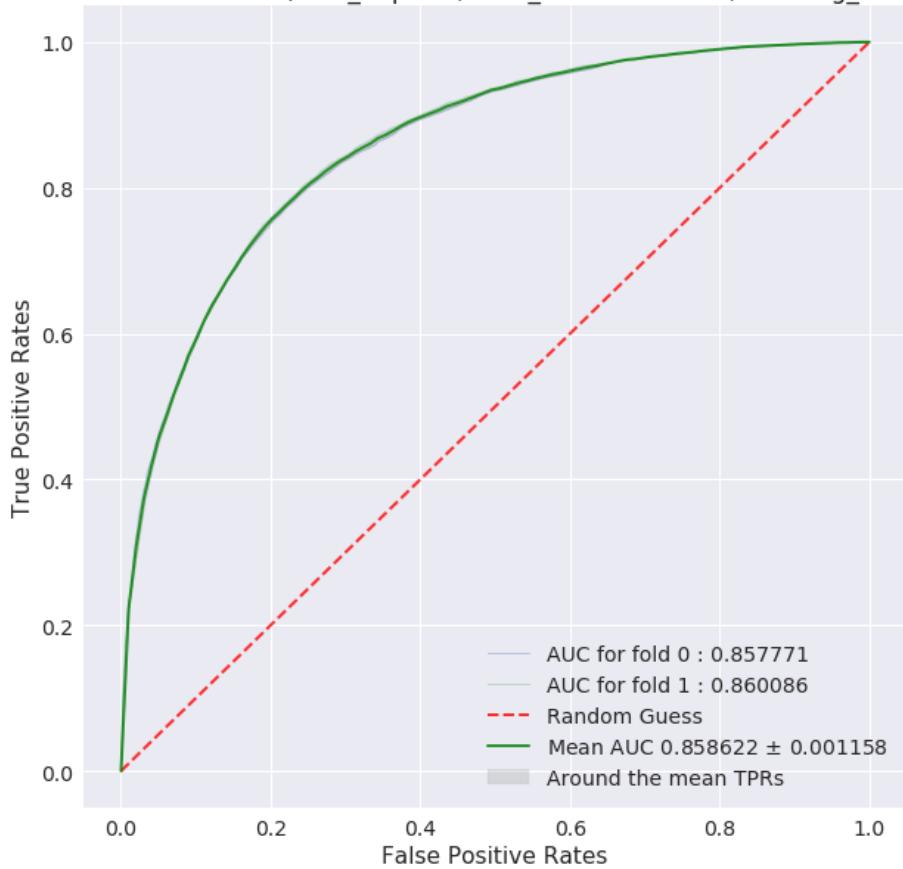


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:2, num\_estimators:200, learning\_rate:0.100000)

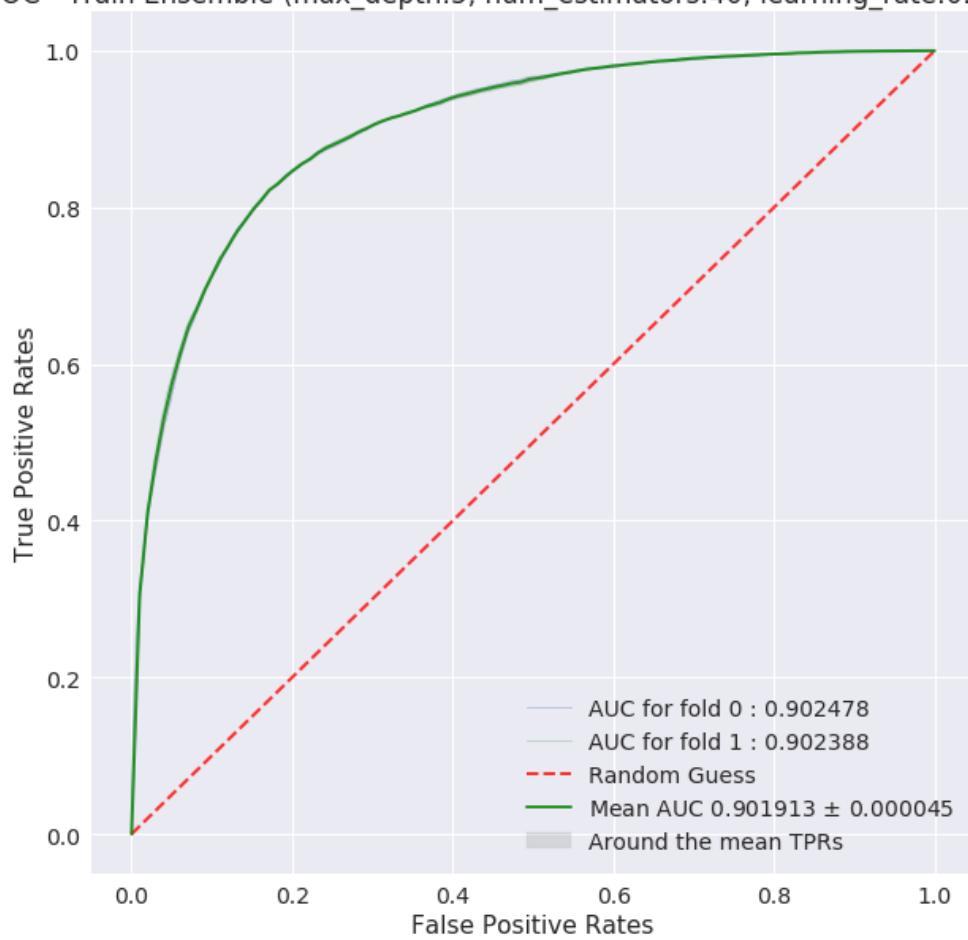


ROC - Validation Ensemble (max\_depth:2, num\_estimators:200, learning\_rate:0.100000)

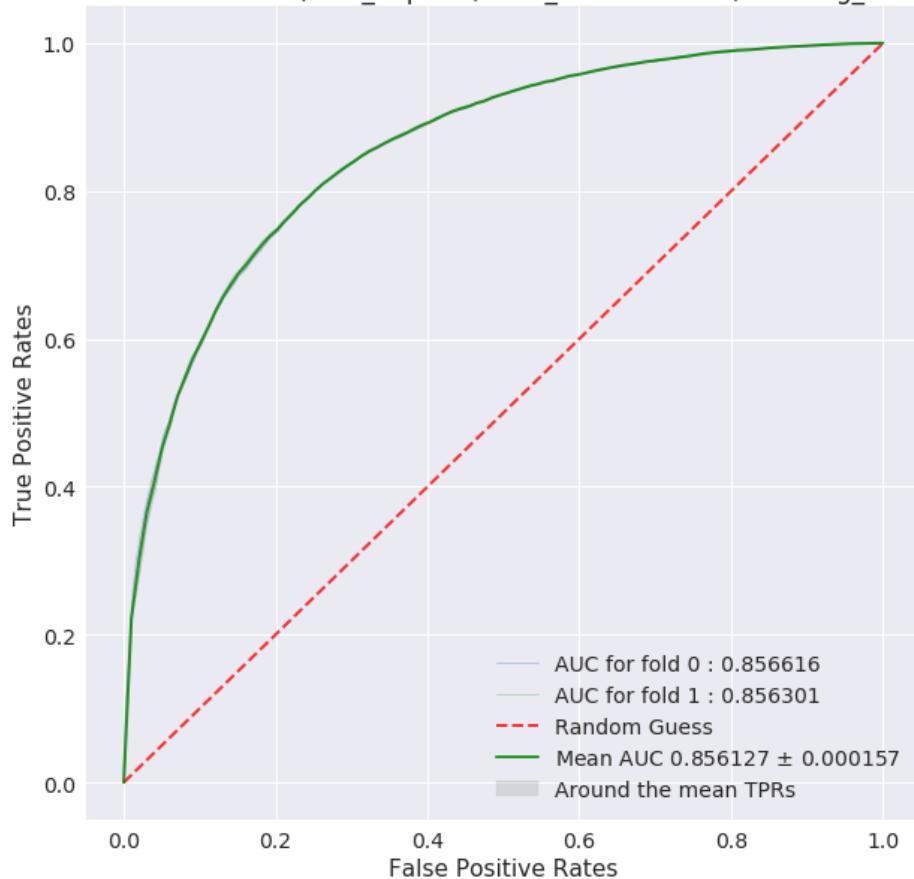


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:40, learning\_rate:0.100000)

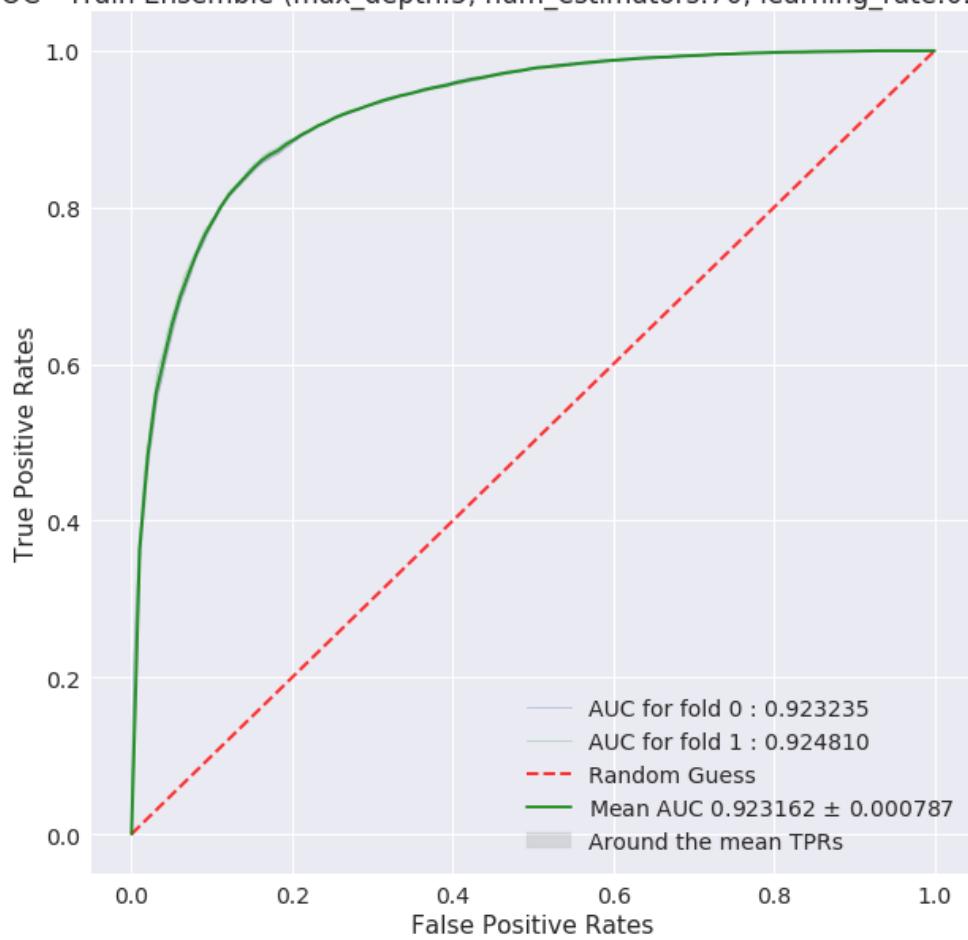


ROC - Validation Ensemble (max\_depth:5, num\_estimators:40, learning\_rate:0.100000)

