Magnimind Machine Learning **Name:** _____
1-to-1 Mentorship Program
Exam 1
6/24/2020
Time Limit: 60 Minutes _____

1. This exam contains 7 pages (including this cover page) and 22 questions.
2. Total of points is 100.
3. Answer all questions. The marks for each question are indicated at the beginning of each question.
4. This **IS an OPEN BOOK** exam. You can only use your notes from your class.
5. Calculators are not allowed.

1 True/False (20 points)

1. (2 points) If highly correlated but relevant features are present in a dataset, Lasso regression will select one of them at random. FALSE
2. (2 points) Tuning two hyper-parameters with four options each using grid-search with 5-fold cross-validation requires exactly 40 model fits.
3. (2 points) It is good practice to standardize sparse datasets so that each feature has zero mean. TRUE
4. (2 points) Kernel support vector machines don't scale well to large datasets. TRUE
5. (2 points) Ridge Regression adds an L_1 norm penalty to the cost function and often sets several of the weights to zero. FALSE
6. (2 points) Using kernel trick, one can get non-linear decision boundaries using algorithms designed originally for linear models. TRUE
7. (2 points) 5-NN has more overfitting (lower bias) than 1-NN. FALSE
8. (2 points) It is important that exactly same scaling transformation is applied to the training set and the test set for the supervised model. TRUE
9. (2 points) Despite its name, LogisticRegressionis a classification algorithm and not a regression algorithm. TRUE
10. (2 points) The distinction between the training set, validation set, and test set is fundamentally important to apply machine learning methods in practice. Any choices made based on the test set accuracy "leak" information from the test set into the model. TRUE

2 Multiple Choice (20 points, 4pts each)

Select **all** choices that apply.

11. (4 points) After training a ridge regression model, you find the training and test accuracies are 0.97 and 0.55, respectively. Which of the following would be the best choice for the next ridge regression model you train?

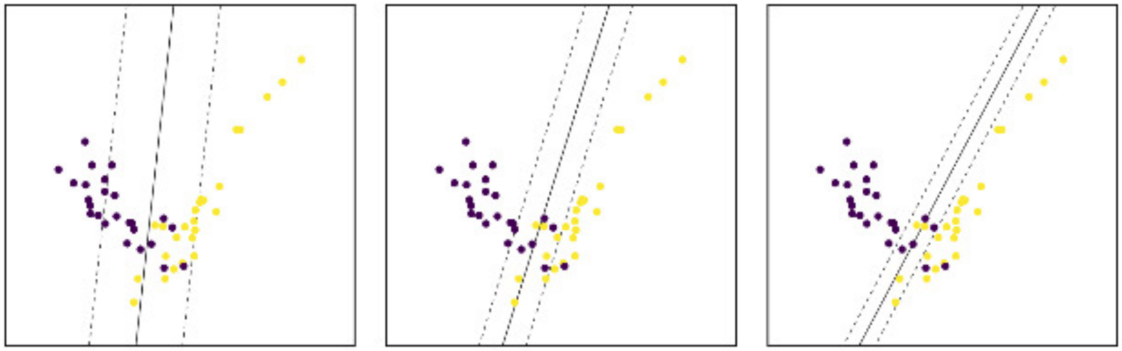
    A. You are overfitting, the next model trained should have a lower value for alpha

    B. You are overfitting, the next model trained should have a higher value for alpha

    C. You are underfitting, the next model trained should have a lower value for alpha

    D. You are underfitting, the next model trained should have a higher value for alpha

12. (4 points) Match the plots of SVM margins below to the values of the C parameter that corresponds to them.



    A. 10,1,0.1

    B. 1,0.1,10

    C. 0.1,1,10

    D. 10,0.1,1

    E. 1,10,0.1

13. (4 points) Which of the following variables should be treated as categorical?

    Income

    Nationality X

    Gender X

    Age X

    ZIP code X

14. (4 points) Suppose you are interested in finding a parsimonious model (the model that accomplishes the desired level of prediction with as few predictor variables as possible) to predict housing prices. Which of the following would be the best choice?

    A. Ridge Regression
    B. Ordinary Least Squares Regression
    C. Logistic Regression
    D. Lasso Regression

15. (4 points) Which of the following is not part of data preprocessing?

    A. Scaling
    B. Data transformation
    C. One-Hot-Encoding
    D. Feature Selection
    E. Cross-validation

3 Debugging

For each code snippet,find and explain **all** errors given the task. There can be more than one errors. You can write your answer on the empty spaces on this page.