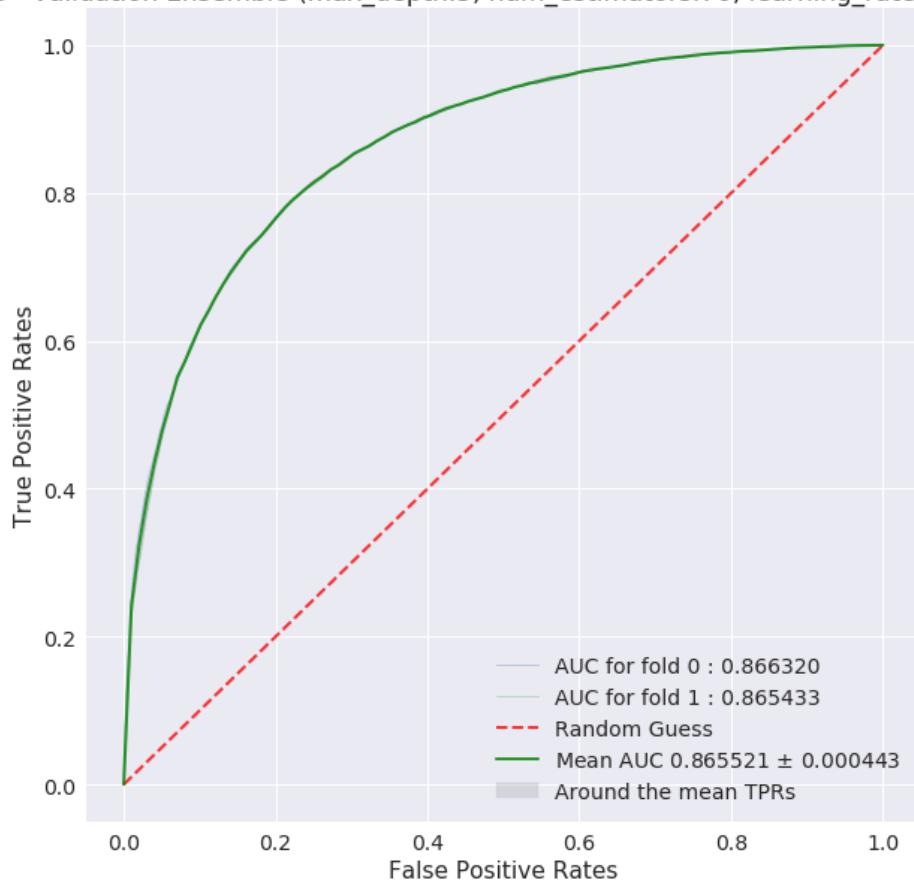


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:70, learning\_rate:0.100000)

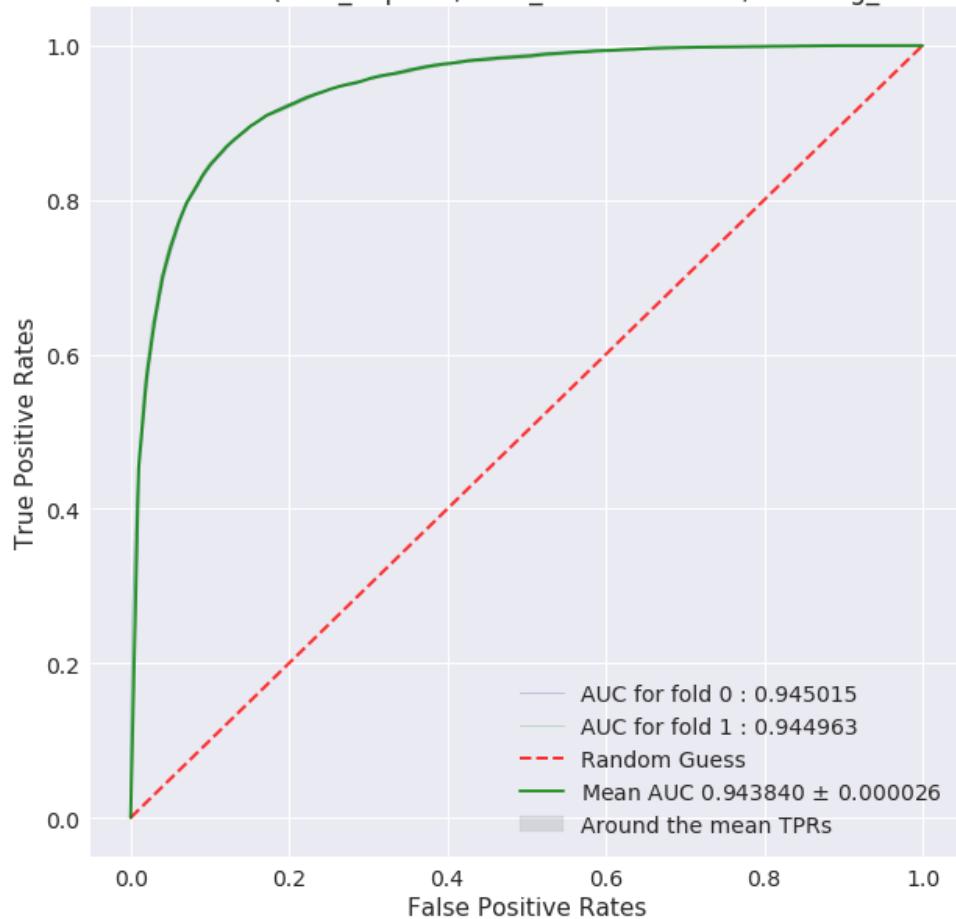


ROC - Validation Ensemble (max\_depth:5, num\_estimators:70, learning\_rate:0.100000)

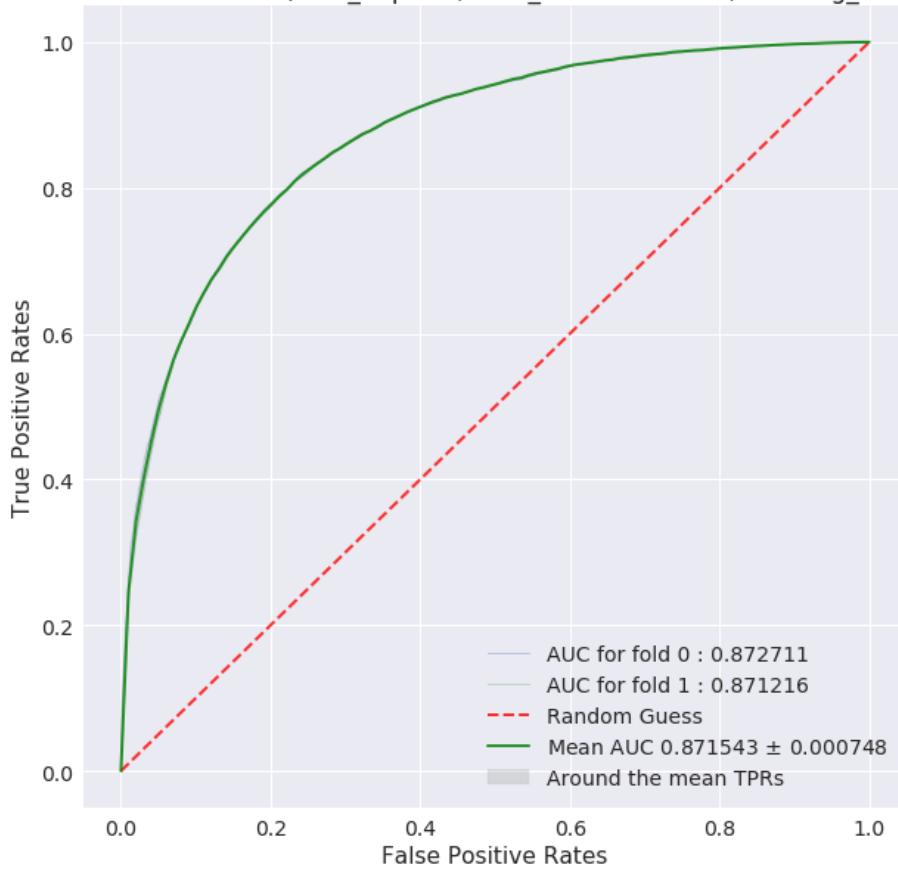


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:120, learning\_rate:0.100000)

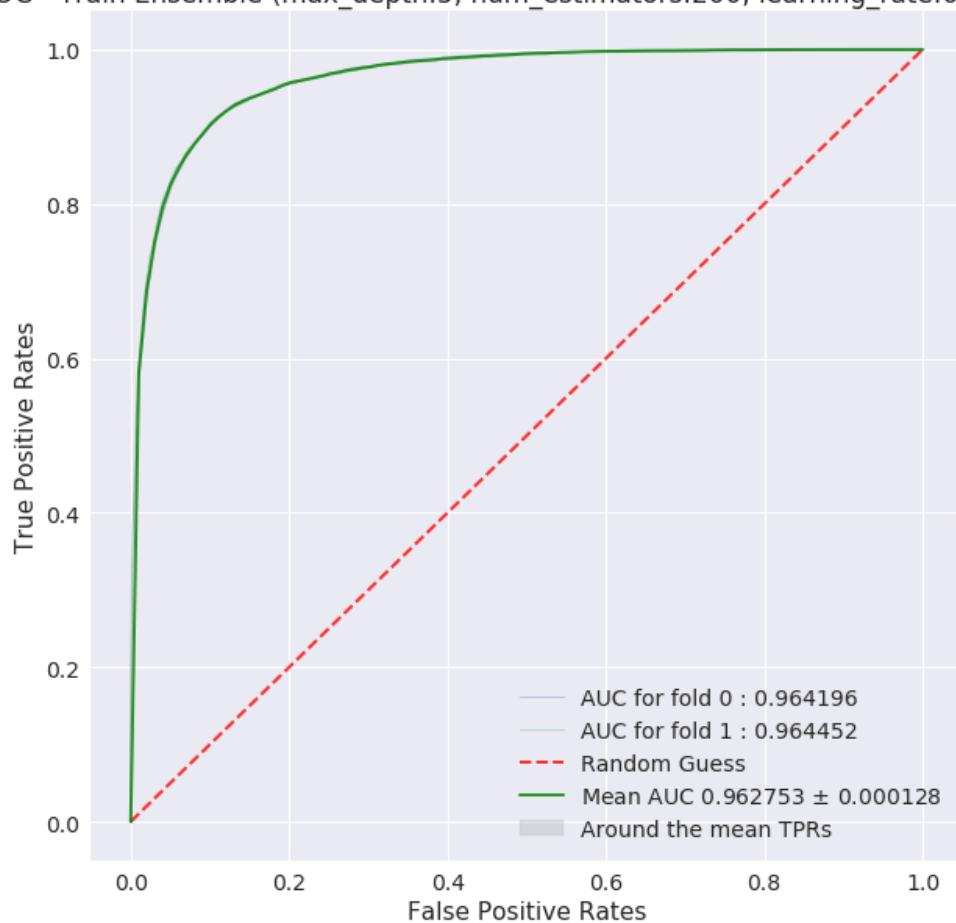


ROC - Validation Ensemble (max\_depth:5, num\_estimators:120, learning\_rate:0.100000)

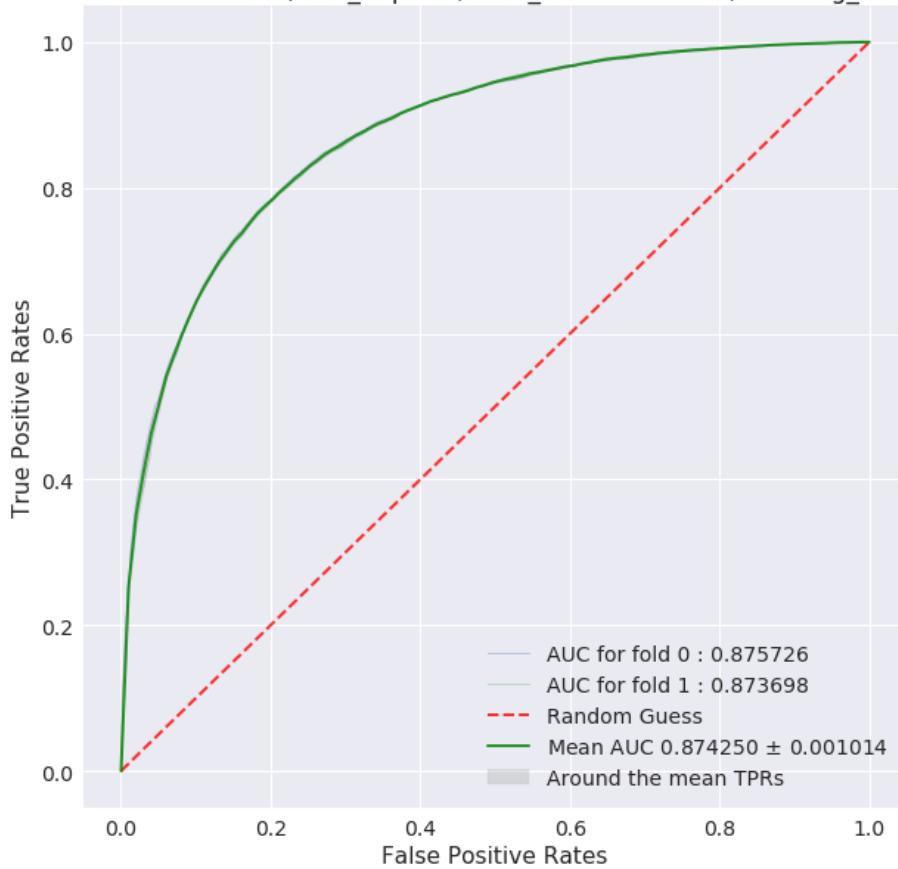


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:5, num\_estimators:200, learning\_rate:0.100000)

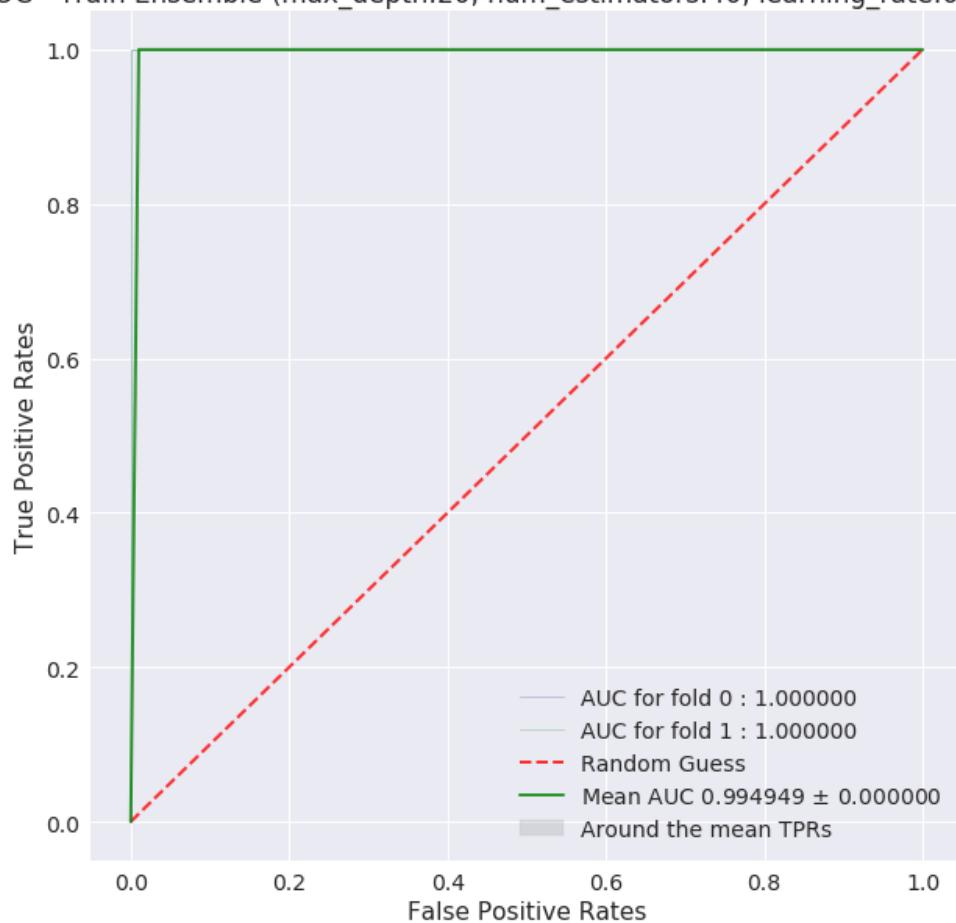


ROC - Validation Ensemble (max\_depth:5, num\_estimators:200, learning\_rate:0.100000)

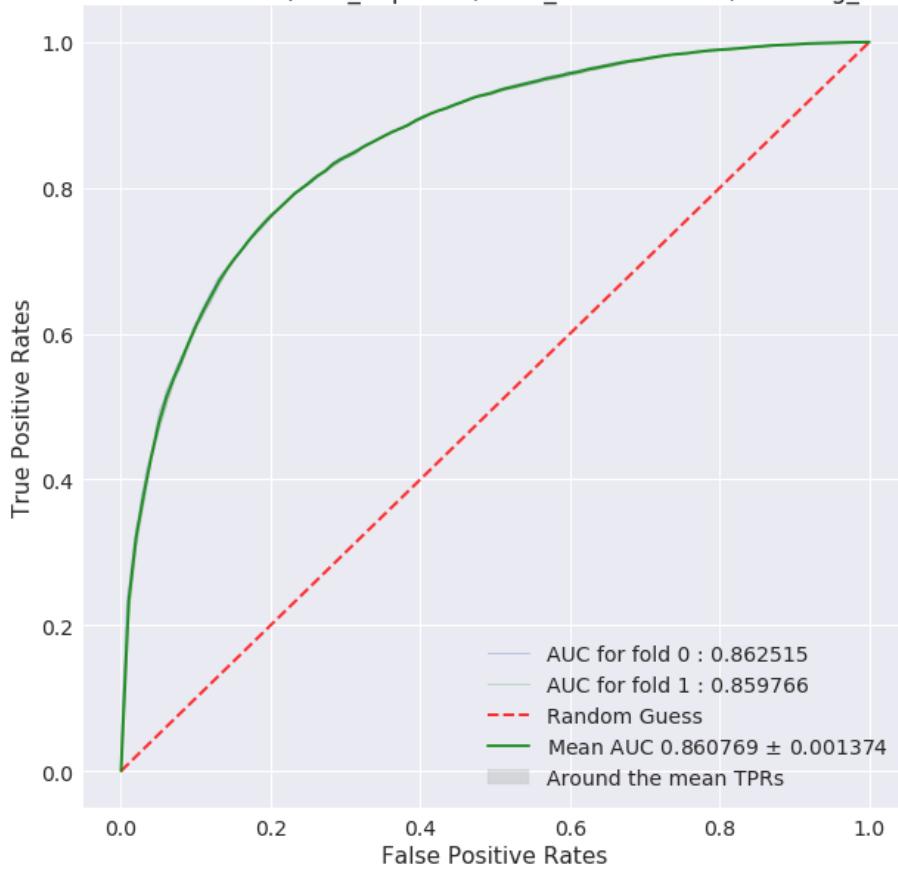


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:40, learning\_rate:0.100000)