16. (10 points) Task: Perform grid search (without using the GridSearchCV class) using a split into training, validation, and test data, with a final valuation on the test data.

```
X_trainval, X_test, y_trainval, y_test=train_test_split(X,y)
X_train, X_valid, y_train, y_valid=
train_test_split(X_trainval, y_trainval)

best_score=0

for C in [0.001, 0.01, 0.1, 1, 10, 100]:
    svm=LinearSVC(C=C)
    svm.fit(X_train, y_train)
    score=svm.score(X_test, y_test)
    if score > best_score:
        best_score=score
        best_C=C

svm=LinearSVC(C=best_C).fit(X_valid, y_valid)
```

17. (10 points) Task: Use the PowerTransformer (implementing the box-cos transformation) transformer to preprocesss data and learn a Ridge model.

```
pipe=make_pipeline (StandardScaler(), PowerTransformer(), Ridge())
scores=cross_val_score(pipe, X_train, y_test, cv=10)
```

16.

```
X_trainval, X_test, y_trainval, y_test = train_test_split(X, y, random_state=42)
X_train, X_valid, y_train, y_valid = train_test_split(X_trainval, y_trainval, random_state=42)

best_score = 0
for C in [0.001, 0.01, 0.1, 1, 10, 100]:
    svm = LinearSVC(C=C, random_state=42)
    svm.fit(X_train, y_train)
    score = svm.score(X_valid, y_valid)
    if score > best_score:
        best_score = score
        best_C = C

svm = LinearSVC(C=best_C, random_state=42).fit(X_trainval, y_trainval)
test_score = svm.score(X_test, y_test)
```

17.

```
pipe = make_pipeline(StandardScaler(), PowerTransformer(), Ridge())
scores = cross_val_score(pipe, X_train, y_train, cv=10)
```

4 Coding

Assume all necessary imports are already made.

18. (10 points) Provide code to build LogisticRegression model and evaluate its performance on a separate test set, given a dataset as numpy arrays X and y.

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Test set accuracy: ", accuracy)
```

19. (10 points) Provide code to implement grid-searching the parameters C and gamma of an SVC in a pipeline with a StandardScaler, and evaluating the best parameter setting on a separate test set, given a dataset as numpy arrays X and y.

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV, train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

pipe = Pipeline([('scaler', StandardScaler()), ('svc', SVC())])

param_grid = {'svc__C': [0.1, 1, 10, 100], 'svc__gamma': [0.1, 1, 10, 100]}

grid_search = GridSearchCV(pipe, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)

best_svc = grid_search.best_estimator_
test_score = best_svc.score(X_test, y_test)

print(f"Best parameters: {grid_search.best_params_}")
print(f"Test score: {test_score:.3f}")
```

5 Concepts (20 Points)

20. (5 points) What is overfitting?

Overfitting is a common problem in machine learning where a model performs well on the training data but performs poorly on new, unseen data. In other words, the model "memorizes" the training data and does not generalize well to new data. Overfitting occurs when a model is too complex relative to the amount and/or noise in the training data, leading it to fit to the noise in the data rather than the underlying patterns. This can result in a model that is not useful for practical applications because it makes inaccurate predictions on new data. To avoid overfitting, it is important to use techniques such as regularization, cross-validation, and early stopping.

21. (5 points) Why are nearest neighbor methods sensitive to the scaling of the data?

Nearest neighbor methods are sensitive to the scaling of the data because they rely on computing distances between data points to make predictions. If one feature has a much larger scale than others, it will dominate the distance metric and make other features less important. As a result, rescaling the features to a common scale is important to ensure that all features contribute equally to the distance metric and prevent the nearest neighbor method from being biased towards one feature. Failure to properly scale the data can lead to poor performance and inaccurate predictions.

22. (10 points) A real estate firm would like to build a system that predicts the sale prices of a house. They create a spreadsheet containing information about 1460 house sales in the Santa Clara area. In addition to the price, there are 79 features describing the house, such as number of bedrooms, total indoor area, lot area, has a garage, location, etc. Explain how you would implement a machine learning model that would solve this prediction task. You don't need to show Python code,but please give a description of the system and explain all steps you would carry out when developing it.

To implement a machine learning model that predicts house sale prices, the following steps can be taken:

Data Cleaning: The data should be checked for missing values and outliers. Missing values can be filled with the median or mean of the feature, and outliers can be removed or corrected.

Feature Selection: Out of the 79 features, not all are relevant in predicting the house sale price. Therefore, the most important features can be selected using techniques such as correlation analysis or regularization.

Data Preprocessing: The features can be scaled to ensure that they are on the same scale. Categorical features can also be transformed into binary variables using one-hot encoding.

Model Selection: Several regression models can be used to predict house prices, including linear regression, decision trees, and random forests. The choice of model should be based on the accuracy and interpretability of the model.

Model Evaluation: The model's performance can be evaluated using metrics such as the mean squared error, mean absolute error, and R-squared. Cross-validation can also be used to ensure that the model's performance is not overfitting to the training data.

Hyperparameter Tuning: The model's hyperparameters can be tuned using techniques such as grid search or random search to find the optimal hyperparameters that give the best performance on the test data.

Deployment: Once the model is trained and tested, it can be deployed into production and used to predict the sale prices of new houses.