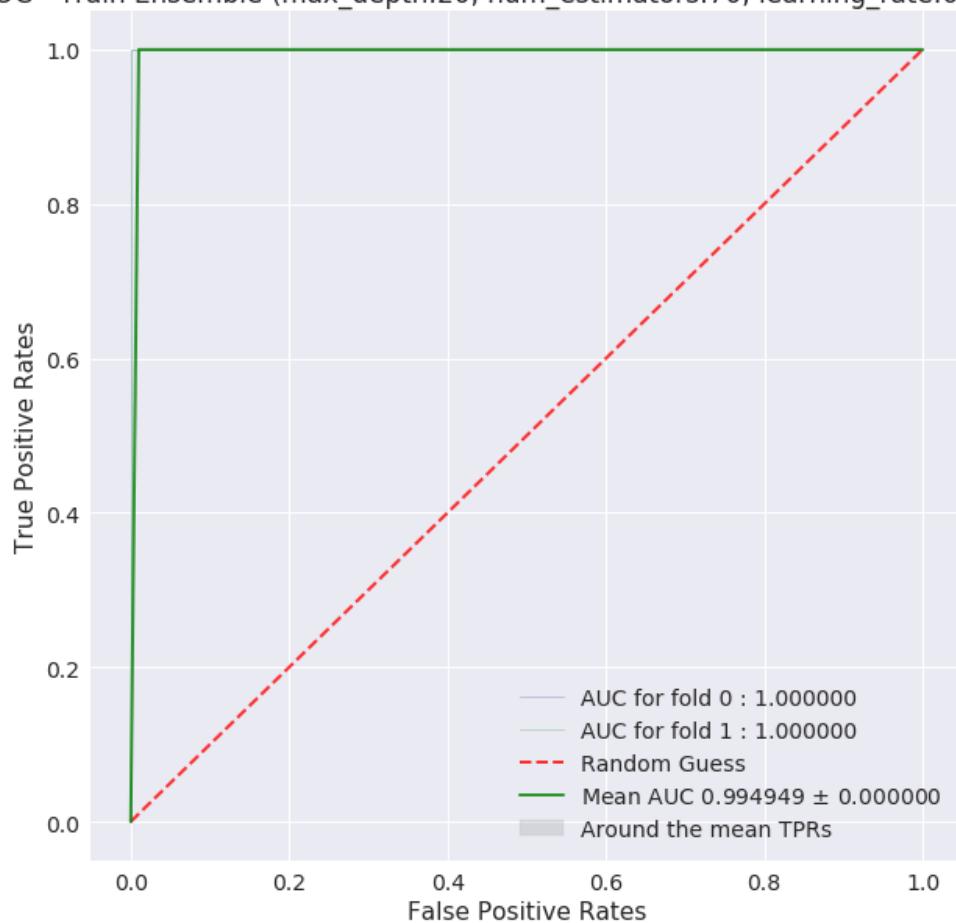


ROC - Validation Ensemble (max\_depth:20, num\_estimators:40, learning\_rate:0.100000)

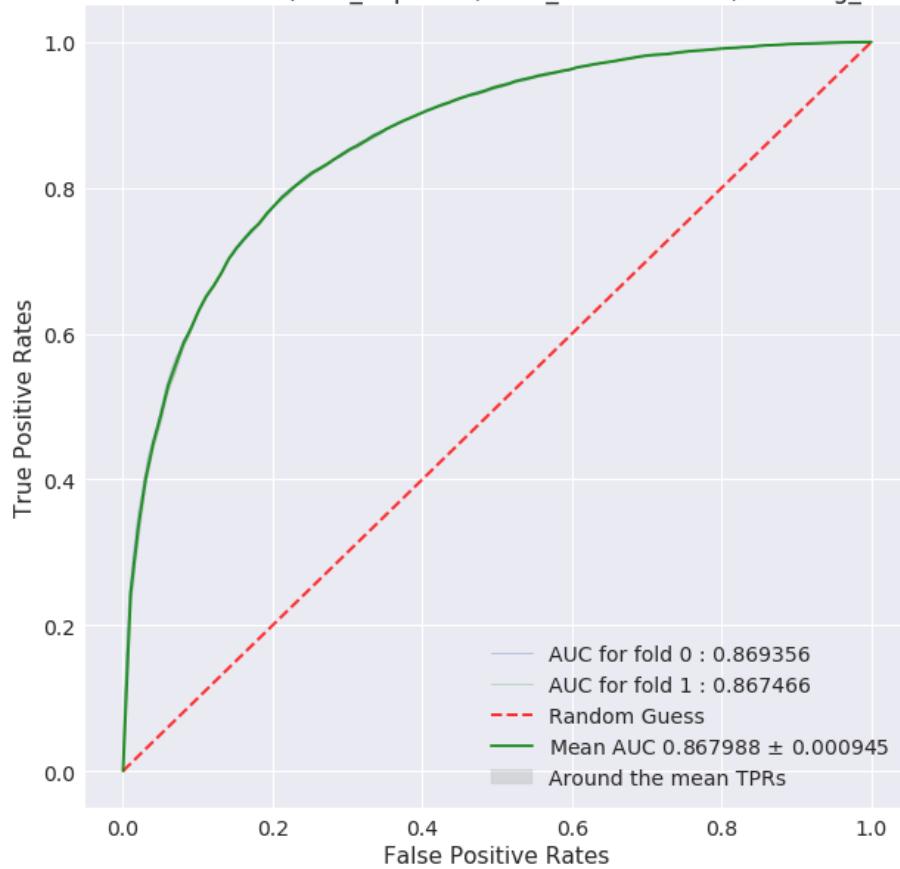


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:70, learning\_rate:0.100000)

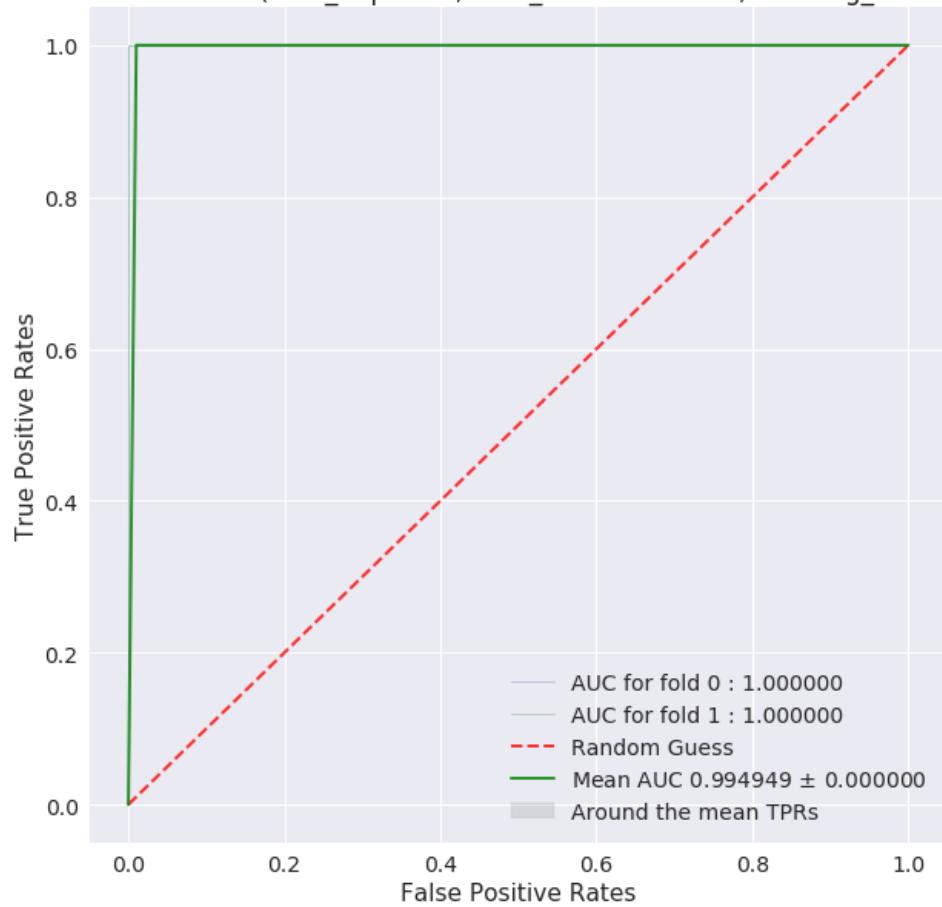


ROC - Validation Ensemble (max\_depth:20, num\_estimators:70, learning\_rate:0.100000)

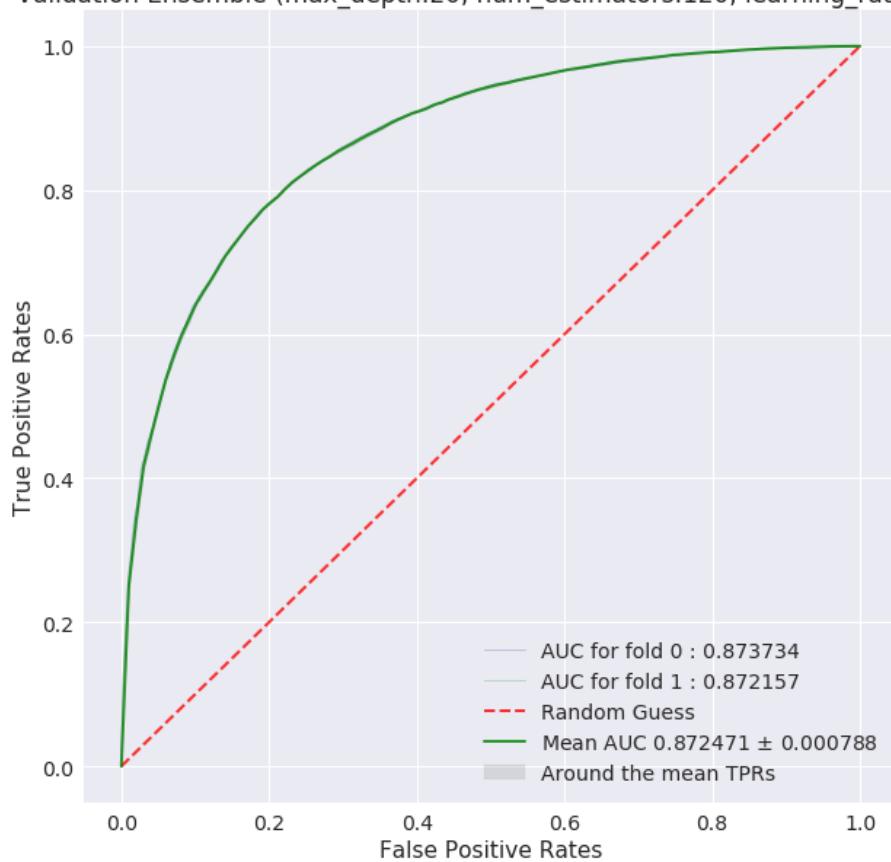


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:120, learning\_rate:0.100000)



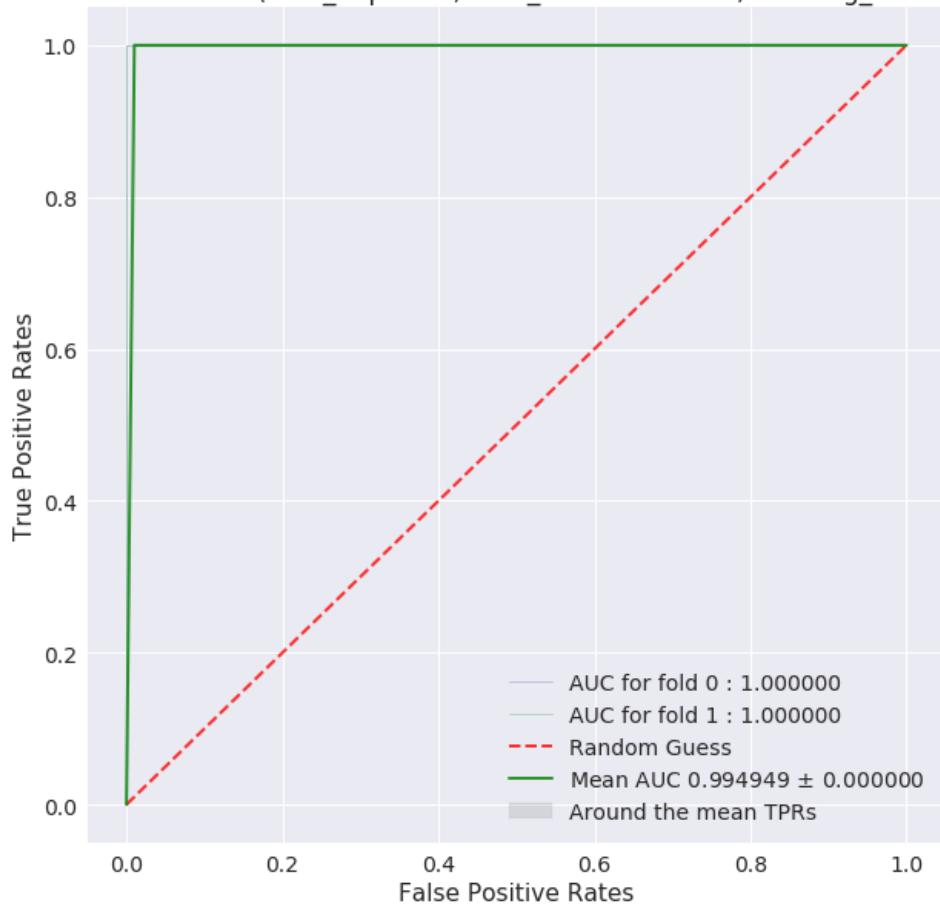
ROC - Validation Ensemble (max\_depth:20, num\_estimators:120, learning\_rate:0.100000)



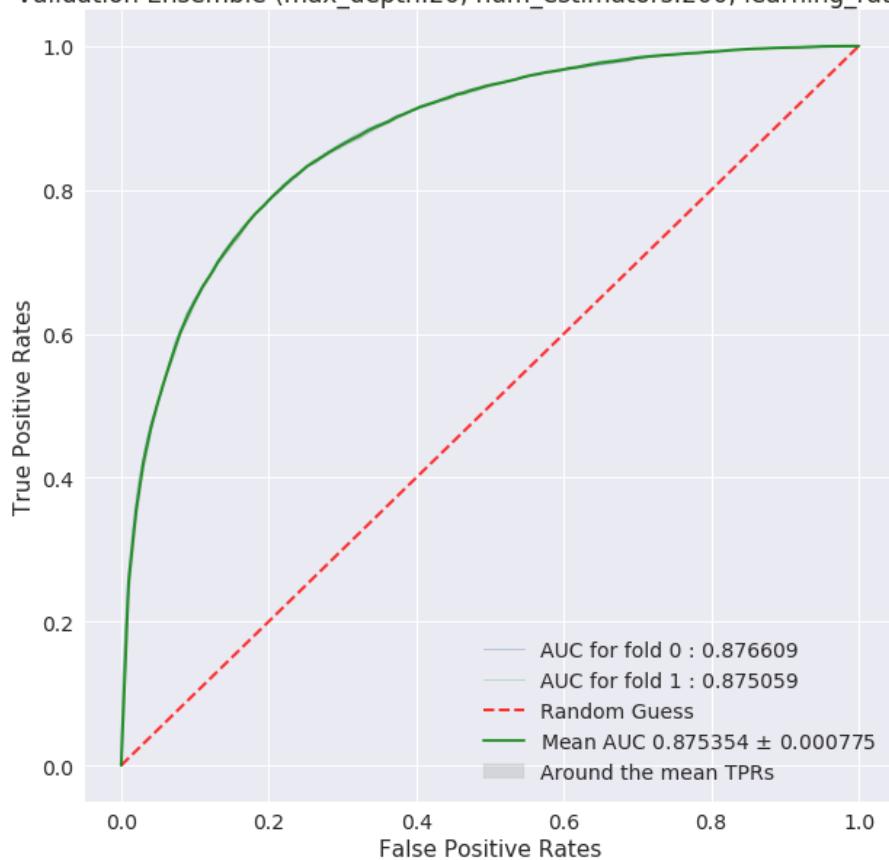
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:20, num\_estimators:200, learning\_rate:0.100000)



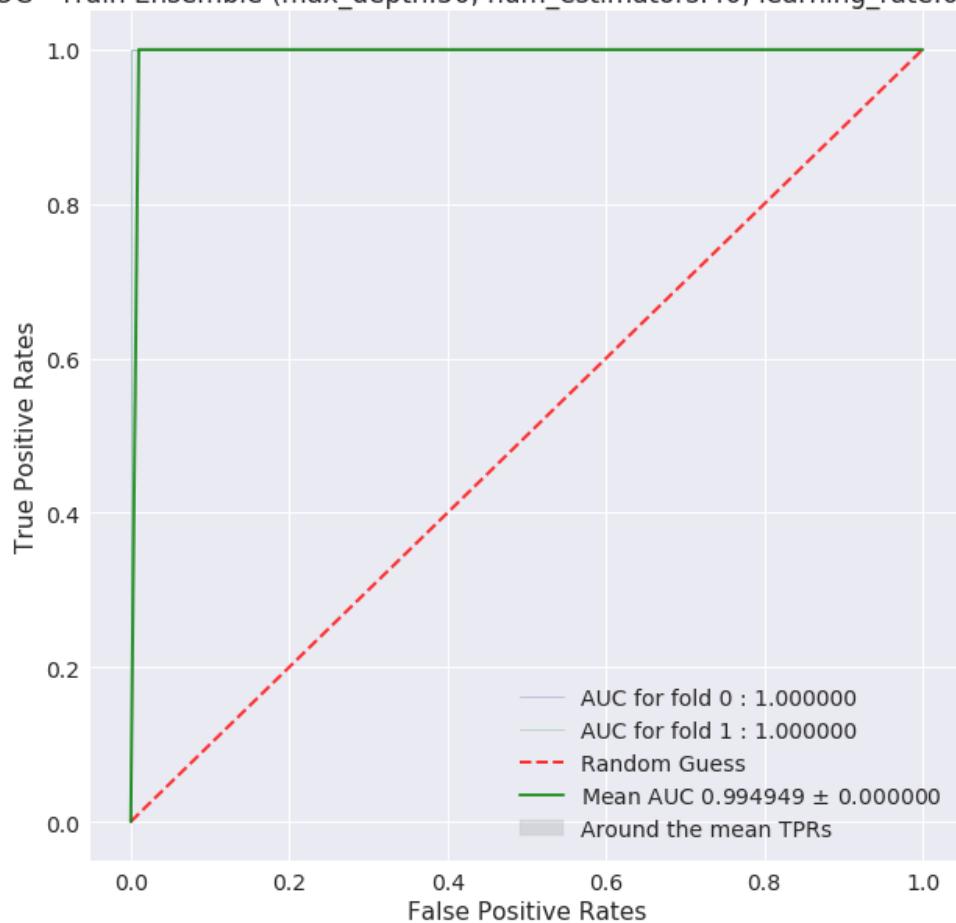
ROC - Validation Ensemble (max\_depth:20, num\_estimators:200, learning\_rate:0.100000)



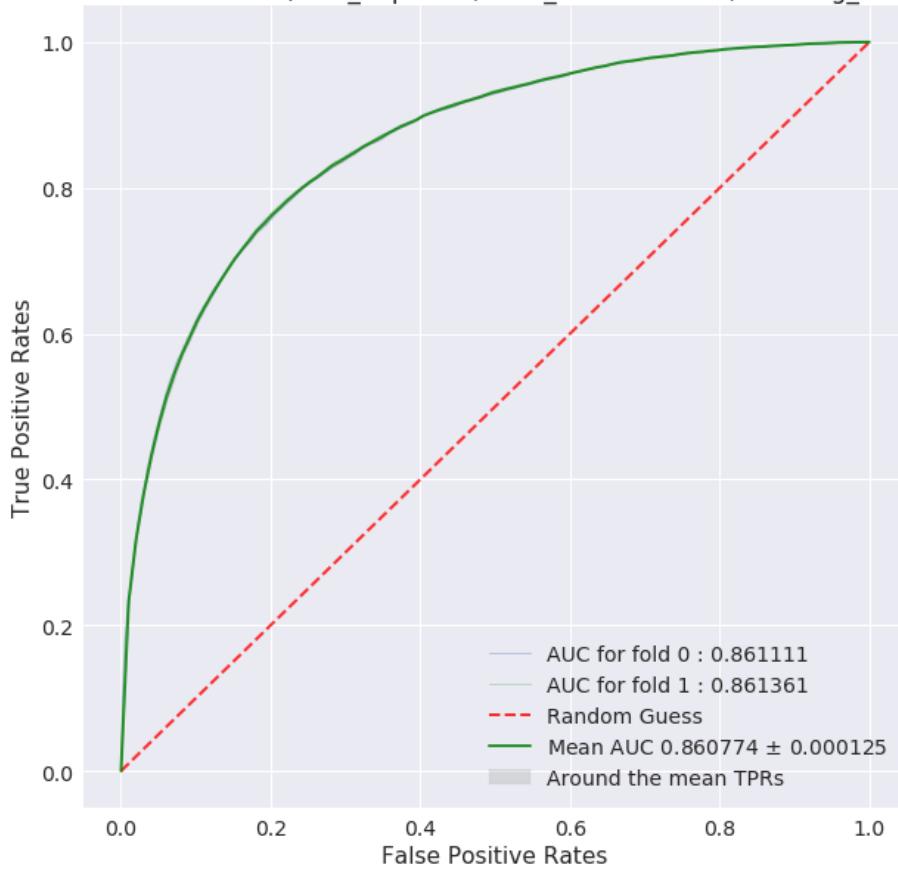
---

```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:  
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:40, learning\_rate:0.100000)

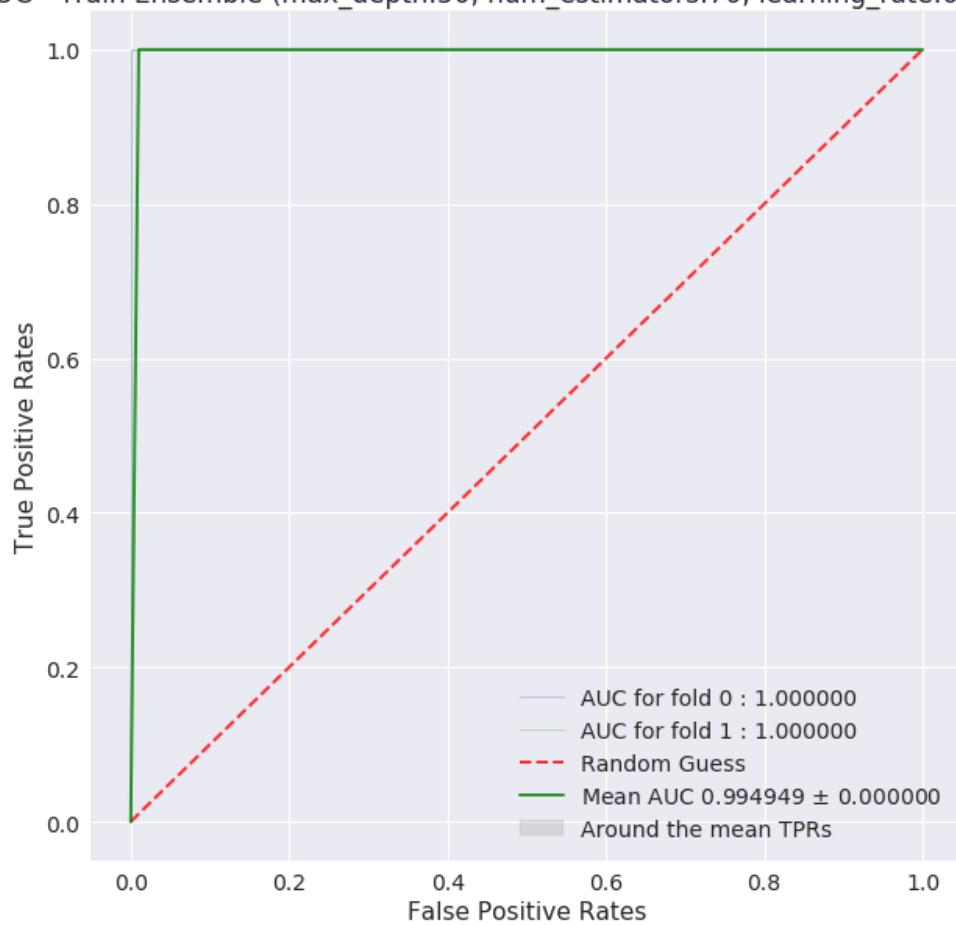


ROC - Validation Ensemble (max\_depth:50, num\_estimators:40, learning\_rate:0.100000)

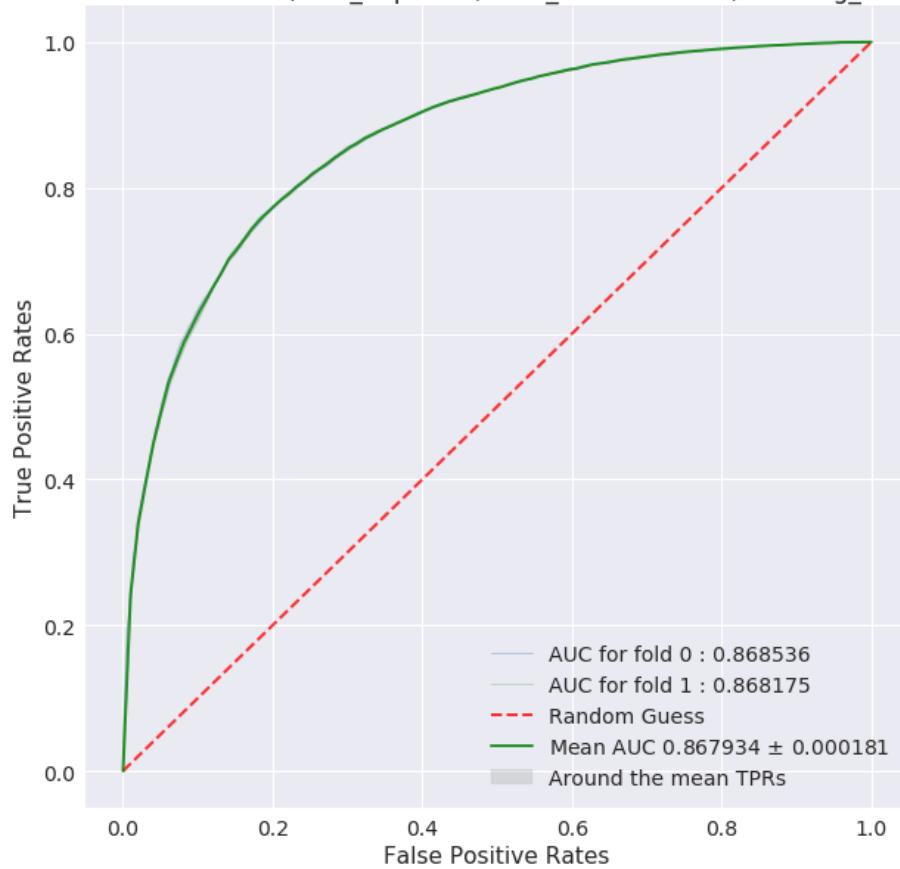


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:70, learning\_rate:0.100000)

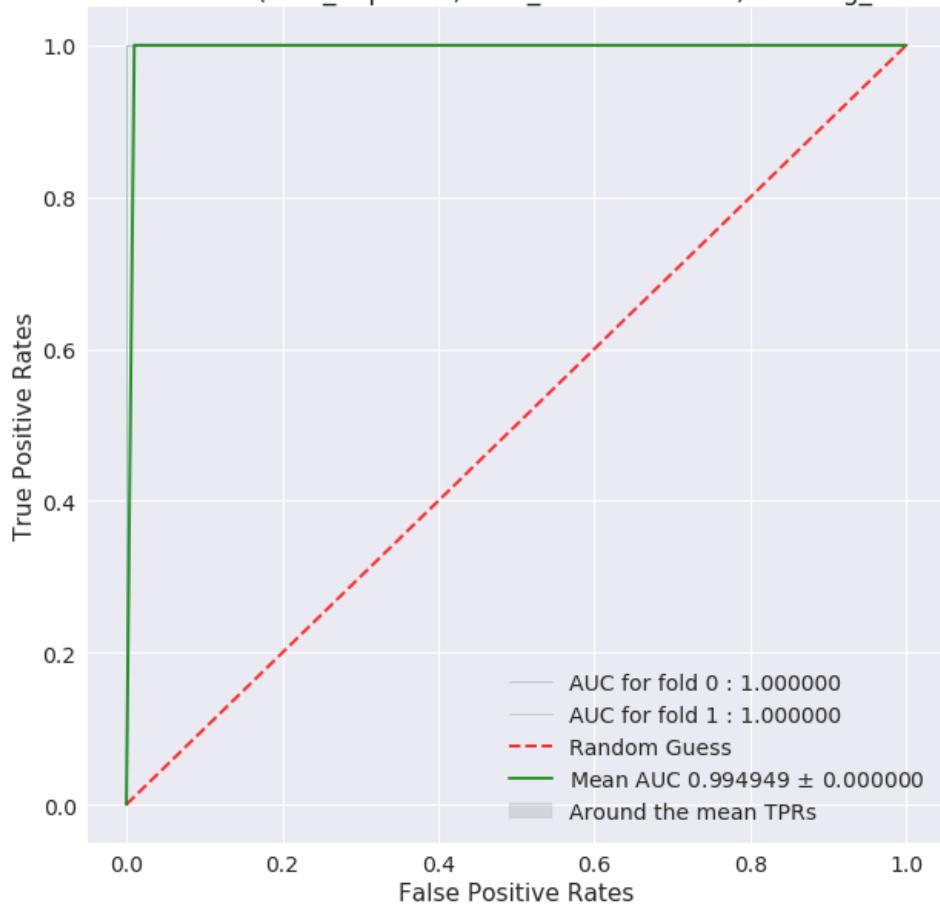


ROC - Validation Ensemble (max\_depth:50, num\_estimators:70, learning\_rate:0.100000)

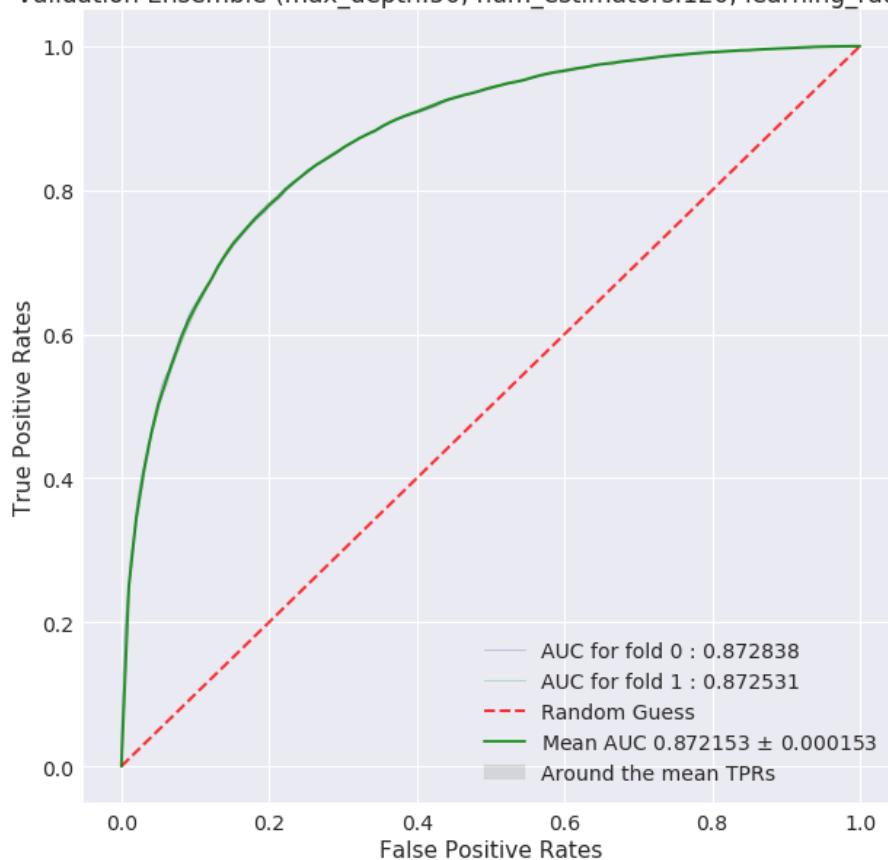


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:120, learning\_rate:0.100000)

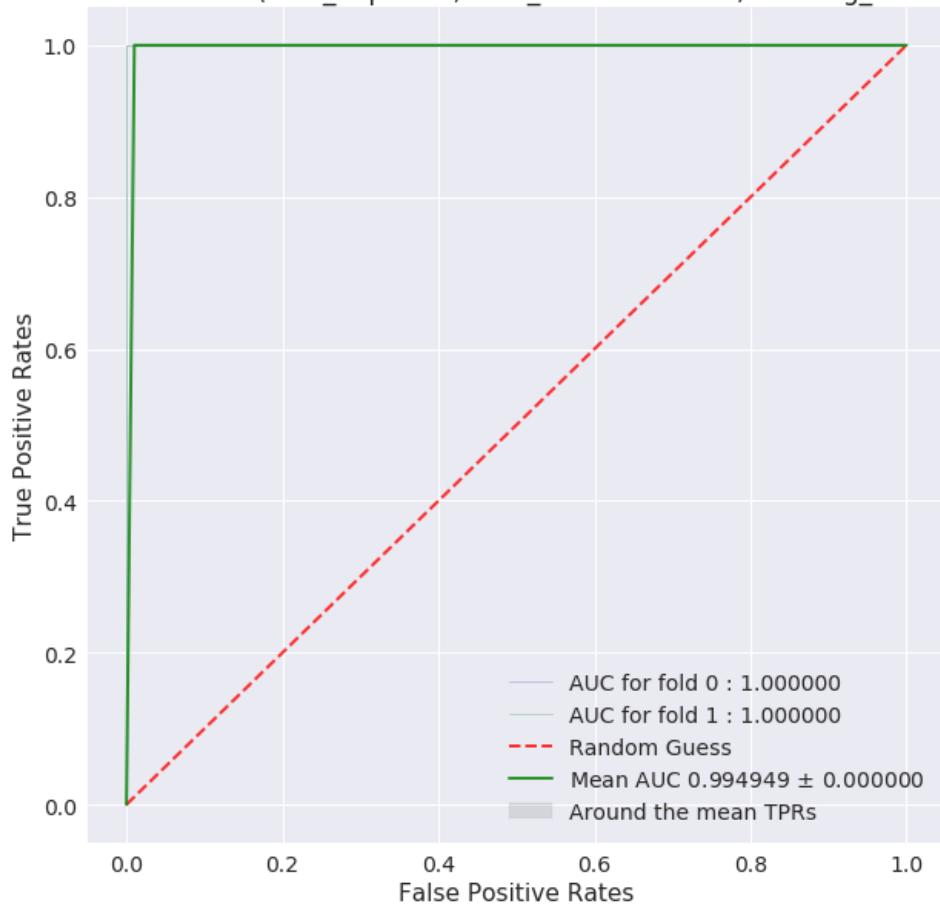


ROC - Validation Ensemble (max\_depth:50, num\_estimators:120, learning\_rate:0.100000)

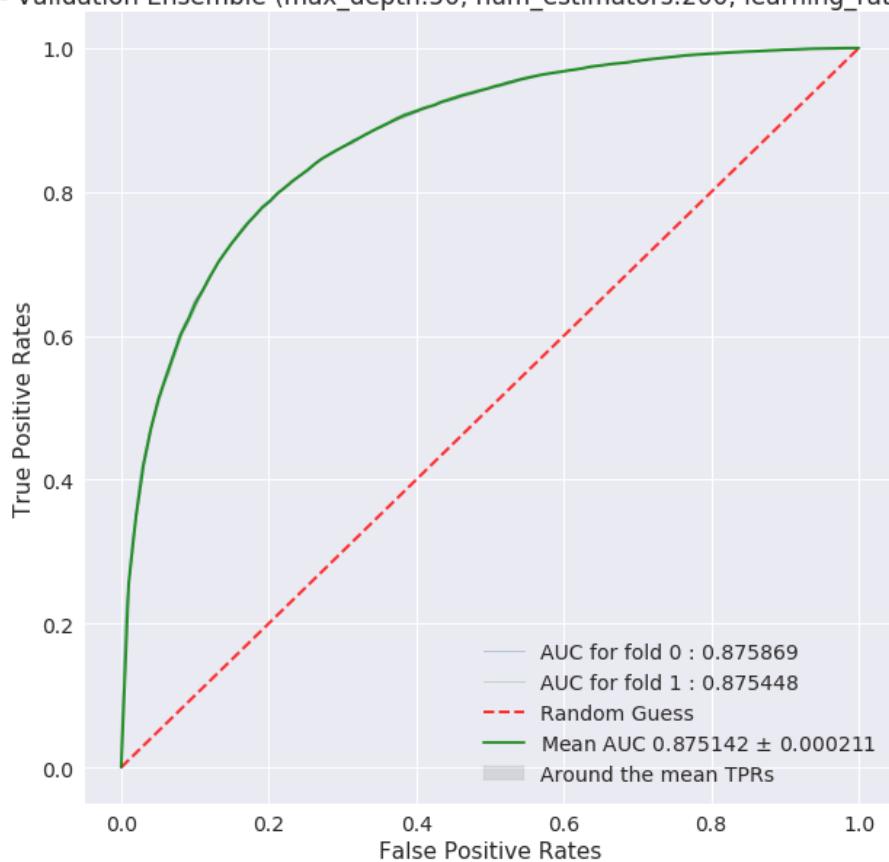


```
=====
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Train Ensemble (max\_depth:50, num\_estimators:200, learning\_rate:0.100000)



ROC - Validation Ensemble (max\_depth:50, num\_estimators:200, learning\_rate:0.100000)



=====  
Train hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.833062
1	(2, 70, 0.1)	0.850426
2	(2, 120, 0.1)	0.866610
3	(2, 200, 0.1)	0.880298
4	(5, 40, 0.1)	0.901913
5	(5, 70, 0.1)	0.923162
6	(5, 120, 0.1)	0.943840
7	(5, 200, 0.1)	0.962753
8	(20, 40, 0.1)	0.994949
9	(20, 70, 0.1)	0.994949
10	(20, 120, 0.1)	0.994949
11	(20, 200, 0.1)	0.994949
12	(50, 40, 0.1)	0.994949
13	(50, 70, 0.1)	0.994949

```
14 (50, 120, 0.1) 0.994949
15 (50, 200, 0.1) 0.994949
```

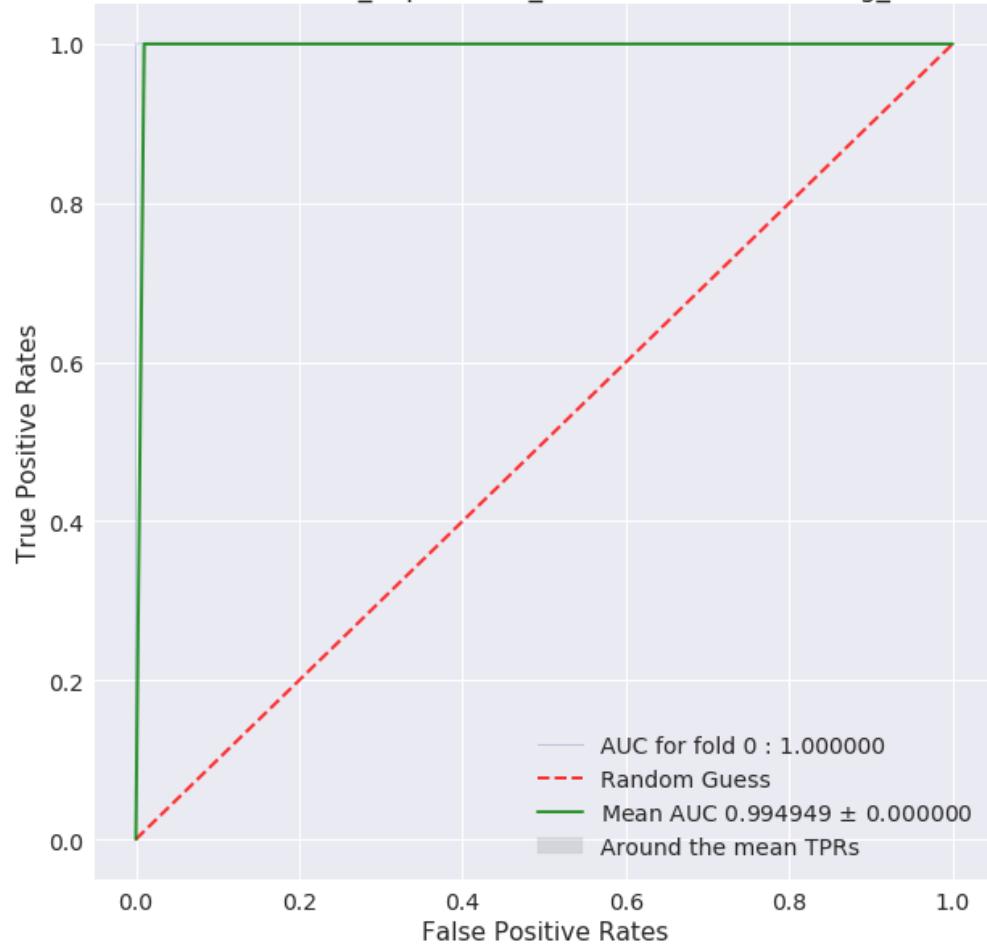
Validation hyper params:

	Hyper Params	AUC
0	(2, 40, 0.1)	0.821436
1	(2, 70, 0.1)	0.836218
2	(2, 120, 0.1)	0.848860
3	(2, 200, 0.1)	0.858622
4	(5, 40, 0.1)	0.856127
5	(5, 70, 0.1)	0.865521
6	(5, 120, 0.1)	0.871543
7	(5, 200, 0.1)	0.874250
8	(20, 40, 0.1)	0.860769
9	(20, 70, 0.1)	0.867988
10	(20, 120, 0.1)	0.872471
11	(20, 200, 0.1)	0.875354
12	(50, 40, 0.1)	0.860774
13	(50, 70, 0.1)	0.867934
14	(50, 120, 0.1)	0.872153
15	(50, 200, 0.1)	0.875142

Best hyperparam value: (20, 200, 0.1)

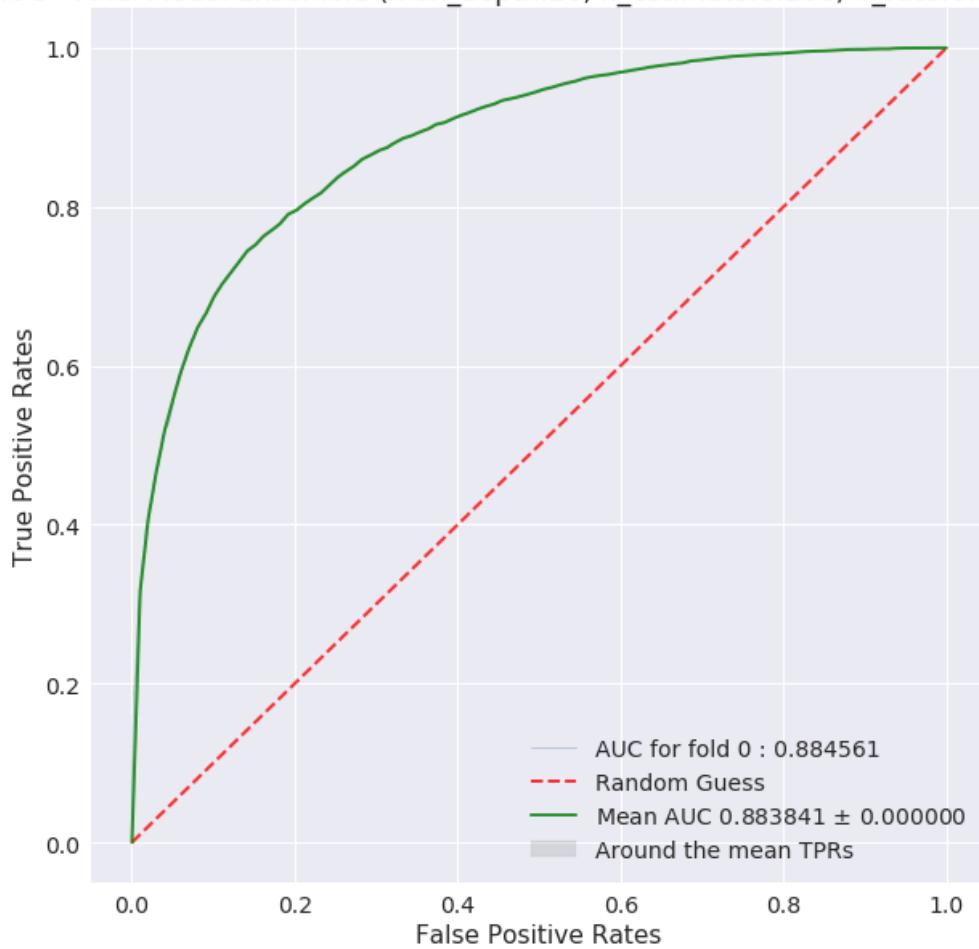
```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: Deprecatio
  if diff:
```

ROC - Final Model DT (max\_depth:20, n\_estimators:200, learning\_rate:0.100000)



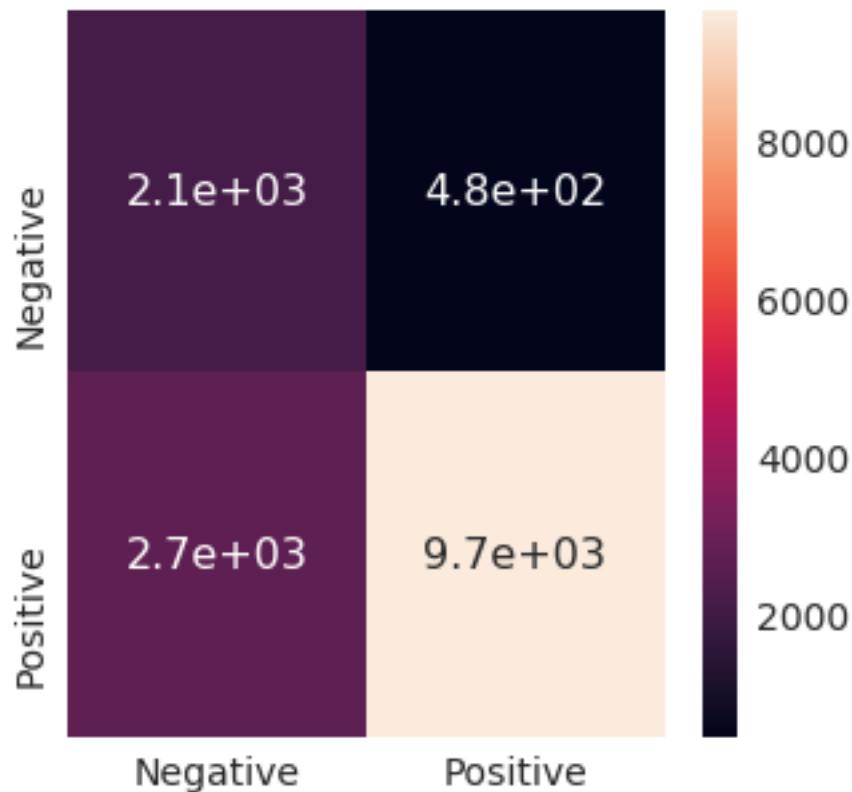
```
/home/amd_3/anaconda3/lib/python3.6/site-packages/sklearn/preprocessing/label.py:151: DeprecationWarning: if diff:
```

ROC - Final Model Ensemble (max\_depth:20, n\_estimators:200, lr\_rate:0.100000)



Test auc score 0.8838408303933897

### Ensemble Model Confusion Matrix



	Negative	Positive
Precision	0.439014	0.953011
Recall	0.817904	0.779428
Fscore	0.571352	0.857524
Support	2614.000000	12386.000000

#### 4.6 Observation on XGB set 1, set 2, set 3, set 4

XGB has more number of hyper parameters to tune and is computationally expensive to search.  
All the xgb model has AUC score above 0.88

## 5 Procedure Summary

- Train random forest with different hyper parameter setting on all 4 datasets
- Identify the best hyper parameter usig cross validation method
- Build final RF model by using the best hyper param identified
- Visualize the performance of model using ROC curve & confusion matrix
- Represent the important features identified by the decision tree in word cloud

Train XGB models with different hyper parameter setting on all 4 datasets  
Identify the best hyper parameter usig cross validation method  
Build final XGB model by using the best hyper param identified  
Visualize the performace of model using ROC curve & confusion matrix

## 6 Results Summary

```
In [2]: from prettytable import PrettyTable
Pret_table = PrettyTable()
Pret_table.field_names = ['Vectorizer', 'Model', '(max_depth, n_estimators,*learning_rate)', 'AUC', 'Fscore (-ve)', 'Fscore (+ve)']
Pret_table.title = 'Ensemble Results Summary'

In [3]: ptabe_entry_a1 = [(50, 500), 0.9171, 62.70, 88.19]
ptabe_entry_a2 = [(50, 500), 0.9169, 62.80, 88.22]
ptabe_entry_a3 = [(10, 500), 0.9116, 62.00, 87.92]
ptabe_entry_a4 = [(10, 500), 0.8617, 54.19, 84.32]

ptabe_entry_b1 = [(20, 200, 0.1), 0.9324, 65.6725, 89.6031]
ptabe_entry_b2 = [(20, 200, 0.1), 0.9305, 65.3120, 89.4664]
ptabe_entry_b3 = [(5, 200, 0.1), 0.9251, 64.3799, 89.1104]
ptabe_entry_b4 = [(20, 200, 0.1), 0.8838, 57.1352, 85.7524]

In [4]: # RF Results Summary
Pret_table.add_row(['BoW', 'RF'] + ptabe_entry_a1)
Pret_table.add_row(['TF-IDF', 'RF'] + ptabe_entry_a2)
Pret_table.add_row(['Avg W2V', 'RF'] + ptabe_entry_a3)
Pret_table.add_row(['TF-IDF W2V', 'RF'] + ptabe_entry_a4)

# XGBoost Results Summary
Pret_table.add_row(['BoW', 'XGB'] + ptabe_entry_b1)
Pret_table.add_row(['TF-IDF', 'XGB'] + ptabe_entry_b2)
Pret_table.add_row(['Avg W2V', 'XGB'] + ptabe_entry_b3)
Pret_table.add_row(['TF-IDF W2V', 'XGB'] + ptabe_entry_b4)

In [5]: print(Pret_table)

+-----+
|                               Ensemble Results Summary
+-----+-----+-----+-----+-----+-----+
| Vectorizer | Model | (max_depth, n_estimators,*learning_rate) | AUC   | Fscore (-ve) | Fscore (+ve) |
+-----+-----+-----+-----+-----+-----+
|    BoW     |    RF  | (50, 500)           | 0.9171 | 62.7      | 88.19       |
|    TF-IDF  |    RF  | (50, 500)           | 0.9169 | 62.8      | 88.22       |
| Avg W2V   |    RF  | (10, 500)          | 0.9116 | 62.0      | 87.92       |
| TF-IDF W2V|    RF  | (10, 500)          | 0.8617 | 54.19     | 84.32       |
|    BoW     |   XGB  | (20, 200, 0.1)     | 0.9324 | 65.6725   | 89.6031    |
|    TF-IDF  |   XGB  | (20, 200, 0.1)     | 0.9305 | 65.3120   | 89.4664    |
```

Avg W2V	XGB	(5, 200, 0.1)	0.9251	64.3799	89.1
TF-IDF W2V	XGB	(20, 200, 0.1)	0.8838	57.1352	85.7

## 7 Conclusions

XGB models outperformed when compared with RF models. The f-score for +ve class as well as -ve class is higher when compared with RF model

The f-score of XGB model for -ve class is approximately 3% more than the f-score of RF model for -ve class